Constrained Control of Large Graph-Based MDPs Under Measurement Uncertainty

Ravi N. Haksar, *Student Member, IEEE*, and Mac Schwager, *Member, IEEE*

***Abstract*—We consider controlling a graph-based Markov decision process (GMDP) with a control capacity constraint given only uncertain measurements of the underlying state. We also consider two special structural properties of GMDPs, called anonymous influence and symmetry. Large-scale spatial processes such as forest wildfires, disease epidemics, opinion dynamics, and robot swarms are well-modeled by GMDPs with these properties. We adopt a certainty-equivalence approach and derive efficient and scalable algorithms for estimating the GMDP state given uncertain measurements, and for computing approximately optimal control policies given a maximum-likelihood state estimate. We also derive suboptimality bounds for our estimation and control algorithms. Unlike prior work, our methods scale to GMDPs with large state-spaces and explicitly enforce a control constraint. We demonstrate the effectiveness of our estimation and control approach in simulations of controlling a forest wildfire using a model with 10^{1192} total states.**

***Index Terms*—Filtering, Markov processes, network analysis and control, stochastic optimal control, variational methods.**

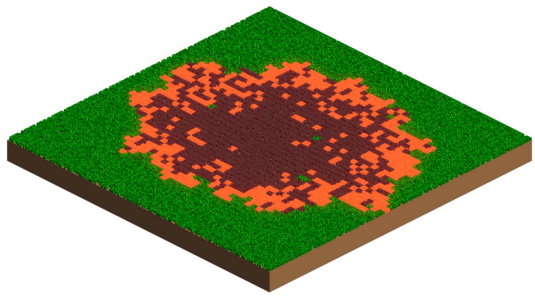

Fig. 1. Forest wildfire modeled by a graph-based Markov decision process (GMDP), where green represents healthy trees, red represents trees on fire, and black represents burnt trees. We propose a scalable framework to produce constrained control actions given noisy measurements to address large-scale phenomena.

I. Introduction

In THIS work, we consider the problem of producing a control action, subject to a capacity constraint, given noisy measurements for a class of discrete space and discrete time graph-based Markov decision processes (GMDPs). Many large-scale, dynamic spatial processes of recent interest are described by these models, such as user interactions in a social network [1], the spread of a contagion in a population [2], and the spread of wildfire in a forest [3]. These applications require a model capable of representing complex time-varying and spatially-varying relationships between a significant number of individual components, which motivates our study of GMDPs. We consider controlling such models by applying localized control effort, for example, supplying medical treatment to a subset of communities in a disease epidemic or supplying fire retardant to a subset of trees to control a forest wildfire. Furthermore, controlling such processes is only meaningful if the total control effort applied at each time step is limited, which we call a control capacity constraint. Otherwise, the optimal unconstrained policy is straightforward, such as applying fire retardant to every tree to extinguish a forest wildfire.

In this work, we develop a certainty-equivalence approach to provide a single framework capable of addressing realistic processes that naturally contain state uncertainty. To the best of our knowledge, our approach is the first framework to consider the control of GMDPs with a control constraint and measurement uncertainty.

For the graph-based models in this work, each vertex in the graph corresponds to an MDP and edges describe the coupling interactions between MDPs. A measurement model is associated with each MDP and describes the likelihood of observing different states of the MDP. While the partially observable MDPs (POMDPs) framework is appropriate for this type of model, it is difficult to develop approximately optimal methods that are suitable for the model sizes we consider using existing tools. In addition, any candidate method must also run in (near) real time to be useful in real-world applications. Therefore, we adopt a certainty-equivalence approach and separate the problem into creating a filter to produce accurate state estimates, and a controller to produce effective constrained control actions given a state estimate.

We develop a fast online filtering method that is tractable for large GMDPs by leveraging variational inference (VI) to derive a message-passing algorithm, which is similar in spirit to belief propagation (BP). Prior work has proposed variations

Manuscript received 30 June 2021; revised 1 July 2021, 19 April 2022, and 23 August 2022; accepted 28 January 2023. Date of publication 6 February 2023; date of current version 26 October 2023. This work was supported by the National Science Foundation under Grant IIS-1646921, in part by DARPA under Grant YFA D18AP00064, and in part by ONR under Grant N00014-18-1-2830. Recommended by Associate Editor S.-S. Jia. *(Corresponding author: Ravi N. Haksar.)*

Ravi N. Haksar is with the Department of Mechanical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: rhaksar@alumni.stanford.edu).

Mac Schwager is with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305 USA (e-mail: schwager@stanford.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAC.2023.3242344.

Digital Object Identifier 10.1109/TAC.2023.3242344

of VI and BP methods, but these methods either do not scale to the model sizes we consider or they perform significantly worse than our approach in terms of accuracy. We prove that our filter approximately optimizes the evidence-based lower bound, and show that it achieves 5%–10% better accuracy in significantly less time than comparable methods.

Our control approach, which produces actions that satisfy a global capacity constraint, is based on approximate linear programming (ALP) for MDPs. ALP methods can circumvent the explicit enumeration of the state space, which is required by standard value and policy iteration methods, by using a basis function approximation of the optimal value or state–action function. We first derive offline approximate dynamic programming methods and prove that our approach produces an approximation with minimum deviation from the optimal solution. We then consider a class of capacity-constrained linear programs that has an efficient solution, so that constrained actions can be produced online quickly.

Our framework is most appropriate for GMDPs with two properties common in large-scale spatial processes, called "anonymous influence" and "symmetry." A GMDP has anonymous influence if the dynamics of a given MDP relies on the number of influencing MDPs in particular states, and not the identity of these influencing MDPs. Symmetry refers to the insight that value approximations for a given MDP can frequently be reused for other MDPs in the model, which greatly reduces the complexity of our methods.

Earlier versions of some of the material in this work appear previously in [3] and [4]. In [3], we considered control of GMDPs with perfect state information and in [4], we considered state estimation of GMDPs without control. The current work brings these two methods together to form a closed-loop estimation and control framework for large-scale GMDPs. In addition, we developed a model fitting approach to learn meaningful GMDPs from real-world data and in this work, we assume a learned model is provided as an input to our framework [5].

The main contributions of this article are as follows.

1) We propose a unified framework to address the constrained control of large GMDP models given uncertain measurements.
2) We derive an approximately optimal filtering method to produce accurate state estimates online.
3) We derive approximate dynamic programming approaches to produce a value function or a state–action function with suboptimality bounds.
4) We propose a class of constrained programs with a closed-form solution for structured value function approximations.
5) We show our approach is suitable for real-time online use, and that other methods are not scalable or effective, on simulations of a forest wildfire with 10^{1192} total states.

The rest of this article is organized as follows. Section II reviews prior work. Section III describes the GMDP framework with uncertain measurements, the anonymous influence and symmetry properties, and a forest wildfire model. Sections IV and V present our certainty-equivalence framework and the derivations of our filtering and control methods. We present numerical results validating our approach in Section VI. Finally, Section VII concludes this article.

II. Prior Work

A. POMDPs

POMDPs are the most appropriate framework for our problem formulation and methods have been proposed for factored models [6], [7], [8]. Notably, the authors of [9] were able to solve models with approximately 10^6 states, but this is still much smaller than the models we wish to address. While prior work suggests some appealing approximation methods, it is not clear how to adopt them for online use. Contrary to POMDP methods, which are dominated by the interleaving of filtering and control, we intentionally separate the two. This leads to our methods being able to scale up to problems with 10^{1192} possible discrete states, well beyond existing POMDP methods. We review individual control and filter methods next.

B. Control Methods

Traditional MDP methods are infeasible for GMDPs as the state and action spaces of the aggregate MDP typically grow exponentially in the number of constituent MDPs. The factored MDP [10] as well as GMDP [11] frameworks have been formulated to compactly represent structured MDPs. Nevertheless, the compact representation does not translate to tractable exact solution methods [12]. Other work [13], [14] has proposed structured policy iteration methods, but do not consider control constraints that are integral to our problem formulation.

ALP circumvents the explicit enumeration of the value function over the state space by using a basis function approximation. Efficient variable elimination methods have been proposed [15], [16], [17]; however, these approaches are applied to model sizes that are several orders of magnitude smaller than the ones we consider in this work. Our control approach can be seen as a generalization of prior work that derives local policies for each MDP based on a specific basis approximation, without considering control constraints or state–action functions [11]. The proposed approximation is not suitable for our problem formulation, as we show in our simulation experiments, and therefore, our framework is necessary to consider a broader class of graph-based models.

The assignment of limited control effort to a set of MDPs has also been considered in literature. Constrained MDP formulations [18] allow explicit control constraints but require traditional MDP descriptions. Several types of approximate methods have been proposed, but are intractable for the high-dimensional state and action spaces of GMDPs [19], [20], [21], [22], [23], [24]. We develop an approximate method for satisfying a strict global capacity constraint that is tractable for large GMDPs, which is similar in spirit to [19].

C. Filter Methods

We consider graph-based models with state uncertainty for which the equivalent graphical model representation contains many cycles. Therefore, methods that rely on a tree structure (e.g., BP) or few cycles cannot be directly applied. Furthermore, we are interested in producing a full posterior distribution over states, in contrast to a maximum-likelihood estimate (e.g., Viterbi algorithm).

Sampling-based methods can address some issues of online inference for GMDPs [2], [25], but are applied to smaller models or require specific dynamics and measurement model forms.

TABLE I
NOMENCLATURE FOR THE GMDP MODELING FRAMEWORK

Symbol	Definition
$a_i^t \in \mathcal{A}_i$	MDP i action and state space
t	Time index
$x_i^t \in \mathcal{X}_i$	MDP i state and state space
$y_i^t \in \mathcal{Y}_i$	MDP i measurement and measurement space
z_i^t	Count aggregator
G	GMDP graph structure
J	GPOMDP objective function
$\mathcal{E}, \mathcal{V}$	GMDP graph edge and vertex sets
$\mathcal{M}(i)$	Influence of other MDPs in measurement model
$\mathcal{N}(i)$	Influence of other MDPs in dynamics model
$\mathbb{E}$	Expectation operator

Proposed approaches that have addressed this issue [26] are appropriate for continuous dynamical system models and not the discrete models we consider. Other methods [27], [28] have been applied to relatively large models but do not scale to the model sizes we consider.

VI methods have been applied to relatively large discrete models for inference [29], [30], [31]. Notably, semi-implicit VI [31] optimizes bounds of the evidence lower bound (ELBO), but these bounds are not suitable for our approach, and thus, we develop our own approximation. While some methods [32] perform approximate inference for large datasets, the adaptation to online use is unclear as applications have been limited to relatively small models [33]. Stochastic gradient methods typically require a differentiable distribution whereas we estimate arbitrary discrete distributions. Other methods [34] are based on exploiting distribution structure that we do not require.

BP methods can be derived using VI with energy approximations (e.g., Bethe or Kikuchi). Loopy belief propagation (LBP) has been shown to be effective in some discrete loopy graphical models [35] and we use LBP as a benchmark method. Generalized belief propagation improves upon LBP [36] but incurs additional (worst-case exponential) complexity and is nontrivial to apply generally. We emphasize that our approach does not use these energy approximations. In contrast, our filtering approach is based on a message-passing scheme that approximates the Kullback–Leibler (KL) divergence typically used in VI methods. The result of this approach is a combination of the computational efficiency of message-passing methods with the theoretical insight and effectiveness of VI approaches.

In the next section, we review the GMDP framework, discuss our structural assumptions, and introduce a model to describe a forest wildfire.

III. GRAPH-BASED MARKOV DECISION PROCESSES

We describe the main aspects of the GMDP framework; see Table I for the nomenclature of this section [2]. Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected graph with vertex set $\mathcal{V} = \{1, \ldots, n\}$ containing n vertices and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Each vertex $i \in \mathcal{V}$ corresponds to an MDP with state $x_i^t \in \mathcal{X}_i$, action $a_i^t \in \mathcal{A}_i$, and measurement $y_i^t \in \mathcal{Y}_i$ at time t. See Fig. 2 for a visualization of the GMDP structure. In a GMDP, the dynamics of MDP i are influenced by its neighbors.

Definition 1 (Neighbor Set): The neighbor set $\mathcal{N}(\mathcal{S}) : \mathcal{S} \subseteq \mathcal{V} \to \mathcal{T} \subseteq \mathcal{V}$ is defined as

$$\mathcal{N}(\mathcal{S}) = \bigcup_{i \in \mathcal{S}} \{j \mid (j, i) \in \mathcal{E}\}.$$

In the case that $\mathcal{S} = \{i\}$, this is the typical neighbor set of vertex i. However, our notion of a neighbor set also expresses the notion of neighbors of a set of vertices $\mathcal{S} \subseteq \mathcal{V}$. For the neighbor set of MDP i, we let $\mathcal{N}(\{i\}) = \mathcal{N}(i)$. Subscripts indicate the states or measurements of a subset of MDPs, for example, x_i^t for MDP i and $x_{\mathcal{N}(i)}^t = \{x_j^t \mid j \in \mathcal{N}(i)\}$ for the neighbors of MDP i. We likewise use subscripts for the domain of variables, e.g., $x_{\mathcal{N}(i)}^t \in \mathcal{X}_{\mathcal{N}(i)} = \prod_{j \in \mathcal{N}(i)} \mathcal{X}_j$. We omit the subscript for the combination of all MDP states or measurements, $x^t = \{x_1^t, \ldots, x_n^t\} \in \mathcal{X}$ and $y^t = \{y_1^t, \ldots, y_n^t\} \in \mathcal{Y}$. Summing out (i.e., marginalizing) all MDP latent states from a distribution is denoted by $\sum_{x^t} = \sum_{x_1^t} \cdots \sum_{x_n^t}$. The marginalization of a subset of variables is specified in the summation, e.g., marginalizing out a neighbor set is $\sum_{x_{\mathcal{N}(i)}^t}$.

The probability of transitioning from a state x_i^t to x_i^{t+1} for an MDP in the GMDP model only depends on the current state of the MDP x_i^t, the state of its neighbors x_j^t in the graph, and the MDP action a_i^t. Hence, the dynamics can be written compactly as

$$p_i(x_i^{t+1} \mid x_i^t, x_{\mathcal{N}(i)}^t, a_i^t). \tag{1}$$

The dynamics for the aggregate state x^t describing the combination of all MDP states is then

$$p(x^{t+1} \mid x^t, a^t) \propto \prod_{i=1}^{n} p_i(x_i^{t+1} \mid x_i^t, x_{\mathcal{N}(i)}^t, a_i^t). \tag{2}$$

Measurements for each MDP are conditionally independent given the state of the underlying MDP, $p_i(y_i^t \mid x_i^t)$, and the measurement likelihood for the aggregate state is described by the distribution

$$p(y^t \mid x^t) \propto \prod_{i=1}^{n} p_i(y_i^t \mid x_i^t). \tag{3}$$

Arbitrary measurement models, with $p_i(y_i^t \mid x_i^t, x_{\mathcal{M}(i)}^t)$ and $\mathcal{M}(i) \subseteq \mathcal{V}$, can be used in our framework. However, for clarity of exposition, we consider each MDP measurement conditionally independent given the MDP state (i.e., $\mathcal{M}(i) = \emptyset$), and we plan to present more general cases in future work.

Finally, the reward function for the GMDP is additively composed of individual reward functions r_i, which are associated with each MDP i

$$R(x^t, a^t, x^{t+1}) = \sum_{i=1}^{n} r_i(x_{O(i) \cup \mathcal{N}(O(i))}^t, a_{O(i)}^t, x_{O(i)}^{t+1}). \tag{4}$$

Each MDP reward function is defined over a subset of variables, $O(i) \subseteq \mathcal{V} \, \forall i \in \mathcal{V}$, and typically $|O(i)| \ll |\mathcal{V}|$. Our goal in controlling the GMDP is to find a control policy to maximize the infinite-horizon discounted reward. Formally, we seek a policy $\pi(x^t)$ which maximizes

$$J_\pi = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R(x^t, \pi(x^t), x^{t+1}) \mid x^1, y^{2:t}\right]$$

where the expectation is conditioned on the history of measurements $y^{2:t}$ and the initial state x^1, and is with respect to the dynamics distribution (2) and the measurement distribution (3). A capacity constraint is enforced on the action $a^t = \pi(x^t)$, and γ is the discount factor. We emphasize that maximizing this objective requires solving a high-dimensional POMDP that is intractable with typical methods, as we cannot directly observe the underlying state x^t. Therefore, we separate the problem into two stages. First, we approximate the Bayesian posterior $p(x^t \mid y^{1:t})$ and determine the maximum-likelihood state $\hat{x}^t$ of the posterior. Second, we solve the control problem assuming perfect state knowledge using the estimate $\hat{x}^t$ in place of the unknown state x^t to generate an action $a^t = \pi(\hat{x}^t)$.

A. Anonymous Influence

We consider the case where (1) is based on the number of neighbors in particular states rather than the identity of these neighbors. This property is called "anonymous influence," and we summarize the relevant ideas [1], [17].

We use $\mathbf{1}_j(x_i)$ to represent the indicator function that equals one when $x_i = j$ and zero otherwise. In addition, we use brackets to refer to the elements of a vector, e.g., $[z]_1$ refers to the first element of vector z. For a set of n discrete variables $x_i \in \{0, 1, \ldots, \mathcal{D} \in \mathbb{Z}_{\geq 0}\}$, the count aggregator (CA) is a vector $z \in \mathbb{Z}_{\geq 0}^{\mathcal{D}}$, where each element describes the number of variables taking on a particular value, $[z]_j = \sum_{i=1}^{n} \mathbf{1}_j(x_i)$ and $j \in \{0, 1, \ldots, \mathcal{D}\}$. A mixed-mode function (MMF) uses a CA, as well as other discrete arguments not part of a CA, and maps to the real numbers $\mathbb{R}$.

For a GMDP where all MDPs have the same discrete domain $x_i^t \in \{0, 1, \ldots, \mathcal{D}\}$, the dynamics (1) for each MDP requires specifying (at most) $(\mathcal{D}+1)^{|\mathcal{N}(i)|+2}$ values. If a CA z_i^{t-1} is used to represent the influence of other MDPs, then (1) can be represented by an MMF

$$p_i(x_i^t \mid x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1}, a_i^{t-1}) = p_i(x_i^t \mid x_i^{t-1}, z_i^{t-1}, a_i^{t-1}) \tag{5}$$

which requires specifying $(\mathcal{D}+1)^2 \cdot \begin{pmatrix} |\mathcal{N}(i)| + \mathcal{D} \\ |\mathcal{N}(i)| \end{pmatrix}$ values, where $\begin{pmatrix} n \\ j \end{pmatrix}$ is the binomial coefficient, a potentially significant reduction. We illustrate this property in the discussion of our wildfire model in Section III-C.

B. Symmetry

If a GMDP consists of a comparatively small number of unique "classes" of MDPs, we say it has symmetry. Two MDPs i and j are in the same class k, denoted by $i, j \in \mathcal{C}_k$, if both MDPs have the same reward and dynamics functions. We describe how these quantities define a class in our optimal control approach in Section V.

The n MDPs in a GMDP are partitioned into $s \leq n$ unique classes, $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset \; \forall i \neq j$, and $\sum_{k=1}^{s} |\mathcal{C}_k| = n$. Many domains contain a graph topology with many MDPs but few classes. Methods that require enumerating the state space scale exponentially with the number of MDPs and quickly become intractable. In contrast, our control approach exploits symmetry to instead scale with the number of classes.

Our framework can be applied without the anonymous influence and symmetry properties, and the benefit of these properties is significant computational improvements. Without these properties, our control and filtering methods still provide high-quality solutions, due to our choice of approximations, as we discuss in Sections IV and V. We validate our approach with simulation experiments presented in Section VI.

TABLE II
NOMENCLATURE FOR OUR FOREST WILDFIRE MODEL

Symbol	Definition
e_i^t	Number of healthy neighbor MDPs for MDP i
f_i^t	Number of MDP neighbors on fire for MDP i
B, F, H	Tree states: "burnt", "on fire", and "healthy"
L, W	Forest lattice dimensions
α, β	Tree dynamics parameters

TABLE III
TREE TRANSITION PROBABILITIES FOR WILDFIRE MODEL

		x_i^{t+1}		
		H	F	B
x_i^t	H	$1 - \alpha f_i^t$	αf_i^t	
	F		$\beta - \Delta\beta a_i^t$	$1 - \beta + \Delta\beta a_i^t$
	B			1

Notes: Blank entries are zero.

C. Example GMDP: Forest Wildfires

We now introduce a model that describes the spread of a wildfire in a forest, to illustrate the GMDP modeling framework and our structural assumptions; see Table II for a summary of model quantities. The forest is modeled as a finite 2-D lattice of dimensions $L \times W$ with LW total nodes (see Fig. 1). Each node $i \in \{1, \ldots, LW\}$ on the lattice represents a tree and the tree state x_i^t is one of three values, $\mathcal{X}_i = \{H, F, B\} = \{\text{healthy, on fire, burnt}\}$. An undirected graph is used to represent the trees and influence between trees, with the vertex set $\mathcal{V} = \{1, \ldots, LW\}$. Edges exist between trees if they are neighbors on the lattice. Table III summarizes the dynamics where $f_i^t = \sum_{j \in \mathcal{N}(i)} \mathbf{1}_F(x_j^t)$ is the number of neighboring trees on fire.

A tree that is healthy transitions to on fire only if at least one tree in its neighbor set is on fire where α describes the likelihood of fire spreading from a tree on fire to a healthy tree. A tree on fire will either remain on fire or transition to burnt in a single time step. The parameters β and $\Delta\beta$ describe the average number of time steps a fire will persist and the effectiveness of control actions, respectively. Control actions are binary and reflect the choice of whether or not to apply fire retardant on a tree, $a_i^t = \{0, 1\}$. Finally, a tree that is burnt will remain burnt for all time.

The measurement model for each tree is parameterized by p_c, which is the probability of the ground truth tree state being observed. The other two tree states are observed with probability $\frac{1}{2}(1 - p_c)$. The measurement model is thus

$$p(y_i^t \mid x_i^t) = \begin{cases} p_c & \text{if } y_i^t = x_i^t \\ \frac{1}{2}(1 - p_c) & \text{if } y_i^t \neq x_i^t. \end{cases} \tag{6}$$

In Section VI, we define the reward functions we use to evaluate our framework using simulation experiments.

TABLE IV
NOMENCLATURE FOR ONLINE FILTER APPROACH

Symbol	Definition
d_i^k	HMM i summary function in message-passing scheme
g	Linear approximation to logarithm
m_i^k	k^{th} message from HMM i
u_i, q_i	HMM i prior and posterior distribution
D_{KL}	Kullbach-Liebler (KL) divergence
E_i^k	k^{th} estimate of expectation of joint probability for HMM i
$\mathcal{L}_i$	MDP i objective function for posterior distribution
$\mathcal{Q}$	Space of approximate distributions
ϵ	Joint probability lower bound

Algorithm 1: Certainty-Equivalence Framework.

1: Offline: Solve linear program(s) for basis functions weights of approximate function $V_w(x^t)$ or $Q_w(x^t)$.
2: Online
3: **for** each time step t **do**
4: Get measurement.
5: Update state estimate using filter with measurement.
6: Solve linear program to get constrained control action for the state estimate.
7: Apply control action.

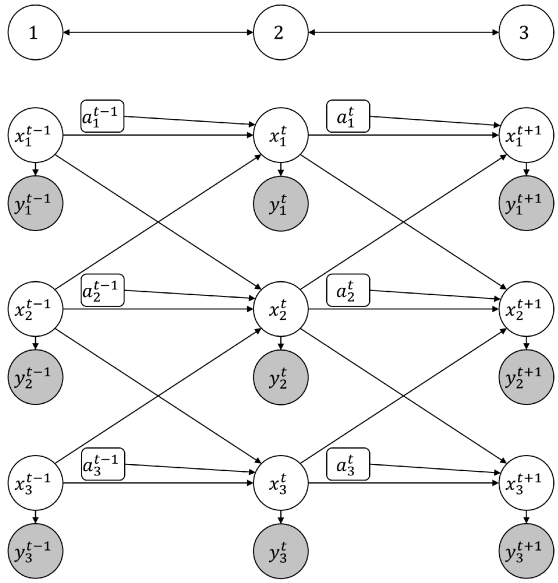

Fig. 2. Top: Example GMDP consisting of three vertices, each of which represents an MDP, where arrows indicate the mutual influence between MDPs. Bottom: Underlying graphical model of the example GMDP, where arrows indicate influence between time steps.

Without MMFs, the MDP dynamics p_i contains $|\mathcal{X}_i||\mathcal{X}_{\mathcal{N}(i)}||\mathcal{A}_i| = 6 \cdot 3^{|\mathcal{N}(i)|}$ elements and is exponential in the number of neighbors $|\mathcal{N}(i)|$. Since the probability for a healthy tree to transition to on fire depends on the number of neighboring trees on fire, and not the identity of these trees, an MMF reduces the representation size. With an MMF, the domain of p_i is reduced to $(|\mathcal{N}(i)|+1)|\mathcal{X}_i||\mathcal{A}_i| = 6(|\mathcal{N}(i)|+1)$ values, which is now linear in $|\mathcal{N}(i)|$.

The aggregate state space of the wildfire model has 3^{LW} state configurations. A 100-tree forest has more states than there are grains of sand on Earth and a 250-tree forest has more states than atoms in the universe [37]. Therefore, using MMFs for the tree dynamics is essential to building a tractable online framework, which we describe next.

IV. EFFICIENT ONLINE FILTER FOR GMDPS

We adopt a certainty-equivalence approach and decompose the problem into two parts: a process filter to produce accurate maximum-likelihood estimates of the underlying state, and a process controller to produce effective constrained actions given a state estimate. Algorithm 1 provides a high-level overview of our framework. First, a value or state–action function approximation is produced offline by solving linear programs to determine the weights of the chosen basis functions. Second, a fast filter produces online estimates of the process state, which is used in a linear program to determine a control action that satisfies a capacity constraint. In the rest of this section, we describe our filtering approach, and provide the nomenclature in Table IV.

Given the previous control action a^{t-1}, the GMDP can be considered a graph-based hidden Markov model (GHMM), where each node in the graph refers to an HMM (previously, an MDP). An HMM can be considered an MDP that has no action input and only noisy state information is available. Therefore, for the following discussion, we use the HMM terminology in our derivation of a scalable approximate filter. The objective of a filter at a single time step is to produce the posterior distribution $p(x^t \mid y^{1:t})$ where $y^{1:t}$ is the history of measurements up to time t, $y^{1:t} = \{y^1, \ldots, y^t\}$. The exact filter is derived via Bayes' rule and is a recursive relationship,

$$p(x^t \mid y^{1:t}) \propto p(y^t \mid x^t) \sum_{x^{t-1}} p(x^t \mid x^{t-1}, a^{t-1}) p(x^{t-1} \mid y^{1:t-1})$$

which is initialized by a prior at the initial time step, $p(x^1)$. Expanding with (2) and (3), the recursive Bayesian filter (RBF) [38] for the models considered in this work is

$$p(x^t \mid y^{1:t}) \propto \left(\prod_{i=1}^{n} p_i(y_i^t \mid x_i^t) \right) \left(\sum_{x^{t-1}} p(x^{t-1} \mid y^{1:t-1}) \prod_{i=1}^{n} p_i(x_i^t \mid x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1}, a_i^{t-1}) \right). \quad (7)$$

The above expression does not simplify to a tractable form as we allow for an arbitrary graph structure. In particular, the graphical model representation of the graph G may contain many cycles (or loops), as is the case for the lattice-based graph in our wildfire model. See Fig. 2 for a graphical model representation of a simple GMDP as well as its equivalent dynamic Bayesian network representation.

VI methods formulate an optimization problem to produce an approximation of an intractable posterior distribution [39]. In the case of an online filter for the models we consider in this work, the distribution representing the belief at each time step is intractable to compute exactly. Therefore, we use VI methods to produce a tractable and accurate approximate distribution at each time step. The RBF (7) requires $(\prod_{i=1}^{n} |\mathcal{X}_i|) - 1$ values to specify $p(x^{t-1} \mid y^{t-1})$ and is intractable to determine exactly for even a single time step despite the graph structure. For example,

our wildfire model with 250 total trees has 10^{119} total states. Therefore, we use VI to introduce a family of distributions $q(x^t) \in \mathcal{Q}$ to best approximate the posterior $p(x^t \mid y^{1:t})$ by minimizing the KL divergence. However, directly minimizing the KL divergence is infeasible due to requiring knowledge of the posterior. Instead, a tractable optimization is maximizing the ELBO that indirectly minimizes the KL divergence. The ELBO is

$$\text{ELBO} = \mathbb{E}_{q(x^t)}\left[\log p(x^t, y^t \mid y^{1:t-1}) - \log q(x^t)\right] \tag{8}$$

where $\mathbb{E}_{q(x^t)}$ indicates the expectation is taken with respect to the distribution $q(x^t)$. Choosing an appropriate form for the approximating distribution $q(x^t)$ results in an optimization problem with a tractable solution, as we explain next.

We leverage the mean-field approximation where the approximating distribution is factored, $q(x^t) = \prod_{i=1}^n q_i(x_i^t)$, and a discrete distribution (or variational factor) is associated with each HMM in the GHMM. This approximation reduces the representation size of the posterior and leads to

$$\begin{aligned}\text{ELBO} = &\sum_{x^t}\left(\prod_{i=1}^n q_i(x_i^t)\right)\log p(x^t, y^t \mid y^{1:t-1}) \\ &- \sum_{i=1}^n \sum_{x_i^t} q_i(x_i^t)\log q_i(x_i^t)\end{aligned}$$

after substitution of $q(x^t)$ and algebraic simplification. A common approach to optimizing the ELBO, known as coordinate ascent VI (CAVI [39, Sec. 2.4]), finds a local optimum by iteratively optimizing each factor $q_i(x_i^t)$ while holding others fixed. For each iteration of CAVI, every factor is updated once using information from the other factors. The method continues to iterate until a termination condition is reached, e.g., the majority of factors do not change between successive iterations. We present a brief derivation of the CAVI method here to illustrate our approach in the next section, and a more detailed treatment can be found in [39, Sec. 2]. The update rule for the factor associated with HMM i is derived by collecting the terms in the ELBO that contains factor $q_i(x_i^t)$

$$\begin{aligned}\text{ELBO} = &\sum_{x_i^t} q_i(x_i^t)\mathbb{E}_{-i}\left[\log p(x^t, y^t \mid y^{1:t-1})\right] \\ &- \sum_{x_i^t} q_i(x_i^t)\log q_i(x_i^t) + \text{other terms}\end{aligned} \tag{9}$$

where $\mathbb{E}_{-i}$ refers to the expectation taken with respect to the distribution $q(x^t)$ excluding factor $q_i(x_i^t)$, i.e., $\prod_{j=1, j\neq i}^n q_j(x_j^t)$. For the optimization of (9) over a single factor, the "other terms" are dropped as they are constant with respect to factor $q_i(x_i^t)$. As a result, the expression in (9) can be rewritten as the following objective function:

$$\mathcal{L}_i = -D_{\text{KL}}(q_i(x_i^t) \,||\, \exp \mathbb{E}_{-i}\left[\log p(x^t, y^t \mid y^{1:t-1})\right]) \tag{10}$$

where D_{KL} is the KL divergence. Since the KL divergence is nonnegative and equals zero when the argument distributions are identical, maximizing $\mathcal{L}_i$ leads to the update expression

$$q_i(x_i^t) \propto \exp \mathbb{E}_{-i}\left[\log p(x^t, y^t \mid y^{1:t-1})\right]. \tag{11}$$

We emphasize that arbitrary graph structure prevents simplification of the previous expression through structure in $\log p(x^t, y^t \mid y^{1:t-1})$. Therefore, the standard CAVI approach cannot directly be applied to the models we consider in this work. Next, we introduce approximations to derive a tractable iterative optimization approach, based on a surrogate objective function that is a lower bound to the ELBO.

A. Approximating the ELBO

For GHMMs with the mean-field assumption, the coordinate ascent update (11) for a single time step requires computing the joint probability

$$\begin{aligned}&p(x^t, y^t \mid y^{1:t-1}) \\ &\propto p(y^t \mid x^t)\sum_{x^{t-1}} p(x^t \mid x^{t-1}, a^{t-1})p(x^{t-1} \mid y^{1:t-1}).\end{aligned} \tag{12}$$

For online filtering, the posterior distribution computed at the previous time $t-1$ is used as the prior distribution for the current time step t. An important feature of our filtering method is that given a factored prior distribution, the posterior distribution is also factored, with one factor associated with each HMM. Denoting the belief at the previous time step as $p(x^{t-1} \mid y^{t-1}) \approx u(x^{t-1}) = \prod_{i=1}^n u_i(x_i^{t-1})$ and using the factored measurement model, the joint probability is

$$\begin{aligned}&p(x^t, y^t \mid y^{1:t-1}) \\ &\propto \prod_{i=1}^n p_i(y_i^t \mid x_i^t)\sum_{x^{t-1}}\prod_{i=1}^n p_i(x_i^t \mid x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1}, a_i^{t-1})u_i(x_i^{t-1}).\end{aligned}$$

Computing the joint probability is typically intractable due to the required marginalization of all n HMMs.

Instead, a tractable computation of $\mathbb{E}_{-i}[p(x^t, y^t \mid y^{1:t-1})]$ is possible using a message-passing scheme. We first discuss necessary approximations to the ELBO (8) and then describe the scheme in the following section. We assume the joint probability satisfies a lower bound, $p(x^t, y^t \mid y^{1:t-1}) \geq \epsilon$ for some $0 < \epsilon < 1$. Given a bound ϵ, an underapproximation to the logarithm function over the interval $[\epsilon, 1]$ is the line

$$g(\theta) = \frac{\log \epsilon}{1-\epsilon}(1-\theta) \tag{13}$$

and $g(\theta) \leq \log \theta$ for $\theta \in [\epsilon, 1]$. Using (13) to approximate $\log p(x^t, y^t \mid y^{1:t-1})$ in (8) results in a surrogate ELBO

$$\underline{\text{ELBO}} = \mathbb{E}_{q(x^t)}\left[g\left(p(x^t, y^t \mid y^{1:t-1})\right) - \log q(x^t)\right]. \tag{14}$$

Theorem 1: Given $\epsilon > 0$ such that $\epsilon \leq p(x^t, y^t \mid y^{1:t-1}) \leq 1$, the surrogate ELBO (14) is a lower bound to the ELBO (8).

Proof: The difference between the surrogate ELBO (14) and the ELBO (8) is

$$\begin{aligned}&\text{ELBO} - \underline{\text{ELBO}} \\ &= \mathbb{E}_{q(x^t)}\left[\log p(x^t, y^t \mid y^{1:t-1}) - \log q(x^t)\right] - \\ &\qquad \mathbb{E}_{q(x^t)}\left[g\left(p(x^t, y^t \mid y^{1:t-1})\right) - \log q(x^t)\right] \\ &= \mathbb{E}_{q(x^t)}\left[\log p(x^t, y^t \mid y^{1:t-1})\right] - \\ &\qquad \mathbb{E}_{q(x^t)}\left[g\left(p(x^t, y^t \mid y^{1:t-1})\right)\right] \geq 0 \\ &\Rightarrow \text{ELBO} \geq \underline{\text{ELBO}} \quad \forall x^t, y^t.\end{aligned}$$

The expectation operator is linear and thus preserves the lower bound relationship of the approximation (13) to the logarithm function. The lower bound is valid for any combination of states

x^t and measurements y^t as the joint probability is bounded below by ϵ. ■

Maximizing the surrogate ELBO (14) over the factors $q_i(x_i^t)$ therefore indirectly maximizes the ELBO (8). Following the same derivation as previously described for the CAVI method, the factor objective for the surrogate ELBO is

$$\hat{\mathcal{L}}_i = -D_{\mathrm{KL}}(q_i(x_i^t) \,||\, \exp \mathbb{E}_{-i}\left[g\left(p(x^t, y^t \mid y^{1:t-1})\right)\right]).$$

The coordinate update for this objective function is

$$\begin{aligned} q_i(x_i^t) &\propto \exp \mathbb{E}_{-i}\left[g\left(p(x^t, y^t \mid y^{1:t-1})\right)\right] \\ &\propto \exp g\left(\mathbb{E}_{-i}\left[p(x^t, y^t \mid y^{1:t-1})\right]\right) \end{aligned} \tag{15}$$

and the factors are now a function of the expectation of the joint probability due to the linear approximation. Next, we develop a message-passing scheme to tractably use this update rule to update the posterior distributions for each HMM.

Remark: Imposing a lower bound on the joint probability (12) precludes combinations of states and observations that have zero probability of occurring. In practice, we round estimates of (12) lower than ϵ up to ϵ, which has the effect of introducing noise into the joint probability. For large GHMM models, probabilities naturally tend to zero, e.g., the aggregate state distribution (2) and observation distribution (3), since the product of probabilities less than one will approach zero. This approximation can therefore be seen as preventing the expectation in (15) from being zero for all states x_i^t prior to updating the posterior factor $q_i(x_i^t)$. In addition, after updating a posterior factor with (15), state probabilities lower than ϵ are rounded to zero before normalizing the distribution. This is used to preserve the idea that some state transitions must be considered impossible, e.g., in our wildfire model a healthy tree cannot transition to on fire without any neighboring trees being on fire. We show through numerical simulations in Section VI that this approach is effective. Finally, ϵ is a tuning parameter and is chosen to be a small positive value to avoid excessively influencing the posterior factors.

B. Message-Passing Scheme

So far, we have introduced an approximation to the logarithm, resulting in a lower bound to the ELBO, and derived the resulting update rule for each HMM's posterior distribution. We now build a tractable message-passing scheme to estimate the expectation $\mathbb{E}_{-i}[p(x^t, y^t \mid y^{1:t-1})]$ required in the update rule (15). For the models, we consider in this work, the expectation required for each HMM i is

$$\begin{aligned} \mathbb{E}_{-i}\left[p(x^t, y^t \mid y^{1:t-1})\right] \propto \sum_{\{x_j^t \mid j \in \mathcal{V}\setminus i\}} \left(\prod_{\substack{j=1 \\ j\neq i}}^{n} q_j(x_j^t)\right) \\ \left(\prod_{i=1}^{n} p_i(y_i^t \mid x_i^t)\right) \sum_{x^{t-1}} \prod_{i=1}^{n} p_i(x_i^t \mid x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1}, a_i^{t-1}) u_i(x_i^{t-1}). \end{aligned} \tag{16}$$

From inspection of this expression, each HMM i must marginalize out all other HMMs, using their posterior distributions. We emphasize here that it is intractable to compute this expectation exactly due to the number of HMMs and the size of their state spaces. We also do not rely on model structure to simplify this expression to a tractable form.

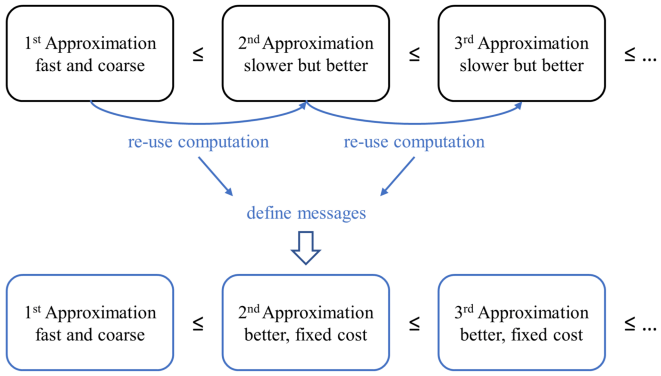

Fig. 3. Visualization of the main idea in our message-passing scheme. Since the expectation in the update rule (15) is intractable to compute exactly, we consider "kth order" approximations, where higher orders correspond to better approximations. By leveraging the structure between successive approximations, we derive a message-passing scheme with a fixed complexity at each iteration.

Therefore, we use a message-passing scheme that efficiently produces successively better approximations of the expectation each iteration by leveraging structure in the approximations; we illustrate this idea in Fig. 3. Each HMM in the GHMM maintains an estimate of its posterior factor, $q_i^k(x_i^t)$, and an estimate of the expectation (16), $E_i^k(x_i^t) \approx \mathbb{E}_{-i}[p(x^t, y^t \mid y^{1:t-1})]$; the superscript k on these quantities, and other quantities below, refers to the kth estimate. For HMM i, a "first-order" approximation assumes that only the immediate neighbors $j \in \mathcal{N}(i)$ influence its posterior

$$\begin{aligned} E_i^1(x_i^t) \propto p_i(y_i^t \mid x_i^t) \sum_{x_i^{t-1}} u_i(x_i^{t-1}) \\ \sum_{x_{\mathcal{N}(i)}^{t-1}} p_i(x_i^t \mid x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1}, a_i^{t-1}) \prod_{j\in\mathcal{N}(i)} \left[u_j(x_j^{t-1})\right]. \end{aligned} \tag{17}$$

Comparing the first-order approximation with the true expectation (16) for HMM i, we have only retained the prior distributions of the neighbors, which are used to marginalize out the neighbors' influence from the dynamics. As a result, we have ignored the measurements of all other HMMs at the current time step, which is the information that influences HMM i's posterior $q_i(x_i^t)$. However, the benefit is that the first-order approximation has very low computational cost. During the first iteration $k = 1$ of the scheme, every HMM $i \in \mathcal{V}$ uses the first-order approximation (17) to estimate the expectation (16) and update its posterior (15).

An improved "second-order" approximation for HMM i includes the measurement information for the neighbors $\mathcal{N}(i)$

$$\begin{aligned} E_i^2(x_i^t) \propto p_i(y_i^t \mid x_i^t) \sum_{x_i^{t-1}} u_i(x_i^{t-1}) \\ \sum_{x_{\mathcal{N}(i)}^{t-1}} p_i(x_i^t \mid x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1}, a_i^{t-1}) \prod_{j\in\mathcal{N}(i)} \Bigg[u_j(x_j^{t-1}) \sum_{x_j^t} q_j^1(x_j^t) \\ p_j(y_j^t \mid x_j^t) \sum_{x_{\mathcal{N}(j)}^{t-1}} p_j(x_j^t \mid x_j^{t-1}, x_{\mathcal{N}(j)}^{t-1}, a_j^{t-1}) \prod_{l\in\mathcal{N}(j)} u_l(x_l^{t-1})\Bigg]. \end{aligned} \tag{18}$$

Including the measurements from the neighbor HMMs helps inform the posterior distribution of HMM i due to the coupling interactions in its dynamics. However, improving the approximation comes at the cost of increased computational complexity, as each HMM i now requires the prior distributions from the neighbors of neighbors $\bigcup_{j\in\mathcal{N}(i)}\mathcal{N}(j)$, and an estimate of the posterior distributions from the neighbors, as shown by the brackets in (18). During the second iteration $k=2$ of the scheme, every HMM $i\in\mathcal{V}$ uses the second-order approximation (18) to estimate the expectation (16) and update its posterior. For the second iteration, we have approximated the influence of the neighbors, by assuming that each HMM i is not a part of its neighbors of neighbors $l\in\mathcal{N}(j)$. We do not require any structure in the graph to apply this approximation, and we show how it allows us to derive a tractable message-passing scheme in the following discussion.

We generalize the procedure of adding the measurements of additional HMMs to better approximate the expectation for HMM i, by identifying a recursive structure between the first- and second-order approximations. This allows us to define messages for the HMMs to share during each iteration of the scheme, to both reduce the computational complexity and to generalize to "kth-order" approximations.

The goal of the messages is to simplify the approximations so that each HMM i only requires information from its neighbors $j\in\mathcal{N}(i)$ for each iteration k. In the first-order approximation (17), each HMM simply requires the prior distributions from its neighbors, as indicated by the brackets. However, in the second-order approximation (18), each HMM requires information from its neighbors of neighbors, as indicated by the brackets. Our insight is that the expression in brackets in the second-order approximation (18) is almost identical to the entire expression of the first-order approximation (17), with the addition of the neighbors posteriors $\{q_j(x_i^t)\mid j\in\mathcal{N}(i)\}$. The key intuition here is that it is possible to reuse the computations of the first-order approximation to reduce the complexity of the second-order approximation. This idea can be generalized, so that higher order approximations do not require an increasing amount of complexity. Rather, we create a scheme with a fixed complexity at each iteration.

Therefore, we define the messages as a recursive form

$$\begin{aligned} m_i^0(x_i^{t-1}) &= u_i(x_i^{t-1})\\ m_i^k(x_i^{t-1}) &\propto u_i(x_i^{t-1})\sum_{x_i^t} q_i^k(x_i^t)p_i(y_i^t\mid x_i^t)\\ &\quad\sum_{x_{\mathcal{N}(i)}^{t-1}} p_i(x_i^t\mid x_i^{t-1},x_{\mathcal{N}(i)}^{t-1},a_i^{t-1})\prod_{j\in\mathcal{N}(i)} m_j^{k-1}(x_j^{t-1}). \end{aligned}\tag{19}$$

After substituting the messages, the first- and second-order approximations are identical except for the messages they use

$$\begin{aligned} E_i^1(x_i^t) &\propto p_i(y_i^t\mid x_i^t)\sum_{x_i^{t-1}} u_i(x_i^{t-1})\\ &\quad\sum_{x_{\mathcal{N}(i)}^{t-1}} p_i(x_i^t\mid x_i^{t-1},x_{\mathcal{N}(i)}^{t-1},a_i^{t-1})\prod_{j\in\mathcal{N}(i)} m_j^0(x_j^{t-1})\\ E_i^2(x_i^t) &\propto p_i(y_i^t\mid x_i^t)\sum_{x_i^{t-1}} u_i(x_i^{t-1})\\ &\quad\sum_{x_{\mathcal{N}(i)}^{t-1}} p_i(x_i^t\mid x_i^{t-1},x_{\mathcal{N}(i)}^{t-1},a_i^{t-1})\prod_{j\in\mathcal{N}(i)} m_j^1(x_i^{t-1}). \end{aligned}$$

Both approximations now only require gathering information from its neighbors, as desired. It is straightforward to show that the recursive structure from the message definition also holds for higher order approximations $k\geq 3$, and we have omitted the equations for additional approximations for brevity and clarity. The general form of the kth-order approximation of the expectation (16) is

$$\begin{aligned} E_i^k(x_i^t) &\propto p_i(y_i^t\mid x_i^t)\sum_{x_i^{t-1}} u_i(x_i^{t-1})\\ &\quad\sum_{x_{\mathcal{N}(i)}^{t-1}} p_i(x_i^t\mid x_i^{t-1},x_{\mathcal{N}(i)}^{t-1},a_i^{t-1})\prod_{j\in\mathcal{N}(i)} m_j^{k-1}(x_i^{t-1}). \end{aligned}\tag{20}$$

We make one additional simplification to further reduce the complexity of our scheme. Note that the definition of the messages (19), and the general form of the expectation estimate (20), share many common terms, including the prior, measurement model, dynamics, and neighbor messages. As a result, we can avoid performing duplicate computations by first combining the common terms that will be used to generate the messages and estimate the expectation. We define a "summary function" for each HMM i as

$$\begin{aligned} d_i^k(x_i^{t-1},x_i^t) &= u_i(x_i^{t-1})p_i(y_i^t\mid x_i^t)\\ &\quad\sum_{x_{\mathcal{N}(i)}^{t-1}} p_i(x_i^t\mid x_i^{t-1},x_{\mathcal{N}(i)}^{t-1},a_i^{t-1})\prod_{j\in\mathcal{N}(i)} m_j^{k-1}(x_j^{t-1}) \end{aligned}\tag{21}$$

which describes the joint probability of different transitions for HMM i, given its current measurement, after marginalizing out the influence from the HMM's neighbors. By leveraging the summary function, the messages for each HMM i are

$$\begin{aligned} m_i^0(x_i^{t-1}) &= u_i(x_i^{t-1})\\ m_i^k(x_i^{t-1}) &\propto \sum_{x_i^t} q_i^k(x_i^t)d_i^k(x_i^{t-1},x_i^t) \end{aligned}\tag{22}$$

and the estimate of the expectation is

$$E_i^k(x_i^t)\propto\sum_{x_i^{t-1}} d_i^k(x_i^{t-1},x_i^t)\approx\mathbb{E}_{-i}\left[p(x^t,y^t\mid y^{1:t-1})\right].\tag{23}$$

We normalize estimates of the expectation to estimate the normalization constant of the joint probability (12), since the approximation (13) does not allow for this constant to be factored. Overall, our scheme is summarized by using the summary (21) to produce expectation estimates (23) and updated messages (22), for each iteration. Next, we show that leveraging anonymous influence further reduces the computational complexity of our message-passing scheme.

C. Simplifying With Anonymous Influence

For a given HMM i, the main source of complexity in our message-passing scheme is computing the summary function d_i^k (21). Specifically, computing d_i^k may be intractable as marginalizing out $x_{\mathcal{N}(i)}^{t-1}$ requires considering $\prod_{j\in\mathcal{N}(i)}|\mathcal{X}_j|$ values. If an HMM has many neighbors (large $|\mathcal{N}(i)|$) or if the neighbors have large state spaces $|\mathcal{X}_j|$, then the computational cost may be significant. Therefore, we exploit anonymous influence to reduce the complexity of marginalizing the neighbors influence on an HMM. If the dynamics of an HMM utilize a CA

$$p(x_i^t\mid x_i^{t-1},x_{\mathcal{N}(i)}^{t-1},a_i^{t-1})=p_i(x_i^t\mid x_i^{t-1},z_i^{t-1},a_i^{t-1})$$

Algorithm 2: Relaxed Anonymous Variational Inference (RAVI) for Time Step t.

1: **Input:** prior factors $u_i(x_i^{t-1})$, graph G, actions a_i^{t-1}, dynamics $p_i(x_i^t \mid x_i^{t-1}, x_{\mathcal{N}(i)}^{t-1}, a_i^{t-1})$, measurements y_i^t and models $p_i(y_i^t \mid x_i^t)$
2: **Output:** posterior factors $q_i(x_i^t)\,\forall i \in \mathcal{V}$
3: **for** each vertex $i \in \mathcal{V}$ **do**
4: initialize message $m_i^0(x_i^{t-1}) = u_i(x_i^{t-1})$
5: initialize factor $q_i^0(x_i^t)$
6: **for** iteration $k = 1, \ldots, K_{\max}$ **do**
7: **for** each vertex $i \in \mathcal{V}$ **do**
8: Receive messages $\{m_j^{k-1}(x_j^{t-1}) \mid j \in \mathcal{N}(i)\}$
9: Compute summary $d_i^k(x_i^{t-1}, x_i^t)$ ((21) or (24))
10: Estimate expectation $E_i^k(x_i^t)$ (23)
11: Update posterior $q_i^k(x_i^t)$ (15)
12: Compute next message $m_i^k(x_i^{t-1})$ (22)
13: **if** factors $q_i^k(x_i^t)$ converge **then** terminate early
14: **return** posterior factors $q_i(x_i^t) = q_i^k(x_i^t)\,\forall i \in \mathcal{V}$

then the summary function d_i^k can be rewritten as

$$d_i^k(x_i^{t-1}, x_i^t) = u_i(x_i^{t-1}) p_i(y_i^t \mid x_i^t) \sum_{z_i^{t-1}} p_i(x_i^t \mid x_i^{t-1}, z_i^{t-1}, a_i^{t-1}) m_{\mathcal{N}(i)}^{k-1}(z_i^{t-1}). \quad (24)$$

In the above expression, we have modified the messages from HMM i's neighbors $\{m_j^{t-1}(x_j^{t-1}) \mid j \in \mathcal{N}(i)\}$ to an aggregate message $m_{\mathcal{N}(i)}^{k-1}(z_i^{t-1})$ summarizing the likelihood of different CA values z_i^{t-1}. Typically, marginalizing the dynamics with respect to the CA has a lower cost than considering all combinations of neighbor states.

Creating the aggregate message depends on the particular CA definition. In our forest wildfire model in Section III, the tree dynamics (see Table III) are based on the number of neighbors on fire $f_i^t \in [0, \ldots, |\mathcal{N}(i)|]$ rather than combinations of the neighbor states $x_{\mathcal{N}(i)}^t \in \prod_{j \in \mathcal{N}(i)} \mathcal{X}_j$. To create the aggregate message $m_{\mathcal{N}(i)}^{k-1}(f_i^{t-1})$ for a given HMM, we consider $2^{|\mathcal{N}(i)|}$ configurations of neighbor trees, where a configuration describes each tree as being on fire or not on fire. For each configuration, we compute the total number of neighbors on fire and the configuration likelihood under the messages $\{m_j^{k-1}(x_i^{t-1}) \mid j \in \mathcal{N}(i)\}$. Using this process, we convert the message information defined over each neighbor's state space, into a message representing the relative likelihoods for different numbers of neighbors on fire. Furthermore, creating an aggregate message has a lower cost than considering all $3^{|\mathcal{N}(i)|}$ possible neighbor state combinations, as required by the HMM dynamics without anonymous influence (1). We note that creating an aggregate message is always possible, as we are simply mapping from one discrete distribution to another. When using anonymous influence, the distribution for the CA results in a lower complexity for our message-passing scheme.

D. Algorithm Summary

Algorithm 2, relaxed anonymous variational inference (RAVI), summarizes the approximate filter for a single time step. The factors $q_i(x_i^t)$ are then used as the priors $u_i(x_i^t)$ for the next time step. The main component is the message-passing scheme, which is relatively straightforward to implement. The posterior factors are initialized to any valid discrete distribution (line 5) and the algorithms runs for a fixed number of iterations $K_{\max}$ unless the factors converge (line 13).

Remark: Our filtering approach is based on two key approximations, an approximate lower bound on the joint probability (12) and a message-passing scheme to approximate the expectation of the joint probability (16). Overall, we have developed a filter which approximately optimizes the ELBO to produce a close approximation to the true posterior at each time step. VI methods commonly rely on approximations and structure, such as ELBO bounds [31], the mean-field approximation [39], conjugate distributions [40], or tractable substructures [41]. Quantifying the error in VI approaches is typically infeasible, due to requiring knowledge of the true posterior distribution, which is intractable for GMDPs of any meaningful size. Therefore, our filtering scheme is not amenable to more rigorous analysis, such as convergence rates and error bounds, as is consistent with prior work [32], [33], [34]. Establishing formal guarantees requires certifying a global solution to a nonconvex and nonlinear optimization problem, which is infeasible other than for relatively simple cases and remains a challenging open research problem. Our contribution is an approximately optimal algorithm that builds on the VI framework, and we validate our approach with simulation experiments that we present in Section VI. In future work, we aim to characterize subclasses of GMDP models for which further analysis is feasible.

V. Scalable Constrained Control of GMDPs

We now derive approximately optimal controllers that use a maximum-likelihood estimate from our approximate filter to produce a constrained control action. We present two approaches, one based on approximate value functions, and another based on approximate state–action functions. The use of approximate value functions is more common in prior work, but requires additional structural assumptions to enforce a capacity constraint on the control action. In contrast, state–action functions are less studied in relevant prior work, but are easier to use with our capacity-constrained formulations. We present both approaches to provide a broader set of tools for controlling large-scale natural phenomena. The nomenclature for this section is provided in Table V.

We assume binary actions, $a_i^t \in \{0, 1\}\,\forall i \in \mathcal{V}$, and enforce a capacity constraint. The feasible action set is

$$\mathcal{A}_c = \left\{ a^t \in \mathcal{A} \mid \sum_{i=1}^n a_i^t \leq C \right\} \quad (25)$$

and $C \in \mathbb{Z}_{\geq 0}$ is the maximum allowed capacity. We first derive an approach based on approximate value functions.

A. Approximate Value Functions

We consider approximate value functions which are a sum of local basis functions

$$V_w(x^t) = \sum_{i=1}^n w_i^T h_i(x_{O(i)}^t) \quad (26)$$

TABLE V
NOMENCLATURE FOR CONSTRAINED CONTROL APPROACH

Symbol	Definition
b_i, c_i	Bias and linear terms for MDP i state-action approximation
w_i, h_i	Weights and basis functions for MDP i value approximation
r_i	MDP i reward function
C	Control capacity
$O(i)$	Domain of basis functions for MDP i
Q, R, V	GMDP state-action, reward, and value functions
$\mathcal{B}$	Bellman operator
$\mathcal{C}$	Equivalence class
γ	Discount factor
δ	Value function approximation error
λ_i, μ_i	Bias and linear terms in capacity-constrained program
π	Control policy
ϕ	Difference between value function and Bellman operator
ψ_i	MDP i expectation of reward and future value

which mirrors the structure of the reward function (4), where $w_i \in \mathbb{R}^{k_i}$ and $h_i : \mathcal{X}_{O(i)} \to \mathbb{R}^{k_i}$. Each basis function h_i typically relies on state information from a few MDPs, $O(i) \subseteq \mathcal{V}$ and $|O(i)| \ll |\mathcal{V}|$. This form is based on the additive structure of the reward function, which will allow us to greatly reduce the complexity of solving a linear program to determine the basis weights. We use the following bound from [42] to derive a tractable method for solving for the weights of the value function, and for determining the approximation error relative to the optimal value function.

Proposition 1 (Value function approximation error [42]): The maximum difference between an approximate value function $V_w(x^t)$ and the optimal value function $V^\star(x^t)$ is $\delta = \max_{x^t \in \mathcal{X}} |V_w(x^t) - V^\star(x^t)|$ and is bounded,

$$\delta \leq \frac{2\gamma}{1-\gamma} \max_{x^t \in \mathcal{X}} |V_w(x^t) - (\mathcal{B}V_w)(x^t)|$$

where $(\mathcal{B}V)(x^t)$ is the Bellman operator

$$(\mathcal{B}V)(x^t) = \max_{a^t \in \mathcal{A}} \mathbb{E}_p \left[R(x^t, a^t, x^{t+1}) + \gamma V(x^{t+1}) \right].$$

This bound is useful as the right-hand side (R.H.S.) involves quantities that can be approximated, and minimizing the R.H.S. explicitly minimizes the approximation error relative to the optimal value function. In the following discussion, we occasionally omit the argument of functions for clarity. Minimizing $\phi = \max_{x^t \in \mathcal{X}} |V_w(x^t) - (\mathcal{B}V_w)(x^t)|$ leads to the nonlinear program

$$\begin{aligned} \min_{\substack{w_i \in \mathbb{R}^{k_i} \\ \phi \in \mathbb{R}}} \quad & \phi \\ \text{s.t.} \quad & \phi \geq V_w(x^t) - (\mathcal{B}V_w)(x^t) \\ & \phi \geq (\mathcal{B}V_w)(x^t) - V_w(x^t) \quad \forall x^t \in \mathcal{X} \end{aligned} \tag{27}$$

where the constrained Bellman operator for the models considered in this work is

$$\begin{aligned} (\mathcal{B}V_w)(x^t) &= \max_{a^t \in \mathcal{A}_c} \mathbb{E}_p \left[\sum_{i=1}^n r_i + \gamma w_i^T h_i(x^{t+1}_{O(i)}) \right] \\ &= \max_{a^t \in \mathcal{A}_c} \sum_{i=1}^n \mathbb{E}_p \left[r_i + \gamma w_i^T h_i(x^{t+1}_{O(i)}) \right] \end{aligned}$$

with the expectation taken with respect to the aggregate dynamics model (2). Computing operations over the full state space and the feasible action set is intractable so we develop upper and lower bounds, $(\underline{\mathcal{B}V_w})(x^t) \leq (\mathcal{B}V_w)(x^t) \leq (\overline{\mathcal{B}V_w})(x^t)$. With these bounds, the following constraints are imposed:

$$\begin{aligned} &\phi \geq V_w(x^t) - (\underline{\mathcal{B}V_w})(x^t) \geq V_w(x^t) - (\mathcal{B}V_w)(x^t) \\ &\phi \geq (\overline{\mathcal{B}V_w})(x^t) - V_w(x^t) \geq (\mathcal{B}V_w)(x^t) - V_w(x^t) \quad \forall x^t \in \mathcal{X} \end{aligned}$$

and the original nonlinear program constraints are still satisfied. Let the expected immediate reward and future value for every MDP be

$$\psi_i(x^t_{O(i)\cup\mathcal{N}(O(i))}, a^t_{O(i)}) = \mathbb{E}_p \left[r_i + \gamma w_i^T h_i \right].$$

A lower bound is any action that satisfies the constraint. We use $a_i^t = 0 \, \forall i \in \mathcal{V}$ and denote this action $\overline{a}^t$

$$(\underline{\mathcal{B}V_w}) = \sum_{i=1}^n \psi_i(x^t_{O(i)\cup\mathcal{N}(O(i))}, \overline{a}^t_{O(i)}).$$

An upper bound is an overapproximation of the constrained Bellman operator by removing the capacity constraint and instead maximizing over the set of actions for each summand

$$(\overline{\mathcal{B}V_w}) = \sum_{i=1}^n \max_{a^t_{O(i)}} \psi_i(x^t_{O(i)\cup\mathcal{N}(O(i))}, a^t_{O(i)}).$$

Removing the constraint over-approximates the value of each MDP but is critical in dividing the original intractable nonlinear program into n tractable programs. The constraints in (27) simplify to

$$\begin{aligned} &\phi \geq \sum_{i=1}^n -w_i^T h_i + \max_{a^t_{O(i)}} \psi_i(x^t_{O(i)\cup\mathcal{N}(O(i))}, a^t_{O(i)}) \\ &\phi \geq \sum_{i=1}^n w_i^T h_i - \psi_i(x^t_{O(i)\cup\mathcal{N}(O(i))}, \overline{a}^t_{O(i)}), \forall x^t \in \mathcal{X}. \end{aligned} \tag{28}$$

We now decompose the approximation error $\phi = \sum_{i=1}^n \phi_i$ to impose the following constraints instead:

$$\begin{aligned} &\phi_i \geq -w_i^T h_i + \max_{a^t_{O(i)}} \psi_i(x^t_{O(i)\cup\mathcal{N}(O(i))}, a^t_{O(i)}) \\ &\phi_i \geq w_i^T h_i - \psi_i(x^t_{O(i)\cup\mathcal{N}(O(i))}, \overline{a}^t_{O(i)}) \quad \forall x^t_{O(i)\cup\mathcal{N}(O(i))}. \end{aligned}$$

This overapproximates ϕ by adding structure to the error contribution of each MDP but reduces the coupled nonlinear program into n separate nonlinear programs. In addition, the constraints in (28) are still satisfied after adding this structure. The maximum operator is then replaced by adding a constraint for each action, which results in the linear program

$$\begin{aligned} \min_{\substack{w_i \in \mathbb{R}^{k_i} \\ \phi_i \in \mathbb{R}}} \quad & \phi_i \\ \text{s.t.} \quad & \phi_i \geq -w_i^T h_i(x^t_{O(i)}) + \psi_i(x^t_{\mathcal{N}(O(i))}, a^t_{O(i)}) \\ & \quad \forall x^t_{O(i)\cup\mathcal{N}(O(i))}, a^t_{O(i)} \\ & \phi_i \geq w_i^T h_i(x^t_{O(i)}) - \psi_i(x^t_{O(i)\cup\mathcal{N}(O(i))}, \overline{a}^t_{O(i)}) \\ & \quad \forall x_{O(i)\cup\mathcal{N}(O(i))}. \end{aligned} \tag{29}$$

Each program contains $k_i + 1$ variables and $|\mathcal{X}_{O(i)\cup\mathcal{N}(O(i))}|(|\mathcal{A}_{O(i)}| + 1)$ constraints, and there is one linear program associated with each MDP in the GMDP. The quantity $\phi = \sum_{i=1}^{n} \phi_i$ is a suboptimality estimate of V_w compared to the optimal constrained value function $V^\star$. Furthermore, our approach yields the optimal constrained value function if all suboptimality estimates are zero.

Theorem 2: If all errors are zero, $\phi_i = 0\,\forall i \in \mathcal{V}$, the approximate value function is optimal, $V_w(x^t) = V^\star(x^t)\,\forall x^t \in \mathcal{X}$.

Proof: If the individual errors are zero, then the total error is also zero, since $\phi = \sum_{i=1}^{n} \phi_i$. In this case, $V_w(x^t) = (\mathcal{B}V_w)(x^t)\,\forall x^t \in \mathcal{X}$, from the definition of the total error, $\phi = \max_{x^t\in\mathcal{X}} |V_w(x^t) - (\mathcal{B}V_w)(x^t)|$. In this case, the statement follows from Proposition 1, as the R.H.S. of the inequality is zero and $\delta = 0$. ■

Simplifying with Symmetry: By inspection of program 31, two MDPs will have the same solution w_i, ϕ_i if they have the same dynamics p_i and reward r_i, and use the same approximation $w_i^T h_i$. Therefore, for all MDPs with the same dynamics and reward, we apply the same value approximation to reduce the number of equivalence classes, and thus, the number of programs that must be solved. We formalize this idea in the following theorem.

Theorem 3: For a GMDP containing $s \leq n$ equivalence classes, the value function ALP method requires solving s linear programs and $\sum_{k=1}^{s} |\mathcal{C}_k|\phi_k$ is the suboptimality error.

Proof: The constraints in Program (29) are uniquely defined by the reward function r_i, dynamics p_i, and basis approximation $w_i^T h_i$. By definition, all MDPs in the same equivalence class have identical reward functions r_i and dynamics p_i. Therefore, using the same basis approximation $w_i^T h_i$ for all MDPs in the same equivalence class results in identical Programs (29). As a result, for n equivalence classes, the solution to each program is unique as no two MDPs share the same class. When there are $s < n$ classes, there are at most s unique solutions for all n linear programs. In this case, only one linear program per equivalence class must be solved as the solution for the per-MDP program (29) is identical for all MDPs within a class. ■

B. Approximate State–Action Functions

We now derive a novel ALP approach to produce a state–action function (i.e., a Q-function) to approximate a constrained value function. Our approximate state–action function approach is based on the following bound from [42].

Proposition 2 (State–action function approximation error [42]): The maximum difference between an approximate state–action function $Q_w(x^t, a^t)$ and the optimal state–action function $Q^\star(x^t, a^t)$ is $\delta = \max_{x^t\in\mathcal{X}, a^t\in\mathcal{A}} |Q_w(x^t, a^t) - Q^\star(x^t, a^t)|$ and is bounded

$$\delta \leq \frac{2}{1-\gamma} \max_{x^t\in\mathcal{X}, a^t\in\mathcal{A}} |Q_w(x^t, a^t) - (\mathcal{B}Q_w)(x^t, a^t)|$$

where $(\mathcal{B}Q)(x^t, a^t)$ is the Bellman operator

$$(\mathcal{B}Q)(x^t, a^t) = \mathbb{E}_p\left[R(x^t, a^t, x^{t+1}) + \gamma \max_{a^{t+1}\in\mathcal{A}} Q(x^{t+1}, a^{t+1})\right].$$

We assume the approximate state–action function Q_w form

$$Q_w(x^t, a^t) = \sum_{i=1}^{n} w_i^T b_i(x^t_{O(i)}) + a_i^t w_i^T c_i(x^t_{O(i)}) \tag{30}$$

with $w_i \in \mathbb{R}^{k_i}$ and $b_i, c_i : \mathcal{X}_{O(i)} \subseteq \mathcal{X} \to \mathbb{R}^{k_i}$, specifically for our capacity constrained formulations that we describe in the next section. Minimizing $\phi = \max_{x^t\in\mathcal{X}, a^t\in\mathcal{A}_c} |Q_w(x^t, a^t) - (\mathcal{B}Q_w)(x^t, a^t)|$ results in the nonlinear program

$$\begin{aligned}
\min_{\substack{w_i\in\mathbb{R}^{k_i}\\ \phi\in\mathbb{R}}} \quad & \phi \\
\text{s.t.} \quad & \phi \geq Q_w(x^t, a^t) - (\mathcal{B}Q_w)(x^t, a^t) \\
& \phi \geq (\mathcal{B}Q_w)(x^t, a^t) - Q_w(x^t, a^t) \\
& \forall x^t \in \mathcal{X}, a^t \in \mathcal{A}_c,
\end{aligned}$$

where the nonlinearity is due to the maximization in the Bellman operator. We follow a similar procedure as before to develop a tractable and scalable method. The constrained Bellman operator is

$$(\mathcal{B}Q)(x^t, a^t) = \mathbb{E}_p\Bigg[\sum_{i=1}^{n} r_i(x^t_{O(i)\cup\mathcal{N}(O(i))}, a^t_{O(i)}, x^{t+1}_{O(i)}) + \cdots \gamma \max_{a^{t+1}\in\mathcal{A}_c} \sum_{i=1}^{n} w_i^T b_i(x^{t+1}_{O(i)}) + a_i^{t+1} w_i^T c_i(x^{t+1}_{O(i)})\Bigg].$$

We construct upper and lower bounds for the constrained Bellman operator to instead impose the constraints

$$\begin{aligned}
& \phi \geq Q_w(x^t, a^t) - (\underline{\mathcal{B}Q_w})(x^t, a^t) \\
& \phi \geq (\overline{\mathcal{B}Q_w})(x^t, a^t) - Q_w(x^t, a^t) \quad \forall x^t \in \mathcal{X}, a^t \in \mathcal{A}_c.
\end{aligned}$$

Function arguments are omitted at times for clarity in the following discussion. A lower bound is any action a^{t+1} that satisfies the control constraint. A convenient choice is $a_i^{t+1} = 0\,\forall i \in \mathcal{V}$ thus

$$(\underline{\mathcal{B}Q_w}) = \sum_{i=1}^{n} \mathbb{E}_p\left[r_i + \gamma w_i^T b_i\right].$$

An upper bound is found by removing the capacity constraint and choosing actions to improve the total value of Q_w

$$(\overline{\mathcal{B}Q_w}) = \sum_{i=1}^{n} \mathbb{E}_p\left[r_i + \gamma w_i^T b_i + \gamma \max\{0, w_i^T c_i\}\right].$$

The maximization in the upper bound is replaced by two linear constraints and the error ϕ is decomposed as a sum $\sum_{i=1}^{n} \phi_i$. The result is the following per-MDP linear program:

$$\begin{aligned}
\min_{\substack{w_i\in\mathbb{R}^{k_i}\\ \phi_i\in\mathbb{R}}} \quad & \phi_i \\
\text{s.t.} \quad & \phi_i \geq w_i^T b_i + a_i^t w_i^T c_i - \mathbb{E}_p\left[r_i + \gamma w_i^T b_i\right] \\
& \phi_i \geq \mathbb{E}_p\left[r_i + \gamma w_i^T b_i\right] - w_i^T b_i - a_i^t w_i^T c_i \\
& \phi_i \geq \mathbb{E}_p\left[r_i + \gamma w_i^T b_i + \gamma w_i^T c_i\right] - w_i^T b_i - a_i^t w_i^T c_i \\
& \forall x^t_{O(i)\cup\mathcal{N}(O(i))}, a^t_{O(i)}.
\end{aligned} \tag{31}$$

Each program contains $k_i + 1$ variables and $3|\mathcal{X}_{O(i)\cup\mathcal{N}(O(i))}||\mathcal{A}_{O(i)}|$ constraints. Solving program (31) for each MDP does not enforce that the total control effort will satisfy the capacity constraint. However, allowing infeasible actions results in a more conservative approximation of the true constrained value function since adding constraints to program (31) cannot lower the error ϕ_i. Our approach also yields the optimal constrained state–action function when the suboptimality errors are zero.

Theorem 4: If all errors are zero, $\phi_i = 0\,\forall i \in \mathcal{V}$, then the approximate state–action function is optimal, $Q_w(x^t, a^t) = Q^\star(x^t, a^t)\,\forall x^t \in \mathcal{X}, a^t \in \mathcal{A}_c$.

Proof: If the individual errors are zero, then the total error is also zero, $\phi = 0$, since $\phi = \sum_{i=1}^n \phi_i$. In this case, $Q_w(x^t, a^t) = (\mathcal{B}Q_w)(x^t, a^t)\,\forall x^t \in \mathcal{X}, a^t \in \mathcal{A}_c$, from the definition of the total error, $\phi = \max_{x^t\in\mathcal{X}, a^t\in\mathcal{A}_c} |Q_w(x^t, a^t) - (\mathcal{B}Q_w)(x^t, a^t)|$. As a result, the statement follows from Proposition 2, as the R.H.S. of the inequality is zero and $\delta = 0$. ■

Simplifying with Symmetry: Similar to Theorem 3, we again apply symmetry to reduce the number of programs that must be solved, by applying the same basis approximation for all MDPs that have the same dynamics and reward.

Theorem 5: For a GMDP containing $s \leq n$ unique equivalence classes, the approximate state–action function Q_w ALP method requires solving s linear programs and $\sum_{k=1}^s |\mathcal{C}_k|\phi_k$ is the approximation error.

Proof: The approximate state–action function Q_w is determined after solving for the weights w_i. Program (31) for two MDPs have identical solutions w_i, ϕ_i if both are in the same equivalence class and the same basis functions are used. Therefore, for n classes, n programs are solved to determine Q_w. Only s programs are solved for $s < n$ classes as the solution is identical for all MDPs in the same class. ■

C. Capacity-Constrained Programs

In the previous two sections, we derived methods for constructing an approximate value or state–action function. However, extracting a policy is nontrivial due to the $\binom{n}{C}$ possible feasible constrained actions. Therefore, we now introduce a class of linear programs that have a capacity constraint and an explicit solution, and then discuss leveraging this class of programs to build a policy for an approximate value or state–action function.

Proposition 3 (Capacity-constrained linear program): The integer linear program

$$\max_{a^t\in\mathcal{A}_c} \lambda + \sum_{i=1}^n \mu_i a_i^t \tag{32}$$

where $\lambda, \mu_i \in \mathbb{R}$, has an explicit solution. Assume that $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$. An optimal solution is

$$a_i^t = \begin{cases} 1 & \text{if } i \leq C \text{ and } \mu_i \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{33}$$

Proof: The values μ_i can always be sorted a priori. The solution is optimal, as changing it cannot improve the objective. Consider an optimal solution described as $a_i^t = 1$ for $i \in \{1, \ldots, j\}$ with $j \leq C$ and zero otherwise. If $j = C$, then choosing $a_k^t = 1$ for any $k > j$ will violate the constraint. If $j < C$, then choosing $a_k^t = 1$ for $j < k \leq C$ lowers the objective as μ_k must be negative. Finally, switching $a_k^t = 1$ to $a_k^t = 0$ for $k \leq j$ does not improve the objective as μ_k must be nonnegative. ■

We now discuss the conditions under which our approximate functions result in the policy described by (33).

Theorem 6: For approximate value functions V_w, if the form of the dynamics (1), reward functions (4), and basis functions (26) result in

$$\mathbb{E}_p\left[R(x^t, a^t, x^{t+1}) + \gamma V_w(x^{t+1})\right] = \lambda + \sum_{i=1}^n \mu_i a_i^t \tag{34}$$

then the constrained policy is determined by (33). Furthermore, for approximate state–action functions of the form in (30), the constrained policy is determined by (33).

Proof: The approximate value function is determined after solving for the weights of the basis function representation. If the relationship in (34) holds, the constrained policy is

$$\pi(x^t) = \arg\max_{a^t\in\mathcal{A}_c} \lambda + \sum_{i=1}^n \mu_i a_i^t.$$

Therefore, the policy $\pi(x^t)$ is determined by (33). For approximate state–action functions, if the assumed form (30) is used then the constrained policy is

$$\pi(x^t) = \arg\max_{a^t\in\mathcal{A}_c} \sum_{i=1}^n w_i^T b_i(x_{O(i)}^t) + a_i^t w_i^T c_i(x_{O(i)}^t). \tag{35}$$

Let $\lambda = \sum_{i=1}^n w_i^T b_i(x_{O(i)}^t)$ and $\mu_i = w_i^T c_i(x_{O(i)}^t)$. The constrained maximization (35) is equivalent to program (32) and so the policy is determined by (33). ■

For approximate value functions, the condition (34) depends on the MDP dynamics p_i, MDP reward functions r_i, and the choice of approximation $w_i^T h_i$. The reward functions and basis function approximation are a design choice in our framework, and can be adjusted to meet this condition. The MDP dynamics p_i are typically specified for each problem domain, and approximations can be considered as well. We present this approach for determining approximate value functions for large GMDPs, as the majority of prior work focuses on algorithms for approximate value functions, without considering state–action functions. We also present an approach for approximate state–action functions, as it is straightforward satisfy the conditions of Theorem 6 by leveraging the structured form (30).

We note that in our control approach, we develop Bellman operator bounds that approximate the capacity constraint. However, the errors $\phi_i \in \mathcal{V}$ describe the suboptimality of a given value or state–action function approximation, and our approach recovers the optimal solution when the errors are zero (see Theorems 2 and 4). If the errors are nonzero, we can iterate on the choice of basis functions to improve the approximation. Our control approach for approximate value functions, including Theorem 6, can be seen as a significant generalization of prior work [11]. We show in our experiments in Section VI that this generalization is necessary, as prior work is ineffective in controlling the wildfire process.

D. Simplifying With Anonymous Influence

Theorems 3 and 5 describe how we leverage symmetry to reduce the total number of linear programs that must be solved to fully determine an approximate value or state–action function. Similarly, anonymous influence can be exploited to simplify the implementation of the programs (29) and (31). For example, consider the wildfire model (see Section III) and let

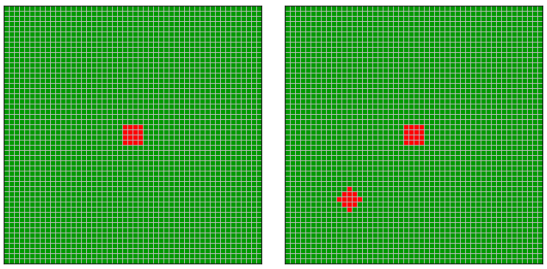

Fig. 4. Both: Visualization of the forest, where green represents healthy trees and red represents trees on fire. Left: Initial condition for our experiments comparing our framework to prior work. Right: Initial condition for the multiple fire locations scenario.

$O(i) = i \cup \mathcal{N}(i)$ and $|\mathcal{N}(i)| = 4$ for all trees. Without MMFs, program (29) requires enumerating on the order of 10^7 state combinations. By using an MMF, for the basis functions we present in Section VI, we only need to consider on the order of 10^3 state combinations. This reduction significantly simplifies the implementation of our framework that is still tractable for graphs where MDPs may have many neighbors or large state spaces. In the next section, we present several simulation experiments to validate the performance of our filter and the combined filter and controller.

VI. Simulation Experiments

We use a forest size of 50×50 with 10^{1192} total states. For comparing with prior work, values of $\alpha = 0.2$ and $\beta = 0.9$ were used in the forest wildfire model (see Section III). We also present experiments with multiple initial fire locations and nonuniform parameters α, β for the tree dynamics. Fig. 4 shows the initial condition and simulations terminate when there are no more trees on fire.

A. Filter Performance

We compare our filter RAVI against LBP that we adapt for online sequential estimation by limiting the model size to a maximum of $H = 3$ time layers. For each time step, if adding a layer exceeds the limit, we remove the oldest layer for LBP. After updating the layers, we retain the beliefs and measurements of the older layers, to re-use previous information. Our implementation of LBP results in an incremental version to limit the computational complexity of performing inference. For RAVI, the value of ϵ is 10^{-10}. Both RAVI and LBP are terminated early if less than 1% of the posterior factors stop changing their maximum likelihood belief. We also compare against taking each measurement as the estimate for each time step, which may produce inconsistent estimates, e.g., a tree transitioning from healthy to burnt in one time step. All filters were initialized with the ground truth.

At each time step of the simulation, the belief produced by the filter is converted to a maximum likelihood estimate and compared to the true state. We compute the percentage of trees whose states are correctly estimated as the "accuracy" of the estimator at that time step. The result is a time history of accuracies for a single simulation, as shown in Fig. 5. We compute the median accuracy for the time history, which we call the simulation accuracy. We run ten total simulations and report the first quartile, the median, and the third quartile simulation

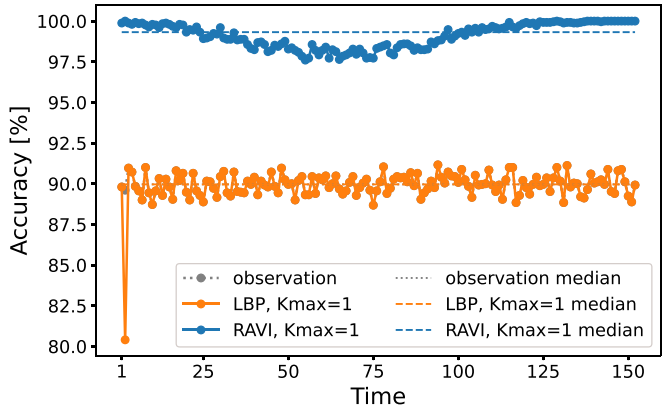

Fig. 5. Example filter results for a single model simulation. The simulation accuracy for a filter is the median accuracy over the entire time series. Here, LBP is the same as taking as the measurement as the estimate, as it overlays the measurement accuracy in the plot. In contrast, RAVI is 9% better.

TABLE VI
Filter Results for Two Different Measurement Accuracies p_c in (6)

Method		Measurement Accuracy 80%	Measurement Accuracy 90%	Estimate Time (seconds)
Measurement		$80.0^{+0.1}_{-0.1}\%$	$90.0^{+0.2}_{-0.1}\%$	—
LBP	$K_{\max} = 1$	$80.0^{+0.0}_{-0.0}\%$	$90.0^{+0.0}_{-0.0}\%$	138.73
	$K_{\max} = 2$	$85.0^{+0.8}_{-1.1}\%$	$94.5^{+0.9}_{-0.8}\%$	251.43
RAVI	$K_{\max} = 1$	$98.0^{+0.4}_{-0.3}\%$	$99.4^{+0.1}_{-0.2}\%$	1.41
	$K_{\max} = 5$	$98.6^{+0.2}_{-0.2}\%$	$99.5^{+0.0}_{-0.1}\%$	4.25
	$K_{\max} = 10$	$98.6^{+0.2}_{-0.2}\%$	$99.5^{+0.0}_{-0.1}\%$	4.23

Notes: Data are the median simulation accuracy for ten simulations, and the subscript and superscript indicate the first and third quartiles, respectively. LBP improves with more iterations but is slow while RAVI is accurate and fast enough to be used online.

accuracy. Table VI shows the results for different message passing iteration limits and measurement model accuracies p_c in (6). While $p_c = 0.9$ may seem like a very accurate measurement model, there is a $0.9^{2500} \approx 10^{-115}$ probability that the true ground state is observed. For the lowest limit $K_{\max} = 1$, LBP is slow and is the same accuracy as simply taking the measurement as the estimate. While we showed in prior work that LBP can be effective when given enough iterations [4], it does not scale to the model size in this work and cannot be used online. For example, the 2018 Camp wildfire in Northern California at one point was spreading at a rate of 80 acres per minute [43]. At this rate, a wildfire burns 184 acres in 138 s, compared to 5.33 acres in 4 s. Therefore, it is critical to use a fast, accurate online filter to enable an effective response to natural disasters. Lastly, Fig. 5 shows that our filtering approach produces better estimates even for a low iteration limit, which is independent of improving the computational time of LBP, e.g., through leveraging our structural assumptions.

In contrast, RAVI is effective even for the low iteration limit, and improves slightly given more iterations. Coordinate ascent methods are known to find local optima and RAVI quickly finds a solution that does not significantly change with more

iterations. In particular, the time to produce an estimate drops for $K_{\max} = 10$, which is likely due to slightly more accurate posterior factors at earlier time steps in each simulation.

B. Closed-Loop Filter and Controller Performance

We call our control approach Approximate Constrained Scalable Allocation of Resources (ACSAR) and present simulation results to demonstrate the performance of our closed-loop filter and control approach, RAVI ACSAR.

Given the filtering results in Table VI, we use the measurement and RAVI (with $K_{\max} = 5$) and $p_c = 0.9$ as filtering methods. We use ACSAR to generate an approximate value and state–action function, and we compare against a prior method for approximate value functions. The control effectiveness parameter used was $\Delta\beta = 0.45$, the discount factor was $\gamma = 0.95$, and the control capacity was $C = 5$. The following reward and basis functions are used for all MDPs $i \in \mathcal{V}$ with our approximate value function approach:

$$r_i(x^t_{i\cup\mathcal{N}(i)}) = \mathbf{1}_H(x^t_i) - \mathbf{1}_F(x^t_i)e^t_i$$
$$w_i^T h_i(x^t_{i\cup\mathcal{N}(i)}) = [w_i]_1 + [w_i]_2\,\mathbf{1}_H(x^t_i) + [w_i]_3\,\mathbf{1}_F(x^t_i)e^t_i$$

where $e^t_i = \sum_{j\in\mathcal{N}(i)} \mathbf{1}_H(x^t_j)$ is the number of healthy trees that are neighbors of tree i. Furthermore, we assume that every tree has four neighbors, $|\mathcal{N}(i)| = 4\,\forall i \in \mathcal{V}$, so that there is one equivalence class. The derived policy is then applied to the original graph model.

Computing the expectation (34) yields the following action weights in the policy, $\mu_i = -\gamma[w_i]_3\mathbf{1}_F(x^t_i)\Delta\beta \sum_{j\in\mathcal{N}(i)} \mathbf{1}_H(x^t_j)(1-\alpha f^t_j)$. Solving program (29) yields $[w_i]_3 = -1.43$, and so this policy treats fires in priority of the number of neighboring healthy trees and their likelihood of remaining healthy at the next time step. We compare our approach with the basis approximation proposed in [11]. The basis functions are

$$w_i^T h_i^{\text{prior}}(x^t_i) = [w_i]_1\,\mathbf{1}_H(x^t_i) + [w_i]_2\,\mathbf{1}_F(x^t_i) + [w_i]_3\,\mathbf{1}_B(x^t_i).$$

The action weight for the policy (33) when using this basis with the same reward function is $\mu_i = \gamma\Delta\beta\mathbf{1}_F(x^t_i)([w_i]_3 - [w_i]_2)$. Therefore, the resulting policy is to randomly treat trees on fire at each time step.

Table VII summarizes the results for two filtering methods in combination with two control approaches. We present the median percent of surviving healthy trees, along with the first and third quartile, to summarize the performance of the overall framework. Without control, nearly the entire forest typically burns down. Although the measurement appears to be accurate enough for control, there are many cases where a tree is believed to be on fire but is actually healthy or burnt. As a result, control actions are wasted as treating a healthy or burnt tree has no effect. Finally, only the combination of our filter and our value function basis approximation is successful in extinguishing the wildfire.

We also construct an approximate state–action function Q_w using ACSAR to illustrate a complementary approach. We use the following reward and basis functions for all MDPs $i \in \mathcal{V}$

$$r_i(x^t_i, x^{t+1}_i) = \mathbf{1}_H(x^t_i) - (1-a^t_i)\mathbf{1}_F(x^{t+1}_i)$$
$$w_i^T b_i(x^t_i) = [w_i]_1 + [w_i]_2\,\mathbf{1}_H(x^t_i) + [w_i]_3\,\mathbf{1}_F(x^t_i)$$
$$a^t_i w_i^T c_i(x^t_{i\cup\mathcal{N}(i)}) = a^t_i\,[w_i]_4\,\mathbf{1}_F(x^t_i)e^t_i. \tag{36}$$

TABLE VII
RESULTS FOR TWO FILTER METHODS AND TWO CHOICES OF VALUE FUNCTION BASIS APPROXIMATIONS

Filter Method	Control Method	Remaining Healthy Trees
—	No Control	$1.0^{+0.0}_{-0.0}\%$
Measurement	Prior Work [11] $V_w(x^t)$	$1.3^{+0.3}_{-0.3}\%$
	ACSAR $V_w(x^t)$	$2.2^{+0.5}_{-0.3}\%$
RAVI $K_{\max} = 5$	Prior Work [11] $V_w(x^t)$	$1.9^{+1.7}_{-0.4}\%$
	ACSAR $V_w(x^t)$	$97.8^{+0.6}_{-1.0}\%$

Notes: Data are the median percent of remaining healthy trees over 100 simulations, with the subscript and superscript denoting the first and third quartile, respectively. Without control, the majority of the forest burns down. An accurate filter is required, as otherwise control effort is wasted on trees that are believed to be on fire but are actually healthy or burnt. Only our filtering method RAVI and our control approach ACSAR is successful in preserving the majority of trees in the forest.

TABLE VIII
APPROXIMATION ERRORS FOR OUR APPROACH AND PRIOR WORK

Control Method	Approximation Error ϕ_i
Prior Work [11] $V_w(x^t)$	2.29
ACSAR $V_w(x^t)$	1.97
ACSAR $Q_w(x^t)$	0.84

Notes: We achieve significantly lower error, which results in effective constrained policies.

TABLE IX
RESULTS FOR ADDITIONAL SCENARIOS, BASED ON MULTIPLE INITIAL FIRE LOCATIONS AND NONUNIFORM TREE DYNAMICS PARAMETERS α AND β

Filter Method	Control Method	Remaining Healthy Trees
Scenario: multiple initial fire locations		
RAVI $K_{\max} = 5$	ACSAR $Q_w(x^t)$	$96.0^{+0.8}_{-0.5}\%$
Scenario: non-uniform parameters α, β		
RAVI $K_{\max} = 5$	ACSAR $Q_w(x^t)$	$98.3^{+0.4}_{-0.5}\%$

Notes: Our framework can directly be applied to these scenarios, and is successful in controlling the wildfire as indicated by the fraction of remaining healthy trees.

Over 100 simulations, the percent of remaining healthy trees was $97.8^{+0.7}_{-1.0}\%$. In Table VIII, we compare the approximation errors for ACSAR and prior work [11]. By developing a more general framework, we are able to achieve lower approximation errors, which translates to effective constrained policies. Furthermore, it is more straightforward to consider approximate state–action functions that fit the form (30), from which the policy is easily determined. Finally, Tables VII and VIII show that while the method in [11] is scalable to large GMDPs, the result is an ineffective constrained control policy. Therefore, only leveraging structural assumptions, such as anonymous influence and symmetry, is insufficient which validates our filtering and control methods.

C. Additional Framework Scenarios

We consider two additional scenarios, to demonstrate the flexibility of our framework, and we present the results in Table IX. For both scenarios, we use RAVI with $K_{\max} = 5$ as the filtering method, and ACSAR for approximate state–action functions as

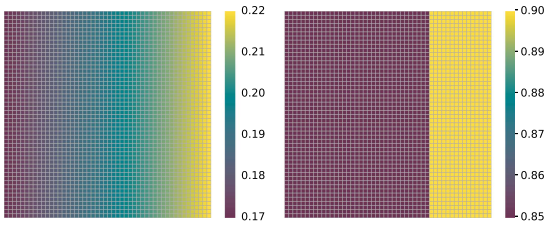

Fig. 6. Nonuniform tree dynamics parameters (left) α and (right) β representing a forest wildfire under a west to east wind, with some trees remaining on fire for longer.

the control method. In the first scenario, we consider multiple initial fire locations, with control capacity $C = 10$. The forest size and tree dynamics parameters are unchanged. We use the same reward and basis functions (36), which results in the same approximation error $\phi_i = 0.84$ for the single equivalence class. Since our approach does not rely on the initial condition, it is straightforward to consider multiple fire locations, and our framework is still effective.

For the second scenario, we vary the tree dynamics parameters α and β across the lattice to simulate wind and trees that remain on fire for longer (see Fig. 6). We use a capacity of $C = 5$ with an initial 4×4 grid of fire in the forest center. We again use the reward and basis functions (36) as before. For this scenario, we consider 50 equivalence classes for the different combinations of (α, β) parameters, which requires solving 50 linear programs to determine the basis weights. Furthermore, each equivalence class assumes neighboring trees use the same dynamics, to avoid enumerating an exponential number of possible parameters for the neighbors. We use this assumption to tractably apply our control approach, which results in an effective constrained policy as shown by the fraction of remaining healthy trees. Previously, we developed a control framework [44] using percolation theory to better address heterogeneous stochastic processes. We plan to extend our framework to more general process models in future work.

VII. Conclusion

In this work, we proposed a certainty-equivalence approach to build a framework capable of addressing large GMDPs with control constraints and measurement uncertainty. After separating the problem into two parts, we derived approximately optimal filtering and control methods. Overall, prior work is unable to address GMDPs with arbitrary graph structure, state uncertainty, large state and measurement spaces, and strict control constraints. Furthermore, scalable methods generally do not provide feedback or insight on the approximation quality. In contrast, we show that our framework is able to address all of these aspects, making it a unique approach to allow GMDPs to be applied to real-world problems. In future work, we plan to investigate other theoretical frameworks to reduce structural assumptions. Prior work on factored POMDP formulations may suggest approaches for a scalable framework with improved analysis and guarantees.

Acknowledgment

The authors would like to thank Adam Caccavale, Eric Cristofalo, Preston Culbertson, and Kunal Shah for their insightful suggestions.

References

[1] V. Raghavan, G. ver Steeg, A. Galstyan, and A. G. Tartakovsky, "Coupled hidden Markov models for user activity in social networks," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, 2013, pp. 1–6.
[2] W. Dong, A. S. Pentland, and K. A. Heller, "Graph-coupled HMMs for modeling the spread of infection," in *Proc. 28th Conf. Uncertainty Artif. Intell.*, 2012, pp. 227–236.
[3] R. N. Haksar and M. Schwager, "Controlling large, graph-based MDPs with global control capacity constraints: An approximate LP solution," in *Proc. IEEE 57th Conf. Decis. Control*, 2018, pp. 35–42.
[4] R. N. Haksar, J. Lorenzetti, and M. Schwager, "Scalable filtering of large graph-coupled hidden Markov models," in *Proc. IEEE Conf. Decis. Control*, 2019, pp. 1307–1314.
[5] R. N. Haksar and M. Schwager, "Learning large graph-based MDPs with historical data," *IEEE Trans. Control Netw. Syst.*, vol. 9, no. 3, pp. 1447–1458, Sep. 2022.
[6] C. Guestrin, D. Koller, and R. Parr, "Solving factored POMDPs with linear value functions," in *Proc. 70th Int. Joint Conf. Artif. Intell. Workshop Plan. Under Uncertainty Incomplete Inf.*, 2001, pp. 67–75.
[7] J. Pajarinen, J. Peltonen, A. Hottinen, and M. A. Uusitalo, "Efficient planning in large POMDPs through policy graph based factorized approximations," in *Machine Learning and Knowledge Discovery in Databases*, J. L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, Eds., Berlin, Germany: Springer, 2010, pp. 1–16.
[8] T. Veiga, M. Spaan, and P. Lima, "Point-based POMDP solving with factored value function approximation," in *Proc. AAAI Conf. Artif. Intell.*, 2014, vol. 28.
[9] P. Poupart and C. Boutilier, "VDCBPI: An approximate scalable algorithm for large POMDPs," in *Advances in Neural Information Processing Systems*, 17 L. K. Saul, Y. Weiss, and L. Bottou, Eds., Cambridge, MA, USA: MIT Press, 2005, pp. 1081–1088.
[10] C. Boutilier, R. Dearden, and M. Goldszmidt, "Stochastic dynamic programming with factored representations," *Artif. Intell.*, vol. 121, no. 1-2, pp. 49–107, 2000.
[11] N. Forsell and R. Sabbadin, "Approximate linear-programming algorithms for graph-based Markov decision processes," in *Proc. Eur. Conf. Artif. Intell.*, 2006, pp. 590–594.
[12] D. Koller and R. Parr, "Computing factored value functions for policies in structured MDPs," in *Proc. Int. Joint Conf. Artif. Intell.*, 1999, pp. 1332–1339.
[13] R. Sabbadin, N. Peyrard, and N. Forsell, "A framework and a mean-field algorithm for the local control of spatial processes," *Int. J. Approx. Reasoning*, vol. 53, no. 1, pp. 66–86, 2012.
[14] Q. Cheng, Q. Liu, F. Chen, and A. T. Ihler, "Variational planning for graph-based MDPs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2976–2984.
[15] C. Guestrin, D. Koller, R. Parr, and S. Venkataraman, "Efficient solution algorithms for factored MDPs," *J. Artif. Intell. Res.*, vol. 19, pp. 399–468, 2003.
[16] F. Chen, Q. Cheng, J. Dong, Z. Yu, G. Wang, and W. Xu, "Efficient approximate linear programming for factored MDPs," *Int. J. Approx. Reasoning*, vol. 63, pp. 101–121, 2015.
[17] P. Robbel, F. A. Oliehoek, and M. J. Kochenderfer, "Exploiting anonymity in approximate linear programming: Scaling to large multiagent MDPs," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 2537–2573.
[18] E. Altman, *Constrained Markov Decision Processes*. Boca Raton, FL, USA: CRC Press, 1999.
[19] N. Meuleau et al., "Solving very large weakly coupled Markov decision processes," in *Proc. AAAI Conf. Artif. Intell.*, 1998, pp. 165–172.
[20] D. Adelman and A. J. Mersereau, "Relaxations of weakly coupled stochastic dynamic programs," *Operations Res.*, vol. 56, no. 3, pp. 712–727, 2008.
[21] H.-J. Schütz and R. Kolisch, "Approximate dynamic programming for capacity allocation in the service industry," *Eur. J. Oper. Res.*, vol. 218, no. 1, pp. 239–250, 2012.
[22] C. B. Browne et al., "A survey of Monte Carlo tree search methods," *IEEE Trans. Comput. Intell. AI Games*, vol. 4, no. 1, pp. 1–43, Mar. 2012.
[23] D. F. Ciocan and V. Farias, "Model predictive control for dynamic resource allocation," *Math. Oper. Res.*, vol. 37, no. 3, pp. 501–525, 2012.
[24] F. Ye, H. Zhu, and E. Zhou, "Weakly coupled dynamic program: Information and Lagrangian relaxations," *IEEE Trans. Autom. Control*, vol. 63, no. 3, pp. 698–713, Mar. 2018.
[25] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statist. Comput.*, vol. 10, no. 3, pp. 197–208, Jul. 2000.

[26] A. Beskos, D. Crisan, A. Jasra, K. Kamatani, and Y. Zhou, "A stable particle filter for a class of high-dimensional state-space models," *Adv. Appl. Probability*, vol. 49, no. 1, pp. 24–48, 2017.
[27] A. Jasra, S. S. Singh, J. S. Martin, and E. McCoy, "Filtering via approximate Bayesian computation," *Statist. Comput.*, vol. 22, no. 6, pp. 1223–1237, Nov. 2012.
[28] K. Fan, C. Li, and K. Heller, "A unifying variational inference framework for hierarchical graph-coupled HMM with an application to influenza infection," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 3828–3834.
[29] D. M. Blei, M. I. Jordan, and J. W. Paisley, "Variational Bayesian inference with stochastic search," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 1367–1374.
[30] M. D. Hoffman and D. M. Blei, "Structured stochastic variational inference," in *Proc. Artif. Intell. Statist.*, vol. 38, pp. 361–369, 2015.
[31] M. Yin and M. Zhou, "Semi-implicit variational inference," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 5660–5669.
[32] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 1303–1347, 2013.
[33] V. Smidl and A. Quinn, "Variational Bayesian filtering," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 5020–5030, Oct. 2008.
[34] J. Winn and C. M. Bishop, "Variational message passing," *J. Mach. Learn. Res.*, vol. 6, pp. 661–694, Dec. 2005.
[35] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Proc. Fifteenth Conf. Uncertainty Artif. Intell.*, 1999, pp. 467–475.
[36] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Generalized belief propagation," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2001, pp. 689–695.
[37] D. Blatner, *Spectrums: Our Mind-Boggling Universe From Infinitesimal to Infinity.* London, U.K.: A&C Black, 2013.
[38] A. H. Jazwinski, *Stochastic Processes and Filtering Theory.* North Chelmsford, MA, USA: Courier Corp., 2007.
[39] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. Amer. Stat. Assoc.*, vol. 112, no. 518, pp. 859–877, 2017.
[40] J. Hensman, M. Rattray, and N. D. Lawrence, "Fast variational inference in the conjugate exponential family Fast variational inference in the conjugate exponential family," in *Advances in Neural Information Processing Systems* 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., Red Hook, NY, USA: Curran Assoc., 2012, pp. 2888–2896.
[41] L. K. Saul and M. I. Jordan, "Exploiting tractable substructures in intractable networks," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 1996, pp. 486–492.
[42] R. Williams and L. C. Baird, "Tight performance bounds on greedy policies based on imperfect value functions," *Tech. Rep.*, 1993.
[43] M. Simon, "The terrifying science behind California's massive camp fire," 2018. Accessed: Jun. 26, 2020. [Online]. Available: https://www.wired.com/story/the-terrifying-science-behind-californias-massive-camp-fire/
[44] R. N. Haksar, F. Solowjow, S. Trimpe, and M. Schwager, "Controlling heterogeneous stochastic growth processes on lattices with limited resources," in *Proc. IEEE 58th Conf. Decis. Control*, 2019, pp. 1315–1322.

Ravi N. Haksar (Student Member, IEEE) received the B.S. degree in mechanical engineering from the Georgia Institute of Technology, Atlanta, GA, USA in 2014 and the M.S. and Ph.D. degrees in mechanical engineering from Stanford University, CA, USA, in 2017 and 2020, respectively.

His research interests include control theory, probabilistic frameworks, decentralized optimization, and cooperative multi-robot teams.

Mac Schwager (Member, IEEE) received the B.S. degree in mechanical engineering from Stanford University, Stanford, CA, USA, in 2000, and the M.S. and Ph.D. degrees in mechanical engineering from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2005 and 2009, respectively.

From 2010 to 2012, He was a Postdoctoral Researcher working jointly with the GRASP Lab, University of Pennsylvania, and CSAIL, MIT, and from 2012 to 2015, he was an Assistant Professor with Boston University, Boston, MA, USA. He is currently an Assistant Professor with the Aeronautics and Astronautics Department, Stanford University. His research interests include distributed algorithms for control, perception, and learning in groups of robots and animals.

Dr. Schwager was the recipient of the NSF CAREER Award in 2014, the DARPA YFA in 2018, and a Google Faculty Research Award in 2018.