

Scalable Decision-Theoretic Planning in Open and Typed Multiagent Systems

Adam Eck,¹ Maulik Shah,² Prashant Doshi,², Leen-Kiat Soh³

¹Oberlin College, ²University of Georgia, ³University of Nebraska
aeck@oberlin.edu, mns28652@uga.edu, pdoshi@uga.edu, lksoh@cse.unl.edu

Abstract

In open agent systems, the set of agents that are cooperating or competing changes over time and in ways that are non-trivial to predict. For example, if collaborative robots were tasked with fighting wildfires, they may run out of suppressants and be temporarily unavailable to assist their peers. We consider the problem of planning in these contexts with the additional challenges that the agents are unable to communicate with each other and that there are many of them. Because an agent’s optimal action depends on the actions of others, each agent must not only predict the actions of its peers, but, before that, reason whether they are even present to perform an action. Addressing openness thus requires agents to model each others presence, which becomes computationally intractable with high numbers of agents. We present a novel, principled, and scalable method in this context that enables an agent to reason about others’ presence in its shared environment and their actions. Our method extrapolates models of a few peers to the overall behavior of the many-agent system, and combines it with a generalization of Monte Carlo tree search to perform individual agent reasoning in many-agent open environments. Theoretical analyses establish the number of agents to model in order to achieve acceptable worst case bounds on extrapolation error, as well as regret bounds on the agent’s utility from modeling only some neighbors. Simulations of multiagent wildfire suppression problems demonstrate our approach’s efficacy compared with alternative baselines.

Introduction

In both cooperative and competitive multiagent systems (MAS), a participating agent benefits from reasoning about how other agents will behave while choosing optimal actions that maximize its chances of accomplishing shared or self-interested goals. However, nuances in real-world environments often challenge straightforward peer modeling. One of these is **agent openness** occurring whenever individual agents join or leave the system (temporarily or permanently) over time. For example, cooperative robots tasked with suppressing wildfires alongside or in place of human fire fighters would need to periodically leave the environment to

recharge their limited suppressants that were spent during firefighting. Likewise, competitive autonomous ride-sharing cars can no longer compete for new passengers while transporting a full ride. Consequently, openness requires that an agent not only predict *what* actions their neighbors will take, but also *whether* they are even present to take actions. Inaccurately predicting the presence or absence of others may cause agents to choose actions that miss their intended benefits or other utility maximizing opportunities.

Furthermore, real-world environments such as wildfire suppression often involve *many agents of various types* working together to put out the wildfires. Thus, the modeling and reasoning must additionally scale with the number of agents. As scalability is a general and ever-present challenge for multiagent planning, investigations have first focused on multiagent planning for smaller groups of agents, e.g., (Amato and Oliehoek 2015; Claes et al. 2017; Hoang and Low 2013; Nair et al. 2005; Oliehoek et al. 2008). Subsequently, there has been some focus on scalable planning in large groups of agents (Oliehoek, Whiteson, and Spaan 2013; Nguyen, Kumar, and Lau 2017; Sonu, Chen, and Doshi 2017; Velagapudi et al. 2011) albeit, in the absence of openness. Indeed, state-of-the-art multiagent planning in open environments has demonstrated successful planning with up to 5 agents (Chandrasekaran et al. 2016; Cohen, Dibangoye, and Mouaddib 2017). Of course, openness compounds the challenges of scalable reasoning since not only should the agent model how its many peers decide their actions, but also how they dynamically join and leave the environment.

In this paper, we focus on achieving scalable multiagent reasoning in open environments. We improve on a previous approach to individual agent planning in the context of agent openness (Chandrasekaran et al. 2016) to consider many-agent settings. To promote flexible scalability, our method enables a subject agent to intelligently select a small subset of neighbors to explicitly model. Then, it extrapolates their expected behaviors to the larger set of all agents. This approach has its roots in surveying and polling theory where surveyors rarely poll all individuals in their target class. As such, this approach represents a novel integration of survey theory in multiagent planning. However, selective modeling may not be enough for many-agent systems. We observe that

in many open domains such as wildfire suppression, which particular firefighter is performing an action is not relevant to the decision making. Similar observations have been made in other domains (Velagapudi et al. 2011; Nguyen, Kumar, and Lau 2017; Sonu, Chen, and Doshi 2017). Consequently, further efficiency is made possible by exploiting the property of anonymity of the other agents (Sonu, Chen, and Doshi 2017). These modeling approaches are utilized in a new generalization of Monte Carlo tree search (Silver and Veness 2010) to plan in open, many-agent settings.

Our results in this paper are both theoretical and experimental. We analytically establish (i) the number of neighbors that a subject agent must model to achieve given worst case extrapolation errors, and (ii) the regret bounds on the agent’s approximate utility function from modeling some neighbors only. To empirically demonstrate scalability and evaluate the benefits of our approach, we extend benchmark simulations of wildfire suppression (Chandrasekaran et al. 2016) to include setups with up to 50 agents – an order of magnitude larger than the prior literature – capable of performing up to 1.27×10^{30} joint actions, for which existing decision-theoretic planning does not scale.

Background

We review the planning framework and a previous approach for integrating anonymity into reasoning about others.

I-POMDP-Lite Framework

A framework for individual agent planning in partially observable MAS is the I-POMDP-Lite (Hoang and Low 2013). This framework retains many of the benefits of the more general I-POMDP framework but mitigates computational costs by modeling the other agents’ reasoning processes as an approximation of their true decision making. Formally,

$$\text{IPOMDP-Lite}_{i,l}^{\mathcal{L}} \triangleq \langle Ag, S, A, \Omega_i, T_i, O_i, R_i, \gamma, b_{i,0}, \{\mathcal{M}_{j,l-1}, \mathcal{M}_{k,l-1}, \dots, \mathcal{M}_{z,l-1}\} \rangle$$

- Ag is a set of agents, consisting of a *subject agent* i using the I-POMDP-Lite to decide how to act and other agents j, \dots, z modeled by subject i . Each agent has a frame θ from the discrete set $\Theta = \{\theta_1, \dots, \theta_{|\Theta|}\}$. A frame represents the agent’s capabilities and preferences.
- S is the set of states of the decision-making problem, possibly factored into variables $F_1 \times F_2 \times \dots \times F_k$, such as the intensities of the k wildfires the agents need to suppress.
- $A = A_i \times A_j \times \dots \times A_z$ is the set of possible joint actions of the agents; e.g., the individual fires that each agent chooses to fight. For notational convenience, $\mathbf{a}_{-i} \in A_j \times \dots \times A_z$ denotes the vector of actions by $Ag \setminus \{i\}$.
- Ω_i is the set of observations of agent i .
- $T_i(s, a_i, \mathbf{a}_{-i}, s') = P(s'|s, a_i, \mathbf{a}_{-i})$ gives the probabilities of stochastic state transitions caused by actions.
- $O_i(s', a_i, \mathbf{a}_{-i}, o_i) = P(o_i|a_i, \mathbf{a}_{-i}, s')$ models the probabilities of stochastic observations revealed to subject agent i after joint action (a_i, \mathbf{a}_{-i}) .
- $R_i(s, a_i, \mathbf{a}_{-i}) \in \mathbb{R}$ is the reward function of agent i dependent on the state and joint actions.

- $\gamma \in (0, 1]$ and $b_{i,0}$ are the discount factor and initial belief state of i , respectively.
- $\{\mathcal{M}_{j,l-1}, \mathcal{M}_{k,l-1}, \dots, \mathcal{M}_{z,l-1}\}$ is the set of mental models ascribed to the other agents j, k, \dots, z at level $l - 1$.¹ Each mental model $\mathcal{M}_{j,l-1}$ is a Nested-MDP. Hoang and Low (2013) define a Nested-MDP as:

$$\mathcal{M}_{i,l} \triangleq \langle S, A, T_i, R_i, \{\pi_{j,d}, \pi_{k,d}, \dots, \pi_{z,d}\}_{d=0}^{l-1}, \gamma \rangle$$

where $\{\pi_{j,d}, \pi_{k,d}, \dots, \pi_{z,d}\}_{d=0}^{l-1}$ is the set of policies followed by other agents j, k, \dots, z at levels 0 to $l - 1$. The policies are obtained by solving their own Nested-MDPs. In solving an I-POMDP-Lite, an agent chooses actions that maximize the cumulative, discounted reward function over a finite horizon H , $r_0 + \gamma r_1 + \gamma^2 r_2 + \dots + \gamma^{H-1} r_{H-1}$, by considering Bellman equations for each belief/action pair:

$$Q_{i,l}^t(b_i, a_i) = \rho_i(b_i, a_i) + \gamma \sum_{s', o_i} T_i^{a_i, o_i}(s', o_i | b_i, a_i) V_{i,l}^{t-1}(b'_i) \quad (1)$$

$$V_{i,l}^t(b'_i) = \max_{a_i \in A_i} Q_{i,l}^t(b_i, a_i) \quad (2)$$

where

$$\rho_i(b_i, a_i) = \sum_{s \in S} \sum_{\mathbf{a}_{-i} \in A_{-i}} \prod_{-i \in \{j,k,\dots,z\}} \pi_{-i,l-1}(s, a_{-i}) \times R_i(s, a_i, \mathbf{a}_{-i}) b_i(s) \quad (3)$$

Policies $\pi_{-i,l-1}(s, a_{-i})$ are solutions of the other agents’ mental models $\mathcal{M}_{-i,l-1}$, and b'_i denotes the updated belief $Pr(s'|o_i, a_i, \mathbf{a}_{-i}, b_i) \propto O_i(s', a_i, o_i) \sum_{s \in S} T_i(s, a_i, \mathbf{a}_{-i}, s') b_i(s)$. Hoang and Low (2013) establish regret bounds on the value function V based on any error introduced through approximating other agents as reasoning using a Nested-MDP (instead of their true reasoning process).

Both the I-POMDP-Lite and decentralized POMDP (Dec-POMDP) (Bernstein et al. 2002) have been proposed for reasoning in open environments (Chandrasekaran et al. 2016; Cohen, Dibangoye, and Mouaddib 2017). The I-POMDP-Lite (as a relative of the more general I-POMDP) differs from the Dec-POMDP in several important ways. First, the I-POMDP-Lite makes no assumptions that a subject agent observes the actual actions taken by other agents during operation in the environment, nor the observations they receive. Indeed, each agent can reason in a vacuum, inferring the behaviors of others through changes to the shared environment state, whereas Dec-POMDPs assume that everyone’s actions and observations are communicated or observed. This makes the I-POMDP-Lite appropriate for a broader range of cooperative (and even competitive) environments, including the wildfire suppression domain where agents might not be able to pre-coordinate their behaviors, directly observe other agents obscured by smoke and fire, nor share information during operation due to damaged communication channels.

¹The level of a reasoning process represents the level in a hierarchy of agents reasoning about their neighbors: level 1 implies all others choose actions randomly, and level $l > 1$ implies that all others reason at level $l - 1$.

Frame-Action Anonymity

In many domains, it does not matter *which* agents perform actions, rather *how many* agents of each frame perform each action. By relaxing agent identities, multiple joint actions may produce the same action counts, and, in turn, equivalent state transitions, observations, and rewards. For example, in wildfire suppression, agents of different frames (e.g., ground firefighters, helicopters) extinguish fires at different rates. But, each agent of the same frame contributes the same amount of fire suppression. Therefore, any k agents of the same frame identically impact the fire. Such domains exhibit **frame-action anonymity** (Sonu, Chen, and Doshi 2017).

Formally, let **configuration** $C = \langle n_{a_1, \theta_1}, \dots, n_{a_{|A|}, \theta_{|\Theta|}} \rangle$, where n_{a_1, θ_1} is the number of agents with frame θ_1 performing action a_1 in a joint action tuple (a_i, \mathbf{a}_{-i}) . Histories with configurations (rather than joint actions) are now sufficient statistics (Sonu, Chen, and Doshi 2017) and the representation of T_i , O_i , and R_i can be greatly compacted since the number of possible configurations $|\mathcal{C}| \propto \binom{|Ag|+m+1}{m+1}$ is polynomial in the number of agents and thus much fewer than the exponential number of possible joint actions $m^{|Ag|}$, where $m = \max\{|A_i|, |A_j|, \dots, |A_z|\}$ (c.f., Figure 1 below for comparisons of the number of configurations vs joint actions in our experiments). Therefore, configurations enabled by frame-action anonymity improve scalability of individual planning in many-agent environments, as shown in the context of I-POMDPs (Sonu, Chen, and Doshi 2017).

Notably, in decision-theoretic planning, configurations are random variables since agents do not directly observe the actions chosen by other agents, but instead the likelihoods of different configurations (i.e., counts of actions per frame) are estimated based on the policies $\pi_{-i, l-1}$ solved for each other agent’s mental model $M_{j, l-1}, \dots, M_{z, l-1}$.

Many-Agent Planning under Openness

In this paper, we limit our attention to systems where agents may leave the environment at any time and possibly reenter, but new agents do not enter the system. Still, this brings unique conceptual and computational challenges.

Challenges of Open Agent Systems

In open systems, individual planning is complicated by the need of each agent to track which other agents are currently present in the system and to reason about the actions of present agents only. In wildfire suppression, each firefighter must know how many others are currently unavailable due to recharging their suppressant, so as to focus on the behaviors of those currently fighting the fires.

Let $N(i) \subseteq Ag$ be the neighborhood of subject agent i , defined as the set of other agents with which i can interact in the environment and that affect its transition, observation, and reward functions (e.g., the other firefighting robots that can suppress the same wildfires as subject agent i). A naive way of tracking the presence of agents in open environments involves each agent i maintaining an additional state variable P_j for each neighbor $j \in N(i)$, where $P_j \in \{\text{present}, \text{absent}\}$. This increases the state space by a factor of $2^{|N(i)|}$ – an exponential increase in the size of

the agent’s state space! To illustrate, modeling the presence or absence of 40 neighbors in this way increases the state space by a factor of $2^{40} \gg 100$ billion states. To avoid this, we place P_j as an *internal state variable within a mental model* $M_{j, l-1}$ attributed by subject agent i to neighbor j . Consequently, the overall increase in the size of the problem is *linear* in the number of agents (scaled by the number of models ascribed to each agent), which promotes scalability.

Nonetheless, even a linear increase in the problem size poses challenges for planning. In offline planning methods such as the interactive point-based value iteration (Chandrasekaran et al. 2016) and interactive expectation-maximization (Qu and Doshi 2015), the subject agent must still consider each neighbor’s current presence and future transition, which adds at least a polynomial coefficient $O(|N(i)|^2)$ to the complexity of planning. Likewise, in online planning approaches such as Monte Carlo tree search algorithms for POMDPs and I-POMDPs (Claes et al. 2017; Hula, Montague, and Dayan 2015), the subject agent must spend time simulating the transitions of each neighbor when sampling an environment outcome, resulting in fewer sampled trajectories in the tree and thus lower quality approximations of the value function V . Overall, these challenges motivate further steps to alleviate the impact of openness on computational complexity.

Selectively Modeling Neighbors

It is well known that surveyors and pollsters do not model every target person individually to understand their attitudes and behaviors. Instead, the collective attitudes and behaviors of the whole are estimated by sampling those of a few, then extrapolating to everyone. Bounds on the approximation error of this approach can be established through statistical analysis and random sampling (Neyman 1934; Frankel and Frankel 1987; Lohr 2010).

Adapting this methodology to open, many-agent settings, we maintain mental models for a small subset of an agent’s neighbors, then extrapolate their predicted behaviors to the entire neighborhood (with bounded error). In the wildfire example, each firefighter models some of the firefighters with which it interacts, then uses their (fewer) predicted behaviors to estimate how many of the full set of firefighters are recharging or choosing to fight each fire. By explicitly modeling some neighbors only, we reduce the reasoning necessary for considering changes to every neighbors’ presence. Of course, the agent’s planning process no longer obtains a policy for *every* neighbor and thereby estimate the distribution of configurations necessary to estimate the state transitions, observations, and rewards (inside the $T_i^{a_i, o_i}(s', o_i | b_i, a_i)$ of Eq. 1 and product in Eq. 3). To mitigate this trade off, we use an alternative way to estimate configuration likelihoods when modeling a subset of the subject agent’s neighborhood.

Estimating configuration probabilities Recall that during planning, configurations C are random variables whose probability distributions typically depend on the policies obtained by the subject agent for its neighbors. Alternately, since $C = \langle n_{a_1, \theta_1}, \dots, n_{a_{|A|}, \theta_{|\Theta|}} \rangle$ is composed of the counts

of the number of agents of each frame performing each action, the distribution over C can also be represented as a *multinomial distribution*, parameterized as

$$P(C|s^t, M^t) \sim \text{Multi}(|N(i)|, \{p_{a_1, \theta_1, N(i)} \cdots p_{a_{|A|}, \theta_{|\Theta|}, N(i)}\})$$

where $p_{a, \theta, N(i)}$ is the probability that neighbors in $N(i)$ of frame θ will perform action a . We may view the configuration C as the concatenation of several multinomial variables, $C_\theta = \langle n_{a_1, \theta}, n_{a_2, \theta}, \dots, n_{a_{|A|}, \theta} \rangle$, one for each frame θ . This allows us to model the multinomial distribution over C using separate multinomial distributions for each C_θ ,

$$P(C_\theta|s^t, M^t) \sim \text{Multi}(|N_\theta(i)|, \{p_{a_1, N_\theta(i)}, \dots, p_{a_{|A|}, N_\theta(i)}\}) \quad (4)$$

where $N_\theta(i) = \{j \in N(i) | j \text{ has frame } \theta\}$ is the set of i 's neighbors with frame θ .

If an agent chooses to model a proper subset of its neighbors $\hat{N}_\theta(i) \subset N_\theta(i)$, it can still estimate for each action the $p_{a, N_\theta(i)}$ values that parameterize the multinomial distribution $P(C_\theta|s^t, M^t)$. Let $\hat{n}_{\pi(s^t)=a, \hat{N}_\theta(i)}$ be the number of agents in $\hat{N}_\theta(i)$ predicted to perform action a at current state s^t based on the policy obtained by solving their Nested-MDP model and their presence variable $P_j \in m_{j, l-1}$. Then

$$\hat{p}_{a, \hat{N}_\theta(i)} = \frac{\hat{n}_{\pi(s^t)=a, \hat{N}_\theta(i)}}{|\hat{N}_\theta(i)|} \quad (5)$$

is how likely an arbitrarily modeled neighbor of frame θ will perform action a .

Proportions $\hat{p}_{a, \hat{N}_\theta(i)}$ estimated using Eq. 5 allow us to estimate the underlying multinomial distribution over configuration likelihoods (Eq. 4). We can then estimate the probability of a given configuration C using the probability mass function of the modeled multinomial distribution.

How many neighbors to model? Equations 4 and 5 are crucial in that they determine how we extrapolate the behaviors of a few modeled agents to the entire neighborhood *without requiring explicit mental models and policies for every neighbor*. At the same time, this approach approximates the true distribution of configurations, which naturally raises questions such as how many neighbors should a subject agent model and how does the number of neighbors impact the error in the estimated configuration probabilities based on extrapolation. We relate these two questions in the following theorem and corollary. As Hoang and Low (2013) previously established bounds on the error introduced by assuming that neighbors reason about others using a Nested-MDP, we focus our attention on errors introduced by modeling a subset of the neighbors.

Let the acceptable worst-case error in each of the $\hat{p}_{a, \hat{N}_\theta(i)}$ proportions in Eqs. 4 and 5 be $e_{\hat{p}}$. We refer to this henceforth as the extrapolation error. From statistical analysis (Lohr 2010), we establish the minimum number of neighbors which an agent needs to explicitly model so that the extrapolation error does not exceed the given bound:

Theorem 1 (Number of modeled neighbors). *Let $N_\theta(i)$ be a neighborhood of agents with frame θ and whose size is N ,*

$e_{\hat{p}}$ be a desired bound on extrapolation error, $(1 - \alpha)$ be a statistical confidence level, and $t_{n-1, \frac{\alpha}{2}}$ come from the Student's t -distribution with $n - 1$ d.o.f. Then if agent i models

$$n_\theta = |\hat{N}_\theta(i)| \geq \frac{N \left(\frac{t_{n-1, \frac{\alpha}{2}}}{2e_{\hat{p}}} \right)^2}{N - 1 + \left(\frac{t_{n-1, \frac{\alpha}{2}}}{2e_{\hat{p}}} \right)^2} \quad (6)$$

neighbors, then it will be confident at the $(1 - \alpha)$ level that $\hat{p}_{a, \hat{N}_\theta(i)}$ for each action a will be within $e_{\hat{p}}$ of the true proportions of all agents choosing action a .

The proof of this theorem is provided in Appendix A of the supplementary material available on arXiv. To illustrate, let $N = 50$. To ensure that the agent's estimated $\hat{p}_{a, \hat{N}_\theta(i)}$ extrapolates to within $e_{\hat{p}} = 0.1$ of the true proportion for the entire neighborhood with frame θ with 95% confidence, we need model at least 34 neighbors. This drops to 18 if instead $e_{\hat{p}} = 0.2$.

If the agent indeed models at least the prescribed number of other agents from Theorem 1, the error in each configuration's probability estimate is bound by the value $\epsilon_{P(C)}$ given below (proof in Appendix B of the supplementary material).

Corollary 1 (Error bound on configuration probability). *Let n_θ be the number of neighbors given by Theorem 1 (for a given confidence level $1 - \alpha$) that subject agent i chooses to model from its neighborhood $N_\theta(i)$ for each $\theta \in \Theta$, and let $\hat{p}_{a, \theta}$ be the resulting estimated proportions of agents within those neighborhoods that will choose action a , given state s and mental models M . Then the estimated probability $P(C|s^t, M^t)$ that the entire neighborhood will exhibit the configuration C has error $\epsilon_{P(C)}$ due to modeling n_θ agents only, which is less than:*

$$\begin{aligned} & |P^*(C|s^t, M^t) - P(C|s^t, M^t)| = \epsilon_{P(C)} \\ & < \frac{\prod_\theta |N_\theta(i)|!}{\prod_{a, \theta} C(a, \theta)!} \left[\prod_{a, \theta} (\hat{p}_{a, \theta} + e_{\hat{p}})^{C(a, \theta)} - \prod_{a, \theta} \hat{p}_{a, \theta}^{C(a, \theta)} \right] \end{aligned} \quad (7)$$

where $P^*(C|s^t, M^t)$ denotes the true likelihood that configuration C will result from state s^t and mental model M^t .

Regret Bounds As modeling a subset of neighbors produces a bounded approximation of the true configuration probabilities considered during planning, the resulting value function V will also be an approximation. Next, we establish that any error in the approximate value function (and hence the discounted cumulative rewards actually earned by the agent following the corresponding policy π_i) is also bounded. Here, $J_{i, k}$ represents the actual discounted cumulative rewards the agent would earn following π_i for k time steps, whereas $V_{i, k}^*$ is the optimal value function.

Theorem 2. (Regret bound). *Maximum regret that agent i incurs $\left\| V_{i, k}^* - J_{i, k} \right\|_\infty$ from following a k -horizon optimal policy π_i (obtained by solving the many-agent I-POMDP-Lite) due to the approximate likelihoods of other agents' configurations $P(C|s^t, M^t)$ is bounded from above:*

$$\|V_{i,k}^* - J_{i,k}\|_\infty \leq 2\epsilon_{P(C)} \cdot |\mathcal{C}| \cdot R_{max} \times \left[\gamma^{k-1} + \frac{1}{1-\gamma} \left(1 + 3\gamma \frac{|\Omega_i|}{1-\gamma} \right) \right] \quad (8)$$

Notice that the bound is linear in the error $\epsilon_{P(C)}$ in the agent’s estimation of configuration likelihoods caused by modeling some neighbors only. The proof is provided in Appendix C of the supplementary material. Though the resulting value function from planning might not be exact, the approximation error is upper bounded and is proportional to only $\sqrt{n_\theta}$ (in the worst case) due to the fact that $\epsilon_{P(C)}$ is at worst linear in $e_{\hat{p}}$ (c.f., Eq. 7 since $e_{\hat{p}} < 1$), and $e_{\hat{p}}$ is proportional to $\sqrt{n_\theta}$ (c.f., Eq. 6).

Which neighbors to explicitly model? After determining the number of agents to model for each frame-action pair, we must select which neighbors will be explicitly modeled, $\hat{N}_\theta(i)$. This selection is performed only when the agent joins the environment. Thus, it does not add to the computational complexity of planning.

To decide which neighbors to model, we first note that we can decompose the agent’s neighborhood $N_\theta(i)$ into (potentially overlapping) sets based on the actions that the neighbors *can* perform: $N_{\theta,a}(i) = \{j \in N_\theta(i) | j \text{ can perform action } a\}$. Each $N_{\theta,a}(i)$ provides a sample set from which we select neighbors in order to estimate $\hat{p}_{a,\hat{N}_\theta(i)}$. To construct $\hat{N}_\theta(i)$, we randomly sample n agents (Eq. 6) for each (θ, a) pair.

Second, many agents may appear in more than one such sample set because agents can perform more than one action. Thus, once a neighbor j has been added to $\hat{N}_\theta(i)$, it contributes to the n needed for each (θ, a) pair corresponding to the actions that agent j can perform. If sampling neighbors starts with the largest $N_{\theta,a}$ and proceeds to the smallest, then agents that have the greater number of possible actions (e.g., can fight the most fires) will have the largest total chance of being sampled for modeling since they appear in most subsets $N_{\theta,a}(i)$. This property is beneficial as such agents are expected to be the *most influential agents* in the system, as well as the *most necessary to explicitly model* because their reasoning will be more complex than other agents, and thus their behaviors are more difficult to predict.

Many-Agent MCTS for Open Systems

Based on our approach for modeling a subset of an agent’s neighbors, we next extend the popular POMCP algorithm (Silver and Veness 2010) to create a MCTS algorithm for many-agent open environments.

Single-agent MCTS A popular approach to quickly approximate the value function and policy of a POMDP is Monte Carlo tree search (MCTS) (Ross et al. 2008; Silver and Veness 2010; Somani et al. 2013). In MCTS, the agent iteratively constructs an AND-OR tree representing the possible beliefs that can be reached by following sequences of actions, and obtains the expected discounted utility of performing those action sequences by simulating fu-

ture rewards. MCTS performs online planning for the beliefs that an agent actually encounters, avoiding the need for computationally-expensive offline planning and scales well to large state spaces due to MC sampling of future state-action-reward trajectories. Until the time budget is met, MCTS revises estimated $Q(b, a)$ values along the trajectories and extending the tree at the leaf using rollout.

Many-agent MCTS We generalize MCTS to open, many-agent settings in the context of the I-POMDP-Lite framework (also applicable to I-POMDPs), exploiting both frame-action anonymity and selective neighbor modeling to further enhance scalability. Our algorithm, shown in Algorithm 1, is **I-POMCP**_O. It considers agent openness in (i) **Simulate**, which additionally simulates the agents leaving and reentering (in M), and in (ii) **SampleConfiguration** when obtaining configuration probabilities.

Our I-POMCP_O algorithm differs from the classic single agent POMCP (Silver and Veness 2010) in the following ways. First, the agent’s rewards R_i , state transitions T_i , and observations O_i depend on not only its own actions and the environment state, but also the configuration of actions by the other agents. Thus, before the agent can simulate the environment on line 6 of the **Rollout** and line 11 of the **UpdateTree** procedures, the agent must sample a configuration C^t of its neighbors’ actions. This addition to standard POMCP occurs on lines 4 and 9 of the two procedures.

Second, the process for sampling a configuration exploits the multinomial probability distribution $P(C^t | s^t, M^t)$ determined from the models of a select subset of the agent’s neighbors in the **SampleConfiguration** procedure, so that the agent need not model every neighbor. Since only some mental models are consulted, there is opportunity to save computation time in this procedure, as opposed to modeling every neighbor². Moreover, since only some mental models are needed by this procedure, the **Simulate** procedure modeling the environment (called on lines 6 and 11 of **Rollout** and **UpdateTree**, respectively) needs to process fewer mental model transitions, saving additional computational time. Overall, this should require less time per trajectory through the AND-OR tree in each call of **UpdateTree** from the root of the tree, and thus more total trajectories sampled and closer approximations of the true value function.

This algorithm significantly differs from previous MCTS algorithms for multiagent POMDPs. Hula et al. (2015) first adapted POMCP to an I-POMDP setting. However, they made several assumptions in their adaptation to their particular problem: the presence of only two agents, agents’ actions occur sequentially rather than simultaneously, and agents directly observe each other actions. Their approach would not scale to many-agent environments as it would have computational complexity that is exponential in the number of agents. Our approach applies to more general settings, assuming only frame-action anonymity, and exhibits complexity that is *linear* in the number of agents: $O(IHN)$ where I is the number of sampled MC trajectories, H is the horizon, and N is the total number of modeled agents. Amato

²Provided that sampling an action for each neighbor is less computationally expensive than interacting with an actual policy.

Algorithm 1 I-POMCP_O: Open Many-Agent MCTS

Note: T is the tree (initially empty), p is a path from the root of the tree (with $p = \emptyset$ signifying the root), B_p is the particle filter signifying the set of state-model pairs encountered at the node at p in the tree, PF is the root particle filter, N is count of the number of visits to each node in the tree initialized to some constant $\nu \geq 0$, Q is the Q function initialized to 0, c a constant from UCB-1.

```
1: procedure I-POMDP-MCTS( $PF, \tau$ )
2:    $time \leftarrow 0$ 
3:   while  $time < \tau$  do
4:      $s^0, M^0 \leftarrow SampleParticle(PF)$ 
5:     UpdateTree( $s^0, M^0, 0, \emptyset$ )
6:     Increment  $time$ 
7:     return  $\operatorname{argmax}_{a \in A_i} Q(\emptyset, a)$ 

1: procedure UPDATETREE( $s^t, M^t, t, p$ )
2:   if  $t \geq H$  then
3:     return 0
4:    $B_p \leftarrow B_p \cup \{(s^t, M^t)\}$ 
5:   if  $p \notin T$  then
6:      $T \leftarrow T + \text{leafnode}(p)$ 
7:     return Rollout( $s^t, M^t, t$ )
8:   else
9:      $C^t \leftarrow SampleConfiguration(s^t, M^t)$ 
10:     $a_i^t \leftarrow \operatorname{argmax}_{a \in A_i} Q(p, a) + c\sqrt{(\log N_p) / N_{p \rightarrow a}}$ 
11:     $s^{t+1}, M^{t+1}, o_i^t, r_i^t \leftarrow Simulate(s^t, M^t, a_i^t, C^t)$ 
12:     $N_p \leftarrow 1 + N_p$ 
13:     $N_{p \rightarrow a_i^t} \leftarrow 1 + N_{p \rightarrow a_i^t}$ 
14:     $p' \leftarrow p + (a_i^t, o_i^t)$ 
15:     $R \leftarrow r_i^t + \gamma \cdot UpdateTree(s^{t+1}, M^{t+1}, t + 1, p')$ 
16:     $Q(p, a_i^t) \leftarrow Q(p, a_i^t) + R - Q(p, a_i^t) / N_{p \rightarrow a_i^t}$ 
17:    return  $R$ 

1: procedure ROLLOUT( $s^t, M^t, t$ )
2:    $R \leftarrow 0, t' \leftarrow t$ 
3:   while  $t < H$  do
4:      $C^t \leftarrow SampleConfiguration(s^t, M^t)$ 
5:      $a_i^t \leftarrow SampleAction(A_i)$ 
6:      $s^{t+1}, M^{t+1}, o_i^t, r_i^t \leftarrow Simulate(s^t, M^t, a_i^t, C^t)$ 
7:      $R \leftarrow R + \gamma^{t-t'} \cdot r_i^t, t \leftarrow t + 1$ 
8:   return  $R$ 

1: procedure SAMPLECONFIGURATION( $s^t, M^t$ )
2:    $C(a, \theta) \leftarrow 0, \hat{n}_{\pi(s^t)=a, \hat{N}_{\theta}(i)} \leftarrow 0 \quad \forall a, \theta$ 
3:   for  $\mathcal{M}_{j, l-1} \in M^t$  do
4:      $a \sim \pi_{j, l-1}(s^t)$ 
5:      $\hat{n}_{\pi(s^t)=a, \hat{N}_{\theta_j}(i)} \leftarrow \hat{n}_{\pi(s^t)=a, \hat{N}_{\theta_j}(i)} + 1$ 
6:   for  $\theta \in \Theta$  do
7:     for  $a \in A$  do
8:        $\hat{p}_{a, \hat{N}(i)} \leftarrow \hat{n}_{\pi(s^t)=a, \hat{N}_{\theta}(i)} / |\hat{N}_{\theta}(i)|$ 
9:     for  $j \in \hat{N}_{\theta}(i)$  do
10:       $a \sim Cat(\hat{p}_{a_1, \hat{N}_{\theta}}, \hat{p}_{a_2, \hat{N}_{\theta}}, \dots, \hat{p}_{a_{|\hat{A}|}, \hat{N}_{\theta}})$ 
11:       $C(a, \theta) \leftarrow C(a, \theta) + 1$ 
12:   return  $C$ 
```

and Oliehoek (2015) and Best et al. (2019) proposed similar algorithms for Dec-POMDPs that scale to more than two agents. However, our algorithm enables planning in settings when the joint actions of other agents are unknown and must

be estimated through individual models, as well as incorporates frame-action anonymity and selective neighbor modeling for improved scalability. Claes et al. (2017) proposed an algorithm that simplified the models of neighbor behavior using domain heuristics to enhance scalability, as well as social laws to help promote cooperation. Here, however, we consider more general models of neighbor behavior that will work across domains without specific heuristics.

Experiments

We evaluate implementations³ of I-POMCP_O on the wild-fire suppression problem. Chandrasekaran et al. (2016) used this domain to study planning in *small* open agent environments; we extend it to include more fires and significantly more firefighters and corresponding joint actions.

Setup Agents are tasked with putting out fires of different sizes in the absence of inter-agent communication and prior coordination. Small, large, and huge fires require at least 10, 20, and 30 agents to act together to reduce their intensity, respectively. The spread of fires is modeled on the dynamics of real wildfires (Boychuk et al. 2009; Ure et al. 2015). Agents have limited amounts of suppressants that stochastically transition from full to half-full to empty, and back to full on recharging during which they temporarily leave the environment for an average of two steps, then rejoin when full. Each agent can use its suppressant on a fire in an adjacent location or take a NOOP action; the latter also represents actions to take while recharging suppressant. Agents earn shared rewards of 20, 40, and 60 for putting out small, large, and huge fires. They receive shared penalties of 1 when a location burns out and individual penalties of 100 for either fighting a non-existent fire or choosing a non-NOOP action without suppressant. Figure 1 illustrates the environments used in our experiments. Notice that our agents are ground firefighters and helicopters, whose frames differ: helicopters are twice as effective in their firefighting ability. Though several agents share the same location, they start with different suppressant levels. As such, they plan and behave differently rather than as one collective agent.

Though in seemingly small grids, these decision problems are non-trivial and substantially complex, especially compared to the prior literature – with 40-50 agents, the number of joint actions ranges from 2.95×10^{21} to 1.27×10^{30} . Even with frame-action anonymity, there are 15,256 to 9,662,576 configurations; thus, these setups are intractable for prior planning algorithms. We intentionally do not consider smaller domains for which existing algorithms are tractable (e.g., with only 5 agents (Chandrasekaran et al. 2016)), as there would be no need for modeling only a few neighbors (Thm. 1 would model every neighbor $\forall e_{\hat{p}} < 1$).

Metrics and baselines Agent performance is measured by: (i) the average rewards earned by agents, (ii) the average number of fire locations put out, and (iii) the average amount of suppressant used. These measures evaluate how well agents *optimize* their planning problems, and how *effectively* and *efficiently* they put out fires, respectively.

³Available at <https://github.com/OberlinAI/ScalableOASYS>

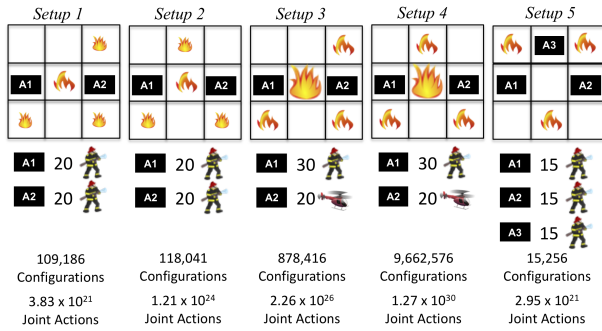


Figure 1: Our environment setups involve a varying number of fires, fire intensities and positions, agents and their types.

To demonstrate both the efficacy and impact of our strategy of intelligently modeling only a subset of neighbors, we consider variants of I-POMCP_O , each planning at level $l = 2$, that use different $e_{\hat{p}} \in [0, 0.3]$ with Theorem 1 providing the numbers of neighbors to model (with $\alpha = 0.05$). Here, $e_{\hat{p}} = 0$ represents I-POMCP_O modeling every neighbor and also provides an upper bound on how the algorithm of Hula et. al (2015) would perform if generalized to many-agent environments and Amato and Oliehoek (2015) if extended to the I-POMDP-Lite with frame-action anonymity.

We compare the performance of our approach with two baselines (using Kruskal-Wallis followed by post-hoc pairwise Mann-Whitney for significance testing). First, **NestedMDP** represents an agent using an adaptation of VI to perform ϵ -bounded planning over configurations at level $l = 1$. This is a strong baseline, which serves as both a scalable, fully observable approximation that can tractably solve our decision problems, as well as a comparison to the models that agent i assumes its neighbors follow. Second, **Heuristic** is an agent that randomly chooses an adjacent fire to fight if one exists and the agent has suppressant, else it takes a NOOP action. It represents semi-intelligent behavior in complex environments (similar to domain heuristics in multi-robot dynamic task allocation, e.g., (Lerman et al. 2006)). Of note, traditional I-POMDP solvers do not scale to many-agent environments. Indeed, we ran the previous method, interactive point-based value iteration (Chandrasekaran et al. 2016), on Setup 1 with only 10 total agents; after almost a week of planning, it did not produce a usable policy.

Results We observe from Fig. 2 that I-POMCP_O improved on NestedMDP and Heuristic by achieving significantly greater rewards in Setups 1-3 and 5, and statistically equivalent reward to NestedMDP in Setup 4. Further investigations (c.f., Figs. 4 and 5 in Appendix D of the supplementary material) reveal that I-POMCP_O agents used slightly more suppressant to put out fires at significantly more locations than NestedMDP (true in all setups). NestedMDP agents earned comparable rewards in Setup 4 by concentrating all of their efforts on a single shared fire, whereas I-POMCP_O agents spread their resources across the most complex environment. Interestingly, in Setup 5, which had the most shared fires, NestedMDP never put out a fire. This was due to the symme-

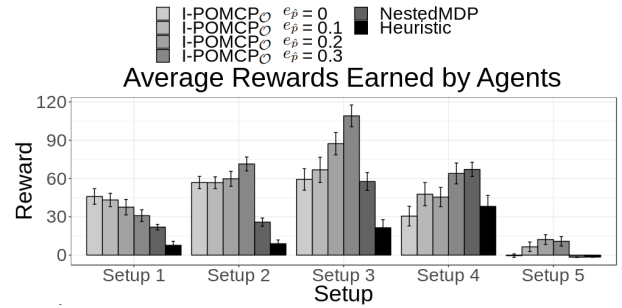


Figure 2: Total reward earned per agent averaged across 100 runs. Error bars represent 95% CIs. I-POMCP_O planned for 5 sec. in Setups 1-3 and 5, and for 20 sec. in Setup 4.

try in the problem causing NestedMDP to behave randomly, whereas I-POMCP_O was more successful.

Comparing the different settings of I-POMCP_O , increasing $e_{\hat{p}}$ so that fewer neighbors were modeled led to *increased performance* in the more complicated Setups 2-5. Moreover, agents were also *more efficient* with their suppressants, using less resources to put out more fires as they modeled fewer neighbors. These results were obtained because MCTS could then run a greater number of trajectories in the fixed planning time τ , in essence trading off modeling accuracy for better value estimates. Furthermore, in Setup 5, modeling only some neighbors biased agents to favor one shared fire over another, breaking the complicated symmetry of the problem. On the other hand, modeling fewer neighbors reduced performance in only the simplest Setup 1, which had only one shared fire and one frame of agents. In this setup, even when modeling all agents, enough trajectories were sampled to adequately estimate the value function.

Concluding Remarks

Real-world domains often exhibit agent openness, where agents may leave and then return. Scaling in the number of agents in such domains continues to remain a challenge for planning in realistic multiagent environments. Our method – consisting of an approach to selectively model neighbors that is new to planning and a generalized MCTS algorithm for many-agent settings – models the presence or absence of agents as it predicts their behaviors. Our key insight for scaling is that we may explicitly model just a few agents and extrapolate the model predictions to others with provable error bounds. This improves on baselines as we demonstrated on a spectrum of firefighting scenarios with an order of magnitude more agents than previously considered in open environments. The idea of extrapolating modeling is new and not limited to open agent settings. It could be combined in future work with other approximations that promote scalability.

Acknowledgments

This research was supported in part by NSF grant #1909513 (to AE, PD, and LS). Experiments were conducted on Oberlin College’s SCIURus cluster (NSF Grant #1427949).

References

- [Amato and Oliehoek 2015] Amato, C., and Oliehoek, F. A. 2015. Scalable planning and learning for multiagent POMDPs. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.
- [Bernstein et al. 2002] Bernstein, D. S.; Givan, R.; Immerman, N.; and Zilberstein, S. 2002. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research* 27(4):819–840.
- [Best et al. 2019] Best, G.; Cliff, O. M.; Patten, T.; Mettu, R. R.; and Fitch, R. 2019. Dec-mcts: Decentralized planning for multi-robot active perception. *The International Journal of Robotics Research* 38(2-3):316–337.
- [Boychuk et al. 2009] Boychuk, D.; Braun, W.; Kulperger, R.; Krougly, Z.; and Stanford, D. 2009. A stochastic forest fire growth model. *Environmental and Ecological Statistics* 16(2):133–151.
- [Chandrasekaran et al. 2016] Chandrasekaran, M.; Eck, A.; Doshi, P.; and Soh, L. 2016. Individual planning in open and typed agent systems. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'16, 82–91. Arlington, Virginia, United States: AUAI Press.
- [Claes et al. 2017] Claes, D.; Oliehoek, F.; Baier, H.; and Tuyls, K. 2017. Decentralised online planning for multi-robot warehouse commissioning. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17, 492–500.
- [Cohen, Dibangoye, and Mouaddib 2017] Cohen, J.; Dibangoye, J.-S.; and Mouaddib, A.-I. 2017. Open decentralized pomdps. In *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, 977–984.
- [Frankel and Frankel 1987] Frankel, M. R., and Frankel, L. R. 1987. Fifty years of survey sampling in the united states. *Public Opinion Quarterly* 51(4):S127–S138.
- [Hoang and Low 2013] Hoang, T. N., and Low, K. H. 2013. Interactive POMDP lite: Towards practical planning to predict and exploit intentions for interacting with self-interested agents. In *23rd International Joint Conference on AI (IJCAI)*.
- [Hula, Montague, and Dayan 2015] Hula, A.; Montague, P.; and Dayan, P. 2015. Monte Carlo planning method estimates planning horizons during interactive social exchange. *PLOS Computational Biology* 11(6).
- [Lerman et al. 2006] Lerman, K.; Jones, C.; Galstyan, A.; and Mataric, M. J. 2006. Analysis of dynamic task allocation in multi-robot systems. *Int. J. Rob. Res.* 25(3):225–241.
- [Lohr 2010] Lohr, S. L. 2010. *Sampling: Design and Analysis*. Boston, MA: Brooks/Cole, 2nd edition.
- [Nair et al. 2005] Nair, R.; Varakantham, P.; Tambe, M.; and Yokoo, M. 2005. Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs. In *Twentieth AAAI Conference on Artificial Intelligence*, 133–139.
- [Neyman 1934] Neyman, J. 1934. On the two different aspects of the representative method: The method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society* 97(4):558–625.
- [Nguyen, Kumar, and Lau 2017] Nguyen, D. T.; Kumar, A.; and Lau, H. C. 2017. Collective multiagent sequential decision making under uncertainty. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, 3036–3043.
- [Oliehoek et al. 2008] Oliehoek, F. A.; Spaan, M. T.; Whiteson, S.; and Vlassis, N. 2008. Exploiting locality of interaction in factored Dec-POMDPs. In *7th International Joint conference on Autonomous agents and Multiagent Systems (AAMAS)*, 517–524.
- [Oliehoek, Whiteson, and Spaan 2013] Oliehoek, F. A.; Whiteson, S.; and Spaan, M. T. 2013. Approximate solutions for factored Dec-POMDPs with many agents. In *International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 563–570.
- [Qu and Doshi 2015] Qu, X., and Doshi, P. 2015. Individual planning in infinite-horizon multiagent settings: Inference, structure and scalability. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems* 28. Curran Associates, Inc. 478–486.
- [Ross et al. 2008] Ross, S.; Pineau, J.; Paquet, S.; and Chaib-draa, B. 2008. Online planning algorithms for POMDPs. *J. Artif. Int. Res.* 32(1):663–704.
- [Silver and Veness 2010] Silver, D., and Veness, J. 2010. Monte-carlo planning in large pomdps. In *23rd International Conference on Neural Information Processing Systems (NIPS)*, NIPS'10, 2164–2172.
- [Somani et al. 2013] Somani, A.; Ye, N.; Hsu, D.; and Lee, W. S. 2013. Despot: Online pomdp planning with regularization. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, 1772–1780. USA: Curran Associates Inc.
- [Sonu, Chen, and Doshi 2017] Sonu, E.; Chen, Y.; and Doshi, P. 2017. Decision-theoretic planning under anonymity in agent populations. *J. Artif. Int. Res.* 59(1):725–770.
- [Ure et al. 2015] Ure, N.; Omidshafiei, S.; Lopez, B.; Agha-Mohammadi, A.-A.; How, J.; and Vian, J. 2015. Online heterogeneous multiagent learning under limited communication with applications to forest fire management. In *Intelligent Robotics and Systems (IROS)*, 5181–5188.
- [Velagapudi et al. 2011] Velagapudi, P.; Varakantham, P.; Sycara, K.; and Scerri, P. 2011. Distributed model shaping for scaling to decentralized POMDPs with hundreds of agents. In *10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, 955–962.