Journal of Artificial Intelligence Research 59 (2017) 725-770 Submitted 01/17; published 08/17

Decision-Theoretic Planning under Anonymity in Agent Populations

Ekhlas Sonu ESONU@STANFORD.EDU
Dept of Aeronautics and Astronautics
Stanford University
Stanford, CA 94305 USA

Yingke Chen YKE.CHEN@GMAIL.COM
College of Computer Science
Sichuan University
Sichuan, China

Prashant Doshi PDOSHI@CS.UGA.EDU
THINC Lab, Dept of Computer Science
University of Georgia
Athens, GA 30602 USA

Abstract

We study the problem of self-interested planning under uncertainty in settings shared with more than a thousand other agents, each of which plans at its own individual level. We refer to such large numbers of agents as an agent population. The decision-theoretic formalism of interactive partially observable Markov decision process (I-POMDP) is used to model the agent's self-interested planning. The first contribution of this article is a method for drastically scaling the finitely-nested I-POMDP to certain agent populations for the first time. Our method exploits two types of structure that is often exhibited by agent populations – anonymity and context-specific independence. We present a variant called the many-agent I-POMDP that models both these types of structure to plan efficiently under uncertainty in multiagent settings. In particular, the complexity of the belief update and solution in the many-agent I-POMDP is polynomial in the number of agents compared with the exponential growth that challenges the original framework.

While exploiting structure helps mitigate the curse of many agents, the well-known curse of history that afflicts I-POMDPs continues to challenge scalability in terms of the planning horizon. The second contribution of this article is an application of the branch-and-bound scheme to reduce the exponential growth of the search tree for look ahead. For this, we introduce new fast-computing upper and lower bounds for the exact value function of the many-agent I-POMDP. This speeds up the look-ahead computations without trading off optimality, and reduces both memory and run time complexity. The third contribution is a comprehensive empirical evaluation of the methods on three new problems domains – policing large protests, controlling traffic congestion at a busy intersection, and improving the AI for the popular Clash of Clans multiplayer game. We demonstrate the feasibility of exact self-interested planning in these large problems, and that our methods for speeding up the planning are effective. Altogether, these contributions represent a principled and significant advance toward moving self-interested planning under uncertainty to real-world applications.

1. Introduction

Decision-theoretic planning by a single agent under partial observability is formalized by the partially observable Markov decision process (POMDP) (Kaelbling, Littman, & Cassandra, 1998). A POMDP is an appealing framework for study because of its mathematical rigor and a theoretical assurance of optimal planning under uncertainty. However, the planning is challenged by high

computational complexity. Two sources contribute to this complexity: the curse of state dimensionality which disproportionately impacts the size of the belief space; and, of particular relevance to this article, the curse of history or policy space that is due to an exponentially growing search tree with increase in planning horizon. As partial observability is encountered in various applications, especially in those involving agents with sensors, POMDPs have been generalized to multiagent settings in different ways (Seuken & Zilberstein, 2008). In this article, we focus on self-interested planning under uncertainty in settings shared with other agents who themselves plan, act and observe. This is formalized by the well-known interactive POMDP (I-POMDP) framework (Gmytrasiewicz & Doshi, 2005).

We situate the problem of self-interested planning under uncertainty in settings shared with more than a *thousand* other agents, each of which plans at its own individual level. For example, consider the decision-making problem of the police when faced with a large protest distributed over multiple protest sites. These settings require us to model many agents in an I-POMDP and solve exactly in this context. Such scalability is critically needed because I-POMDPs cover an important portion of the multiagent planning problem space (Doshi, 2012; Durfee & Zilberstein, 2013). Applications in diverse areas such as security (Seymour & Peterson, 2009; Ng, Meyers, Boakye, & Nitao, 2010), robotics (Woodward & Wood, 2012; Wang, 2013), ad hoc teams (Chandrasekaran, Doshi, Zeng, & Chen, 2014) and human behavior modeling (Doshi, Qu, Goodie, & Young, 2010; Wunder, Kaisers, Yaros, & Littman, 2011; Hula, Montague, & Dayan, 2015) testify to the wide appeal of I-POMDPs while motivating better scalability.

Both of the previously mentioned sources of complexity present in POMDPs are inherited by I-POMDPs, and exacerbated because of multiple agents. Mitigating these sources to promote scalability has been the focus of attention of methods so far, as we review in the next section. While the above sources of complexity are well understood, additional lesser-known barriers emerge in contexts populated by *many* agents. Potent among these is the fact that the space of models in the interactive state space grows exponentially with the number of agents. We refer to this complexity afflicting multiagent planning as the "curse of many agents".

We identify two problem structures that afford tractability in the context of interactions involving many agents. For illustration, consider again the decision-making problem of the police when faced with a large protest. The degree of the police response is often decided by how many protestors of which type (disruptive or not) are participating. The individual identity of the protestor within each type seldom matters. This key observation of frame-action anonymity, where a *frame* encapsulates the agent's capabilities and preferences, [1] motivates us in how we model the agent population in the planning. Such anonymity is also observed in other problems that require decision making in the presence of many other agents such as a smart grid operator seeking to time shift flexible energy loads of many consumers, devising strategy for a massively multiplayer online game, or in intelligently managing dense vehicular traffic for congestion control. However, as we identify the frames of agents, an implicit assumption is that there are far fewer distinct frames than agents. Furthermore, the planned degree of response at a protest site is influenced, in part, by how many disruptive protestors are predicted to converge to the site and much less by some other actions of the protestors such as movement between other distant sites. This example illustrates two known and powerful types of problem structure in domains involving many agents: *action anonymity* (Roughgarden & Tardos, 2002) and *context-specific independence* (Boutilier, Friedman, Goldszmidt, & Koller, 1996).

1. I-POMDPs distinguish between an agent's frame and its type with the latter including beliefs as well. Frames are similar in semantics to the colloquial use of types.

Action anonymity allows the exponentially large joint action space to be substituted with a much more compact space of action configurations where a configuration is a tuple representing the number of agents performing each action. Consequently, the run-time complexity is no longer exponential in the number of agents but is instead polynomial. Context-specific independence wherein given a context such as the state and agent's own action, not all actions performed by other agents are relevant toward determining the next state permits the space of configurations to be compressed by projecting counts over a limited subset of others' actions. We extend both action anonymity and context-specific independence to allow considerations of an agent's frame as well.

Consequently, our first contribution in this article is a variant of finitely-nested I-POMDPs referred to as the **many-agent** I-POMDP framework that models both types of structure – frame-action anonymity and context-specific independence – to plan efficiently and exactly in multiagent settings. In particular, the complexity of the belief update and solution in the many-agent I-POMDP is *polynomial* in the number of agents under certain assumptions compared with the exponential growth that challenges the original framework. A key feature of the framework is a systematic way of modeling the context-specific independence in transition, observation and reward functions using *frame-action hypergraphs*, and exploiting it in an exact method for solving I-POMDPs that models other agents using finite-state machines. The Bellman equation modified to include frame-action configurations and independences continues to remain optimal given the I-POMDP with explicated problem structure.

While exploiting problem structure helps mitigate the curse of many agents, the well-known curse of history that afflicts I-POMDPs continues to challenge scalability in terms of the planning horizon. Specifically, the look-ahead search tree for planning has a branching factor that is proportional to the number of subject agent's actions and observations, and grows exponentially with the horizon. The second contribution of this article is a first application of the branch-and-bound scheme in the context of I-POMDPs to significantly reduce the size of the look-ahead search tree. We introduce new fast-computing upper and lower bounds for the exact value function of the many-agent I-POMDP. Both these are utilized in the branch and bound scheme. The lower bound is obtained by blindly pursuing a single policy oblivious to the observation and the upper bound is the result of assuming that the initial state is known but not the subsequent states; both these bounds are appropriately generalized to the many-agent I-POMDP.

The third contribution of this article is a theoretical verification of the savings in computational time and memory due to the anonymity and context-specific independence, and its demonstration on three large problem domains that naturally contain agent populations. The first domain pertains to policing a large protest involving more than a thousand protestors of different types, which are distributed over multiple protest sites. The second domain requires managing vehicular traffic at multiple intersections for congestion control. The third domain is to play defense in the popular massively multiplayer game *Clash of Clans* (`http://clashofclans.com`). The subject agent is tasked with defending its settlement and other resources against an invasion by other players' armies. All of these domains are new and could potentially serve as testbeds for evaluating both self-interested and cooperative planning in many-agent settings. We demonstrate the efficiency of the branch and bound method by exactly solving these problems for up to 2,000 agents in less than 6 hours on a standard computing platform.

The rest of this article is structured as follows. Section 2 presents the background briefly reviewing the definition of the finitely-nested I-POMDP and its solution, and reviewing the game-theoretic framework of action graph games. Using a running example, we introduce the frame-action

hypergraph as a model of frame-action anonymity and context-specific independence and how it is used in the many-agent I-POMDP framework, in Section 3. Section 4 gives the relevant algorithms and discusses the run time complexity. We derive upper and lower bounds for the exact value function and present the branch-and-bound scheme in Section 5. The problem domains are introduced in Section 6 followed by a comprehensive discussion of the experimentation. We discuss related work in detail in Section 7 in particular emphasizing how our frame-action anonymity differs from the use of anonymity in other approaches. We conclude this article with some remarks in Section 8. Appendix 1 summarizes key notation in a table for easy reference, while Appendix 2 gives proofs of the theorems.

2. Background

In this section, we first briefly review POMDPs and some bounds on its value function. Then, we briefly describe how I-POMDPs generalize POMDPs to multiagent settings. Finally, we briefly review action graph games that model the problem structures, which we expand in our framework.

2.1 Partially Observable Markov Decision Processes

Decision-theoretic planning under uncertainty when the physical state is not perfectly observed is formalized using a POMDP (Smallwood & Sondik, 1973; Kaelbling et al., 1998). Mathematically, a POMDP is defined as the following tuple of parameters:

$$\text{POMDP} := \langle S, A, T, \Omega, O, R, OC \rangle$$

where:

- S is the set of physical states of the environment and the agent relevant to the planning;
- A is the set of actions that the agent may perform;
- $T : S \times A \times S \to [0, 1]$, is a stochastic transition function which gives the distribution over the next physical states given the current state and agent's action;
- Ω is the set of observations that the agent may receive;
- $O : S \times A \times \Omega \to [0, 1]$, is a stochastic observation function which gives the probability with which the agent receives an observation conditioned on its action and the resulting state;
- $R : S \times A \to \mathbb{R}$, is the reward function that gives the reward (or cost) for the agent given its state and the action from the state;
- OC is the optimality criterion, which can be to maximize a sum of the (discounted) rewards obtained over a fixed number of steps H (called horizon), or a discounted sum over an infinite number of steps. In this article, we maximize a discounted finite-horizon sum, $\max \sum_{t=0}^{H-1} \gamma^t \, E[r^t]$, where the discount factor is denoted by $\gamma \in [0, 1]$ and $E[r^t]$ is the expected reward at time step t.

Due to the partial observability, the agent maintains a probability distribution over the current state, which is referred to as its *belief*. The agent updates its belief on performing an action and receiving an observation using the well-known Bayes filter. An agent's belief over the state is a sufficient statistic for its history of observations.

The solution of a POMDP is an optimal *policy*, which is a function that maps the agent's beliefs to a distribution over actions, $\pi : \mathcal{B} \times A \to [0, 1]$, where $\mathcal{B}$ is the belief simplex. If the horizon is finite,

the policy is not stationary and is also indexed by the horizon ranging from H to 1. A finite-horizon policy may be conveniently represented as a labeled tree whose root node gives the action to perform, labeled edges represent observations each of which leads to a node denoting an action to perform given the observation. Thus, a tree-based representation assumes that the sequence and space of observations are finite. To obtain the policy exactly, we define a value function, $V : \mathcal{B} \rightarrow \mathbb{R}$, which gives the expected long-term reward from a belief according to the optimality criterion:

$$V^h(b^t) = \max_{a^t \in A} \left[\sum_{s^t \in S} b^t(s^t) R(s^t, a^t) + \gamma \sum_{\omega^{t+1} \in \Omega} \max_{\alpha^{h-1}} \left\{ \sum_{s^t \in S} b^t(s^t) \sum_{s^{t+1} \in S} O(s^{t+1}, a^t, \omega^{t+1}) \right. \right.$$
$$\left. \left. T(s^t, a^t, s^{t+1})\, \alpha^{h-1}(s^{t+1}) \right\} \right] \tag{1}$$

where h is the horizon and $\alpha : S \rightarrow \mathbb{R}$ is an *alpha vector* of values of dimension $|S|$. For computation purposes, we may factorize this value function as an inner product between the belief and optimizing alpha vectors. We can rewrite the above value function more succinctly as, $V^h(b^t) = \max_{a \in A} \sum_{s^t \in S} b^t(s)\ \alpha^h_{b^t,a,\omega}(s)$. An optimal policy is the one whose actions optimize the value function. Various algorithms (Hauskrecht, 1997) show how the value function may be computed in different ways – some do it more efficiently than others.

The value function defined above also admits several lower and upper bounds, which are utilized in faster approximation methods (Smith & Simmons, 2004). We briefly review one upper and lower bound, introduced by Hauskrecht (2000), as we will generalize these later in this article. A fast-computing lower bound is the value of a policy that performs a root action from the current belief followed by the next action regardless of the observation; as such it is blind to the observations. The value function for the blind-policy lower bound is:

$$\underline{V}^h(b^t) = \max_{a^t \in A} \left[\sum_{s^t \in S} b^t(s^t) R(s^t, a^t) + \gamma \max_{\alpha^{h-1}} \left\{ \sum_{\omega^{t+1} \in \Omega} \sum_{s^t \in S} b^t(s^t) \sum_{s^{t+1} \in S} O(s^{t+1}, a^t, \omega^{t+1}) \right. \right.$$
$$\left. \left. \times\, T(s^t, a^t, s^{t+1})\, \alpha^{h-1}(s^{t+1}) \right\} \right]$$
$$= \max_{a^t \in A} \left[\sum_{s^t} b^t(s^t) R(s^t, a^t) + \gamma \max_{\alpha^{h-1}} \left\{ \sum_{s^t} b^t(s^t) \sum_{s^{t+1}} T(s^t, a^t, s^{t+1})\, \alpha^{h-1}(s^{t+1}) \right\} \right]$$
$$= \max_{a^t \in A} \sum_{s^t} b^t(s^t) \left[R(s^t, a^t) + \gamma \sum_{s^{t+1} \in S} T(s^t, a^t, s^{t+1})\, \alpha^{h-1}_{b^t,a^t}(s^{t+1}) \right]$$
$$= \max_{a^t \in A} \sum_{s^t \in S} b^t(s^t)\, \alpha^h_{b^t,a^t}(s^t) \tag{2}$$

Clearly, not considering the observations in determining the next action is suboptimal and Hauskrecht establishes that Eq. 2 lower bounds the exact value function. Furthermore, this update of the value function requires generating $\mathcal{O}(|A||\Gamma^{h-1}|)$ possible alpha vectors, which can be much less than the $\mathcal{O}(|A||\Gamma^{h-1}|^{|\Omega|})$ many alpha vectors in the exact case. Here, Γ^{h-1} is the set of alpha vectors in the previous horizon. The update takes time $\mathcal{O}(|A||S|^2|\Gamma^{h-1}|)$.

An upper bound that is computed quickly is obtained using the following update rule. It uses the maximizing alpha vector that is the $\arg\max_{\alpha^{h-1}} \left\{ \sum_{s^{t+1} \in S} O(s^{t+1}, a^t, \omega^{t+1})\, T(s^t, a^t, s^{t+1})\, \alpha^{h-1}(s^{t+1}) \right\}$.

Compare this to the maximizing alpha vector used in the exact update rule, $\arg\max_{\alpha^{h-1}} \Big\{ \sum_{s^{t+1}\in S} O(s^{t+1}, a^t, \omega^{t+1}) \sum_{s^t\in S} b(s^t)\, T(s^t, a^t, s^{t+1})\, \alpha^{h-1}(s^{t+1}) \Big\}$, and we note that the latter selection is more informed leading to an upper bound:

$$\begin{aligned}
\overline{V}^h(b^t) &= \max_{a^t\in A} \Bigg[\sum_{s^t\in S} b^t(s^t) R(s^t, a^t) + \gamma \sum_{\omega^{t+1}\in\Omega} \sum_{s^t\in S} b(s^t) \max_{\alpha^{h-1}} \Bigg\{ \sum_{s^{t+1}\in S} O(s^{t+1}, a^t, \omega^{t+1}) \\
&\qquad \times T(s^t, a^t, s^{t+1}) \alpha^{h-1}(s^{t+1}) \Bigg\} \Bigg] \\
&= \max_{a^t\in A} \sum_{s^t\in S} b^t(s^t) \Bigg[R(s^t, a^t) + \gamma \sum_{\omega^{t+1}\in\Omega} \max_{\alpha^{h-1}} \Bigg\{ \sum_{s^{t+1}\in S} O(s^{t+1}, a^t, \omega^{t+1})\, T(s^t, a^t, s^{t+1}) \\
&\qquad \times \alpha^{h-1}(s^{t+1}) \Bigg\} \Bigg] \\
&= \max_{a^t\in A} \sum_{s^t\in S} b^t(s^t)\, \alpha_{a^t}(s^t) \qquad (3)
\end{aligned}$$

Notice that an update of the value function involves computing $|A|$ many vectors only compared to the exponentially-growing $|A||\Gamma^{h-1}|^{|\Omega|}$ vectors in the exact case. An alpha vector in this bound utilizes a vector from the next time step for each observation and current state. The update in this bound takes time $\mathcal{O}(|A|^2|S|^2|\Omega|)$ because $|\Gamma^{h-1}| = |A|$, which is significantly less than the worst-case $\mathcal{O}(|A||S|^2|\Gamma^{h-1}|^{|\Omega|})$ time taken by the exact update. Thus, this informed bound is computed quickly, and as Hauskrecht (2000) shows empirically, is tight as well.

2.2 Interactive POMDP

Moving from a self-interested single- to multi-agent setting, the framework of interactive POMDP (I-POMDP) generalizes POMDPs. An agent in the I-POMDP framework predicts actions of the other agents by reasoning about possible models that could explain the agent's observations. A model may itself be an I-POMDP, which gives rise to recursive reasoning. For purposes of computability, the recursive reasoning is limited to finite depths; the corresponding finitely-nested I-POMDP for an agent 0 situated with N other agents is defined below:

$$\text{I-POMDP}_{0,l} \coloneqq \langle IS_{0,l}, A, T_0, \Omega_0, O_0, R_0, OC_0 \rangle$$

where:

- $IS_{0,l}$ denotes the set of *interactive states* defined as, $IS_{0,l} = S \times \prod_{j=1}^{N} M_{j,l-1}$, $l \geq 1$, where S is the set of shared physical states relevant to all agents and $M_{j,l-1}$ is the set of models ascribed to the other agent j; $IS_{0,0} = S$. We describe the model space after this definition;
- $A = A_0 \times A_1 \times \ldots \times A_N$ is the set of joint actions of all agents. Let $\mathbf{a}_{-0}$ denote a joint action of the N other agents, $\mathbf{a}_{-0} \in \prod_{j=1}^{N} A_j$;
- $T_0 : S \times A_0 \times \prod_{j=1}^{N} A_j \times S \to [0, 1]$ is the transition function which gives the distribution over the next physical states given the current state and a joint action;
- Ω_0 is the set of agent 0's observations;

- $O_0 : S \times A_0 \times \prod_{j=1}^{N} A_j \times \Omega_0 \rightarrow [0, 1]$ is the observation function giving the likelihood of agent 0's observations conditioned on a joint action and the resulting state;

- $R_0 : S \times A_0 \times \prod_{j=1}^{N} A_j \rightarrow \mathbb{R}$ is the reward function that specifies the reward agent 0 receives given a joint action performed by all agents from a state.

- OC_0 is the optimality criterion, which is similar to that for a POMDP. In this article, we limit to optimizing over a discounted finite horizon.

As we mentioned, besides the physical state the I-POMDP's interactive state contains a model for each other agent. The space of models, $M_{j,l-1} = \{\Theta_{j,l-1} \cup SM_j\}$, for $l \geq 1$, where $\Theta_{j,l-1}$ is the set of computable, intentional models ascribed to agent j: $\theta_{j,l-1} = \langle b_{j,l-1}, \hat{\theta}_j \rangle$, where $b_{j,l-1}$ is agent j's level $l-1$ belief, $b_{j,l-1} \in \triangle(IS_{j,l-1})$, and $\hat{\theta}_j := \langle A, T_j, \Omega_j, O_j, R_j, OC_j \rangle$, is j's frame. Here, j is assumed to be Bayes-rational. At level 0, $b_{j,0} \in \triangle(S)$ and a level-0 intentional model reduces to a POMDP. SM_j is the set of subintentional models of j, an example of which is a *finite state automaton*.

Analogous to POMDPs, an agent in a I-POMDP maintains a belief that is a distribution over the interactive states, $b_{0,l} \in \Delta(IS_{0,l})$. Solution of a finitely-nested I-POMDP is a policy that maps agent 0's beliefs to a distribution over actions. If agent 0's initial belief is given, then the policy can be equivalently redefined as a mapping from the agent's history of observations to a distribution over its actions. The optimal policy is obtained by unrolling a finite tree of all possible beliefs that the agent may have given all possible actions and observations. A value is then assigned to each belief using a value function, and the action(s) that leads to the highest value at each belief constitutes the optimal policy. Belief update to find the next beliefs and value function are both given below.

In its belief update, the agent updates its belief about the physical states as well as about the other agents' models based on an estimation of other agents' observations and how they update their models. If the models of N agents are intentional, and $b^t_{0,l}(is^t)$ denotes $Pr(s^t, m^t_{1,l-1}, m^t_{2,l-1}, \ldots, m^t_{N,l-1} | \omega^{t+1}_0, a^t_o, b^t_{0,l})$, the belief update $SE(b^t_{0,l}, a^t_0, \omega^{t+1}_0)$ to obtain $b^{t+1}_{0,l}$ is written as:

$$Pr(s^{t+1}, m^{t+1}_{1,l-1}, m^{t+1}_{2,l-1}, \ldots, m^{t+1}_{N,l-1} | \omega^{t+1}_0, a^t_o, b^t_{0,l}) = \sum_{is^t \in IS_{0,l}} b^t_{0,l}(is^t) \prod_{j=1}^{N} Pr(a_j | m^t_{j,l-1})$$

$$\times T_0(s^t, a^t_0, \mathbf{a}^t_{-0}, s^{t+1})\, O_0(s^{t+1}, a^t_0, \mathbf{a}^t_{-0}, \omega^{t+1}_0) \prod_{j=1}^{N} \Bigg(\sum_{\omega^{t+1}_j} O_j(s^{t+1}, a^t_j, \mathbf{a}^t_{-j}, \omega^{t+1}_j)$$

$$\times\ \tau(b^{t+1}_{j,l-1}, SE(b^t_{j,l-1}, a^t_j, \omega^{t+1}_j)) \Bigg) \tag{4}$$

Here, $\tau(b^{t+1}_{j,l-1}, SE(b^t_{j,l-1}, a^t_j, \omega^{t+1}_j))$ is 1 if $SE(b^t_{j,l-1}, a^t_j, \omega^{t+1}_j)$ results in $b^{t+1}_{j,l-1}$, otherwise it is 0. If a subset of agents' behaviors are correlated, the model space can be expanded to include models that predict the joint actions of these agents possibly using various correlation devices. The optimal policy may be obtained by assigning a value to each belief of agent 0:

$$V^h(b^t_{0,l}) = \max_{a^t_0 \in A_0} \Bigg[\sum_{is^t \in IS_{0,l}} b^t_{0,l}(is^t) ER_0(is^t, a^t_0) + \gamma \sum_{\omega^{t+1}_0 \in \Omega} \max_{\alpha^{h-1}} \Bigg\{ \sum_{is^t \in S} b^t_{0,l}(is^t) \sum_{is^{t+1} \in IS_{0,l}} \times \prod_{j=1}^{N} Pr(a_j | m^t_{j,l-1})\, T_0(s^t, a^t_0, \mathbf{a}^t_{-0}, s^{t+1}) \prod_{j=1}^{N} \Bigg(\sum_{\omega^{t+1}_j} O_j(s^{t+1}, a^t_j, \mathbf{a}^t_{-j}, \omega^{t+1}_j) \tau(b^{t+1}_{j,l-1}, SE(b^t_{j,l-1}, a^t_j, \omega^{t+1}_j)) \Bigg) \alpha^{h-1}(is^{t+1}) \Bigg\} \Bigg] \tag{5}$$

where $\alpha : IS_{0,l} \rightarrow \mathbb{R}$ is a vector of values of dimension $|IS_{0,l}|$.

In the context of N agents, interactive bounded policy iteration (Sonu & Doshi, 2015) generates good quality solutions for an agent interacting with 4 other agents (total of 5 agents) without considering any problem structure. To the best of our knowledge, this result illustrates the maximum scalability of I-POMDPs so far for $N > 2$ agents.

2.3 Action Graph Games

Building on graphical games (Kearns, Littman, & Singh, 2001), action graph games (AGG) (Jiang, Leyton-Brown, & Bhat, 2011) utilize problem structures such as action anonymity and context-specific independence to concisely represent the reward function of single shot complete-information games and to scalably solve for Nash equilibrium in the number of agents.

The independence is modeled using a directed action graph whose nodes are actions and an edge between two nodes indicates that the reward of an agent performing an action indicated by one node is affected by other agents performing action of the other node. Lack of edges between nodes encodes the context-specific independence where the context is the specific action. Action anonymity is useful when the action sets of agents overlap substantially. In AGGs, the rewards on performing a certain action depend on the count of agents performing each of its neighboring actions in the action graph. Significant savings are obtained because the space of the vectors of counts over the set of distinct actions called a *configuration*, is much smaller than the space of joint-action profiles; the latter is exponential in the number of agents. An AGG is formally defined using the tuple $\langle N, A, (\Psi, E), R \rangle$, where N is the number of agents, identified by an integer beginning at 0; A is the set of joint actions of all agents as defined previously; the pair (Ψ, E) defines the graph where the set of nodes is, $\Psi = \bigcup_{i=0}^{N} A_i$, each of which provides a context, and E is the set of edges between nodes. Finally, R maps each configuration over an agent's action and the actions (of others) in its neighborhood as determined by the graph, to a real number indicating the payoff to the agent.

3. Many-Agent Interactive POMDP

As the number of agents sharing the environment grows, the sizes of the joint action and joint model spaces increases exponentially. Therefore, the memory requirement for representing the transition, observation and reward functions grows exponentially as well as does the complexity of performing the belief update over the interactive states. We call this source of exponentially increasing complexity due to more agents as the *curse of many agents*. To facilitate an understanding of the challenges with N agents and experimentation, we introduce a pragmatic running example that also forms one of our evaluation domains.

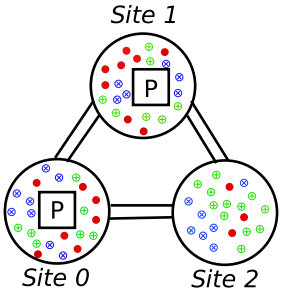

Figure 1: Protesters of different frames (colors) and police troops (denoted by P enclosed in a box) distributed across 3 sites in the policing protest domain. The state space of police decision making is factored into the protest intensity levels at the sites.

Example 1 (Policing Protest)**.** *Consider a policing scenario where police (agent* 0*) must maintain order in 3 geographically distributed and designated protest sites (labeled 0, 1, and 2) as shown in Fig. 1. A total population of* N *agents is protesting at these sites. A protestor may be peaceful or disruptive and this does not change over time. Police may dispatch one or two riot-control troops to either the same or different locations. Protests with differing intensities –* low*,* medium *and* high *– occur at each of the three sites. The goal of the police is to deescalate protests to the low intensity at each site. Protest intensity at any site is influenced by the number of protestors, how many are disruptive, and the number of police troops at that location. In the absence of adequate policing, we presume that the protest intensity escalates. On the other hand, two police troops at a location are adequate for deescalating protest of any intensity.*

We may model the decision making of the self-interested police in this quasi-adversarial situation using a finitely-nested I-POMDP. [2] However, both the problem representation and solution are challenged by the presence of a large number of other agents (protestors) in the context. Consequently, ways of reducing the complexity are needed. We begin by factoring the subject agent's belief over state and models, as shown in Section 3.1. This factorization paves the way to model and exploit the action anonymity of other agents (protestors), as shown in Section 3.2. Finally, we systematically introduce further efficiency by utilizing context-specific independence in Section 3.3. Coalesced together, these structures lead to the definition of the many-agent I-POMDP framework in Section 3.4.

3.1 Factored Belief and Update

As we mentioned previously, the subject agent in the I-POMDP framework maintains a belief over the physical state and joint models of other agents, $b_{0,l} \in \Delta(S \times \prod_{j=1}^{N} M_{j,l-1})$, where $\Delta(\cdot)$ is the space of probability distributions. For settings where N is large, the size of the interactive state space is exponentially larger, $|IS_{0,l}| = |S||M_{j,l-1}|^N$, and the belief representation becomes unwieldy. In fact, $N > 30$ would require many gigabytes of memory to store the belief in joint form. For problem domains involving thousands of agents such as the one described in Example 1,

2. The decision making in policing protests may be modeled differently. For example, it is tempting to model it as a large Stackelberg game (Fudenberg & Tirole, 1991). However, we do not view this problem as a leader-follower game; rather it is a simultaneous-move game where protester and troop movements co-occur.

it is intractable to represent beliefs in this form. However, the representation becomes manageable for large N if the belief is factored:

$$b_{0,l}(s, m_{1,l-1}, m_{2,l-1}, \ldots, m_{N,l-1}) = Pr(s)\ Pr(m_{1,l-1}, m_{2,l-1}, \ldots, m_{N,l-1}|s) \\ \approx Pr(s)\ Pr(m_{1,l-1}|s)\ Pr(m_{2,l-1}|s) \times \ldots \times Pr(m_{N,l-1}|s) \tag{6}$$

The factorization in the second line assumes conditional independence between models of different agents given the physical state. Consequently, beliefs that correlate behaviors of different agents due to which the belief over those agents' models cannot be factored, may not be directly represented. However, correlation could be alternately supported by introducing models that predict joint behaviors using say, a correlating device.

The memory consumed in storing a factored belief is $\mathcal{O}(|S| + N|S||M_j|^*)$, where $|M_j|^*$ is the largest size of the model space among all other agents. This is *linear* in the number of agents, and is significantly dominated by the exponentially-growing memory required to represent the belief as a joint distribution over the interactive state space, $\mathcal{O}(|S||M_j|^N)$.

Given agent 0's belief at time t, $b^t_{0,l}$, its action a^t_0 and the subsequent observation it makes ω^{t+1}_0, the updated belief at time step $t+1$, $b^{t+1}_{0,l}$, may be obtained as:

$$b^{t+1}_{0,l}(s^{t+1}, m^{t+1}_{1,l-1}, \ldots, m^{t+1}_{N,l-1}) = Pr(s^{t+1}, m^{t+1}_{1,l-1}, \ldots, m^{t+1}_{N,l-1}|b^t_{0,l}, a^t_0, \omega^{t+1}_0) \\ = Pr(s^{t+1}|b^t_{0,l}, a^t_0, \omega^{t+1}_0) Pr(m^{t+1}_{1,l-1}|s^{t+1}, m^{t+1}_{2,l-1}, \ldots, m^{t+1}_{N,l-1}, b^t_{0,l}, a^t_0, \omega^{t+1}_0) \\ \times \ldots \times Pr(m^{t+1}_{N,l-1}|s^{t+1}, b^t_{0,l}, a^t_0, \omega^{t+1}_0) \quad \text{(Chain rule)} \tag{7}$$

Each factor in the product of Eq. 7 may be obtained as follows. The update over the physical state is:

$$Pr(s^{t+1}|b^t_{0,l}, a^t_0, \omega^{t+1}_0) = \frac{1}{Pr(\omega^{t+1}_0|b^t_{0,l}, a^t_0)} Pr(s^{t+1}, \omega^{t+1}_0|b^t_{0,l}, a^t_0) \quad \text{(Bayes Rule)} \\ \propto Pr(s^{t+1}, \omega^{t+1}_0|b^t_{0,l}, a^t_0) \\ = \sum_{s^t} \sum_{\mathbf{m}^t_{-0,l-1}} \sum_{\mathbf{a}^t_{-0}} Pr(s^{t+1}, \omega^{t+1}_0, s^t, \mathbf{m}^t_{-0,l-1}, \mathbf{a}^t_{-0}|b^t_{0,l}, a^t_0) \\ = \sum_{s^t} b^t_{0,l}(s^t) \sum_{\mathbf{m}^t_{-0,l-1}} b^t_{0,l}(m^t_{1,l-1}|s^t) \times \ldots \times b^t_{0,l}(m^t_{N,l-1}|s^t) \times \sum_{\mathbf{a}^t_{-0}} Pr(a^t_1|m^t_{1,l-1}) \times \ldots \times \\ Pr(a^t_N|m^t_{N,l-1}) \times O_0(s^{t+1}, a^t_0, \mathbf{a}^t_{-0}, \omega^{t+1}_0)\ T_0(s^t, a^t_0, \mathbf{a}^t_{-0}, s^{t+1}) \tag{8}$$

where $\mathbf{m}^t_{-0,l-1}$ and $\mathbf{a}^t_{-0}$ denote the vectors of models and actions of agents other than 0, respectively. The update over the model of each other agent, $j = 1 \ldots N$, conditioned on the state at $t+1$ is:

$$Pr(m^{t+1}_{j,l-1}|s^{t+1}, m^{t+1}_{j+1,l-1}, \ldots, m^{t+1}_{N,l-1}, b^t_{0,l}, a^t_0, \omega^{t+1}_0) \\ = \sum_{s^t} b^t_{0,l}(s^t) \sum_{\mathbf{m}^t_{-0}} b^t_{0,l}(m^t_{1,l-1}|s^t)\ b^t_{0,l}(m^t_{2,l-1}|s^t) \times \ldots \times b^t_{0,l}(m^t_{N,l-1}|s^t) \sum_{a^t_j} \sum_{\mathbf{a}^t_{-j}} Pr(a^t_1|m^t_{1,l-1}) \\ \times \ldots \times Pr(a^t_N|m^t_{N,l-1}) \sum_{\omega^{t+1}_j} O_j(s^{t+1}, a^t_j, \mathbf{a}^t_{-j}, \omega^{t+1}_j)\ Pr(m^{t+1}_{j,l-1}|m^t_{j,l-1}, a^t_j, \omega^{t+1}_j) \tag{9}$$

If the models are intentional, then $Pr(m^{t+1}_{j,l-1}|m^t_{j,l-1}, a^t_j, \omega^{t+1}_j)$ is equal to $\tau(b^{t+1}_{j,l-1}, SE(b^t_{j,l-1}, a^t_j, \omega^{t+1}_j))$ that appeared in Eq. 4. Derivations of Eqs. 8 and 9 are straightforward and not included here for brevity. In particular, note that models of agents other than j at $t+1$ do not impact j's model update in the absence of correlated behavior. Thus, under the assumption of a factored prior as in Eq. 6 and absence of model correlations, the I-POMDP belief update may be decomposed into an update of the physical state and the models of the N agents conditioned on the state.

3.2 Frame-Action Anonymity

As noted by Jiang et al. (2011), many noncooperative and cooperative problems exhibit the structure that rewards depend on the *number* of agents acting in particular ways rather than which agent is performing the act. This is particularly evident in Example 1 where the outcome of policing at any given site largely depends on the number of peaceful and disruptive protesters converging at that site. Building on this, we additionally observe that the transient state of the protests and observations of the police at a site are also largely influenced by the number of peaceful and disruptive protesters moving from one location to another. This is noted in the example below:

Example 2 (Frame-action anonymity of protesters)**.** *The transient state of protests reflecting the intensity of protests at each site depends on the previous intensity at a site and the* number *of peaceful and disruptive protesters entering the site. Police (noisily) observes the intensity of protest at each site which is again largely determined by the* number *of peaceful and disruptive protesters at a site. Finally, the outcome of policing at a site is contingent on whether the protest was largely peaceful or disruptive. Consequently, the identity of the individual protesters beyond their frame and action is disregarded.*

Here, the peaceful or disruptive nature of the protesters are captured by different *frames* of others in agent 0's I-POMDP, and this modeling may be extended to any number of frames. Frame-action anonymity is an important attribute of the above domain. We formally define it in the context of agent 0's transition, observation and reward functions next:

Definition 1 (Frame-action anonymity)**.** *Let $\mathbf{a}^p_{-0}$ be a joint action of all peaceful protesters and $\mathbf{a}^d_{-0}$ be a joint action of all disruptive ones. Let $\dot{\mathbf{a}}^p_{-0}$ and $\dot{\mathbf{a}}^d_{-0}$ be some permutations of the two joint action profiles, respectively. An I-POMDP models frame-action anonymity iff for any a_0, s, s', $\mathbf{a}^p_{-0}$ and $\mathbf{a}^d_{-0}$:*

$$T_0(s, a_0, \mathbf{a}^p_{-0}, \mathbf{a}^d_{-0}, s') = T_0(s, a_0, \dot{\mathbf{a}}^p_{-0}, \dot{\mathbf{a}}^d_{-0}, s'),$$

$$O_0(s', a_0, \mathbf{a}^p_{-0}, \mathbf{a}^d_{-0}, \omega_0) = O_0(s', a_0, \dot{\mathbf{a}}^p_{-0}, \dot{\mathbf{a}}^d_{-0}, \omega_0), \quad \textit{and}$$

$$R_0(s, a_0, \mathbf{a}^p_{-0}, \mathbf{a}^d_{-0}) = R_0(s, a_0, \dot{\mathbf{a}}^p_{-0}, \dot{\mathbf{a}}^d_{-0}) \quad \forall \dot{\mathbf{a}}^p_{-0}, \dot{\mathbf{a}}^d_{-0}.$$

Recall the definition of an action configuration, which we will denote by $\mathcal{C}$, as the vector of action counts across an agent population. A permutation of joint actions of others having the same frame, say $\dot{\mathbf{a}}^p_{-0}$, differently assigns actions from the same set of actions to individual agents with the frame. Despite this, the fact that the transition and observation probabilities, and the reward remains unchanged indicates that these probabilities and reward are conditionally independent of the identities

of the agents performing the actions given the configuration. Importantly, the action configuration of the joint action and its permutation stay the same: $\mathcal{C}(\mathbf{a}^p_{-0}) = \mathcal{C}(\dot{\mathbf{a}}^p_{-0})$. This combined with Definition 1 allows redefining the transition, observation and reward functions to be over configurations as: $T_0(s, a_0, \mathcal{C}(\mathbf{a}^p_{-0}), \mathcal{C}(\mathbf{a}^d_{-0}), s')$, $O_0(s', a_0, \mathcal{C}(\mathbf{a}^p_{-0}), \mathcal{C}(\mathbf{a}^d_{-0}), \omega)$, and $R_0(s, a_0, \mathcal{C}(\mathbf{a}^p_{-0}), \mathcal{C}(\mathbf{a}^d_{-0}))$.

Let $A^p_1, \ldots, A^p_n$ be the sets of actions of n peaceful protesters, and A^p_{-0} is the Cartesian product of these sets. Let $\mathcal{C}(A^p_{-0})$ be the set of all action configurations for A^p_{-0}. *Observe that multiple joint actions from A^p_{-0} may result in a single configuration; these joint actions are configuration equivalent.* Consequently, the equivalence partitions the joint action set A^p_{-0} into $|\mathcal{C}(A^p_{-0})|$ classes. Furthermore, when other agents of same frame have overlapping sets of actions, the number of configurations could get much smaller than the number of joint actions. Consequently, definitions of the transition, observation and reward functions involving configurations would be more compact. Of course, as we identify actions with frames, the scenario gradually loses anonymity with more frames. Thus, an implicit assumption, which is also realistic, is that there are far fewer frames than the numbers of agents. For example, protestors in the policing protest example exhibit two frames.

For the sake of convenience, let $\mathcal{C}$ be a tuple representing configurations over the actions performed by agents of all frames. Then, we may write the transition, observation, and reward functions as $T_0(s, a_0, \mathcal{C}, s')$, $O_0(s', a_0, \mathcal{C}, \omega)$ and $R_0(s, a_0, \mathcal{C})$, respectively.

3.3 Frame-Action Hypergraphs

In addition to frame-action anonymity, domains involving agent populations often exhibit another form of structure – context-specific independences. This is a broad category and includes the context-specific independence found in conditional probability tables of Bayesian networks (Boutilier et al., 1996) and in action-graph games. It offers significant additional structure for computational tractability. We begin by illustrating this in the context provided by Example 1.

Example 3 (Context-specific independence in policing)**.** *At a protest site, payoff for policing is independent of the movement of the protesters to other sites. Similarly, the transient intensity of the protest at a site given the level of policing at the site as context is independent of the movement of protesters between other sites.*

The context-specific independences above build on the similar independence modeled in action graphs in two ways: (i) We model such partial independence in the transitions of factored states and in the observation function as well, in addition to the reward function. (ii) We allow the context-specific independence to be mediated by the frames of other agents in addition to their actions. For example, the reward from passive policing at a site is independent of the number of *peaceful* protesters, instead influenced primarily by the number of *disruptive* protesters.

These differences imply that the context has expanded to include nodes for transitions (when independence is used in transition function) and other agent's frame. Therefore, action graphs are no longer sufficient because each edge has more than two nodes. We generalize the action graphs into *frame-action hypergraphs*, and specifically 3-uniform hypergraphs where each edge is a set of 3 nodes. We formally define it below:

Definition 2 (Frame-action hypergraph)**.** *A frame-action hypergraph for agent 0 is a 3-uniform hypergraph $\mathcal{G} = \langle \Psi, A_{-0}, \hat{\Theta}_{-0}, E \rangle$, where Ψ is a set of nodes that represent the context, A_{-0} is a set of action nodes with each node representing an action that any other agent may take; $\hat{\Theta}_{-0}$ is a*

set of frame nodes, each node representing a frame ascribed to an agent, and E is a set of 3-uniform hyperedges where each hyperedge contains one node from each set Ψ, A_{-0}, and $\hat{\Theta}_{-0}$, respectively.

Both context and action nodes differ based on whether the hypergraph applies to the transition, observation or reward functions:

- For the transition function, the context is a *set* containing a pair of states between which a transition may occur and an action of agent 0, $\Psi = S \times A_0 \times S$, and the action nodes include actions of all other agents, $A_{-0} = \bigcup_{j=1}^{N} A_j$. Neighbors of a context node $\psi = \langle s, a_0, s' \rangle$ are all the frame-action pairs that affect the probability of the transition. An edge $(\langle s, a_0, s' \rangle, a_{-0}, \hat{\theta})$ indicates that the probability of transitioning from s to s' on performing a_0 is affected (in part) by the number of other agents of frame $\hat{\theta}$ performing the particular action in a_{-0}.

- The context for agent 0's observation function is the set of state-action-observation triplets, $\Psi = S \times A_0 \times \Omega_0$, and the action nodes are identical to those in the transition function. Neighbors of a context node, $\langle s, a_0, \omega_0 \rangle$, are all those frame-action pairs that affect the observation probability. Specifically, an edge $(\langle s, a_0, \omega_0 \rangle, a_{-0}, \hat{\theta})$ indicates that the probability of observing ω_0 from state s on performing a_0 is affected (in part) by the number of other agents performing action a_{-0} who possess frame $\hat{\theta}$.

- For agent 0's reward function, the context is the set of pairs of state and action of agent 0, $\Psi = S \times A_0$, and the action nodes are the same as those in transition and observation functions. An edge $(\langle s, a_0 \rangle, a_{-0}, \hat{\theta}_{-0})$ in this hypergraph indicates that the reward for agent 0 on performing action a_0 at state s is affected (in part) by the agents of frame $\hat{\theta}_{-0}$ who perform action a_{-0}.

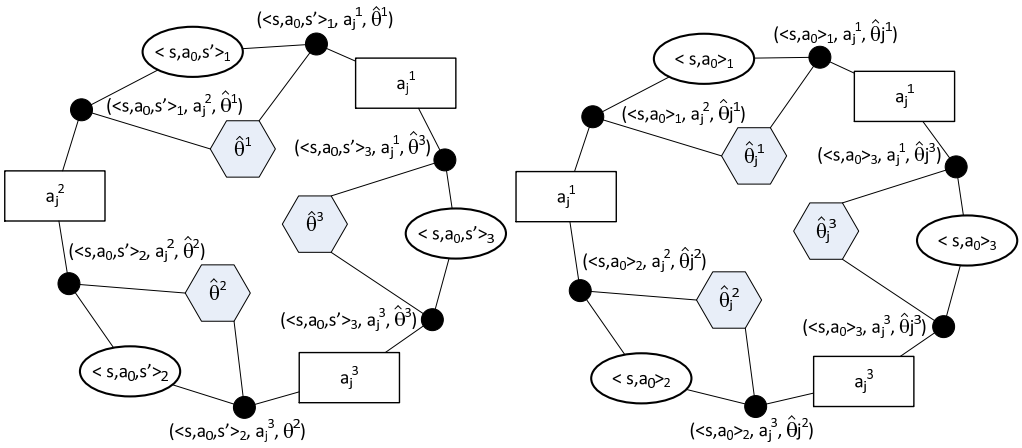

Figure 2: Levi (incidence) graph representation of a generic frame-action hypergraph for (a) the transition function, and (b) the reward function. The shaded black nodes represent edges in the hypergraph. Each edge has the context, ψ, denoted in bold, agent's action, a, and its frame, $\hat{\theta}$, incident on it. For example, the reward for a state and agent 0's action, $\langle s, a_0 \rangle_1$ is affected by others' actions a_j^1 and a_j^2 performed by any other agent of frame $\hat{\theta}_j^1$ only.

We illustrate a schematic frame-action hypergraph for context-specific independence in a transition function and in a reward function as Levi graphs in Figs. $2(a)$ and (b), respectively. We point out that the hypergraph for the reward function comes closest in semantics to the graph in action graph games (Jiang et al., 2011) although the former adds the state to the context as well as nodes for frames. Hypergraphs for the transition and observation functions differ substantially in semantics and form from action graphs.

To use these hypergraphs in our algorithms, we first define the general *frame-action neighborhood* of a context node.

Definition 3 (Frame-action neighborhood)**.** *The frame-action neighborhood of a context node $\psi \in \Psi$, $\nu(\psi)$, given a frame-action hypergraph $\mathcal{G}$ is defined as a subset of $A_{-0} \times \hat{\Theta}$ such that $\nu(\psi) = \{(a_{-0}, \hat{\theta}) | a_{-0} \in A_{-0}, \hat{\theta} \in \hat{\Theta}, (\psi, a_{-0}, \hat{\theta}) \in E\}$.*

As an example from Fig. $2(b)$, the frame-action neighborhood of a state-action pair, $\langle s, a_0 \rangle$ in a hypergraph for the reward function is the set of all action and frame nodes incident on each hyperedge anchored by the node $\langle s, a_0 \rangle$.

We move toward integrating frame-action anonymity introduced in the previous subsection with the context-specific independence as modeled above by introducing frame-action configurations.

Definition 4 (Frame-action configuration)**.** *A configuration over the frame-action neighborhood of a context node, ψ, given a frame-action hypergraph is a vector,*

$$\mathcal{C}^{\nu(\psi)} \triangleq \langle\, \mathcal{C}(\mathbf{a}_{-0}^{\hat{\theta}_1}), \mathcal{C}(\mathbf{a}_{-0}^{\hat{\theta}_2}), \ldots, \mathcal{C}(\mathbf{a}_{-0}^{\hat{\theta}_{|\hat{\Theta}|}}), \mathcal{C}(\phi) \,\rangle$$

where each a included in $\mathbf{a}_{-0}^{\hat{\theta}}$ is an action in $\nu(\psi)$ with frame $\hat{\theta}$, and $\mathcal{C}(\mathbf{a}_{-0}^{\hat{\theta}})$ is a configuration over actions by agents other than 0 whose frame is $\hat{\theta}$. All agents with frames other than those in the frame-action neighborhood are assumed to perform a dummy action, ϕ.

Definition 4 allows further inroads into compacting the transition, observation and reward functions of the I-POMDP using context-specific independence. Specifically, we may redefine these functions one more time (see previous redefinition in the prior subsection) to limit the configurations only over the frame-action neighborhood of the context as, $T_0(s, a_0, \mathcal{C}^{\nu(s,a_0,s')}, s')$, $O_0(s', a_0, \mathcal{C}^{\nu(s',a_0,\omega_0)}, \omega_0)$ and $R_0(s, a_0, \mathcal{C}^{\nu(s,a_0)})$. [3]

3.4 I-POMDP with Anonymity and Context-Specific Independence

In order to benefit from structures of anonymity and context-specific independence, we switch to a factored representation of the state space and redefine I-POMDP for agent 0 as follows:

$$\text{I-POMDP}_{0,l} = \langle IS_{0,l}, A, \Omega_0, \mathcal{T}_0, \mathcal{O}_0, \mathcal{R}_0, OC_0 \rangle$$

where:

- $IS_{0,l}$, A, Ω_0 and OC_0 remain the same as before except that the physical states are factored as, $S = \prod_{k=1}^{K} X_k$, where X_k is a set of values of the k^{th} state variable, and let x_k denote a value.

3. Context in our transition function is $\langle s, a_0, s' \rangle$ compared with the context of just $\langle s, a_0 \rangle$ in Varakantham et al's (2014) transition function.

- $\mathcal{T}_0$ is the transition function, $\mathcal{T}_0(x, a_0, \mathcal{C}^{\nu(x,a_0,x')}, x')$ where $\mathcal{C}^{\nu(x,a_0,x')}$ is the configuration over the frame-action neighborhood of context $\langle x, a_0, x' \rangle$ obtained from a hypergraph that holds for the transition function.

  This transition function is significantly more compact than the original that occupies space $\mathcal{O}(|X|^2|A_0||A_j|^N)$ compared to the $\mathcal{O}(|X|^2|A_0|(\frac{N}{|\nu^*|})^{|\nu^*|})$ of $\mathcal{T}_0$, where the fraction is the complexity of $\binom{N+|\nu^*|+1}{|\nu^*|+1}$; $|\nu^*|$ is the cardinality of the largest neighborhood of any context, and $(\frac{N}{|\nu^*|})^{|\nu^*|} \ll |A_j|^N$. The value $\binom{N+|\nu^*|+1}{|\nu^*|+1}$ is obtained from combinatorial compositions and represents the number of ways $|\nu^*| + 1$ non-negative values can be weakly composed such that their sum is N.

- The redefined observation function is $\mathcal{O}_0(x', a_0, \mathcal{C}^{\nu(x',a_0,\omega_0)}, \omega_0)$ where $\mathcal{C}^{\nu(x',a_0,\omega_0)}$ is the configuration over the frame-action neighborhood of context $\langle x', a_0, \omega_0 \rangle$ obtained from a hypergraph that holds for the observation function. Analogously to the transition function, the original observation function consumes space $\mathcal{O}(|X||\Omega||A_0||A_j|^N)$, which is much larger than space $\mathcal{O}(|X||\Omega||A_0|(\frac{N}{|\nu^*|})^{|\nu^*|})$ occupied by this redefinition.

- $\mathcal{R}_0$ is the reward function defined as $\mathcal{R}_0(x, a_0, \mathcal{C}^{\nu(x,a_0)})$ where $\mathcal{C}^{\nu(x,a_0)}$ is defined analogously to the configurations in the previous parameters. The reward for a state and actions may simply be the sum of local rewards for the state factors and actions (or a more general function if needed). As with the transition and observation functions, this reward function is compact occupying space $\mathcal{O}(|X||A_0|(\frac{N}{|\nu^*|})^{|\nu^*|})$ that is much less than $\mathcal{O}(|X||A_0||A_{-0}|^N)$ of the original.

3.4.1 BELIEF UPDATE WITH ANONYMITY AND CONTEXT-SPECIFIC INDEPENDENCE

For this extended I-POMDP, we compute the updated belief over a physical state as a product of its factors using Eq. 12 and belief update over the models of each other agent using Eq. 13 as shown below:

$$
\begin{aligned}
&Pr(\mathbf{s}^{t+1}|b^t_{0,l}, a^t_0, \omega^{t+1}_0) \propto Pr(\mathbf{s}^{t+1}, \omega^{t+1}_0|b^t_{0,l}, a^t_0) \\
&= Pr(\omega^{t+1}_0|\mathbf{s}^{t+1}, b^t_{0,l}, a^t_0)\ Pr(\mathbf{s}^{t+1}|b^t_{0,l}, a^t_0) \\
&= Pr(\langle \omega^{t+1}_{0,1}, \omega^{t+1}_{0,2}, \ldots, \omega^{t+1}_{0,k} \rangle | \langle x^{t+1}_1, x^{t+1}_2, \ldots, x^{t+1}_k \rangle, b^t_{0,l}, a^t_0)\ Pr(\langle x^{t+1}_1, x^{t+1}_2, \ldots, x^{t+1}_k \rangle | b^t_{0,l}, a^t_0) \\
&= \left\{ \prod_{k=1}^{K} Pr(\omega^{t+1}_{0,k} | \langle x^{t+1}_1, x^{t+1}_2, \ldots, x^{t+1}_k \rangle, b^t_{0,l}, a^t_0) \right\} \times \left\{ \prod_{k=1}^{K} Pr(x^{t+1}_k | b^t_{0,l}, a^t_0) \right\} \qquad (10) \\
&= \left\{ \prod_{k=1}^{K} Pr(\omega^{t+1}_{0,k} | x^{t+1}_k, b^t_{0,l}, a^t_0) \right\} \times \left\{ \prod_{k=1}^{K} Pr(x^{t+1}_k | b^t_{0,l}, a^t_0) \right\} \qquad (11)
\end{aligned}
$$

Without loss of generality, let the observation be decomposed into factors. For simplicity, we assume that there as many factors as the state factors. Equation 10 results from supposing that both the observation and state factors are independent. Additionally, we suppose that an observation factor $\omega_{0,k}$ is conditionally independent of state factors other than x_k given the latter, leading to Eq. 11. We discuss the feasibility of these suppositions in the context of our test problem domains and more generally in practice, in Section 6. Next, we seek to replace the belief $b^t_{0,l}$ in the condition

of each probability term in Eq. 11 with the state using marginalization. This step allows us to proceed toward including the transition and observation functions in the factored belief update. Thus, $Pr(\mathbf{s}^{t+1}|b^t_{0,l}, a^t_0, \omega^{t+1}_0)$ becomes

$$= \left\{ \prod_{k=1}^{K} \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) Pr(\omega^{t+1}_{0,k}|x^{t+1}_k, b^t_{0,l}, a^t_0, \mathbf{s}^t) \right\} \times \left\{ \prod_{k=1}^{K} \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) Pr(x^{t+1}_k|b^t_{0,l}, a^t_0, x^t) \right\}$$

$$= \left\{ \prod_{k=1}^{K} \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{\mathbf{m}^t_{-0}} b^t_{0,l}(\mathbf{m}^t_{-0}|\mathbf{s}^t) \sum_{\mathbf{a}^t_{-0}} Pr(\mathbf{a}^t_{-0}|\mathbf{m}^t_{-0}) Pr(\omega^{t+1}_{0,k}|x^{t+1}_k, a^t_0, \mathbf{a}^t_{-0}) \right\} \times$$

$$\left\{ \prod_{k=1}^{K} \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{\mathbf{m}^t_{-0}} b^t_{0,l}(\mathbf{m}^t_{-0}|\mathbf{s}^t) \sum_{\mathbf{a}^t_{-0}} Pr(\mathbf{a}^t_{-0}|\mathbf{m}^t_{-0}) Pr(x^{t+1}_k|x^t, a^t_0, \mathbf{a}^t_{-0}) \right\}$$

Let us introduce a projection function $\delta^{\nu(\psi)}$ that maps joint actions of others to the corresponding frame-action configurations as defined in Def. 4. Formally $\delta^{\nu(\psi)} : A_{-0} \times \hat{\Theta} \rightarrow \mathbf{C}^{\nu(\psi)}$, where $\mathbf{C}^{\nu(\psi)}$ is the set of all possible frame-action configurations. Notice that $\delta^{\nu(\psi)}$ is a many-one map because there may be many joint action vectors that map to the same frame-action configuration due to anonymity; these joints are configuration-equivalent as mentioned previously. Thus, $\delta^{\nu(\psi)}$ induces an equivalence partition of A_{-0}. An equivalence set for a configuration $\mathcal{C}^{\nu(\psi)} \in \mathbf{C}^{\nu(\psi)}$ is $\{A^{\hat{\theta}_1}_{-0}, A^{\hat{\theta}_2}_{-0}, \ldots, A^{\hat{\theta}_{|\hat{\Theta}|}}_{-0}\}$, where $A^{\hat{\theta}_1}_{-0}$ for instance is a set of joint actions of all other agents with frame $\hat{\theta}_1$ that induce the same $\mathcal{C}(\mathbf{a}^{\hat{\theta}_1}_{-0})$. Therefore, for each $\mathbf{a}_{-0}$ in $A^{\hat{\theta}_1}_{-0} \times A^{\hat{\theta}_2}_{-0} \times \ldots \times A^{\hat{\theta}_{|\hat{\Theta}|}}_{-0}$, $T_0(x^t_k, a^t_0, \mathbf{a}^t_{-0}, x^{t+1}_k) = \mathcal{T}_0(x^t_k, a^t_0, \mathcal{C}^{\nu(x^t_k, a^t_0, x^{t+1}_k)}, x^{t+1}_k)$, and analogously $O_0(x^{t+1}_k, a^t_0, \mathbf{a}^t_{-0}, \omega^{t+1}_{0,k}) = \mathcal{O}_0(x^{t+1}_k, a^t_0, \mathcal{C}^{\nu(x^{t+1}_k, a^t_0, \omega^{t+1}_{0,k})}, \omega^{t+1}_{0,k})$. Previous equation for the belief update becomes, $Pr(\mathbf{s}^{t+1}|b^t_{0,l}, a^t_0, \omega^{t+1}_0) \propto$

$$\left\{ \prod_{k=1}^{K} \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{c=1}^{|\mathbf{C}^{\nu(x^{t+1}_k, a^t_0, \omega^{t+1}_{0,k})}|} \sum_{\mathbf{m}^t_{-0}} b^t_{0,l}(\mathbf{m}^t_{-0}|\mathbf{s}^t) \sum_{\mathbf{a}^t_{-0} \in \mathbf{A}^c_{-0}} Pr(\mathbf{a}^t_{-0}|\mathbf{m}^t_{-0}) \; Pr(\mathcal{C}^{\nu(x^{t+1}_k, a^t_0, \omega^{t+1}_0)}| \right.$$

$$\left. x^{t+1}_k, a^t_0) \; \mathcal{O}_0(x^{t+1}_k, a^t_0, \mathcal{C}^{\nu(x^{t+1}_k, a^t_0, \omega^{t+1}_0)}, \omega^{t+1}_{0,k}) \right\} \times \left\{ \prod_{k=1}^{K} \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{c=1}^{|\mathbf{C}^{\nu(x^t_k, a^t_0, x^{t+1}_k)}|} \sum_{\mathbf{m}^t_{-0}} b^t_{0,l}(\mathbf{m}^t_{-0}|\mathbf{s}^t) \right.$$

$$\left. \times \sum_{\mathbf{a}^t_{-0} \in \mathbf{A}^c_{-0}} Pr(\mathbf{a}^t_{-0}|\mathbf{m}^t_{-0}) \; Pr(\mathcal{C}^{\nu(x^t_k, a^t_0, x^{t+1}_k)}|x^t_k, a^t_0) \; \mathcal{T}_0(x^t_k, a^t_0, \mathcal{C}^{\nu(x^t_k, a^t_0, x^{t+1}_k)}, x^{t+1}_k) \right\}$$

where $\mathbf{A}^c_{-0}$ denotes the equivalence set $A^{\hat{\theta}_1}_{-0} \times A^{\hat{\theta}_2}_{-0} \times \ldots \times A^{\hat{\theta}_{|\hat{\Theta}|}}_{-0}$. By construction of $\mathbf{A}^c_{-0}$, the observation and transition probabilities remain the same for every $\mathbf{a}^t_{-0} \in \mathbf{A}^c_{-0}$. Furthermore, we may compute the probability of a given configuration using the probability of the models of other agents

and the probability of their actions given the models. Therefore, we may rewrite the belief update as:

$$Pr(\mathbf{s}^{t+1}|b^t_{0,l}, a^t_0, \omega^{t+1}_0)$$
$$\propto \Bigg\{ \prod_{k=1}^{K} \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{\mathcal{C}^{\nu(x_k^{t+1}, a_0^t, \omega_0^{t+1})}} Pr(\mathcal{C}^{\nu(x_k^{t+1}, a_0^t, \omega_0^{t+1})} | b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))$$
$$\mathcal{O}_0(x^{t+1}_k, a^t_0, \mathcal{C}^{\nu(x_k^{t+1}, a_0^t, \omega_0^{t+1})}, \omega^{t+1}_0) \Bigg\} \times \Bigg\{ \prod_{k=1}^{K} \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{\mathcal{C}^{\nu(x_k^t, a_0^t, x_k^{t+1})}} Pr(\mathcal{C}^{\nu(x_k^t, a_0^t, x_k^{t+1})} |$$
$$b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))\ \mathcal{T}_0(x^t_k, a^t_0, \mathcal{C}^{\nu(x_k^t, a_0^t, x_k^{t+1})}, x^{t+1}_k) \Bigg\} \tag{12}$$

Here, $Pr(\mathcal{C}^{\nu(x_k^{t+1}, a_0^t, \omega_0^{t+1})}|b^t_{0,l}(M_{1,l-1}|s^t), \ldots, b^t_{0,l}(M_{N,l-1}|s^t))$ is the probability of a frame-action configuration that is context specific to the triplet, $\langle x^{t+1}_k, a_0, \omega^{t+1}_{0,k} \rangle$. It is computed using the factored belief distributions over the models of each other agent using a dynamic program as outlined later. A benefit of the belief factorization is that the program takes as input just N beliefs each of size $|M_j|$ compared to a single large belief of exponential size $|M_j|^N$.

Analogously, the factored belief update over the models of each other agent $j = 1 \ldots N$ conditioned on the state at $t+1$ as previously shown in Eq. 9 now becomes:

$$Pr(m^{t+1}_{j,l-1}|s^{t+1}, b^t_{0,l}, a^t_0)$$
$$= \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{m^t_j} b^t_{0,l}(m^t_j|s^t) \sum_{a^t_j} Pr(a^t_j|m^t_j) \sum_{\omega^{t+1}_j} \Bigg\{ \prod_{k=1}^{K} \sum_{\mathcal{C}^{\nu(x_k^{t+1}, a_j^t, \omega_{j,k}^{t+1})}} Pr(\mathcal{C}^{\nu(x_k^{t+1}, a_j^t, \omega_{j,k}^{t+1})} |$$
$$a^t_0, b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{j-1,l-1}|\mathbf{s}^t), b^t_{0,l}(M_{j+1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))$$
$$\mathcal{O}_j(x^{t+1}_k, a^t_j, \mathcal{C}^{\nu(x_k^{t+1}, a_j^t, \omega_{j,k}^{t+1})}, \omega^{t+1}_{j,k}) \Bigg\}\ Pr(m^{t+1}_j|m^t_j, a^t_j, \omega^{t+1}_j) \tag{13}$$

The probability of configuration given action a^t_0 and beliefs over models of all others except j is computed as mentioned previously using dynamic programming.

3.4.2 VALUE FUNCTION

The finite-horizon value function of the many-agent I-POMDP continues to be the sum of agent 0's immediate reward and the discounted expected reward over the future:

$$V^h(b^t_{0,l}) = \max_{a^t_0 \in A_0} ER_0(b^t_{0,l}, a^t_0) + \gamma \sum_{\omega^{t+1}_0} Pr(\omega^{t+1}_0|b^t_{0,l}, a^t_0)\ V^{h-1}(b^{t+1}_{0,l}) \tag{14}$$

where $ER_0(b^t_{0,l}, a^t_0)$ is the expected immediate reward of agent 0 and γ is the discount factor. In the context of the redefined reward function of the I-POMDP framework in this section, the expected

immediate reward is obtained as:

$$ER_0(b^t_{0,l}, a^t_0) = \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t)\Bigg(\sum_{k=1}^{K} \sum_{\mathcal{C}^{\nu(x^t_k,a^t_0)}} Pr(\mathcal{C}^{\nu(x^t_k,a^t_0)}|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t),\ldots,b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t)) \\ R_0(x^t_k, a^t_0, \mathcal{C}^{\nu(x^t_k,a^t_0)})\Bigg) \tag{15}$$

where the inner sum is over all the state factors, $\mathbf{s}^t = \langle x^t_1, \ldots, x^t_K\rangle$, and the term $Pr(\mathcal{C}^{\nu(x^t_k,a^t_0)}|$ $b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t),\ldots,b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))$ denotes the probability of a frame-action configuration that is context-specific to the factor x^t_k and action a^t_0.

Notice that Eq. 14 involves updating agent 0's belief $b^t_{0,l}$ to obtain $b^{t+1}_{0,l}$. This is accomplished using the belief update derived in Section 3.4.1, which relied on the property of frame-action anonymity (Definition 1) and the resulting structure (Definitions 3 and 4), as well as context-specific independence as modeled in frame-action hypergraphs (Definition 2). Importantly, Proposition 1 establishes that the Bellman equation above is exact.

Proposition 1 (Optimality)**.** *The Bellman equation in* (14) *provides an exact computation of the value function for the many-agent* I-POMDP *under the conditions of Definitions 1 and 4.*

Proof. The proof of Proposition 1 proceeds by induction on the horizon. For base case of horizon 1 the value function may be written as:

$$V^1(b^t_{0,l}) = \max_{a^t_0} ER_0(b^t_{0,l}, a^t_0)$$

$$= \max_{a^t_0} \sum_{\mathbf{s}^t,\mathbf{m}^t_{-0},\mathbf{a}^t_{-0}} Pr(\mathbf{s}^t, \mathbf{m}^t_{-0}, \mathbf{a}^t_{-0}) \sum_{k=1}^{K} R_0(x^t_k, a^t_0, \mathbf{a}^t_{-0})$$

$$= \max_{a^t_0} \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{\mathbf{m}^t_{-0},\mathbf{a}^t_{-0}} Pr(\mathbf{m}^t_{-0}, \mathbf{a}^t_{-0}|\mathbf{s}^t) \sum_{k=1}^{K} R_0(x^t_k, a^t_0, \mathbf{a}^t_{-0})$$

$$= \max_{a^t_0} \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{k=1}^{K} \sum_{\mathbf{m}^t_{-0},\mathbf{a}^t_{-0}} Pr(\mathbf{m}^t_{-0}, \mathbf{a}^t_{-0}|\mathbf{s}^t)\, R_0(x^t_k, a^t_0, \mathbf{a}^t_{-0})$$

where $\mathbf{s}^t = \langle x^t_1, x^t_2, \ldots, x^t_K\rangle$. Frame-action anonymity and context-specific independence as stated in Definitions 1 and 4 allows us to introduce configurations. Thus, $V^1(b^t_{0,l})$ becomes

$$= \max_{a^t_0} \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \Bigg\{ \sum_{k=1}^{K} \sum_{\mathcal{C}^{\nu(x^t_k,a^t_0)}} \sum_{\mathbf{m}^t_{-0},\mathbf{a}^t_{-0}} Pr(\mathcal{C}^{\nu(x^t_k,a^t_0)}, \mathbf{m}^t_{-0}, \mathbf{a}^t_{-0}|\mathbf{s}^t)\, R_0(x^t_k, a^t_0, \mathcal{C}^{\nu(x^t_k,a^t_0)}) \Bigg\}$$

$$= \max_{a^t_0} \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \Bigg\{ \sum_{k=1}^{K} \sum_{\mathcal{C}^{\nu(x^t_k,a^t_0)}} Pr(\mathcal{C}^{\nu(x^t_k,a^t_0)}|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t),\ldots,b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t)) \\ \times R_0(x^t_k, a^t_0, \mathcal{C}^{\nu(x^t_k,a^t_0)}) \Bigg\}$$

As we mentioned previously, the probability of a configuration can be computed using the conditional belief over other agents' models.

As the inductive hypothesis, let the value function be exact for horizon $h-1$. Then, from Eq. 14 the horizon h value function is composed as

$$V^h(b^t_{0,l}) = \max_{a^t_0 \in A^t_0} ER_0(b^t_{0,l}, a^t_0) + \gamma \sum_{\omega^{t+1}_0} Pr(\omega^{t+1}_0 | b^t_{0,l}, a^t_0) V^{h-1}(b^{t+1}_{0,l})$$

The first term is the horizon 1 value function which is exact from the base case. Term $V^{h-1}(b^{t+1}_{0,l})$ is exact by inductive hypothesis. Finally, the term $Pr(\omega^{t+1}_0 | b^t_{0,l}, a^t_0)$ is merely the normalization constant for the belief update. Equations 12 and 13 that constitute the belief update for the many-agent I-POMDP are derived from first principles under the condition that the observation and state factors are independent. Hence, $V^h(b^t_{0,l})$ is exact. □

4. Algorithm

In this section, we present our simple method for solving the many-agent I-POMDP defined previously. The section that follows improves on this initial method. To assist in understanding the remaining sections better, we recapitulate key notation introduced so far and some new notation in Appendix A.

4.1 Look-Ahead Tree for Solution

We utilize a straightforward method for solving the many-agent I-POMDP given an initial belief: each other agent is modeled using a finite-state controller which is included as part of the interactive state space. A tree consisting of beliefs as nodes, which can be reached is projected for as many steps as the horizon using Eqs. 12 and 13. Value iteration (Eq. 14) is then performed on the tree in a bottom up manner to obtain the exact value at the root node.

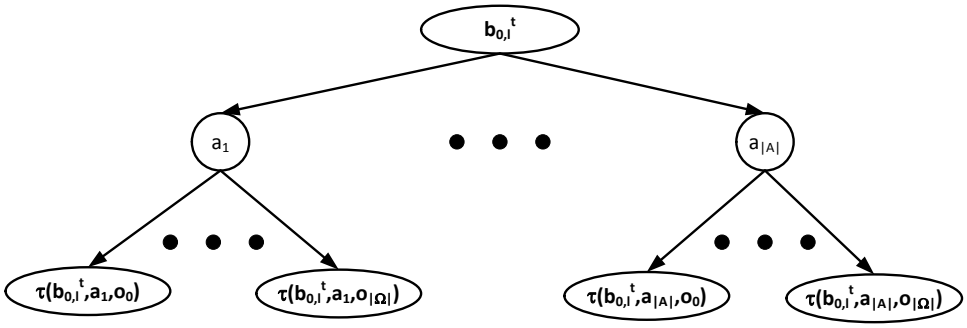

Figure 3: Computing beliefs reachable from a given belief in one step. A belief is obtained for each action-observation pair $\langle a^t_0, \omega^{t+1}_0 \rangle$.

Figure 3 illustrates the computation of beliefs reachable from initial belief $b^t_{0,l}$ in one step. For each action a^t_0 and observation ω^{t+1}_0, the updated belief $\tau(b^t_{0,l}, a^t_0, \omega^{t+1}_0)$ is obtained. A horizon h reachability tree is projected by computing 1 step reachability for the root node and all inner

nodes. Note that the number of beliefs in the reachability tree is $\mathcal{O}((|A_0||\Omega_0|)^h)$ which affects the computational complexity exponentially; this is the well-known curse of history. As such, it is easy to see that this solution method does not scale with horizon.

Equations 12, 13 and 15, which form key steps in the value iteration, utilize distributions over frame-action configurations. Next, we present an algorithm that computes this distribution using the conditional belief over the models of each other agent, which gives the probability of their action(s) given the model.

4.2 Computing Distribution Over Frame-Action Configurations

Algorithm 1 generalizes the dynamic programming by Jiang and Lleyton-Brown (2011) for computing configurations over actions given mixed strategies of other agents to include frames and conditional beliefs over models of other agents. It computes the probability distribution of configurations over the frame-action neighborhood of a context given the belief over the agents' models. Specifically, the algorithm allows us to obtain $Pr(\mathcal{C}^{\nu(x_k^{t+1},a_0,\omega_{0,k}^{t+1})}|b_{0,l}^t(M_{1,l-1}|\mathbf{s}^t),\ldots,b_{0,l}^t(M_{N,l-1}|\mathbf{s}^t))$ and $Pr(\mathcal{C}^{\nu(x_k^t,a_0,x_k^{t+1})}|b_{0,l}^t(M_{1,l-1}|\mathbf{s}^t),\ldots,b_{0,l}^t(M_{N,l-1}|\mathbf{s}^t))$ in Eq. 8, $Pr(\mathcal{C}^{\nu(x_k^{t+1},a_j^t,\omega_{j,k}^{t+1})}|b_{0,l}^t(M_{1,l-1}|\mathbf{s}^t),$ $\ldots, b_{0,l}^t(M_{j-1,l-1}|\mathbf{s}^t), b_{0,l}^t(M_{j+1,l-1}|\mathbf{s}^t),\ldots b_{0,l}^t(M_{N,l-1}|\mathbf{s}^t))$ in Eq. 9, and $Pr(\mathcal{C}^{\nu(x_k^t,a_0)}|$ $b_{0,l}^t(M_{1,l-1}|\mathbf{s}^t),\ldots,b_{0,l}^t(M_{N,l-1}|\mathbf{s}^t))$ in Eq. 15.

Algorithm 1 Computing $Pr(\mathcal{C}^{\nu(\cdot)}|b_{0,l}(M_{1,l-1}|s),\ldots,b_{0,l}(M_{N,l-1}|s))$

Input: $\nu(\cdot)$, $\langle b_{0,l}(M_{1,l-1}|s),\ldots,b_{0,l}(M_{N,l-1}|s)\rangle$
Output: Trie P_n representing distribution over frame-action configurations in $\nu(\cdot)$

Initialize $c_0 = (0,\cdots,0)$ one value for each frame-action pair in $\nu(\cdot)$ and one for dummy action ϕ representing all other actions. Insert it into an empty trie P_0
Initialize $P_0[c_0] \leftarrow 1$
for $j = 1$ to N **do**
  Initialize P_j to be an empty trie
  for all c_{j-1} from P_{j-1} **do**
    for all $m_{j,l-1} \in M_{j,l-1}$ **do**
      for all $a_j \in A_j$ such that $Pr(a_j|m_{j,l-1}) > 0$ **do**
        $c_j \leftarrow c_{j-1}$
        if $\langle\hat{\theta}_j, a_j\rangle \in \nu(\cdot)$ **then**
          $c_j[\hat{\theta}, a_j] \overset{+}{\leftarrow} 1$
        else
          $c_j[\phi] \overset{+}{\leftarrow} 1$
        if $P_j[c_j]$ does not exist **then**
          Initialize $P_j[c_j] \leftarrow 0$
        $P_j[c_j] \overset{+}{\leftarrow} P_{j-1}[c_{j-1}] \times Pr(a_j|m_{j,l-1}) \times b_{0,l}(m_{j,l-1}|s)$
return P_n

At a high level, all configurations supported by the set of models ascribed to each other agent and their solutions, are generated and receive a non-zero probability. Algorithm 1 computes a probability for a configuration, which is the result of multiplying the probabilities of actions of N agents scaled by the belief over the corresponding models, which contributed to the configuration.

The final probability assigned to a configuration is a sum of all probabilities previously assigned to that configuration.

Algorithm 1 adds the predicted actions of each agent one at a time. It utilizes a trie data structure to store the probabilities of configurations. The trie enables efficient insertion and access of the configuration probabilities in the algorithm. We begin by initializing the configuration space for agent 0 (P_0) to contain one tuple of integers (c_0) with $|\nu| + 1$ 0s and assign its probability to be 1 (lines 1-2). Using the configurations from the previous step, we construct the configurations over the actions performed by N agents by adding 1 to the relevant element depending on agent j's action and its frame (lines 3-15). If an action a_j performed by an agent j with frame $\hat{\theta}_j$ is in the frame-action neighborhood $\nu(\cdot)$ then we increment its corresponding count by 1. Otherwise, it is considered as a dummy action and the count of ϕ is incremented (lines 9-12). Similarly, we update the probability of a configuration using the probability of a_j and that of the base configuration c_{j-1} (line 15). This algorithm may be invoked multiple times for different contexts $\nu(\cdot)$ as needed in computing the belief update and value function.

4.3 Analysis of Computational Savings

The complexity of accessing an element in a ternary search trie with $|\nu| + 1$ elements is $\Theta(|\nu|)$. The maximum number of configurations encountered at any iteration is upper bounded by total number of configurations for N agents, i.e. $\mathcal{O}((\frac{N}{|\nu^*|})^{|\nu^*|})$. Importantly, the complexity of Algorithm 1 is *polynomial* in N, $\mathcal{O}(N|M_j^*||A_j^*||\nu^*|(\frac{N}{|\nu^*|})^{|\nu^*|})$ where M_j^* and A_j^* are the largest sets of models and actions among any other agent.

For the traditional I-POMDP belief update, the complexity of computing Eq. 8 is $\mathcal{O}(|S||M_j^*|^N |A_j^*|^N)$ and that for computing Eq. 9 is $\mathcal{O}(|S||M_j^*|^N|A_j^*|^N|\Omega_j^*|)$ where * denotes the maximum cardinality set for any agent. For a factored representation, the belief update operator invokes Eq. 8 for each value of all state factors and it invokes Eq. 9 for each model of each other agent j and for all values of updated states. Hence, the total complexity of belief update is $\mathcal{O}(N|M_j^*||S|^2|M_j^*|^N|A_j^*|^N|\Omega_j^*|)$. The complexity of computing updated belief over state factor x^{t+1} using Eq. 12 is $\mathcal{O}(|S|NK|M_j^*||A_j^*| |\nu^*|(\frac{N}{|\nu^*|})^{|\nu^*|})$ (recall the complexity of Algorithm 1). Similarly, the complexity of computing updated model probability using Eq. 13 is $\mathcal{O}((|S|N|M_j^*||A_j^*||\nu^*| + |\Omega_j^*|)(\frac{N}{|\nu^*|})^{|\nu^*|})$. These complexity terms are polynomial in N for small values of $|\nu^*|$ as opposed to exponential in N as in Eqs. 8 and 9. The overall complexity of belief update is also polynomial in N.

Complexity of computing the immediate expected reward in the absence of problem structure is $\mathcal{O}(|S|K|M_j^*|^N|A_j^*|^N)$. On the other hand, the complexity of computing expected reward using Eq. 15 is $\mathcal{O}(|S|KN|M_j^*||A_j^*||\nu^*|(\frac{N}{|\nu^*|})^{|\nu^*|})$, which is again polynomial in N thereby providing significant savings for low values of $|\nu^*|$.

5. Branch and Bound for Scaling Exact Solution

In order to mitigate the curse of history, we present a novel branch-and-bound based method that utilizes upper and lower bounds on the value function of the many-agent I-POMDP to prune action nodes of the reachability tree while avoiding exactly evaluating the subtree below them. This method is a first application of the branch-and-bound scheme in the context of I-POMDPs, and in particular

addresses the challenge of deriving fast value function bounds. We derive these bounds and outline the branch-and-bound algorithm in this section.

Hauskrecht (2000) discusses various lower and upper bounds for POMDPs some of which are tight and can be computed quickly. Among these, the *blind-policy* lower bound and the *informed* upper bound, briefly reviewed in Section 2, can be computed quickly. We adapt these to the framework of many-agent I-POMDP.

5.1 Lower Bound Using Pessimistic Blind Policy

A blind policy chooses the same action at a time step regardless of the observation it receives. In other words, it ignores the information contained in the observation. This loss of information leads to an underestimation of the expected value. We extend this lower bound to many-agent I-POMDPs. Furthermore, to quickly compute this bound, for any given context we adopt a pessimistic perspective by utilizing the configuration that would lead to the least expected value for the agent. This shares similarities with the maximin way of estimating a player's utility in a game. A direct benefit of this step is that it avoids running Algorithm 1 to compute the distribution over configurations.

Proposition 2 (Pessimistic Blind Policy)**.** *The pessimistic blind policy yields a lower bound on the exact value of the many-agent I-POMDP.*

The proof of this proposition uses mathematical induction on the horizon, and is detailed in Appendix B. We show that the lower bound obtained by selecting the configuration $\underline{C}^{\nu(x_k^t,a_0^t)}$ that minimizes R_0 for the horizon 1 action-value is: $\underline{Q}^1_{0,l}(b^t_{0,l},a^t_0) = \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{k=1}^{K} R_0(x^t_k,a^t_0,\underline{C}^{\nu(x^t_k,a^t_0)})$. Analogously to POMDPs, we may decompose this lower bound value using alpha vectors as $\underline{Q}^1_{0,l}(b^t_{0,l},a^t_0) = \sum_{\mathbf{s}^t} \underline{\alpha}^1(\mathbf{s}^t)\cdot b^t_{0,l}(\mathbf{s}^t)$, where $\underline{\alpha}^1(\mathbf{s}^t) = \sum_{k=1}^{K} R_0(x^t_k,a^t_0,\underline{C}^{\nu(x^t_k,a^t_0)})$.

For horizon $h > 1$, the action-value lower bound in full is,

$$\underline{Q}^h_{0,l}(b^t_{0,l},a^t_0) = \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{k=1}^{K} R_0(x^t_k,a^t_0,\underline{C}^{\nu(x^t_k,a^t_0)}) + \gamma \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{\mathbf{s}^{t+1}} \prod_{k=1}^{K} \mathcal{T}_0(x^t_k,a^t_0, \underline{C}^{\nu(x^t_k,a^t_0,x^{t+1}_k)},x^{t+1}_k)\ \underline{\alpha}^{h-1}_*(\mathbf{s}^{t+1})$$

The lower-bound vectors for horizon h may be obtained for all actions by using lower bound vectors from horizon $h-1$ as follows:

$$\underline{\alpha}^h(s^t) = \sum_{k=1}^{K} R_0(x^t_k,a^t_0,\underline{C}^{\nu(x^t_k,a^t_0)}) + \gamma \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{\mathbf{s}^{t+1}} \prod_{k=1}^{K} \mathcal{T}_0(x^t_k,a^t_0,\underline{C}^{\nu(x^t_k,a^t_0,x^{t+1}_k)},x^{t+1}_k) \times \underline{\alpha}^{h-1}_*(\mathbf{s}^{t+1})$$

Notice that the vectors computed for the lower bound are over the physical states only (and not over the interactive state space). Hence, they are not impacted by the curse of many agents and can be computed quickly. Given agent 0's belief over the physical states $b_{0,l}$, a lower bound on the horizon h value function is obtained as:

$$\underline{V}^h_{0,l}(b^t_{0,l}) = \max_{a^t_0 \in A_0} \underline{Q}^h_{0,l}(b^t_{0,l},a^t_0) \tag{16}$$

5.2 Fast Informed Upper Bound with Optimistic Configurations

As we mentioned in Section 2, the fast informed bound provides an upper bound to the POMDP value function. Hauskrecht (1997) shows that this bound is tighter than the one obtained by using the MDP based method. We generalize this bound to the many-agent I-POMDP and enable faster computation by choosing configurations that maximize reward and probabilities while upholding the upper bound property.

Proposition 3 (Optimistic Fast Informed Bound)**.** *The fast informed update with maximizing configurations gives an upper bound to the exact many-agent I-POMDP value function.*

Analogously to the previous proposition, the proof here also uses mathematical induction on the horizon, and is detailed in Appendix B. An upper bound for the horizon 1 value is obtained as follows: $\overline{Q}^1_{0,l}(b_{0,l}, a^t_0) = \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{k=1}^{K} \mathcal{R}_0(x^t_k, a^t_0, \overline{C}^{\nu(x^t_k, a^t_0)})$, where $\overline{C}^{\nu(x^t_k, a^t_0)}$ is the configuration that maximizes $R_0(x^t_k, a^t_0, C^{\nu(x^t_k, a^t_0)})$. These upper bounds may be decomposed into a set of vectors one for each action a_0, $\overline{\alpha}^1_{a_0}(\mathbf{s}^t) = \sum_{k=1}^{K} \mathcal{R}_0(x^t_k, a^t_0, \overline{C}^{\nu(x^t_k, a^t_0)})$, and each vector has as many components as states.

For horizon h, the action-value upper bound is $\overline{Q}^h_{0,l}(b^t_{0,l}, a^t_0) = \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t)\, \overline{\alpha}^h_{a^t_0}(s^t)$. Each action value is associated with a single alpha vector due to which there are as many vectors as the number of actions. We write out the full definition of the alpha vector for completeness:

$$\overline{\alpha}^h_{a^t_0}(\mathbf{s}^t) = \sum_{k=1}^{K} \mathcal{R}_0(x^t_k, a^t_0, \overline{C}^{\nu(x^t_k, a^t_0)}) + \gamma \sum_{\omega^{t+1}_0} \sum_{\mathbf{s}^{t+1}} \prod_{k=1}^{K} \mathcal{O}_0(x^{t+1}_k, a^t_0, \overline{C}^{\nu(x^{t+1}_k, a^t_0, \omega^{t+1}_0)}, \omega^{t+1}_0) \times \mathcal{T}_0(x^t_k, a^t_0, \overline{C}^{\nu(x^t_k, a^t_0, x^{t+1}_k)}, x^{t+1}_k)\, \overline{\alpha}^{h-1}_*(\mathbf{s}^{t+1}) \Bigg\} \tag{17}$$

The upper bound on the value function is obtained as:

$$\overline{V}^h_{0,l}(b^t_{0,l}) = \max_{a^t_0 \in A_0} \overline{Q}^h_{0,l}(b^t_{0,l}, a^t_0) \tag{18}$$

Similar to the lower bound, the upper bound vectors are defined over the physical states only. Therefore, joint models or joint actions are not needed in their computation. Hence, these bounds are unaffected by the *curse of many agents*. [4]

5.3 Improved Efficiency using Branch and Bound

Branch and bound (Land & Doig, 1960) is a well known and general scheme for finding optimal solutions to typically discrete problems. It involves iterating over the two steps of assigning values

4. A minor modification to the backup rules of the lower and upper bounds introduced in Sections 5.1 and 5.2 respectively, improves the tightness of these bounds. Let the horizon 1 action values that form the lower and upper bounds respectively, remain unchanged. Note that the update rules for horizon $h > 1$ utilize these action values. We may replace these lower and upper bound horizon 1 action values in the update rules with their exactly computed horizon 1 counterparts. Clearly, the modified action values continue to upper and lower bound the exact action values, respectively. Simultaneously, including the exact horizon-1 action value tightens both the bounds. However, these bounds are no longer independent of the joint model space because of the exact component and cannot be computed quickly.

to an increasing subset of variables (branch) and establishing bounds on the values of the solutions based on the partial assignment. The bounds are utilized in pruning portions of the search tree that are guaranteed to lead to suboptimal solutions. Therefore, it requires a flexible way to lower and upper bound the exact value of the solution that becomes increasingly tight with a growing partial assignment.

Algorithm 2 InitializeNode

Input: Current belief $b^t_{0,l}$, horizon h
Output: A node n in the reachability tree and the subset of actions to be expanded from n

1: Initialize a new node n, and set its belief to be $b^t_{0,l}$ and look ahead to h
2: **for all** $a_0 \in A_0$ **do**
3: $\quad LB[a_0] \leftarrow \underline{Q}^h(b^t_{0,l}, a_0)$ (*Initialize to pessimistic blind policy value as in Section 5.1*)
4: $\quad UB[a_0] \leftarrow \overline{Q}^h(b^t_{0,l}, a_0)$ (*Initialize to optimistic fast-informed value as in Section 5.2*)
5: $\underline{V} \leftarrow \max_{a_0} LB[a_0]$
6: $\overline{V} \leftarrow \max_{a_0} UB[a_0]$
7: **for all** $a'_0 \in A_0$ **do**
8: $\quad$ **if** $UB[a'_0] < \underline{V}$ **then**
9: $\quad\quad$ Do not expand the tree for action a'_0

We will use branch and bound to compute more efficiently the optimal expected value of the initial belief at the root node of the tree in Fig. 3 and the corresponding policy tree. Observe that the reachability tree of Fig. 3 explores all possible actions from a node in order to find the action with the largest value. However, computing the exact value of a node requires knowing the value of the best action only at each of its children. Consequently, there is potential for improvement by avoiding expanding the tree for suboptimal actions.

Algorithm 2 initializes the root node of the reachability tree using an iteration of branch and bound. It takes the belief and horizon as input. For each action of agent 0, a_0, the algorithm initializes the upper and lower bound values for policies with a_0 as the initial action as described in Sections 5.1 and 5.2. The upper and lower bound on the overall value of the node is initialized to be the largest values of these bounds currently as $\overline{V}$ and $\underline{V}$, respectively. Actions for which the corresponding upper bound is smaller than the lower bound on the overall value function are guaranteed to produce sub-optimal policy. Hence they need not be explored further without affecting the solution quality.

Next, we recursively tighten the lower and upper bounds on the value of the root node using the branch and bound approach as outlined in Algorithms 3, 4 and 5. Particularly, we repeatedly invoke Algorithm 3 on the root node of the reachability tree to update its bound until the upper and lower bounds have the same value and the algorithm returns true.

It partially unfolds the reachability tree for some action a_0 and computes tighter bounds on its action-value based on the unfolded tree. We base the choice of action for which to expand the tree on a heuristic function (line 3 in Algorithm 3). Both the upper and lower bounds are informative about which action is likely optimal given the current belief. Subsequently, while we may pick either bound to provide a heuristic ordering of actions, we select the relative upper bounds for prioritizing actions. This heuristic is also referred to as the IE-Max (Kaelbling, 1993) and it makes an appearance in several other heuristic-based POMDP methods (Smith & Simmons, 2004; Shani, Brafman, &

Algorithm 3 UpdateBounds

Input: A node n of the reachability tree
Output: **True** if the node's optimal value is available otherwise **false**

1: **if** $\underline{V} = \overline{V}$ for n **then**
2: return **true** (*Terminate*)
3: $a_0 \leftarrow$ Pick action according to a heuristic ordering
4: **if** did not previously expand on action a_0 **or** do not expand is not flagged for a_0 **then**
5: NextNodes[a_0][] $\leftarrow$ Branch (n, a_0)
6: Set branched[a_0] $\leftarrow$ **true**
7: **else**
8: **for all** $\omega_0 \in \Omega_0$ **do**
9: Recursively invoke UpdateBounds on NextNode[a_0][ω_0]
10: Get $LB[a_0], UB[a_0] \leftarrow$ Bound (n, a_0)
11: $\underline{V} \leftarrow \max_{a_0} LB[a_0]$
12: $\overline{V} \leftarrow \max_{a_0} UB[a_0]$
13: **for all** $a_0' \in A_0$ **do**
14: **if** $UB[a_0'] < \underline{V}$ **then**
15: Do not expand the tree for action a_0' at node n
16: return **false**

Shimony, 2007; Kurniawati, Hsu, & Lee, 2008). It is preferred because as the upper bound for an action a_0 is updated, its value may drop below the upper bound for some other action thereby allowing another action to be explored. On the other hand, picking an action based on the lower bound will often result in the same action because its value will increase monotonically as the lower bound is tightened. This makes it difficult to discover its suboptimality till the entire subtree rooted at that node has been expanded and evaluated.

Algorithm 4 Branch

Input: A node n of the reachability tree and action a_0
Output: Next nodes on performing a_0 and all observations if horizon is not 1

1: **if** $h > 1$ **then**
2: **for all** $\omega_0 \in \Omega_0$ **do**
3: NextNode[a_0][ω_0] $\leftarrow$ InitializeNode ($\tau(b_{0,l}, a_0, \omega_0)$, $h-1$)

The algorithm then expands the tree for the chosen action if it hasn't done so previously by invoking Algorithm 4 in line 5. Branch() expands a subtree of depth 1 for that action by performing the belief update for each observation, and initializes the bounds at the child nodes. This is followed by updating the bound for the chosen action (line 10) and using the largest of the lower and upper bounds for all explored actions so far to obtain the exact value bounds at node n. If the upper bound for an action at any iteration is less than the overall lower bound on the value function, the action is flagged so that it is no longer explored.

Algorithm 5 Bound

Input: A node n of the reachability tree and action a_0
Output: Updated lower and upper bounds for n

1: $ER_0 \leftarrow$ Compute expected immediate reward for $b_{0,l}$
2: $LB[a_0] \leftarrow ER_0$
3: $UB[a_0] \leftarrow ER_0$
4: **if** $h > 1$ **then**
5: **for all** $\omega_0 \in \Omega_0$ **do**
6: $LB[a_0] \leftarrow \gamma \, Pr(\omega_0|b_{0,l}, a_0) \times \underline{V}$ of NextNode$[a_0][\omega_0]$
7: $UB[a_0] \leftarrow \gamma \, Pr(\omega_0|b_{0,l}, a_0) \times \overline{V}$ of NextNode$[a_0][\omega_0]$

We may encounter an action that has been previously explored because Algorithm 3 is called repeatedly. In this case, the bounds on the nodes in the subtree are updated during a depth-first traversal (lines 8-9).

Bound() described in Algorithm 5 tightens the lower and upper bound on the action-value for a node n and action a_0. These values are updated by considering the updated value function bounds of the child nodes in the subtree if node n is not a leaf node.

Note that once a subtree has been fully expanded for an action a_0 and all its subtrees have been exactly evaluated, the upper and lower bound value for a_0 are the same (i.e., $UB[a_0] = LB[a_0]$). At this point, either the lower bound value for a_0 is greater than upper bound value for all other actions or there exists some other action a_0' such that its upper bound value is greater than $UB[a_0]$ (i.e., either $LB[a_0] \geq UB[a_0'], \forall a_0'$ or $\exists a_0'$ such that $UB[a_0'] > UB[a_0]$). In the former, we may conclude that a_0 is the optimal action and its corresponding value is the exact solution for the subtree. In the latter case, the heuristics will pick a_0' to explore if feasible the next time Algorithm 3 is invoked for the node.

6. Experiments

The theoretical analysis establishes exact solutions for the many-agent I-POMDP as well as bounds on the value that can be quickly computed. In this section, we empirically evaluate these methods on three problem domains. In particular, we focus on the run time for the planning, how we may reduce it, the impact of each problem structure, and the expected value of the solution.

6.1 Many-Agent Problem Domains

We seek problem domains that exhibit large agent populations. Such domains often naturally allow for anonymity and context-specific independence.

6.1.1 POLICING LARGE PROTESTS

Our first problem domain for evaluation is the previously characterized non-cooperative policing of a large protest by two troops. The problem exhibits 27 physical states, 9 actions for the police and 4 for the protestors, 8 observations each for the police and protestors. Three factors each of which informs whether the protest intensity at a site is low or high comprises an observation. The protests are distributed across 3 sites all connected with each other. The police must decide on where

to send its two troops while a protestor must decide on whether to protest or not, and move to a different site. Each protestor has one of two frames, *peaceful* or *disruptive*, and its protest action impacts the intensity of the protest at that site based on its frame. An observation factor in the policing protest scenario is the protest intensity at a site. As the intensity is determined by the actions of the protesters present at the site, it is independent of the intensity at any other site. Thus, the independence supposition of Eq. 10 is easily satisfied in this scenario.

6.1.2 TRAFFIC CONGESTION CONTROL

Our second domain pertains to an intelligent traffic management system that seeks to reduce vehicular congestion at a busy intersection. We study a typical road network that consists of 3 managed intersections all of which feed into another intersection which we seek to decongest. This network is illustrated in Fig. 4. The traffic management system, which is the subject agent in this domain, has the choice of either increasing or decreasing the duration of the red lights at three nearby intersections, A, B and C. All red lights at these intersections are configured to have the same duration. This regulates the traffic flowing into the intersection of interest because increasing the red light duration should reduce the traffic flowing from that intersection, otherwise the traffic escalates. Thus, the subject agent has $|A_0| = 8$ actions at any time step.

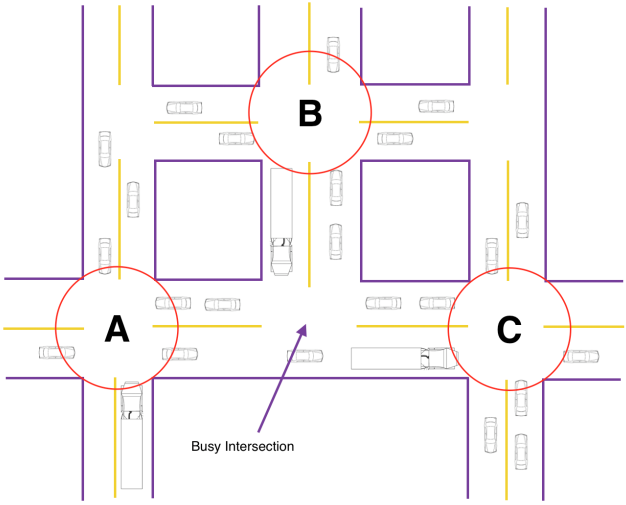

Figure 4: Our second problem domain is about decongesting traffic at a busy intersection of a road network. Traffic from three nearby intersections can flow into the busy intersection. These are two-lane roads and each passenger vehicle or commercial truck can choose to go straight, turn left, turn right, or make a U turn.

The state of the problem consists of three factors. Each of these factors represents the congestion status of the traffic flowing into the busy intersection from each of the three intersections. This could be low, medium, or high. While there is no consensus on how to accurately measure vehicular traffic congestion, we could use travel time index, which is a ratio of the number of vehicles passing through the intersection under congested and free flow conditions and categorize it into the three levels. At

each time step, the system attempts to ascertain the density of traffic flowing into the intersection of interest from each of the three directions. It does this using its overhead cameras and object recognition system, which is typically inaccurate. This leads to $|\Omega_0| = 8$ observations. Because there are various ways of entering or exiting the intersections, the assumption of independence among the state and observation factors as in Eq. 10 is met. The reward function of the system encodes its objective to keep the traffic flowing smoothly through the intersection. Therefore, quicker flows are given a higher reward.

The traffic consists of passenger vehicles and commercial trucks. These are tracked using overhead cameras and object recognition as they move through the road network over many time steps. The number of passenger vehicles is larger than commercial trucks, however, commercial trucks have a disproportionate impact on traffic congestion due to their larger size and slower speed. Each vehicle may choose either to go straight ahead, turn left, turn right, or make a U turn at an intersection ($|A_j| = 4$). The vehicle also observes whether the red light duration at its current intersection is longer than usual. It prefers to perform its usual action, which may cause it to travel through the intersection (dislikes detours). Notice that the transition of the state of the system depends on the joint actions of all agents. The next state is not perfectly determined because the system does not know how many vehicles may precisely turn into the roads leading to the intersection.

6.1.3 MULTIPLAYER VIDEO GAMING

We base a third problem domain on the popular mobile game *Clash of Clans (CoC)* (`http://clashofclans.com`), which is a massively multiplayer online strategy game. The game has two aspects: the first is to gather resources through various activities including raiding settlements of other players using own armies and the second is to build a strong settlement to defend own resources against the invasion of other players' armies. For our purpose, we focus on the latter aspect of the game.

Figure 5: A screen shot of the game Clash of Clans. The image shows a simple settlement in which the resources are stored in the central structure which is surrounded by walls on four sides. The walls are guarded using cannons which are situated on the outside of the walls.

The objective of the subject agent (agent 0) is to defend its resources against any invasion by the armies of other players. To do so it may erect boundary walls in the four cardinal directions around its settlement and install cannons alongside each wall that may ward off the attack by invading armies. Figure 5 is a screen shot of the game depicting an example settlement.

We model the physical state of the game as composed of four factors. The value of each factor represents the defense status in the respective cardinal direction: is it completely *unprotected* in that direction, just *walled*, or is it walled and *guarded* using canons. These represent increasing levels of protection. Therefore $|S| = 81$ (3 values for each cardinal direction). At any time step, agent 0 may fortify its defenses in any one cardinal direction only ($|A_0| = 4$) and receive an observation providing information about the direction with the weakest defense status ($|\Omega_0| = 4$). If more than one cardinal direction is unprotected, agent 0 is informed one of these directions at random. Similarly, if more than one direction is walled while others are guarded with cannons, one of the walled direction is told to agent 0 at random. On the other hand, if a single direction is the weakest, the agent is informed about this direction perfectly. This motivates a probability distribution over observations in some scenarios, and the observation function is stochastic. The action of fortifying defenses in any direction raises the level of protection from *unprotected* to *walled* and from *walled* to *guarded* with some probability. Note that the defending agent is unable to fortify its defenses due to insufficient gold. Consequently, the fortify action may fail occasionally and this non-determinism makes the transition function stochastic (note that we do not include collectibles such as amount of gold, gems and elixir in the state to keep the state space bounded). When attacked by an opponent's army, the subject agent may suffer losses and its defenses may be weakened or even completely destroyed.

An opponent army consists of N agents that may have one of two fixed frames: *tier 1* or *tier 2*. CoC defines tier 1 to be a class of weaker soldiers (consisting of barbarians, archers and goblins) that are injured easily by the canons but could recuperate quickly as well. Tier 2 attackers (consisting of giants and wizards) are more resilient to attack by the canons and cause more damage than the tier 1 attackers. But when injured, they recover slowly. These agents may attack from one of the four cardinal directions or they may recover when injured ($|A_j| = 5$). A common strategy while attacking a settlement is to target the least protected side. An agent continues to target the same side until it is injured by a canon. When injured the agent retreats to recover. On recovering, it may pick another side to target. The state of the defenses in any direction depends on the action of agent 0 and the number of agents of each type attacking that direction.

6.2 Performance Evaluation

We begin by evaluating the performance of the exact many-agent I-POMDP solution technique discussed in Section 4. It expands the look-ahead tree to include all reachable beliefs over the finite horizon and then calculates the optimal value at the root node using the Bellman equation for the many-agent I-POMDP framework in a bottom up fashion starting at the leaf nodes, as outlined in Section 4.1. We evaluate its performance in the aforementioned policing protest and traffic control domains. Other agents are modeled as POMDPs and their predicted behavior is obtained using bounded policy iteration (Poupart & Boutilier, 2003). This represents the models as finite state controllers, which enables a compact model space (Sonu & Doshi, 2015).

We set the maximum planning horizon to 5 in all the experiments. The transition, observation and reward functions of the many-agent I-POMDP are all compactly encoded as frame-action

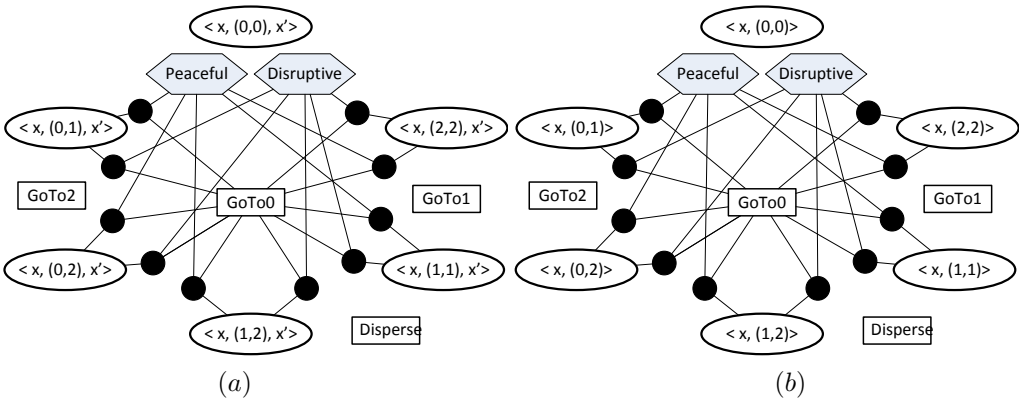

Figure 6: A compact Levi graph representation of the *Policing Protest* scenario as a frame-action hypergraph for (a) the transition function, and (b) the reward function for site 0. The variables x and x' represent the start and end intensity of the protest at site 0 and the action shows the location of the two police troops. As two police troops are sufficient to deescalate any protest, the contexts in which both troops are at site 0 are independent of the actions of other agents. All other contexts depend only on the agents belonging to either frame choosing to protest at site 0.

hypergraphs; example hypergraphs are shown in Fig. 6. All computations are carried out on a RHEL platform with 2.80 GHz processor and 4 GB of main memory.

An important *baseline* that allows evaluating the computational gain due to the problem structures is an exact method that solves the many-agent I-POMDP but without leveraging the problem structures of anonymity and context-specific independence. This method enumerates the transition, observation and reward functions for N agents instead of using hypergraphs and also expands the look-ahead reachability tree in order to perform exact value iteration.

In our first experimental setting, we consider the policing protest and traffic control problems with up to N=5 protestors and cars, respectively, exhibiting different frames. Table 1 shows the run time and expected value of exactly solving the policing problem using the original I-POMDP and the many-agent I-POMDP that models the problem structures. To assess the impact on time of modeling each problem structure, we also report the planning time using I-POMDP with action anonymity but not context-specific independence. All three frameworks produce policies with the same expected value. However, as the latter losslessly compresses joint actions to configurations and precludes reasoning about some actions of others based on context, the many-agent I-POMDP requires the least amount of running time. Exploiting action anonymity but not context-specific independences reduces the run time compared to the original I-POMDP. The additional impact of considering context-specific independences can now be ascertained from the run times of using the many-agent I-POMDP. Furthermore, notice that the gradient of increase in run time with increase in number of agents is much less for many-agent I-POMDPs compared to the original framework. This result strongly suggests that the many-agent I-POMDP framework may scale to significantly larger number of agents. Interestingly, an increase in the number of protestors causes the value to drop suggesting that policing more protestors is harder, as we may expect.

No. of protesters (N)	H	Planning time (secs)			Speed up	Exp. Value
		I-POMDP	I-POMDP with action anonymity	many-agent I-POMDP		
2	2	1	0.7	0.55	1.8	77.42
	3	19	19	17	1.1	222.42
3	2	3	0.7	0.56	5.4	77.34
	3	38	22	17	2.2	222.32
4	2	39	0.7	0.57	68	76.96
	3	223	22	17	13.1	221.87
5	2	603	0.80	0.60	1,005	76.88
	3	2,480	24	18	137.8	221.77

Table 1: Comparison between the traditional I-POMDP, I-POMDP exhibiting action anonymity only, and many-agent I-POMDP frameworks, on the domain of policing protest. All frameworks follow the same solution approach of computing a reachability tree and performing value iteration on the tree. Note that the last framework exhibits both action anonymity and context-specific independence. We report the speed up in run time due to exploiting these structures as compared to the traditional I-POMDP.

No. of vehicles (N)	H	Planning time (secs)			Speed up	Exp. Value
		I-POMDP	I-POMDP with action anonymity	many-agent I-POMDP		
2	2	0.8	1	0.6	1.33	171.76
	3	16	14	9	1.78	267.25
3	2	3	1	0.7	4.29	171.77
	3	41	15	9	4.56	267.25
4	2	32	1	1	32	171.76
	3	328	18	12	27.33	267.26
5	2	449	2	1	449	171.76
	3	26,970	23	13	2,075	267.25

Table 2: Comparison between the traditional I-POMDP, I-POMDP exhibiting action anonymity only, and many-agent I-POMDP frameworks, on the traffic control domain. All run times reflect the same solution approach of computing a reachability tree and performing value iteration on the tree. Notice that the speed up for 5 agents at a horizon of 3 is more than three orders of magnitude.

Table 2 reports on analogous results for the traffic congestion control domain. Compared to the disproportionate growth in the run time of the traditional I-POMDP, the many-agent I-POMDP experiences only slight increases in run time as the number of vehicles on the roads grows. Here, context-specific independence has a significant impact on the run time as observed by comparing the run times of using I-POMDP with action anonymity and the many-agent I-POMDP frameworks. This is because two of the four actions of each vehicle at an intersection leads it away from the busy

intersection of interest; the traffic control system need not consider these actions in its deliberations. To further understand and test the behavior of the traffic control system, we varied the ratio of light to heavy vehicles from its default of 8:2 when $N = 20$. As we increased the proportion of heavy vehicles in the traffic leading to ratios of 5:5 and 0:10, we observed a slight drop in the expected values, indicating some increased difficulty in clearing up the congestion caused by more slow-moving vehicles. The resulting policy resorted to increasing the duration of the red light at the intersections with more slow-moving vehicles in the traffic.

No. of protesters (N)	**H**	**Planning time (secs)**		**Exp. Value**
		Exhaustive	**Branch&Bound**	
20	3	149	7	275.13
	4	2,816	27	384.47
	5	—	86	482.88
50	3	157	8	274.90
	4	3,062	33	384.23
	5	—	118	482.62
100	3	193	10	274.41
	4	3,651	44	383.72
	5	—	187	482.11
200	3	350	19	273.38
	4	6,217	108	382.66
	5	—	416	481.02
500	3	1,137	73	270.41
	4	18,087	285	379.56
	5	—	1,462	477.82
1000	3	6,782	465	265.02
	4	—	2,135	373.96
1500	3	—	1,851	259.74
	4	—	9,858	368.44
2000	3	—	4,189	254.9
	4	—	21,534	363.35

Table 3: Comparison of the planning times between the performance of the *Exhaustive* and *Branch&Bound* methods for the policing problem. A '—' indicates that the program ran out of memory.

In our second set of experiments, we evaluate on settings involving many more agents. As Table 1 indicates, the traditional I-POMDP does not realistically scale to $N > 5$ agents. On the other hand, modeling and considering problem structures as we do in the many-agent I-POMDP helps. Indeed, Table 3 indicates that we may obtain exact policing solutions using tree search for as many as 1,000 protestors, and ease traffic congestion for up to 500 vehicles as we show in Table 4. We may find a defense strategy in CoC when agent 0's settlement is under attack by an army with up to 1,000 attackers (Table 5).

No. of vehicles (N)	H	Planning time (secs)		Exp. Value
		Exhaustive	Branch&Bound	
20	3	16	2	215.48
	4	374	12	316.72
	5	10,480	91	412.34
50	3	90	7	140.26
	4	46,268	58	185.24
	5	—	502	221.66
100	3	256	15	140.28
	4	—	119	185.24
200	3	1,182	42	140.92
	4	—	332	186.10
500	3	8,410	243	141.00
	4	—	5,047	186.09
1000	3	104,393	12,439	140.25

Table 4: Comparison of the planning times between the performance of the *Exhaustive* and *Branch&Bound* methods for the traffic control problem.

Nevertheless, while the exploitation of the problem structures reduces the curse of many agents that plagues I-POMDPs, the curse of history is unaffected by such approaches. To mitigate the curse of history we implemented the more efficient branch-and-bound based solution technique presented in Section 5.3 using the pessimistic blind policy as a lower bound and the optimistic informed bound as an upper bound. Tables 3, 4, and 5 demonstrate that branch and bound enables planning with more agents in all three domains. In particular, we can effectively police large protests with up to 2,000 protestors, control traffic for 1,000 vehicles, and plan for an attack with up to 2,000 individuals in the army. A 4-horizon plan can be obtained for the police in under 6 hours while a 3-horizon plan for the latter consumed under 3 hours. We were unable to solve beyond 1,000 vehicles for traffic control as fewer actions of others could be deemed contextually independent. Note that the branch and bound approach is exact and yields the same expected value as value iteration on the full tree. Furthermore, it collapses into the exhaustive approach if bounds are not used for pruning.

6.3 Discussion

A comparison of the performance of the many-agent I-POMDP with the original I-POMDP yields two important results: (i) When there are few other agents, the many-agent I-POMDP provides exactly the same solution as the original I-POMDP but with reduced planning times because the problem structure is exploited. (ii) Many-agent I-POMDP scales to larger agent populations, from 100 to more than 1,000, and the new framework delivers promising results within reasonable time on standard computing platforms. The branch-and-bound method outperforms the exhaustive method consistently without any loss in value of the outcome policy. However, the increase in run time with number of agents is not proportionate and the method eventually becomes infeasible as the number of agents is further increased. While the horizon of 5 may not appear to be large, the look-ahead search tree for the policing protest problem as an example exhibits a branching factor of 72. Consequently, a

No. of Attackers (N)	H	Planning time (secs)		Exp. Value
		Exhaustive	Branch&Bound	
20	3	4	<1	-3.25
	4	33	2	-3.99
	5	228	7	-4.67
50	3	6	1	-3.31
	4	45	3	-4.09
	5	304	9	-4.83
100	3	14	2	-3.43
	4	97	7	-4.30
	5	648	27	-5.14
200	3	56	9	-3.65
	4	445	53	-4.66
	5	3,359	293	-5.71
500	3	349	93	-4.62
	4	3065	818	-6.23
	5	—	4,776	-8.01
1000	3	1897	1,190	-6.21
	4	—	14,087	-8.79
1500	3	—	5,043	-7.89
2000	3	—	11,089	-8.88

Table 5: Comparison between planning time performances of the *Exhaustive* and *Branch&Bound* methods on the CoC gaming problem.

tree of horizon 3 has more than 373K nodes, and a tree of horizon 4 has more than 268 million nodes. Experiments reported in Table 3 demonstrate that branch-and-bound consistently offers a speed up that is greater than an order of magnitude. As such, this is evidence of a significant reduction in the size of the look-ahead tree. For less numbers of agents, the branch-and-bound also allows scaling the planning horizon to greater than 5 with proportionate increase in run time.

In general, the policing domain is defined such that the numbers of troops are sufficient to police the protests even as the number of protestors grows. However, it does become increasingly difficult to police. This is reflected in the fact that the exact expected value reduces as the number of protestors increases for the same horizon. For example, for horizon 3 the value reduces from 275.13 for $N = 20$ to 254.9 for $N = 2,000$, and for horizon 4, it reduces from 384.47 for $N = 20$ to 363.35 for $N = 2,000$. The expected rewards in CoC are negative because our planning focuses solely on one aspect of the game – defending the resources. Specifically, agent 0 incurs costs for fortifying its defenses and fending off enemy's attacks. We do not plan for the task of attacking other agents' settlements or other activities that could yield positive rewards.

For relatively low numbers of protestors, $N \leq 50$, if there is more than one site with a high protest intensity, the policy recommends sending the two police troops to two different high protest sites. This enables successfully controlling those protests. But, for greater numbers of protestors, the policy recommends sending both troops to a single site of high protest intensity even if there are

more such sites. While two troops are sufficient to control the protests for large N, a single troop is not. Analogously, the policy for the traffic control domain does not increase the red light durations at intersections when the traffic flows are low or medium, and the number of vehicles is lower than 50. However, for more vehicles in the system, the policy recommends increasing the red light durations at one or two intersections (but not all three) when traffic flows are medium or high.

7. Related Work

We substantially build on AGGs in this article by extending anonymity and context-specific independence to include agent frames, and generalizing their use to a partially observable stochastic game solved using decision-theoretic planning as formalized by I-POMDPs. Indeed, Bayesian AGGs (Jiang & Leyton-Brown, 2010) extend the original AGG formulation to include agent types, which results in type-specific action sets. The number of nodes in the action graph grows with types: $|\hat{\Theta}||A|$ nodes for agents with $|\hat{\Theta}|$ types each having $|A|$ actions that may overlap. Two actions from different type-action sets may share a node, in which case these actions are interchangeable and the count may involve agents of both types. A key difference in our representation is that we explicitly model frames in the graphs due to which context-specific independence is modeled using frame-action *hypergraphs*. Benefits are that we naturally maintain the distinction between two similar actions but performed by agents of different frames, and we add less additional nodes: $|\hat{\Theta}| + |A|$. However, a hypergraph is a more complex data structure for operation. Finally another extension of AGGs, temporal AGGs (Jiang, Leyton-Brown, & Pfeffer, 2009) extend AGGs to a repeated game setting and allow decisions to condition on chance nodes. These nodes may represent the action counts from previous step (similar to observing the actions in the previous game). Temporal AGGs come closest to multiagent influence diagrams (Koller & Milch, 2001) although the former can additionally model the anonymity and contextual independence structure. Overall, I-POMDPs with frame-action anonymity and context-specific independence significantly augment the combination of Bayesian and temporal AGGs in expressivity by utilizing the structures in a *partially observable stochastic game* setting with agent types. As such, these structures additionally benefit transition and observation functions.

Varakantham et al. (2014) building on previous work (Varakantham, Cheng, Gordon, & Ahmed, 2012) recently introduced a decentralized MDP that models a simple form of anonymous interactions: transition probability and reward specific to some state-action pairs are affected by the number of agents regardless of their identities. The considered numbers of agents are contextual to the state-action pair, and may not involve all the agents. In contrast, we consider context-specific independence where not all actions performed by other agents are relevant depending on the context. Very recently, Robbel et al. (2016) utilized anonymous influence to scale variable elimination for factor graphs exhibiting large tree widths. Specifically, structure that relies on the number of binary variables being active during variable elimination rather than their identity is exploited. Interestingly, this exploit is used to scale the solution of multiagent MDPs using asynchronous linear programming (Guestrin, Koller, & Parr, 2001) to allow up to 25 agents. In both these efforts, the interaction influence is not distinguished into how many agents are performing which actions as in action-specific anonymity and independence, due to which neither configurations nor hypergraphs are used. Furthermore, agent types are not considered. Rather, the context in Robbel et al. includes the state variables and agents that are active.

Various representations of context have played key roles in decoupling multiagent decision making, thereby promoting scalability. For example, Witwicki and Durfee (2010) introduce a decentralized POMDP model that is decoupled into individual POMDPs for the most part except for transition dependencies due to state factors affected by other agents. Both Varakantham et al. (2009) and, Mostafa and Lesser (2009) introduce frameworks that allow context-specific interactions with the benefit that outside these contexts, the decision making of each is independent of others. Finally, the interaction hypergraphs in networked-distributed POMDPs (Nair, Varakantham, Tambe, & Yokoo, 2005) model complete reward independence between agents that are not linked, analogous to the independence structure in graphical games. This model differs from the hypergraphs in this article (and action graphs) that model independence in reward (and transition, observation probabilities) along a different dimension: actions.

8. Concluding Remarks

The many-agent I-POMDP introduced in this article is a compelling framework because of its ability to scale exact planning under uncertainty to beyond 1,000 agents by exploiting problem structures. We formalize widely existing problem structures – frame-action anonymity and context-specific independence – and encode it in frame-action hypergraphs. Other real-world examples exhibiting such problem structure are found in economics where the value of an asset depends on the number of agents vying to acquire it and their financial standing (frame), in real estate where the value of a property depends on its demand, the valuations of neighboring properties, as well as the economic status of the neighbors because an upscale neighborhood is desirable. Compared to the previous best approach (Sonu & Doshi, 2015), which scales to an extension of the simple tiger problem involving 5 agents only, the presented framework is far more scalable in terms of number of agents.

An alternate way of modeling problems involving frame-action anonymity would be to include the configuration vectors in the state space S and represent the decision-making problem as a POMDP. A major problem with a representation that directly includes action counts in the state space is how to generate an accurate transition function that models the probabilities of transitioning between various action count vectors. Here, I-POMDPs offer the benefit of including models of other agents, which we update over time to obtain updated distributions over actions. These are then used to obtain action counts for the next time step. We are not aware of any approach for transitioning between frame-action configurations *directly*.

A limitation of the presented method for solving the many-agent I-POMDP is that it proceeds requiring a prior belief of the subject agent. Updated beliefs are necessary at each step to compute distributions over the configurations (Algorithm 1) and to evaluate the value of a policy subtree, which is in turn used to compute the value of the entire policy. This makes it ill-suited for settings requiring a general solution for any prior in the form of policies and corresponding value vectors. The belief update in the many-agent I-POMDP assumes that the observation and state factors are respectively independent. Factors that constitute an observation or a state are typically orthogonal to each other because each factor represents another "dimension" of the observation or state. Thus, these independence assumptions are usually satisfied in practice. Another minor limitation is the assumption that the large set of agents are associated with few frames. This assumption is not particularly strict with this often being the case in practice, as exemplified in our scenarios. While the policing protest domain is inspired by contemporary events around the world, our modeling of this

problem is simplistic. It does not capture the complex nuances and issues involved in real protests; rather we think of it as an academic abstraction for studying many-agent decision making.

Our future work involves continuing to find ways to scale principled multiagent planning to larger agent populations. In this regard, we are exploring and modeling other types of problem structures often found in large problems. We are also investigating new approximation algorithms for I-POMDPs. A candidate is to replace the exact branch-and-bound with MCTS to scale the method presented in this article to larger state and action spaces as well. An integration with existing multiagent simulation platforms (Luke, Cioffi-Revilla, Panait, Sullivan, & Balan, 2005; Bogert, Solaimanpour, & Doshi, 2015) to illustrate the behavior of agent populations would be interesting.

Acknowledgments

This research was supported in part by a NSF CAREER grant, IIS-0845036, and a grant from ONR, N000141310870. We thank Brenda Ng at Lawrence Livermore National Laboratory, Pradeep Varakantham at Singapore Management University, and several anonymous reviewers for valuable feedback that led to improvements in the article.

Appendix A.

We summarize key notation and its meaning as used in this article, in the table below.

S, s	Set of physical states, member of this set
A	Set of joint actions of all agents
a_0, $\mathbf{a}_{-0}$	Agent 0's action, joint action of all agents other than 0
T_0	Agent 0's transition function
Ω_0, ω_0	Set of agent 0's observations, member of this set
O_0	Agent 0's observation function
R_0	Agent 0's reward function
OC_0	Agent 0's optimality criterion
N	Number of agents
$M_{j,l-1}$, $m_{j,l-1}$	Set of level $l-1$ models of some agent, member of this set
$b_{0,l}$	Agent 0's level 0 belief
$\mathbf{m}_{-0,l-1}$	Joint model of all agents other than 0
$\hat{\Theta}_j$, $\hat{\theta}_j$	Frame of some agent, member of this set
$\mathcal{C}(\cdot)$	Action configuration derived from the argument
Ψ, ψ	Set of nodes in a graph, member of this set
E	Set of edges in the graph
$\nu(\cdot)$	Frame-action neighborhood of the argument
X_k, x_k	Set of values of the k^th state variable, member of this set

$\overline{V}^h_{0,l}$, $\overline{Q}^h_{0,l}$	Upper bound on agent 0's horizon-h value function, upper bound on its action value
$\underline{V}^h_{0,l}$, $\underline{Q}^h_{0,l}$	Lower bound on agent 0's horizon-h value function, lower bound on its action value

Table 6: Key notation and its meaning.

Appendix B.

We present detailed proofs for Propositions 2 and 3 in this appendix.

Proposition 2 (Pessimistic Blind Policy)**.** *The pessimistic blind policy yields a lower bound on the exact value of the many-agent I-POMDP.*

Proof. The proof uses mathematical induction on the horizon. We begin with the horizon 1 action-value function and obtain a lower bound for it.

$$Q^1_{0,l}(b^t_{0,l}, a^t_0) = \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t)\Big\{ \sum_{k=1}^{K} \sum_{C^{\nu(x^t_k, a^t_0)}} Pr(C^{\nu(x^t_k,a^t_0)}|b_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b_{0,l}(M_{N,l-1}|\mathbf{s}^t))$$
$$R_0(x^t_k, a^t_0, C^{\nu(x^t_k,a^t_0)})\Big\}$$

Let $\underline{C}^{\nu(x^t_k,a^t_0)}$ be the configuration that minimizes R_0. We get,

$$Q^1_{0,l}(b^t_{0,l}, a^t_0) \geq \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t)\Big\{ \sum_{k=1}^{K} \sum_{C^{\nu(x^t_k, a^t_0)}} Pr(C^{\nu(x^t_k,a^t_0)}|b_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b_{0,l}(M_{N,l-1}|\mathbf{s}^t))$$
$$\min_{C^{\nu(x^t_k,a^t_0)}} R_0(x^t_k, a^t_0, C^{\nu(x^t_k,a^t_0)})\Big\}$$
$$= \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t)\Big\{ \sum_{k=1}^{K} R_0(x^t_k, a^t_0, \underline{C}^{\nu(x^t_k,a^t_0)}) \sum_{C^{\nu(x^t_k,a^t_0)}} Pr(C^{\nu(x^t_k,a^t_0)}|b_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b_{0,l}(M_{N,l-1}|$$
$$\mathbf{s}^t))\Big\}$$
$$= \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{k=1}^{K} R_0(x^t_k, a^t_0, \underline{C}^{\nu(x^t_k,a^t_0)})$$
$$= \underline{Q}^1_{0,l}(b^t_{0,l}, a^t_0)$$

Therefore the lower bound obtained by selecting the minimizing configuration for the horizon 1 action-value is: $\underline{Q}^1_{0,l}(b^t_{0,l}, a^t_0) = \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{k=1}^{K} R_0(x^t_k, a^t_0, \underline{C}^{\nu(x^t_k,a^t_0)})$. Analogously to POMDPs, we may decompose this lower bound value using alpha vectors as $\underline{Q}^1_{0,l}(b^t_{0,l}, a^t_0) = \sum_{\mathbf{s}} \underline{\alpha}^1(\mathbf{s}) \cdot b^t_{0,l}(\mathbf{s})$, where $\underline{\alpha}^1(\mathbf{s}) = \sum_{k=1}^{K} R_0(x^t_k, a^t_0, \underline{C}^{\nu(x^t_k,a^t_0)})$.

Let us assume that for horizon $h-1$, the pessimistic blind policy provides a lower bound to the value function and could be represented using a set of alpha vectors over states only similar to

the above decomposition. We derive the lower bound value vectors for horizon h. For simplicity of derivation, we do not mention the configuration contexts for convenience; these are same as before.

$$Q^h_{0,l}(b^t_{0,l}, a^t_0) = Q^1_{0,l}(b^t_{0,l}, a^t_0) + \gamma \sum_{\omega_0^{t+1}} \max_{\alpha^{h-1}} \Bigg[\sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{\mathbf{s}^{t+1}} \Bigg\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \\ \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))\ \mathcal{O}_0(\mathbf{s}^{t+1}, a^t_0, C, \omega_0^{t+1}) \Bigg\} \Bigg\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t)) \\ \times \mathcal{T}_0(\mathbf{s}^t, a^t_0, C, \mathbf{s}^{t+1}) \Bigg\} \sum_{\mathbf{m}_{-0}^{t+1}} \prod_{j=1}^{N} Pr(m^{t+1}_{j,l-1}|\mathbf{s}^{t+1}, b^t_{0,l}, a^t_0)\ \alpha^{h-1}(\mathbf{s}^{t+1}, \mathbf{m}^{t+1}_{-0}) \Bigg]$$

At this point, we make the blind policy modification. The agent chooses the same policy regardless of the observations. Therefore, we may take the maximization over α^{h-1} outside the summation over observations analogously to Eq. 2 in Section 2.

$$Q^h_{0,l}(b^t_{0,l}, a^t_0) \geq Q^1_{0,l}(b^t_{0,l}, a^t_0) + \gamma \max_{\alpha^{h-1}} \Bigg[\sum_{\omega_0^{t+1}} \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{\mathbf{s}^{t+1}} \Bigg\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \\ \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))\ \mathcal{O}_0(\mathbf{s}^{t+1}, a^t_0, C, \omega_0^{t+1}) \Bigg\} \Bigg\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t)) \\ \times \mathcal{T}_0(\mathbf{s}^t, a^t_0, C, \mathbf{s}^{t+1}) \Bigg\} \sum_{\mathbf{m}_{-0}^{t+1}} \prod_{j=1}^{N} Pr(m^{t+1}_{j,l-1}|\mathbf{s}^{t+1}, b^t_{0,l}, a^t_0)\ \alpha^{h-1}(\mathbf{s}^{t+1}, \mathbf{m}^{t+1}_{-0}) \Bigg] \\ = Q^1_{0,l}(b^t_{0,l}, a^t_0) + \gamma \max_{\alpha^{h-1}} \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{\mathbf{s}^{t+1}} \Bigg\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t)) \\ \times \mathcal{T}_0(\mathbf{s}^t, a^t_0, C, \mathbf{s}^{t+1}) \Bigg\} \sum_{\mathbf{m}_{-0}^{t+1}} \prod_{j=1}^{N} Pr(m^{t+1}_{j,l-1}|\mathbf{s}^{t+1}, b^t_{0,l}, a^t_0)\ \alpha^{h-1}(\mathbf{s}^{t+1}, \mathbf{m}^{t+1}_{-0}) \\ = Q^1_{0,l}(b^t_{0,l}, a^t_0) + \gamma \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{\mathbf{s}^{t+1}} \Bigg\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t)) \\ \times \mathcal{T}_0(\mathbf{s}^t, a^t_0, C, \mathbf{s}^{t+1}) \Bigg\} \sum_{\mathbf{m}_{-0}^{t+1}} \prod_{j=1}^{N} Pr(m^{t+1}_{j,l-1}|\mathbf{s}^{t+1}, b^t_{0,l}, a^t_0)\ \alpha^{h-1}_*(\mathbf{s}^{t+1}, \mathbf{m}^{t+1}_{-0})$$

where α^{h-1}_* is the vector that maximizes the second term given belief $b^t_{0,l}$ and action a^t_0.

Next, we select minimizing configurations while computing Q^1 and α^{h-1}_* to obtain their pessimistic counterparts, which form corresponding lower bounds as we know from the base case and inductive hypothesis. The action-value above becomes,

$$\geq \underline{Q}^1_{0,l}(b^t_{0,l}, a^t_0) + \gamma \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{\mathbf{s}^{t+1}} \Bigg\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))$$
$$\times \mathcal{T}_0(\mathbf{s}^t, a^t_0, C, \mathbf{s}^{t+1}) \Bigg\} \sum_{\mathbf{m}^{t+1}_{-0}} \prod_{j=1}^{N} Pr(m^{t+1}_{j,l-1}|\mathbf{s}^{t+1}, b^t_{0,l}, a^t_0)\ \underline{\alpha}^{h-1}_*(\mathbf{s}^{t+1})$$
$$= \underline{Q}^1_{0,l}(b^t_{0,l}, a^t_0) + \gamma \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{\mathbf{s}^{t+1}} \Bigg\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))$$
$$\times \mathcal{T}_0(\mathbf{s}^t, a^t_0, C, \mathbf{s}^{t+1}) \Bigg\} \underline{\alpha}^{h-1}_*(\mathbf{s}^{t+1}) \sum_{\mathbf{m}^{t+1}_{-0}} \prod_{j=1}^{N} Pr(m^{t+1}_{j,l-1}|\mathbf{s}^{t+1}, b^t_{0,l}, a^t_0)$$
$$= \underline{Q}^1_{0,l}(b^t_{0,l}, a^t_0) + \gamma \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{\mathbf{s}^{t+1}} \Bigg\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))$$
$$\times \mathcal{T}_0(\mathbf{s}^t, a^t_0, C, \mathbf{s}^{t+1}) \Bigg\} \underline{\alpha}^{h-1}_*(\mathbf{s}^{t+1})$$

Notice that the trailing term $\sum_{\mathbf{m}^{t+1}_{-0}} \prod_{j=1}^{N} Pr(m^{t+1}_{j,l-1}|\mathbf{s}^{t+1}, b^t_{0,l}, a^t_0)$ sums to 1. We replace the configuration in $\mathcal{T}_0$ for each context with the one that minimizes the transition probability. The above action-value is

$$\geq \underline{Q}^1_{0,l}(b^t_{0,l}, a^t_0) + \gamma \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{\mathbf{s}^{t+1}} \Bigg\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))$$
$$\times \mathcal{T}_0(\mathbf{s}^t, a^t_0, \underline{C}, \mathbf{s}^{t+1}) \Bigg\} \underline{\alpha}^{h-1}_*(\mathbf{s}^{t+1})$$
$$= \underline{Q}^1_{0,l}(b^t_{0,l}, a^t_0) + \gamma \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{\mathbf{s}^{t+1}} \mathcal{T}_0(\mathbf{s}^t, a^t_0, \underline{C}, \mathbf{s}^{t+1})\ \underline{\alpha}^{h-1}_*(\mathbf{s}^{t+1})$$

Finally, for completeness we write the action-value lower bound in full.

$$\underline{Q}^h_{0,l}(b^t_{0,l}, a^t_0) = \sum_{s^t} b^t_{0,l}(s^t) \sum_{k=1}^{K} R_0(x^t_k, a^t_0, \underline{C}^{\nu(x^t_k, a^t_0)}) + \gamma \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{\mathbf{s}^{t+1}} \prod_{k=1}^{K} \mathcal{T}_0(x^t_k, a^t_0,$$
$$\underline{C}^{\nu(x^t_k, a^t_0, x^{t+1}_k)}, x^{t+1}_k)\ \underline{\alpha}^{h-1}_*(\mathbf{s}^{t+1})$$

□

Proposition 3 (Optimistic Fast Informed Bound)**.** *The fast informed update with maximizing configurations gives an upper bound to the many-agent I-POMDP value function.*

Proof. The proof here also uses mathematical induction on the horizon. The base case is the horizon 1 action-value function and we obtain an upper bound for it.

$$Q^1_{0,l}(b^t_{0,l}, a^t_0) = \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{k=1}^{K} \Bigg\{ \sum_{C^{\nu(x^t_k,a^t_0)}} Pr(C^{\nu(x^t_k,a^t_0)}|b_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b_{0,l}(M_{N,l-1}|\mathbf{s}^t))$$

$$R_0(x^t_k, a^t_0, C^{\nu(x^t_k,a^t_0)}) \Bigg\}$$

Let $\overline{C}^{\nu(x^t_k,a^t_0)}$ be the configuration that maximizes R_0. We get,

$$Q^1_{0,l}(b^t_{0,l}, a^t_0) \geq \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{k=1}^{K} \Bigg\{ \sum_{C^{\nu(x^t_k,a^t_0)}} Pr(C^{\nu(x^t_k,a^t_0)}|b_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b_{0,l}(M_{N,l-1}|\mathbf{s}^t))$$

$$\max_{C^{\nu(x^t_k,a^t_0)}} \mathcal{R}_0(x^t_k, a^t_0, C^{\nu(x^t_k,a^t_0)}) \Bigg\}$$

$$= \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{k=1}^{K} \Bigg\{ \mathcal{R}_0(x^t_k, a^t_0, \overline{C}^{\nu(x^t_k,a^t_0)}) \sum_{C^{\nu(x^t_k,a^t_0)}} Pr(C^{\nu(x^t_k,a^t_0)}|b_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b_{0,l}(M_{N,l-1}|$$

$$\mathbf{s}^t)) \Bigg\}$$

$$= \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{k=1}^{K} \mathcal{R}_0(x^t_k, a^t_0, \overline{C}^{\nu(x^t_k,a^t_0)})$$

$$= \overline{Q}^1_{0,l}(b^t_{0,l}, a^t_0)$$

Therefore an upper bound for the horizon 1 value is obtained as follows: $\overline{Q}^1_{0,l}(b_{0,l}, a^t_0) = \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t)$ $\times \sum_{k=1}^{K} \mathcal{R}_0(x^t_k, a^t_0, \overline{C}^{\nu(x^t_k,a^t_0)})$. These upper bounds may be decomposed into a set of vectors one for each action a_0, $\overline{\alpha}^1_{a_0}(\mathbf{s}) = \sum_{k=1}^{K} \mathcal{R}_0(x^t_k, a^t_0, \overline{C}^{\nu(x^t_k,a^t_0)})$, and each vector has as many components as states.

Let $\overline{\Gamma}^{h-1}_*$ be the set of alpha vectors for horizon $h-1$ that are obtained by taking the fast informed approximation with maximizing configurations. There are as many vectors in this set as the number of actions $|A_0|$. As the inductive hypothesis, let the vectors in $\overline{\Gamma}^{h-1}_*$ form an upper bound.

Next we derive the upper bound on horizon h value function.

$$Q^h_{0,l}(b^t_{0,l}, a^t_0) = Q^1_{0,l}(b^t_{0,l}, a^t_0) + \gamma \sum_{\omega^{t+1}_0} \max_{\alpha^{h-1}} \Bigg[\sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \sum_{\mathbf{s}^{t+1}} \Bigg\{ \sum_{C} Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t),$$

$$\ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))\ \mathcal{O}_0(\mathbf{s}^{t+1}, a^t_0, C, \omega^{t+1}_0) \Bigg\} \Bigg\{ \sum_{C} Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))$$

$$\times\ \mathcal{T}_0(\mathbf{s}^t, a^t_0, C, \mathbf{s}^{t+1}) \Bigg\} \sum_{\mathbf{m}^{t+1}_{-0}} \prod_{j=1}^{N} Pr(m^{t+1}_{j,l-1}|\mathbf{s}^{t+1}, b^t_{0,l}, a^t_0)\ \alpha^{h-1}(\mathbf{s}^{t+1}, \mathbf{m}^{t+1}_{-0}) \Bigg]$$

We replace the backup above with the fast-informed backup that pushes the maximization over α^{h-1} deeper into the backup.

$$Q^h_{0,l}(b^t_{0,l}, a^t_0) \leq Q^1_{0,l}(b^t_{0,l}, a^t_0) + \gamma \sum_{\omega^{t+1}_0} \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \max_{\alpha^{h-1}} \Big[\sum_{\mathbf{s}^{t+1}} \Big\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \\ \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))\ \mathcal{O}_0(\mathbf{s}^{t+1}, a^t_0, C, \omega^{t+1}_0) \Big\} \Big\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t)) \\ \times \mathcal{T}_0(\mathbf{s}^t, a^t_0, C, \mathbf{s}^{t+1}) \Big\} \sum_{\mathbf{m}^{t+1}_{-0}} \prod_{j=1}^{N} Pr(m^{t+1}_{j,l-1}|\mathbf{s}^{t+1}, b^t_{0,l}, a^t_0)\ \alpha^{h-1}(\mathbf{s}^{t+1}, \mathbf{m}^{t+1}_{-0}) \Big]$$

Analogously to Section 2, we may factor out the current belief. The above Q-value becomes,

$$\sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \Big\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t)) \mathcal{R}_0(\mathbf{s}^t, a^t_0, C) + \gamma \sum_{\omega^{t+1}_0} \max_{\alpha^{h-1}} \Big[\sum_{\mathbf{s}^{t+1}} \\ \Big\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))\ \mathcal{O}_0(\mathbf{s}^{t+1}, a^t_0, C, \omega^{t+1}_0) \Big\} \Big\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}| \\ \mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t)) \mathcal{T}_0(\mathbf{s}^t, a^t_0, C, \mathbf{s}^{t+1}) \Big\} \sum_{\mathbf{m}^{t+1}_{-0}} \prod_{j=1}^{N} Pr(m^{t+1}_{j,l-1}|\mathbf{s}^{t+1}, b^t_{0,l}, a^t_0)\ \alpha^{h-1}(\mathbf{s}^{t+1}, \mathbf{m}^{t+1}_{-0}) \Big] \Big\} \\ = \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t) \Big\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t)) \mathcal{R}_0(\mathbf{s}^t, a^t_0, C) + \gamma \sum_{\omega^{t+1}_0} \sum_{\mathbf{s}^{t+1}} \\ \Big\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))\ \mathcal{O}_0(\mathbf{s}^{t+1}, a^t_0, C, \omega^{t+1}_0) \Big\} \Big\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}| \\ \mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t)) \mathcal{T}_0(\mathbf{s}^t, a^t_0, C, \mathbf{s}^{t+1}) \Big\} \sum_{\mathbf{m}^{t+1}_{-0}} \prod_{j=1}^{N} Pr(m^{t+1}_{j,l-1}|\mathbf{s}^{t+1}, b^t_{0,l}, a^t_0)\ \alpha^{h-1}_*(\mathbf{s}^{t+1}, \mathbf{m}^{t+1}_{-0}) \Big\}$$

We do not show the contexts of the configurations above to promote readability; these are easy to infer. Let us replace the configuration in $\mathcal{R}_0$ with the one that maximizes it given the context (as in the base case) and use the inductive hypothesis to replace the optimizing horizon $h-1$ vectors with their upper bounds. The above Q-value is,

$$\leq \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t)\Big\{\mathcal{R}_0(\mathbf{s}^t, a^t_0, \overline{C}) + \gamma \sum_{\omega_0^{t+1}} \sum_{\mathbf{s}^{t+1}} \Big\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))$$
$$\times \mathcal{O}_0(\mathbf{s}^{t+1}, a^t_0, C, \omega_0^{t+1})\Big\}\Big\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))\mathcal{T}_0(\mathbf{s}^t, a^t_0, C, \mathbf{s}^{t+1})\Big\}$$
$$\times \sum_{\mathbf{m}^{t+1}_{-0}} \prod_{j=1}^{N} Pr(m^{t+1}_{j,l-1}|\mathbf{s}^{t+1}, b^t_{0,l}, a^t_0)\, \overline{\alpha}^{h-1}_*(\mathbf{s}^{t+1})\Big\}$$
$$= \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t)\Big\{\mathcal{R}_0(\mathbf{s}^t, a^t_0, \overline{C}) + \gamma \sum_{\omega_0^{t+1}} \sum_{\mathbf{s}^{t+1}} \Big\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))$$
$$\times \mathcal{O}_0(\mathbf{s}^{t+1}, a^t_0, C, \omega_0^{t+1})\Big\}\Big\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))\mathcal{T}_0(\mathbf{s}^t, a^t_0, C, \mathbf{s}^{t+1})\Big\}$$
$$\times\ \overline{\alpha}^{h-1}_*(\mathbf{s}^{t+1}) \sum_{\mathbf{m}^{t+1}_{-0}} \prod_{j=1}^{N} Pr(m^{t+1}_{j,l-1}|\mathbf{s}^{t+1}, b^t_{0,l}, a^t_0)\Big\}$$
$$= \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t)\Big\{\mathcal{R}_0(\mathbf{s}^t, a^t_0, \overline{C}) + \gamma \sum_{\omega_0^{t+1}} \sum_{\mathbf{s}^{t+1}} \Big\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))$$
$$\times \mathcal{O}_0(\mathbf{s}^{t+1}, a^t_0, C, \omega_0^{t+1})\Big\}\Big\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))\mathcal{T}_0(\mathbf{s}^t, a^t_0, C, \mathbf{s}^{t+1})\Big\}$$
$$\times\ \overline{\alpha}^{h-1}_*(\mathbf{s}^{t+1})\Big\}$$

Thus, the updated models of the other agents sum out. Next, we replace configurations in the observation and transition functions of agent 0 with those that maximize the respective probabilities given the context. Action value $Q^h_{0,l}(b^t_{0,l}, a^t_0)$ is,

$$\leq \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t)\Big\{\mathcal{R}_0(\mathbf{s}^t, a^t_0, \overline{C}) + \gamma \sum_{\omega_0^{t+1}} \sum_{\mathbf{s}^{t+1}} \Big\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))$$
$$\times \mathcal{O}_0(\mathbf{s}^{t+1}, a^t_0, \overline{C}, \omega_0^{t+1})\Big\}\Big\{ \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))\ \mathcal{T}_0(\mathbf{s}^t, a^t_0, \overline{C}, \mathbf{s}^{t+1})\Big\}$$
$$\times\ \overline{\alpha}^{h-1}_*(\mathbf{s}^{t+1})\}$$
$$= \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t)\Big\{\mathcal{R}_0(\mathbf{s}^t, a^t_0, \overline{C}) + \gamma \sum_{\omega_0^{t+1}} \sum_{\mathbf{s}^{t+1}} \Big\{ \mathcal{O}_0(\mathbf{s}^{t+1}, a^t_0, \overline{C}, \omega_0^{t+1}) \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots,$$
$$b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))\Big\}\Big\{\mathcal{T}_0(\mathbf{s}^t, a^t_0, \overline{C}, \mathbf{s}^{t+1}) \sum_C Pr(C|b^t_{0,l}(M_{1,l-1}|\mathbf{s}^t), \ldots, b^t_{0,l}(M_{N,l-1}|\mathbf{s}^t))\Big\}$$
$$\times\ \overline{\alpha}^{h-1}_*(\mathbf{s}^{t+1})\}$$

This becomes,

$$Q^h_{0,l}(b^t_{0,l}, a^t_0) = \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t)\Big\{\mathcal{R}_0(\mathbf{s}^t, a^t_0, \overline{C}) + \gamma \sum_{\omega^{t+1}_0} \sum_{\mathbf{s}^{t+1}} \mathcal{O}_0(\mathbf{s}^{t+1}, a^t_0, \overline{C}, \omega^{t+1}_0)\, \mathcal{T}_0(\mathbf{s}^t, a^t_0, \overline{C}, \mathbf{s}^{t+1}) \times\ \overline{\alpha}^{h-1}_*(\mathbf{s}^{t+1})\Big\}$$
$$= \sum_{\mathbf{s}^t} b^t_{0,l}(\mathbf{s}^t)\, \overline{\alpha}^h_{a^t_0}(s^t)$$

□

References

Bogert, K., Solaimanpour, S., & Doshi, P. (2015). Aerial robotic simulations for evaluation of multi-agent planning in gatac (demonstration). In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pp. 1919–1920.

Boutilier, C., Friedman, N., Goldszmidt, M., & Koller, D. (1996). Context-specific independence in bayesian networks. In *Twelfth international conference on Uncertainty in artificial intelligence (UAI)*, pp. 115–123.

Chandrasekaran, M., Doshi, P., Zeng, Y., & Chen, Y. (2014). Team behavior in interactive dynamic influence diagrams with applications to ad hoc teams (extended abstract). In *Autonomous Agents and Multi-Agent Systems Conference (AAMAS)*, pp. 1559–1560.

Doshi, P. (2012). Decision making in complex multiagent settings: A tale of two frameworks. *AI Magazine*, *33*(4), 82–95.

Doshi, P., Qu, X., Goodie, A., & Young, D. (2010). Modeling recursive reasoning in humans using empirically informed interactive POMDPs. In *International Autonomous Agents and Multiagent Systems Conference (AAMAS)*, pp. 1223–1230.

Durfee, E., & Zilberstein, S. (2013). *Multiagent Systems* (Second edition)., chap. Multiagent Planning, Control and Execution, pp. 485–546. MIT Press.

Fudenberg, D., & Tirole, J. (1991). *Game Theory*. MIT Press.

Gmytrasiewicz, P. J., & Doshi, P. (2005). A framework for sequential planning in multiagent settings. *Journal of Artificial Intelligence Research*, *24*, 49–79.

Guestrin, C., Koller, D., & Parr, R. (2001). Multiagent planning with factored mdps. In *Neural Information Processing Systems (NIPS)*, pp. 1523–1530.

Hauskrecht, M. (1997). *Planning and control in stochastic domains with imperfect information*. Ph.D. thesis, MIT.

Hauskrecht, M. (2000). Value-function approximations for partially observable Markov decision process. *Journal of Artificial Intelligence*, *13*, 33–94.

Hula, A., Montague, P. R., & Dayan, P. (2015). Monte carlo planning method estimates planning horizons during interactive social exchange. *PLOS Computational Biology*, *11*, 1–38.

Jiang, A. X., & Leyton-Brown, K. (2010). Bayesian action-graph games. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 991–999.

Jiang, A. X., Leyton-Brown, K., & Bhat, N. A. (2011). Action-graph games. *Games and Economic Behavior*, *71*(1), 141–173.

Jiang, A. X., Leyton-Brown, K., & Pfeffer, A. (2009). Temporal action-graph games: A new representation for dynamic games. In *Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 268–276.

Kaelbling, L., Littman, M., & Cassandra, A. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*, 99–134.

Kaelbling, L. P. (1993). *Learning in Embedded Systems*. MIT Press.

Kearns, M., Littman, M., & Singh, S. (2001). Graphical models for game theory. In *Uncertainty in Artificial Intelligence (UAI)*, pp. 253–260.

Koller, D., & Milch, B. (2001). Multi-agent influence diagrams for representing and solving games. In *IJCAI*, pp. 1027–1034.

Kurniawati, H., Hsu, D., & Lee, W. S. (2008). SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems*.

Land, A. H., & Doig, A. G. (1960). An automatic method of solving discrete programming problems. *Econometrica*, *28*(3), 109–132.

Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., & Balan, G. (2005). Mason: A multi-agent simulation environment. *Simulation: Transactions of the Society for Modeling and Simulation*, *82*(7), 517–527.

Mostafa, H., & Lesser, V. (2009). Offline planning for communication by exploiting structured interactions in decentralized mdps. In *International Conference on Web Intelligence and Intelligent Agent Tech- nology (IAT)*, pp. 193–200.

Nair, R., Varakantham, P., Tambe, M., & Yokoo, M. (2005). Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs. In *Twentieth AAAI Conference on Artificial Intelligence*, pp. 133–139.

Ng, B., Meyers, C., Boakye, K., & Nitao, J. (2010). Towards applying interactive POMDPs to real-world adversary modeling. In *Innovative Applications in Artificial Intelligence (IAAI)*, pp. 1814–1820.

Poupart, P., & Boutilier, C. (2003). Bounded finite state controllers. In *Neural Information Processing Systems*.

Robbel, P., Oliehoek, F., & Kochenderfer, M. J. (2016). Exploiting anonymity in approximating linear programming: Scaling to large multiagent MDPs. In *Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*, pp. 2537–2543.

Roughgarden, T., & Tardos, E. (2002). How bad is selfish routing?. *Journal of ACM*, *49*(2), 236–259.

Seuken, S., & Zilberstein, S. (2008). Formal models and algorithms for decentralized decision making under uncertainty. *Journal of Autonomous Agents and Multiagent Systems*, *17*(2), 190–250.

Seymour, R., & Peterson, G. L. (2009). A trust-based multiagent system. In *IEEE International Conference on Computational Science and Engineering*, pp. 109–116.

Shani, G., Brafman, R., & Shimony, S. (2007). Forward search value iteration for POMDPs. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2619–2624.

Smallwood, R., & Sondik, E. (1973). The optimal control of partially observable Markov decision processes over a finite horizon. *Operations Research*, *21*, 1071–1088.

Smith, T., & Simmons, R. (2004). Heuristic search value iteration for POMDPs. In *Uncertainty in Artificial Intelligence (UAI)*, pp. 520–527.

Sonu, E., & Doshi, P. (2015). Scalable solutions of interactive POMDPs using generalized and bounded policy iteration. *Journal of Autonomous Agents and Multi-Agent Systems*, *29*(3), 455–494.

Varakantham, P., Adulyasak, Y., & Jaillet, P. (2014). Decentralized stochastic planning with anonymity in interactions. In *AAAI Conference on Artificial Intelligence*, pp. 2505–2511.

Varakantham, P., Cheng, S., Gordon, G., & Ahmed, A. (2012). Decision support for agent populations in uncertain and congested environments. In *Uncertainty in artificial intelligence (UAI)*, pp. 1471–1477.

Varakantham, P., Kwak, J. Y., Taylor, M., Marecki, J., Scerri, P., & Tambe, M. (2009). Exploiting coordination locales in distributed pomdps via social model shaping. In *International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 313–320.

Wang, F. (2013). An I-POMDP based multi-agent architecture for dialogue tutoring. In *International Conference on Advanced Information and Communication Technology for Education (ICAICTE)*, pp. 486–489.

Witwicki, S. J., & Durfee, E. H. (2010). Influence-based policy abstraction for weakly-coupled Dec-POMDPs. In *20th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 185–192.

Woodward, M. P., & Wood, R. J. (2012). Learning from humans as an I-POMDP. *CoRR*, *abs/1204.0274*.

Wunder, M., Kaisers, M., Yaros, J., & Littman, M. (2011). Using iterated reasoning to predict opponent strategies. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pp. 593–600.