Intelligent Knowledge Distribution: Constrained-Action POMDPs for Resource-Aware Multi-Agent Communication

Michael C. Fowler[1], T. Charles Clancy[2], and Ryan K. Williams[3]

Abstract—This paper addresses a fundamental question of multi-agent knowledge distribution: what information should be sent to whom and when, with the limited resources available to each agent? Communication requirements for multi-agent systems can be rather high when an accurate picture of the environment and the state of other agents must be maintained. To reduce the impact of multi-agent coordination on networked systems, e.g., power and bandwidth, this paper introduces two concepts for partially observable Markov decision processes (POMDPs): 1) *action-based constraints* which yield constrained-action POMDPs (CA-POMDPs); and 2) *soft probabilistic constraint satisfaction* for the resulting infinite-horizon controllers. To enable constraint analysis over an infinite horizon, an unconstrained policy is first represented as a Finite State Controller (FSC) and optimized with policy iteration. The FSC representation then allows for a combination of Markov chain Monte Carlo and discrete optimization to improve the probabilistic constraint satisfaction of the controller while minimizing the impact to the value function. Within the CA-POMDP framework we then propose *Intelligent Knowledge Distribution (IKD)* which yields per-agent policies for distributing knowledge between agents subject to interaction constraints. Finally, the CA-POMDP and IKD concepts are validated using an asset tracking problem where multiple unmanned aerial vehicles (UAVs) with heterogeneous sensors collaborate to localize a ground asset to assist in avoiding unseen obstacles in a disaster area. The IKD model was able to maintain asset tracking through multi-agent communications while only violating soft power and bandwidth constraints 3% of the time, while greedy and naive approaches violated constraints more than 60% of the time.

Index Terms—Autonomous Agents, Markov Decision Processes, Multi-agent Collaboration, Probabilistic Constraint Satisfaction, Wireless Communications

I. INTRODUCTION

DECENTRALIZED coordination of autonomous systems in the field requires a delicate balance between system objectives and the burden on limited resources such as power, bandwidth, computation, etc. The resilience of decentralized approaches is clear, allowing agents to act independently with varying levels of information about other agents' states, observations, and actions. However, decentralized systems by their nature must *intelligently* consume resources to remain feasible. In this paper, we propose a framework for *Intelligent Knowledge Distribution* (IKD) which decides on what information is transmitted to whom and when, while constraining the impact coordination has on the limited resources available to each agent.

In disaster response or military operations, remote sensing by autonomous systems provides a stream of data back to critical personnel. Disaster recovery and other active sensing regimes often rely on high bandwidth links that are stretched to their limits under unpredictable channel and link qualities in support of electrooptical, chemical sensors, electronic sensing, etc. [1]. At the same time, the communication necessary for multiple agents to coordinate needs to stay within constrained allowances as to not impact the flow of critical information in the system. Along with typical bandwidth constraints, unmanned aerial vehicles (UAVs) for example have the additional limitation of battery power that needs to be preserved to maximize their flight time in support of field operations.

With the above concerns in mind, in this work we assume each agent is executing its own decision-making in a decentralized network, i.e., our underlying model is a decentralized partially observable Markov decision process (Dec-POMDP) [2], [3], [4]. In the Dec-POMDP setting, each agent has local state observations but only indirect observations of the environment and other agents. Thus, an agent can run independently without the necessity of communicating or having information about other agents' states or rewards, but this will lead to sub-optimal behavior. Indeed, the key problem is to determine from local observations what information neighboring agents will need and when they will need it, improving the overall effectiveness of a coordination objective. We propose methods to achieve these goals while respecting the resource constraints of fielded systems, eliminating the common assumption that communications are instantaneous and free [5]. In particular, we aim to abstract communication decisions away from other joint agent policies (e.g., for motion, tasking, etc.) to construct an agnostic "plug-and-play" capability to mitigate combinatorial explosion and provide an approach that may utilize information from other models similar to Concurrent MDPs [6].

In this paper, the Constrained-Action POMDP (CA-POMDP) proposed for *Intelligent Knowledge Distribution* (IKD) yields per-agent policies for distributing knowledge with other agents based upon the *value of the information* subject to interaction constraints, with a case study in disaster recovery with a heterogeneous multi-robot team. An optimal POMDP policy is solved through policy iteration and then a combination of Markov chain Monte Carlo (MCMC) and

[1]Michael C. Fowler is a research scientist with the Hume Center for National Security and Technology, and a Ph.D. Candidate in Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24060, USA, mifowler@vt.edu

[2]T. Charles Clancy is with the Department of Electrical Engineering, Virginia Tech, Blacksburg, VA 24060, USA, tcc@vt.edu

[3]Ryan K. Williams is with the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24060, USA, rywilli1@vt.edu

discrete optimization are applied to alter the controller to improve the probability it will not violate resources thresholds over a certain horizon. Soft constraints are relevant in communication decisions compared to hard constraints for multiple reasons. Infinite-horizon POMDPs are often represented as a finite state controller [7] and evaluating hard constraints on cyclic graphs is intractable due to the stochastic nature of the environment [7], [8]. In addition, resource availability is often not a hard constraint over short periods of time, as a system may not utilize maximum instantaneous availability or temporary resource over-commitment is permissible (e.g., using queuing protocols such as ZeroMQ). In [9], preliminary work in the application of CA-POMDPs to solving IKD was presented and this work expands on it by: 1) Performing a Monte Carlo simulation comparison between intelligent, naive, and greedy communication strategies; 2) Utilizing Kalman filters as a state estimator to drive information relevance and observation models for IKD; 3) A discrete optimization algorithm that improves the edge observation transitions of the CA-POMDP; and, 4) Adapting the CA-POMDP finite state controller policy with Bayesian predictive estimation in response to realized observation probabilities and resource utilization.

To evaluate the effectiveness of our solution, we require a model that provides a means to gauge relevance of information for communication decisions in our disaster recovery case study. Kalman filters provide a well established method for state estimation of a ground asset and a means to determine the usefulness of information and the accuracy of our disaster response use case for CA-POMDPs and IKD. As a result of IKD, the drones monitoring the disaster site were able to satisfy mission objectives, which involved having the ground asset avoid environmental hazards while searching for survivors, while also drastically reducing the consumption of available resources. In comparison to naive and greedy communication models, the IKD communication model was able to maintain accurate state estimations of the ground vehicle while utilizing significantly less power and bandwidth.

The remainder of this paper is organized as follows. Section II provides a perspective of our concept in relation to the state of the art. In Section III, review the fundamental models at the core of our architecture and evaluation case study. Our approach for Constrained-Action POMDPs is formulated in Section IV, which is the basis for providing Intelligent Knowledge Distribution (IKD) in Section V. Section VI describes how the CA-POMDP was validated to perform IKD, with the results reported in Section VII. The paper wraps up with future and continuing research in Section VIII.

II. RELATED WORK

Capitan's paper [5] is a key comparison in analyzing the performance of our paper in a multi-agent coordination setting. However, [5] assumes that point-to-point communications are instantaneous and cost-free between nodes, while applying a collaboration policy similar to consensus. In this work, we instead remove the assumption of instantaneous and cost-free communication to yield multi-agent communication that respects resource constraints. This results in knowledge distribution that is more nuanced than consensus, i.e., data flooding vs. putting data where it needs to be.

Regarding constraints in decision-making, constrained Markov Decision Processes (MDPs) have been used historically to solve two major drawbacks of standard discrete MDPs: Multiple objectives and limited resources [10], [11], [12], [13], [14], [15], [16], [17]. Applying constraints to MDPs has been well established in restricting utilization of states to prevent collisions [12], and has been expanded to hierarchical cases to reduce the complexity of the linear programming formulation [13], [14]. The primary objective of the Constrained MDP (CMDP) is to find a policy that is within a restricted state, cost, and/or reward structure of an unconstrained MDP. Constrained POMDPs (CPOMDPs), which provide constraints for partially observable environments, have also been explored using mixed integer linear programming (MILP) [16], [17]. Both [16] and [17] solve the POMDP using value iteration, though [17] also describes a point-based value iteration (PBVI) approach to provide an upper bound to a heuristic search. An online algorithm has also been proposed [17] to address drawbacks related to the propagation of risk aversion. A finite horizon policy is computed off-line with a "constraint-penalty-to-go" for every step of dynamic programming value iteration as defined in [16] for solving the MILP.

The above approaches to CMDPs and CPOMDPs apply constraints to the state-space of the model and projection into the value space, but our formulation requires *action-based* constraints as we are limiting the utilization of resources that an action consumes which cannot be tied to physical constructs, such as states that represent "no-fly" or "stay-away" zones. As many scenarios require an indefinite length of operation and no predefined goal states (as in our case study), our CA-POMDP approaches needs to be solved as an infinite-horizon policy. Infinite horizon POMDPs are solved by policy iteration with a finite state controller representation [18], and thus in our context will require analyzing how a cyclic graph utilizes resources with respect to soft constraints. To the best of the authors' knowledge, it is not possible to represent soft resource constraints on actions of a cyclic controller in the state or value space of state-of-the-art CMDPs and CPOMDPs.

The modeling of multi-agent systems (MASs) that have common objectives can be represented as a Decentralized Markov decision process (Dec-POMDP). A Dec-POMDP is a construct that allows multiple independent POMDPs running on different agents to act independently while working towards an objective function that is dependent on all the agents' actions [3]. In brief, the agents only have access to their local observations and as the objective function is dependent on the behavior of all the agents, each agent must maintain a *belief* of the other agents policies. There are many subclasses of the Dec-POMDP that address joint and local observations, communications, model independence, etc [3], [19]. The Dec-POMDP-Comm is the dominant subclass related to this paper where the agents are reasoning on when to communicate due to delays and costs. The most commonly seen approach is to make the decision a part of Dec-POMDP model based upon joint state and actions [3]. There is also an approach where

there is a centralized POMDP plan that is communicated between the agents which triggers communications when their individual Dec-POMDP plan differs from the centralized one. Unlike Dec-POMDP-Comm, our CA-POMDP formulation does not utilize communications to inform the joint policy of multiple agents driving communications, instead the model is an *agnostic "plug-and-play"* approach that separates other collaborative tasks of an agent from communication decisions while enforcing soft resource constraint satisfaction.

Driving communications from the *value of information* similar to CA-POMDP is not unique, but has been the focus of research behind reward shaping and belief-dependent rewards [20], [21]. These techniques use information-theoretic measures between probability distributions, like KL-Distance, as part of the reward function for belief-dependent rewards [21] and for determining communication [20]. [20] focuses on restricting communications to when they are needed but does not provide soft constraint satisfaction to the policy controller which is the focus of this work. The authors in [20] use the Dec-POMDP-Comm as their basis but change the perspective from a cost to a reward for communication to highlight an opportunity to use a resource. They state efficient policy generation techniques will be adapted to allow for scalability, where they use the information-theoretic concepts from Dec-POMDP-Value-Comm [22] as a measure of belief divergence. Alternatively, [23] purposely restructures the action space so they can remain in a classic POMDP problem. In this paper, we approach the IKD problem similarly to [23] where information rewards drive cooperative perception, but instead of restructuring the action space we restructure the observation model to describe belief on the value of information.

III. BACKGROUND

In this section, we review the principles behind our formulation and provide a foundation of concepts, terminology, and lexicon used in the paper. Afterwards, we explain how these concepts are integrated into a single framework to provide a Constrained-Action POMDP (CA-POMDP) leading to an *Intelligent Knowledge Distribution* model.

A. Partially Observable Markov Decision Processes

Markov decision processes (MDPs) are a decision-theoretic approach to determining the best actions over a time horizon based upon the current state s of a system, the probability of transition $T(s'|s, a)$ to another state s' based on action a, and the reward $R(s, a)$ associated taking actions in a given state [7], [8]. This model is used to determine actions that maximize rewards, i.e., a policy π, when the rewards are delayed or realized at some future point in time. If the state of the agent is not fully observable then the model becomes a Partially Observable MDP (POMDP) where belief states, $b \in \boldsymbol{B}$, are used to represent the probability distribution of being in any particular state based upon observations [19]. These belief states are tied to an observation model, $\boldsymbol{O}$, that describes the probability of observations based on the underlying state. Formulating a POMDP in this manner is often referred to as a *belief-state MDP*. A POMDP can be described formally by
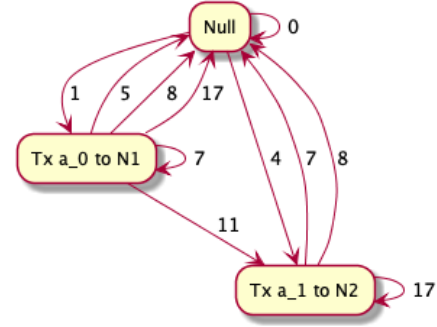

Fig. 1: A graphical representation of a Finite-state Controller for a POMDP policy where the vertices $v \in \mathcal{V}$ represent actions and the edges $e \in \mathcal{E}$ are transitions based upon environmental observation $o \in \boldsymbol{O}$. The example has three actions: (1) transmit message a_0 to Node 1, transmit message a_1 to Node 2, or no transmission $Null$.

the vector $< \boldsymbol{S}, \boldsymbol{O}, \boldsymbol{B}, \boldsymbol{A}, \boldsymbol{T}, \boldsymbol{R}, \boldsymbol{\Pi} >$, where $\boldsymbol{S}$ is the set of states an agent can be in, $\boldsymbol{A}$ is the set of actions the agent can take, $\boldsymbol{T}$ is the set of transition probabilities $T(s'|s, a)$ between states based on an action $a \in \boldsymbol{A}$, $\boldsymbol{R}$ is the set of rewards $R(s, a)$ for taking an action in a state, and $\boldsymbol{\Pi}$ is the set of policies, π, that are feasible consisting of a set of vectors of actions, $\{\pi_0, \pi_1, \ldots, \pi_t\}$. The objective is to select a policy, π, that maximizes the expected utility (reward) over time: $\pi^*(s) = \arg\max_\pi U^\pi(s)$ given a utility function $U^\pi(s)$. Since we do not know the state we are currently in but maintain a belief b, an immediate one-step policy would select an action that maximizes the expected reward $\max_a \sum_s b(s)R(s, a)$. If we let let α_a represent $R(\cdot, a)$ as a vector, often referred to as the *alpha vector*, and the current belief state as a vector $\boldsymbol{b}$, then an immediate one-step policy would become $\max_a \alpha_a^T \boldsymbol{b}$. An alpha vector thus represents a hyperplane in the belief space that is piecewise linear and convex.

In order to represent infinite-horizon POMDP policies, we assume the typical policy representation of a finite state controller (FSC). In Figure 1 we show a simple policy, π, represented as an FSC, which is a directed cyclic graph with the vertices $v \in \mathcal{V}$ representing *machine states* consisting of an action $a \in \boldsymbol{A}$ per vertex and edges $e \in \mathcal{E}$ representing transitions from an action of machine state a_i to an action of machine state a_j determined by the observation $o \in \boldsymbol{O}$ seen after executing that action a_i. An action a_i of machine state i can be the same action a_j of another machine state j such that $\exists a_i = a_j : i, j \in \mathcal{V}(\pi)$ though they will consist of unique alpha vectors, $\alpha_i \neq \alpha_j$.

The policy iteration (PI) algorithm of Hansen in [18] is the basis for improving a FSC through transformations which searches within the policy space until epsilon convergence is observed [18]. In particular, a dynamic programming update is iteratively used to generate a new set of candidate machine states that the policy improvement loop uses to compare the alpha vectors of existing value function V and the new value function V' and modifies the FSC accordingly. The set of alpha vectors can be run through a linear programming (LP) formulation to prune the set of alpha vectors that are dominated in the belief space by any combination of other alpha vectors, which is explained in [18]. We have chosen

Hansen's algorithm as the starting point for CA-POMDP for its basic implementation to validate the concept and the availability of point-based policy iteration methods [24] for Hansen's approach that reduce the computational complexity.

B. Markov Chain Monte Carlo

To analyze the probabilistic constraint satisfaction of an FSC policy from an *unconstrained* POMDP, the Markov chain Monte Carlo (MCMC) sampling methodology is applied. A MCMC is a particle-based approximate inference sampling technique that uses a sequence of samples to iteratively approach a desired posterior distribution [25]. For CA-POMDPs, the MCMC provides a mechanism to continuously sample an FSC policy seeking a posterior that represents its resource utilization. When samples are initially likely to be from the prior distribution, the sampling of a sequence of samples will successively approach the posterior, whereas likelihood based approaches are unlikely to account for this fact and place undue weighting on earlier samples. Another gain in utilizing MCMCs in determining the posterior distributions is the versatility in the type of distribution that can be inferred without being bound to Gaussian distributions or Gaussian approximations.

The Metropolis-Hastings algorithm to solving MCMC was selected because of the difficulty of direct sampling from the probability distribution of a cyclic controller and calculating the normalizing factor of the distribution. The algorithm uses a random walk approach that either accepts or denies a proposal rather than trying to track importance weights. The acceptance ratio reduces simply to

$$\mathcal{A} = \frac{P(x|\mu)P(\mu)}{P(x|\mu_0)P(\mu_0)} \tag{1}$$

of the proposed posterior distribution, $P(x|\mu)P(\mu)$, over the current posterior distribution, $P(x|\mu_0)P(\mu_0)$. The most difficult part of calculating the posterior with the Bayes formula, the evidence $P(x)$, is common to both the proposed and current posterior and therefore conveniently cancels out.

If a random number from $[0, 1]$ is lower than the acceptance ratio (1), then the proposed posterior is accepted. In cases where the proposed distribution is larger than the current distribution, then the acceptance ratio will be greater than one $\mathcal{A} \geq 1$ and therefore the proposed distribution will always be accepted. On the contrary when the acceptance ratio is less than one $\mathcal{A} < 1$, then there is a uniform probability that it will accept the proposed distribution.

C. Graph Theory

Finally, we require a model for interactions in a decentralized multi-robot system. Consider a multi-robot system composed of n robots with indices $\mathcal{I} = \{1, \ldots, n\}$, operating in $\mathbb{R}^d$, each having position $\boldsymbol{x}_i \in \mathbb{R}^d$. As an example in determining collaboration between agents, it can be assumed that the robots can intercommunicate in a proximity-limited way, inducing interactions (topology) of a time varying nature. Specifically, letting $d_{ij} \triangleq \|x_{ij}\| \triangleq \|x_i - x_j\|$ denote the distance between robots i and j, and (i, j) a link between

connected robots, the neighborhood $\mathcal{N}_i$ of each robot i is defined by the interaction radius $\rho(x_i)$, which has a dependence on the location of robot i. Note that such an interaction radius encodes typical sensing and communication constraints that vary spatially. The assumed spatial interaction model is formalized by the *directed graph*, $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ with nodes indexed by the robots, and edges $\mathcal{E} \subseteq \mathcal{I} \times \mathcal{I}$ such that $(i, j) \in \mathcal{E} \iff \|x_{ij}\| \leq \rho(x_i)$ [26]. The interaction graph $\mathcal{G}$ of a decentralized multi-robot system describes the neighbors $\mathcal{N}_i$ that an agent i transmits information to in IKD-based collaboration.

IV. CONSTRAINED-ACTION POMDP

The *Constrained-Action POMDP* is a formulation that seeks to find a near optimal policy in a partially observable environment with action-based constraints that are probabilistically guaranteed to stay within specified *soft* limits. The framework first solves for an unconstrained optimal policy then improves the probabilistic constraint satisfaction of the controller. Any action performed by an autonomous system will have a distribution representing the utilization of resources, and thus we aim to determine the probability that a series of actions will respect a soft constraint through an analysis of the cumulative distribution function (CDF) of policy resource utilization. In the formulation of the CA-POMDP, the analysis of probabilistic constraint satisfaction is performed by sampling a policy with Markov chain Monte Carlo (Section IV-C) and improving the constrained policy through discrete optimization (Section IV-B).

A. CA-POMDP Model

Our goal is to adapt an infinite-horizon FSC of a POMDP policy π from policy iteration [18] to probabilistically stay within soft constraints over a desired period of time T, i.e., $p(\sum_{t=0}^{T} u_h(a_t) \leq c_h) \geq \eta_h$, for all constraints $h \in \mathcal{H}$, with resource limits $c_h \in \boldsymbol{C}$, where $u_h \in \boldsymbol{U}$ is resource utilization used by the action a_t. Policies π are represented as a finite-state controller (FSC), since an FSC is a cyclic graph that represents an infinite-horizon POMDP well [27]. The FSC representation can also accelerate the convergence to a policy, since an infinite horizon policy in value iteration is not guaranteed to converge [7]. The CA-POMDP can be described formally by the tuple

$$< \boldsymbol{S}, \boldsymbol{O}, \boldsymbol{B}, \boldsymbol{A}, \boldsymbol{T}, \boldsymbol{R}, \boldsymbol{\Pi}, \mathcal{H}, \boldsymbol{C}, \boldsymbol{U}, \boldsymbol{E} > \tag{2}$$

where $\boldsymbol{S}$, $\boldsymbol{O}$, $\boldsymbol{B}$, $\boldsymbol{A}$, $\boldsymbol{T}$, $\boldsymbol{R}$, and $\boldsymbol{\Pi}$ are the same as defined in a POMDP, $\mathcal{H}$ is the set of resources being used by the agents, $\boldsymbol{C}$ is the matrix defining the constraints for each resource, $\boldsymbol{U}$ defines the utilization of resources for an action $a \in \boldsymbol{A}$ and its variation or uncertainty, and $\boldsymbol{E}$ is the matrix of edge observation probabilities for transitioning from a machine state i to another j in an FSC based upon action observation histories (AOH).

The constraint model consists of a matrix of soft constraints $\boldsymbol{C}$ (e.g., power and bandwidth), the utilization of resources $\boldsymbol{U}$ per time epoch Δt (a decision is made each epoch of

```
 1: procedure CONSTRAINTIMP(C, U, E, π₀)
 2:     π*, V*, Ω ← POMDP POLICY ITERATION(π₀)
 3:     N₀ ← CONSTRAINTEVAL(π*, C, U, E)
 4:     if N₀ ≥ η then
 5:         return π*, N₀
 6:     end if
 7:     π̂₀ ← π*;   t ← 0
 8:     repeat
 9:         t ← t + 1
10:         Π' ← ALLFEASIBLEREGIONS(π̂ₜ₋₁, Ω)
11:         πₘᵢₙ ← minπ π ∈ Π'
12:         g(Π') ← ‖ V* − POLICYEVAL(πₘᵢₙ) ‖
13:         Q ← Π'
14:         repeat
15:             Π' ← arg minq∈Q g(q)
16:             πₘₐₓ ← maxπ π ∈ Π'
17:             f(Π') ← ‖ V* − POLICYEVAL(πₘₐₓ) ‖
18:             N ← CONSTRAINTEVAL(πₘₐₓ, C, U, E)
19:             R ← BRANCHCONTROLLER(π', U)
20:             for R ∈ R do
21:                 πₘᵢₙ ← minπ π ∈ R
22:                 g(R) ← ‖ V* − POLICYEVAL(πₘᵢₙ) ‖
23:             end for
24:             Q ← Q ∪ R
25:             Q.Pop(Π')
26:             PRUNE(Q)
27:         until Q = ∅
28:         π̂ᵢ ← arg maxπ∈Q g(π)
29:     until π̂ᵢ = π̂ᵢ₋₁
30:     return π̂ᵢ, N(π̂ᵢ)
31: end procedure
```

Fig. 2: Algorithm for performing constraint improvement of an optimal finite state controller.

Fig. 3: An example of the feasible regions for a finite state controller for use in the Branch and Bound Discrete Optimization Algorithm.

time), and the matrix of observation probabilities E. Specifically, $C \in \Re^{|\mathcal{H}| \times 2}$ is a matrix representing the desired soft constraints within a specified time period T where each row $h \in \mathcal{H}$ of C is the vector $[c_h, \eta_h]$, with c_h the desired upper limit for resource h, and $0 < \eta_h < 1$ is the probabilistic constraint satisfaction for that resource based upon operational scenarios. The resource utilization U defines for each resource $h \in \mathcal{H}$ a matrix of Gaussian probability distribution parameters for all actions, i.e., $u_h(a) \leftarrow \mathcal{N}(\mu_h^a, \sigma_h^a)$. The matrix $E \in \Re^{|\pi| \times |O|}$ consists of edge observation probabilities $p(o|a_{i \in \pi})$ for encountering an observation $o \in O$ while at a machine state i of the finite state controller policy $\pi \in \Pi$. Edge observation probabilities are tied to the FSC policy representation as a means to track the stochastic nature of the environment causing observation transitions from one machine state to another. Their probabilistic distribution is initially assumed to be uniform and the true distribution is learned online, see Section IV-D.

Our core mechanism in CA-POMDP is the FSC representing the optimal policy calculated using Hansen's POMDP PI algorithm [18], along with discrete optimization which acts to constrain the policy. During the discrete optimization phase, CA-POMDP will introduce *constraint states* j into an unconstrained controller to bring the controller within a probabilistic constraint satisfaction η (analyzed via MCMC) while minimizing the impact on the optimal unconstrained value function V^*. To effectively alter the controller with constraint states, the non-dominant alpha vectors per action, defined as set Ω, are maintained during the dynamic programming updates for unconstrained policy optimization. These alpha vectors are not
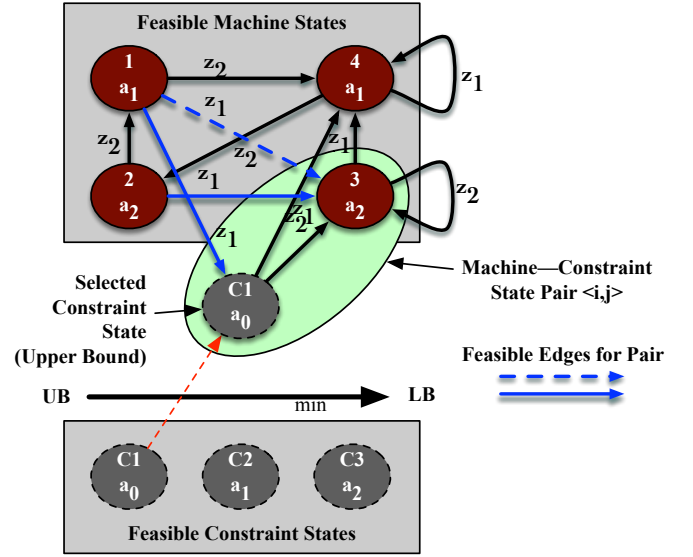
dominant at any dynamic programming update, but may utilize fewer resources and therefore should be considered for altering the controller for probabilistic constraint satisfaction.

B. Constraint Improvement

Improving the constraint satisfaction of an ϵ-optimal unconstrained policy is solved using a branch and bound (BnB) algorithm variant of [28], a well-known technique for discrete and combinatorial optimization. The underlying goal of the BnB is to determine a set of constraint states, our previously stored set of alpha vectors Ω, to "inject" into the optimal controller which utilize fewer resources than the existing machine states in the FSC while trying to minimize the loss in the expected value from ϵ-optimal.

Remark 1. *By calculating the optimal policy first, we also have an initial optimal policy to recalculate from as resource utilization and observation edge probabilities are learned online (Section IV-D).*

The algorithm (Figure 2) initializes with the ϵ-optimal FSC policy π^* calculated from policy iteration which is immediately checked for probabilistic constraint satisfaction (Line 3 of Figure 2). If the policy is non-compliant, the root of a BnB Tree Q (Line 10) is initialized with the *feasibility space* of modifications that can be made to the unconstrained policy π^*, i.e., an ordered combinatorial set of all of machine states, constraint states, and edge redirection variables, see Figure 3.

The first variable in the feasibility region is the set of existing machine states i in the policy $\hat{\pi}_{t-1}$, shown in the upper gray box of Figure 3, which is initially π^*. Each machine state's dominant region $V_i b_i$, where b_i is the region in the belief space b that machine state i has dominance over all other machine states, is used in creating an ordered set of machine – constraint state pairs, marked in Figure 3.

The set of alpha vectors Ω saved from the policy iteration algorithm define the second set of variables in the feasibility

region. Each alpha vector becomes a potential constraint state j, shown in the lower gray box of Figure 3, that can be introduced into the controller and will result in a loss of value depending on the value of the new alpha vector and chosen edge redirection (the third feasibility region variable). A feasible constraint state j is an alpha vector whose action a_j utilizes less resources $U(a_j) < U(a_i)$ than machine state i. The upper $f(\pi)$ and lower bound $g(\pi)$ of a constraint state and machine state pair $<i, j>$ provides the BnB maximum (UB) and minimum (LB) value difference between the constraint state's alpha vector $V_j b_i$ and the machine state's $V_i b_i$ for the machine state's dominant region b_i: $\|V_i b_i - V_j b_i\|$.

The third and final variable in the feasibility region is the set of viable edge redirection probabilities. During the branch and bound, the feasible constraint states for a particular machine state pair $<i, j>$ are injected into the controller and the edges leading to this new constraint state from other machine states already in the controller are selected as either maintaining their transition to the existing machine state i or redirecting to the new constraint state j. The edge redirections are solved for last and until then a simple assumption is made for the calculation of upper and lower bounds for machine–constraint state pairs. The lower bound function assumes that none of the edges are redirected which results in a controller that has no value impact. The upper bound assumes all edges are redirected to the constraint state, which causes the greatest impact on the value function as the original machine state that dominated has been completely substituted or replaced by the constraint state that maximizes $\|V_i b_i - V_j b_i\|$.

The blue lines in Figure 3 indicate feasible edges that could be redirected or maintained for the $<3, C1>$ machine – constraint pair. The constraint state inherits the outgoing edges of the original machine state i since the outgoing edges already transition to dominant machine states after a belief update, and any constraint state introduced into the controller has already been determined to be non-dominant. Optimally redirecting an edge from another machine state to the new constraint state is not trivial due to the combinatorial explosion of redirection options, which drastically reduces computational efficiency. Instead, we allow the BnB to select from a finite set of the probabilities of redirecting an edge from an original machine state to the new constraint state, i.e., a vector of ordered probabilities $\overrightarrow{P} = [p_0, p_1, \ldots, p_n]$ where $p_0 < p_1 < \cdots < p_n$ and $0 < p_i < 1$. The BnB algorithm searches for the correct probability $p_i \in \overrightarrow{P}$ of edge redirection, yielding a solution that satisfies probabilistic constraints while minimizing the impact to the value of the constrained controller. As the edge redirection probability increases, more edges from an existing machine states k in the controller to the machine state i will be redirected to the constraint state j.

Formally, the objective of the Branch and Bound algorithm of Figure 2, is to minimize the impact on the constrained controller value $V(\hat{\pi})$ compared to the optimal controller V^* while ensuring that the probabilistic constraint satisfaction $N(\hat{\pi})$ is greater than or equal to the soft constraints $\eta \in \boldsymbol{C}$.
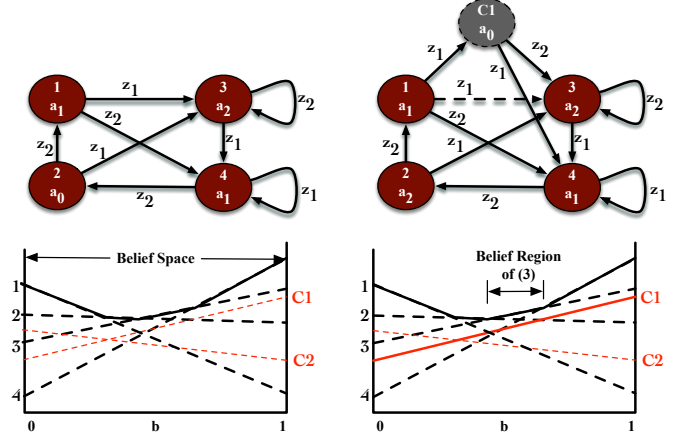

Fig. 4: The constraint $C1$ has been injected into the controller to reduce the resource utilization of machine state 3. The edge redirection function has selected the $1 \to 3$ to be reconnected as $1 \to C1$ and $C1$ inherits the outgoing edges of 3. Notice the loop for observation z_2 at 3 becomes a transition of $C1 \to 3$

That is,

$$\begin{aligned} Objective: \quad & \min_{\hat{\pi}} \|V^* - V(\hat{\pi})\| \\ s.t.: \quad & N(\hat{\pi}) \geq \eta \quad \eta \in \boldsymbol{C}. \end{aligned} \tag{3}$$

During each step of the algorithm, a set of feasible solutions Π' is removed from the tree Q based on a priority metric (best-first search) provided by the lower bound function $g(\Pi')$ (Line 12). The branching function (line 18) simply divides the region of feasible solutions in half, similar to constrained integer programming [29], creating two new nodes Π' in the tree Q until there is no region to divide creating a "leaf" node. The probabilistic constraint satisfaction N of the upper bound policy $\pi_{\max}$ of a node $\Pi' \in Q$ is evaluated via MCMC (Line 17), described in the next section, since we are interested in the best satisfaction the worst controller can provide. The pruning function (Line 19) removes any node in the tree where: 1) the lower bound of that node is greater than the upper bound of any other $g(\Pi'_i) < f(\Pi'_j) \quad \forall (i, j) \in Q_{i \neq j}$; or, 2) the upper bound does not meet the constraint satisfaction $N_i < \eta \quad \forall i \in Q, \eta \in \boldsymbol{C}$.

There are two loops in the constraint improvement algorithm with the inner loop terminating when there are no further nodes to consider in the tree Q (Line 20). The outer loop (Line 22) introduces a single constraint state during each iteration until the desired probabilistic constraint satisfaction has been achieved. Afterwards, a constraint state can be introduced into the finite state machine with an objective to increase the value function without violating the constraint. Once there are no changes to the controller to improve constraint satisfaction or value function, the function returns the new controller.

Figure 4 provides an example of a constraint improvement step with the original optimal FSC shown on the left with machine states 1 through 4. Notice that the machine states 1 and 4 have the same action, a_1, which is an indicator that the same action during policy iteration was dominant in two different belief regions. During the constraint improvement, the branch and bound is analyzing the possibility of a constraint state $C1$ concurrent with machine state 3 utilizing

action a_0, which utilizes fewer resources than action a_2. $C1$ was initially selected because $C1$ has the greatest value in the same belief region as 3, since in a best-first search it minimizes $\|V_3 b_3 - V_{C1} b_3\|$ and can be seen in the red line of the right graph of Figure 4. If $C1$ fails to satisfy the constraints, then the algorithm will examine using $C2$ assuming $C2$ utilizes fewer resources then $C1$.

The edge redirection function has selected to redirect the edge $(1, 3)$ to constraint state $C1$ and $C1$ inherits the outgoing edges from machine state 3. It is important to note that the loop at machine state 3 for observation z_2 in the original controller does not create a loop in $C1$ for z_2 but is redirected to the original machine state since 3. This occurs since machine state 3 is the dominant machine state for the belief update of z_2, thus the BnB procedure avoids a non-optimal loop in the constrained controller.

C. Constraint Satisfaction Evaluation

Evaluating constraint satisfaction, shown in Figure 5, is accomplished by random sampling of the finite state controller to estimate the resource utilization through a Markov chain Monte Carlo (MCMC) Metropolis-Hastings algorithm. Using the length of time or epochs over which the constraints are being applied T, the finite state controller is sampled $T/\Delta t$ times, where an action is taken every Δt. Starting from a random initial machine state, this sampling process follows the FSC edge transitions around the controller and records the resource utilization per action by sampling from distributions $\mathcal{N}(\mu_h^a, \sigma_h^a)$, for each resource $h \in \mathcal{H}$. The samples are combined into an aggregate sample x that represents the total mean and deviation of resource utilization for the entire time period T. This is repeated from the same initial machine state to gather a sequence of samples $\boldsymbol{x}(\alpha)$ until the chain has approached a representative posterior $P^T(\alpha)$ of the FSC resource utilization for that particular initial machine state, where α represents the distribution of sample data. The posterior has been considered "mixed" when the variational distance is within epsilon [25] (Line 13)

$$\mathbf{D}(P^{(T)}; \boldsymbol{x}(\alpha)) = \max_\alpha |P^{(T)}(\alpha) - \boldsymbol{x}(\alpha)| \leq \epsilon_D. \quad (4)$$

To determine when the algorithm needs to stop taking samples from random initial machine states, the algorithm continues until the autocovariance of the lagged MCMC distributions, a generalization of the central limit theorem to Markov chains, has converged [25] to within ϵ_{cov}

$$Cov\left[\boldsymbol{P}^{(T)}[m]; \boldsymbol{P}^{(T)}[m+l]\right] \approx \quad (5)$$

$$\frac{1}{M-l} \sum_{m=1}^{M-l} (\boldsymbol{P}^{(T)}[m] - \hat{\boldsymbol{E}}(m))(\boldsymbol{P}^{(T)}[m+l] - \hat{\boldsymbol{E}}(m+l)),$$

where $\mathbf{M}$ is the set of samples collected, l the number of samples to lag, $\boldsymbol{P}^{(T)}$ is the vector of $P^{(T)}(\alpha)$, and $\hat{\mathbf{E}}(\cdot)$ is the unbiased estimator $\frac{1}{M}\sum_{m=1}^M P^{(T)}(m)$.

Once the MCMC has terminated, the resource utilization posterior distribution is used to determine the probability that the current FSC satisfies the soft resource constraints η.

```
1: procedure CONSTRAINTEVAL(π*, U, A, C, B, l)
2:     P^(T) ← ∅
3:     repeat
4:         b_0 ← RANDOM(b ∈ B)
5:         i* ← arg max_{i∈π*} V_i · b_0
6:         x ← ∅
7:         while True do
8:             for t ← 1 to T step Δt do
9:                 x ← x ∪ U(a_{i*} ∈ A)
10:                i* ← RANDOMEDGE(p(o|i*), P^(T)(α))
11:            end for
12:            x(α) ← GAUSSIANFIT( x )
13:            if D(P^(T); x(α)) ≤ ε_D then          ▷ Equation (4)
14:                break
15:            end if
16:            if RANDOM(0, 1) ≤ A then              ▷ Equation (1)
17:                P^(T)(α) ← x(α)
18:            end if
19:        end while
20:        P^(T) ← P^(T) ∪ P^(T)(α)
21:    until Cov[P^(T)[m]; P^(T)[m + l]] < ε_cov     ▷ Equation (5)
22:    return ∫_{-∞}^{η∈C} P^(T) dx
23: end procedure
```

Fig. 5: Algorithm for Evaluating a Finite State Controller for its Probabilistic Constraint Satisfaction.

D. Learning Edge Probabilities and Resource Utilization

When sampling the controller during the evaluation of constraint satisfaction, an observation edge o is randomly chosen during each of the samples with a probability $p(o|a_i)$ depending on the action a_i of machine state i. This Monte Carlo sampling of observation edges through the controller is initially a uniform *a priori* distribution, but the actual conditional probability of an observation given an action during operation needs to be learned so that the controller can be updated to ensure proper constraint satisfaction as the environment changes. Observations o given an action a are described by a categorical distribution (aka, a generalized Bernoulli or multinoulli distribution), which is used to represent the likelihood of some finite set of possible observations. The conjugate prior of a multinoulli distribution is a Direchlet distribution, which is also a Jeffrey's Prior for an N-sided die with biased probabilities, $\overrightarrow{\gamma} = (\gamma_1, \ldots, \gamma_N)$. The closed form solution for calculating the posterior distribution for the categorical distribution and a Direchlet prior is

$$p(o|A) = \frac{c(o, a) + \alpha(o, a)}{\sum_{j=1}^{|c(j)|} c(o, j) + \sum_{j=1}^{K} \alpha(o, j)}, \quad (6)$$

where c is the *a priori* count for an observation o with state action a, α the observed occurrence of observation o, and K the total number of occurrences seen. This method allows a deployed system to track the relationship between actions and observations over time and then recalculate a constrained policy to maintain constraint satisfaction or maximize the value of the controller if it is utilizing fewer resources than expected.

Another consideration in adapting the controller for constraint satisfaction is validating the *a priori* resource utilization models are still applicable during operation. Situations in the field change over time and the *a priori* models used for resources will not be valid during the entire operational life of

the autonomous system. As an example, as the battery of an autonomous system is drained during its operation, an action may consume more battery power than when the battery is fully charged. To track the resource utilization, a simple Bayes estimator with Gaussian priors and likelihood is used to track the current resource utilization per action.

It is unrealistic to continuously update the controller with every edge observation or resource distribution change due to limitations in computational resources and to prevent short-term instabilities or unpredictability in the controller. However, we still need to adapt online to ensure constraint satisfaction or improve the FSC value. To define an appropriate trigger for recomputing a new controller, the information-theoretic concept of Variation of Information is utilized. When the variation of information

$$d(X, Y) = H(X|Y) + H(Y|X) \tag{7}$$

$$H(X|Y) = -\sum_{i,j} p(x_i, y_i) \log \frac{p(x_i, y_i)}{p(y_i)} \ \forall i \in X, j \in Y \tag{8}$$

exceeds a desired threshold, the algorithm will recompute a constrained controller from the precomputed optimal controller, where X is either the *a priori* probability distribution of edges $\boldsymbol{E}$ or resource utilization $\boldsymbol{U}$ and Y is the associated probability distribution of the learned distribution. When the learned distributions are significantly different than those previously used to compute a constrained FSC, we recompute the controller with the new learned probability distributions.

V. INTELLIGENT KNOWLEDGE DISTRIBUTION MODEL

Intelligent Knowledge Distribution (IKD), in this paper, is applying a Constrained-Action POMDP to control communications between multiple independent agents that must stay within quality of service limitations. The goal of IKD is to answer the questions: (i) What information should we send, (ii) when should we send it, and (iii) to whom should we send it to?

A. Action & State Model

The IKD Model builds on the CA-POMDP formulation by extending the CA-POMDP tuple to include $< \boldsymbol{\Sigma}_i, \mathcal{I}, \mathcal{N}_i, \boldsymbol{F} >$, where we assume each agent independently runs CA-POMDP. Each agent i has a set of neighbors $\mathcal{N}_i$ in the environment that can actively collaborate. $\boldsymbol{\Sigma}_i$ is the alphabet of communications $\sigma_k \in \boldsymbol{\Sigma}_i$ that agent i can transmit to a neighbor $j \in \mathcal{N}_i$. $\boldsymbol{F}$ is a set of states $s \in \boldsymbol{S}$ that place mission objectives at risk of failure, which we will use to drive information relevance. We define the set of actions $\boldsymbol{A}_i$ for an agent i that indicate the decision to transmit the kth element of information to agent $j \in \mathcal{N}_i$, denoted as action a_k^j related to the information σ_k:

$$\mathbf{A_i} = \left\{ \cup_{\sigma_k \in \boldsymbol{\Sigma}_i, j \in \mathcal{N}_i}, \quad a_k^j \to \{0, 1\} \right\} \cup \varnothing \tag{9}$$

along with the single $\varnothing$ (or Silence) action to not transmit at all.

The states of the IKD model are defined by levels of relevance S_r of locally observed information for each agent i (e.g., $S_r = \{LOW, MEDIUM, HIGH\}$), and levels of

collaboration S_c for each agent i with its neighbors $j \in \mathcal{N}_i$ (e.g., $S_c = \{LOW, MEDIUM, HIGH\}$). Relevance is described by a set of discrete states that indicate how important information is to global mission objectives, and is calculated per-agent based on local observations only. The confidence of an agent in its current collaboration with its neighbors $j \in \mathcal{N}_i$ is a set of discrete states indicating a level of confidence that the current level of communication will maintain the ability for the agents to achieve global mission objectives by sharing local observations. Formally, the set of states for an agent i in the IKD model is:

$$\mathbf{S_i} = \{s_r^{k=1}, \dots, s_r^{k=|\Sigma_i|}, s_c^{j=1}, \dots, s_c^{j=|\mathcal{N}_i|}\} \tag{10}$$

with relevance states $s_r^k \in S_r$ for each of information element k and a set of collaboration states $s_c^j \in S_c$ with a neighbor $j \in \mathcal{N}_i$ indicating whether the neighbor is "up to date" with relevant information. The states in an agent's state set therefore becomes a representation of how important information is and how timely the agent's information is for neighboring agents.

B. Reward Model

The reward functions for IKD need to be defined specifically by the model designer or learned online; what follows are simply basic guidelines for reward formulation. The reward for relevance $\hat{R}_r(s, a)$ is a normalized reward based upon the product of 1) the likelihood $\mathcal{L}$ that the next state s' is not a critical state $f \in \boldsymbol{F}$ where global mission objectives are at risk when taking action a in state s, which is either known *a priori* or learned online, and 2) the information-theoretic metric on the maximal value of the information $\rho(k)$ an agent could convey to a neighbor:

$$\hat{R}_r(s, a) = \mathcal{L}(s' \notin \boldsymbol{F}|a, s) \max_{\sigma_k \in \Sigma_i} \rho(k). \tag{11}$$

The basic concept is to increase the reward for communications when the mission objectives are at risk and the value of the local information is high.

Remark 2. *The utilization of a reward function with the components ρ and $\mathcal{L}$ by its nomenclature appears similar to belief-dependent rewards [20], however we argue here why this is not the case. The reward functions are not tied to a state belief but instead are calculated for each state of the IKD model and assume a linear function between states, as seen with any belief-state MDP [3], [19]. The reward functions can be learned online and the FSC policy recalculated but they will remain piecewise linear and convex within the hyperplane of belief space and do not change as the belief does.*

Collaboration is a normalized reward $\hat{R}_c(a, s)$ based upon the product of the proximity of a quantized state to approaching the heuristic bound where collaboration will diverge and become unbounded $\mathcal{Q}(\cdot)$, as with intermittent communication in controls with a Kalman filter's estimation error covariance [30], and the maximal value of information that can be shared with a particular neighbor:

$$\hat{R}_c(a, s) = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathcal{S}(\mathcal{Q}(\lambda_j|a, s)) \cdot \max_{\sigma_k \in \Sigma_i} \rho(k), \tag{12}$$

where $\mathcal{S}(\cdot)$ is the sigmoid function, $\mathcal{Q}(\lambda_j|a,s)$ is the probability that the communication action a with neighbor $j \in \mathcal{N}$ will approach the proximity of an unbounded heuristic $\lambda \in \mathbf{\Lambda}$, $\mathbf{\Lambda}$ is the limits of the upper $\bar{\lambda}$ and lower $\underline{\lambda}$ bound heuristic of a critical probability λ_c that is dependent on the model (See Section VI-A), and $\max_{\sigma_k \in \Sigma_i} \rho(k)$ is the best value of information $\rho(k)$ that an agent $j \in \mathcal{N}$ could receive from i. Thus, as the lack of communications drives a global objective towards an unbounded heuristic (e.g., a diverging Kalman filter estimate due to intermittent communications) and the value of information for neighboring agents increases, the reward for communication increases. The final reward $R(s,a)$ is the sum of the collaboration and relevance rewards with one exception. If the action is not to transmit ($a = \varnothing$), the reward is zero for all states:

$$R(s,a) = \begin{cases} \hat{R}_r + \hat{R}_c & if \; a \neq \varnothing \\ 0 & if \; a = \varnothing \end{cases} . \qquad (13)$$

This at first appears counter intuitive, because the action to not transmit will be dominated by all other actions which is intentional. If the agent stops transmitting then it always is going to lead to information loss and therefore provides no benefit not to continuously transmit. It is the constraints to preserve resources that will introduce a do not transmit action or one that utilizes fewer resources into the controller as a viable constraint state to satisfy soft resource constraints.

C. Transition & Observation Models

As with the rewards model, transitions and observations are divided into relevance and collaboration components. Transition probabilities from one state s to another state s' based upon an action a are assumed to be independent to reduce computational complexity and therefore are the product of the independent transitions of the components

$$T(a,s,s') = \prod_{i=1}^{n} \prod_{\sigma_k \in \Sigma_i} T(a, s_r^k, s_r^{k'}) \cdot \prod_{j \in N_i} T(a, s_c^j, s_c^{j'}) \quad (14)$$

where T is the transition matrix, and s_r and s_c are the relevance and collaboration state components. The component transition functions $T(a, s_r, s_r')$ and $T(a, s_c, s_c')$ are determined by the knowledge being shared and are discussed for our use case in Section VI-B.

The observation probabilities are similarly constructed as the transition probabilities with relevance and collaboration components as follows:

$$O(a,s,o) = O(a, s_r, o_r) \cdot \prod_{j \in \mathcal{N}_i} O(a, s_c^j, o_c^j) \qquad (15)$$

where O the observation matrix, and o_r and o_c are the relevance and collaboration components of observations. [1]

Remark 3. *Though the relevance of the information and the collaboration between agents are conditionally dependent in reality, the formulation of the model as a decentralized POMDP reduces the dependence with the focus on local observations allowing to validate the approach before addressing a conditional dependence transition and observation model.*

As discussed in related work, the observation model is key in restructuring the POMDP to avoid belief-dependent rewards and its associated solution methods. In the construction of an IKD model, an observation model should be designed to provide the ability to map observations to the appropriate belief space for relevance and collaboration. Any mapping υ from observations in the environment to an observation state is the mapping of a continuous observation space to discrete observations through cluster-based techniques, such as K-medoids [31] or DBScan. See Section VI-B for how an observation model was constructed to address IKD with asset localization and Kalman Filters.

VI. EXPERIMENTAL MODEL

The use case for validating the Constrained-Action POMDP for *Intelligent Knowledge Distribution* (IKD) was the aerial monitoring of ground assets during a disaster response to ensure they safely avoid hazards and dangerous situations they may not be aware of from their perspective on the ground. Though a ground vehicle could potentially have accurate location information from GPS, we consider the asset tracking viable, because: 1) There are situations where GPS information in a disaster site is either unavailable or degraded due to obstructions or other factors; 2) The actual location of the hazards are not known and the monitoring drones need to maintain accurate relative positioning of the ground asset; and, 3) The ground asset needs to update its motion planning to optimize for known hazards as they are estimated by monitoring drones.

In this context, our IKD agents are unmanned aerial vehicles (UAVs) performing a continuous predefined search pattern around the disaster site while communicating across a first responders mobile ad-hoc network (MANET) as shown in Figure 6. In this initial study to validate the CA-POMDP and IKD concepts, the agents are only tracking a single ground asset and need to determine what sensor information needs to be shared between them to ensure they maintain situational awareness of the hazard risk. They also provide a relative position warning to the ground asset so that it can update the edge costs of its motion planning algorithm appropriately, e.g., D*-Lite [32].

The IKD model, as described in Section V, is a combinatorial problem between the neighboring nodes, the *heterogeneous* sensor data, the relevance of the information, and the current level of collaboration. Since an agent can only communicate with a single node at a time due to routing protocols with WiFi Mesh Networking, it needs to determine the risk it believes an asset is under (relevance) and how confident it is in the data it has received so far from other nodes (collaboration) to ascertain the appropriate information to send and to whom to maintain accurate situational awareness, all while not overly consuming the limited resources of bandwidth in the MANET and power available from the UAV battery.

---

[1] In future work, this assumption will also be relaxed as with the transition assumption of independence.
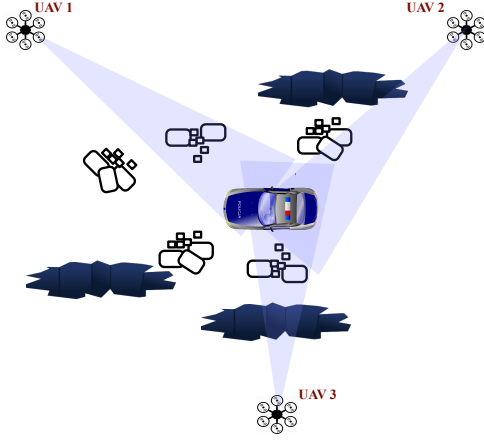
Fig. 6: The conceptual diagram of the situation being modeled with an autonomous emergency vehicle traversing a disaster site with blocked views of potential hazards (e.g. sinkholes)

A. Asset Localization & Kalman Filter

An unscented Kalman filter (UKF) was utilized in performing localization of the ground vehicle from multiple drones, where each drone is providing a bearing and distance to the ground vehicle with variances per the capabilities of the sensor packages that are aboard that particular drone. Kalman filters (KF) are a common approach to guidance, navigation, and control of vehicles and robots [33] as a linear quadratic estimator of the state of a system from a series of measurement or observation samples over time. The Unscented Kalman filter (UKF) uses a deterministic sampling technique known as an unscented transform around a minimal set of sample points about the mean, which performs well in highly non-linear systems as compared to Extended Kalman Filters (EKF) [34]. A nonlinear Kalman filter was necessary due to 1) the non-linearity of the motion model with arcs of motion depending on the angle of the front wheel and 2) the nonlinearity of triangulating the position of the ground vehicle from drones with sensor bearings and distances.

The motion model $f(x, u)$ of the ground vehicle is a standard *bicycle model* with static rear wheel and variable front wheel. The state model $\boldsymbol{x}$ maintains information on the position and orientation of the ground vehicle as the vector $\boldsymbol{x} = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$, where (x, y) is the location of the ground vehicle in the disaster site as measured in meters and θ is the bearing in radians. The state transition function $\bar{x}$ is a non-linear motion model $f(x, u)$ with Gaussian white noise $\mathcal{N}(0, Q)$ defined as

$$\bar{x} = x + f(x, u) + \mathcal{N}(0, Q), \tag{16}$$

where $\boldsymbol{u}$ is the command input $\begin{bmatrix} v & \alpha \end{bmatrix}^T$ to the ground vehicle control system and is defined by the linear velocity v and the steering angle α of the ground vehicle. The measurement model $\boldsymbol{z}$

$$\boldsymbol{z} = h(\boldsymbol{x}, \boldsymbol{P}) + \mathcal{N}(0, R) \tag{17}$$

$$h(\boldsymbol{x}, \boldsymbol{P}) = \begin{bmatrix} \sqrt{(p_x - x)^2 + (p_y - y)^2} \\ \tan^{-1}\left(\frac{y - p_y}{x - p_x}\right) - \theta \end{bmatrix} \tag{18}$$

involves the bearing and distance to the ground vehicle $\boldsymbol{x} = [x, y]$ from the current observation location of drone $[p_x, p_y]$, in which the bearing and range measurement noise

$$\boldsymbol{R} = \begin{bmatrix} \sigma_{range}^2 & 0 \\ 0 & \sigma_{bearing}^2 \end{bmatrix} \tag{19}$$

is assumed to be independent and may or may not have line of sight to the target. In cases where a drone does not have a line of sight of the ground asset nor any observation communications from another drone, this is a missing measurement.

For the UKF, the unscented transform uses a particle-based technique, which requires the ability to calculate the mean of the particles. Calculating the mean of the positions is a simple mathematical average, whereas calculating the mean of the bearing is

$$\bar{\theta} = \text{atan2}\left(\frac{\sum_{i=1}^{n} \sin \theta_i}{n}, \frac{\sum_{i=1}^{n} \cos \theta_i}{n}\right). \tag{20}$$

There are two occurrences that need to be addressed in the application of Kalman Filters because of intermittent communications and limited line of sight: missing measurements and missing observations. In missing measurements, there is no information at a discrete time step to update the state estimate and the predicted measurement is propagated forward as an observation [30], [35]. For missing observations, there are observations at a given time step but not of all the state information. In the case of geolocation through triangulation, a bearing and distance measurement is not available to accurately pinpoint the robot, but there are enough observations to get a less accurate measurement. With UKF, the unscented transform is performed as usual, because the model can be constructed "on-the-fly" to replicate the variance that would be seen with the lack of observation for a fully accurate triangulation by adjusting the measurement noise, R [36].

If the probability of arrival λ of an observation in a Bernoulli process is less than a critical probability $\lambda \leq \lambda_c$ for a Kalman Filter, then the expectation of the estimation error covariance is unbounded [30]. The lower bound $\underline{\lambda}$ of the critical probability λ_c has a closed form solution but the upper bound $\bar{\lambda}$ requires solving a linear matrix inequality (LMI) [30]. Therefore, the authors used a simplified simulation involving fixed observation points to experimentally approximate upper $\bar{\lambda}$ and lower $\underline{\lambda}$ heuristics for $\boldsymbol{\Lambda}$ of the IKD reward function $\mathcal{Q}$ (12) as applied to Kalman filters for use in the full simulation.

B. Model Formulation

Due to size, weight, and power (SWAP) restrictions of the UAVs, they are assumed to have heterogeneous sensors with overlapping sensor support. Each UAV has been assigned to carry two sensors on their platform from a total possibility of three: 1) RF geolocation, 2) optical tracking, and 3) laser range finder. Therefore each node has the option of sending the result of a sensor to any one of the nodes that it's connected to. This creates an action space that as previously described combines the sensors' data on-board and the number of neighbors $\mathcal{N}_i$, including the action of not transmitting any information.

Remark 4. *The combinatorial explosion caused by scaling the number of agents in the system can be combatted with the*

use of collaboration graphs to factor the action space between agents by clustering capabilities, location, etc.

As an example, a UAV connected to two other drones with RF geolocation and optical tracking capabilities, there are five total actions to choose during any time epoch:

1) Do not communicate (Silence or Null)
2) Send RF Geolocation results to Node A
3) Send Optical Tracking results to Node A
4) Send RF Geolocation results to Node B
5) Send Optical Tracking results to Node B

The state information is assumed to be a combinatorial set of the relevance of the data, the risk to the ground asset, and the state of collaboration based on the expected value of information from previous communications and the quality of the data. The rewards are also combinatorial as formulated in IKD with the information theoretic metric ρ based upon the information matrix of the sensor variances $\rho = 1/\kappa_a$. This leads to an updated relevance of

$$R_r(a,s) = \mathcal{L}(s' \notin \boldsymbol{F}|a,s) \cdot \frac{1}{\kappa_a}, \tag{21}$$

where $\kappa_a = \min_{j \in \mathcal{N}}(\sigma_i \sigma_j)$, σ_i is the variance of the local sensor information being transmitted, and σ_j is the variance of a collaborative agent $j \neq i \in \mathcal{N}$. This also leads to an updated collaboration reward of

$$\hat{R}_c(a,s) = \frac{1}{|\mathcal{N}|}\sum_{j \in \mathcal{N}} \mathcal{S}(\mathcal{Q}(\lambda_j|a,s)) \cdot \frac{1}{\kappa_a}, \tag{22}$$

where the sigmoid function $\mathcal{S}(\cdot)$ has a center point $x_0 = \frac{1}{2}(\bar{\lambda} + \underline{\lambda})$ and k is set to create a switching function beyond the heuristic bounds of $\boldsymbol{\Lambda}$. The function $\mathcal{Q}$ adjusts λ_j based upon the current state s of communication collaboration and the value of information κ_a in the action a to improve the covariance of the Kalman filter.

The transition probabilities for relevance are driven by the relationship between the monitoring UAVs and the ground asset. The ground asset is assumed to be reachable by the UAVs through the MANET or a multi-hop network so that it can be influenced in finding a more conservative route by updating the edge costs between waypoints around the relative position of the hazard. The timeliness and usefulness of the data being transmitted and received was utilized to calculate the collaboration transition probabilities via repeated simulation.

Observation probabilities are defined through subject matter expertise and repeated simulation. The Kalman filter and a sufficient statistic of observation histories, a Bayesian estimator, were processed through a belief function v to create the *relevance* and *collaboration* mapping to the observation space $o \in \boldsymbol{O}$. The observation states themselves are a discrete categorization of local observations using a K-means clustering algorithm learned unsupervised from numerous Monte Carlo simulations. In particular, the mapping of observations v_r is performed solely by cluster classification of the state estimation and its covariance, driven by sensor inaccuracies of the ground asset's proximity to hazards to observation relevance states o_r. Whereas, the observation mapping v_c for collaboration is based on multiple observations: 1) a Bayes estimation of the current arrival probability $\hat{\lambda}_{ij}$ of messages $\sigma : j \rightarrow i$ or $\sigma : i \rightarrow j$ used to ascertain proximity to heuristic bounds; 2) the usefulness of information σ_k recently received from or sent to another agent j according to κ_a; and, 3) the lagged autocovariance $cov(\boldsymbol{P}[m], \boldsymbol{P}[m + l])$ of the Kalman filter covariances $\boldsymbol{P}$ to evaluate the impact of information exchange over time in maintaining mission objectives involved in asset tracking. A *collaboration* mapping v_c is the cluster classification of $\hat{\lambda}$, σ_k, and $cov(\cdot)$ to the observation space o_c. The relevance o_r and collaboration o_c observations from the mappings v are indexed to the observation o in the combinatorial set, which then drives the edge transitions of the finite state controller policy.

C. Constraints on Communication

The constraints being analyzed by CA-POMDP are the utilization of bandwidth (bytes per second) and power (Watt per second) over one second, which is a sampling length of 10 epochs as the UAV makes a collaboration decision every $100ms$. The constraints are considered soft constraints because there is often more available bandwidth in the network than is being allocated to multi-agent collaboration. Also, network protocols like ZeroMQ permit queueing techniques to account for myopic link saturation. The power utilization is considered an average power consumption needed for the communication system to use to ensure the UAV can maintain the required flight time needed to maintain situational awareness of the disaster site.

Resource utilization by the UAVs is dependent on the type of information they are sending. It is assumed that sending optical data requires more data and therefore higher power requirements to achieve the SNR needed for higher bandwidth physical layer protocols, e.g., 64QAM. The data for laser range finding then RF geolocation are assumed to take sequentially less data and power than optical data.

VII. Experimental Results & Analysis

The applicability of Constrained-Action POMDPs to address resource awareness in Intelligent Knowledge Distribution was investigated through a series of Monte Carlo simulations. From these simulations, the probabilistic constraint satisfaction was analyzed between intelligent (IKD), greedy, and naive communication models while confirming their ability to maintain acceptable state estimations.

In our experiment, greedy communications assume a constant communication of the best information observed at all times alternating the destination between each time epoch among collaborating neighbors. Naive communications applies a probability of communicating depending on the relevance of the information, the estimated proximity to a hazard. In previous work [9], we already examined the trade off of constraint satisfaction and optimal value, Figure 8, along with an analysis of how the finite state controller is impacted by constraint states, Figure 7.

The probabilistic constraint satisfaction between the three communication models was analyzed in a Monte Carlo simulation where the simulation would randomly determine a

(a) An unconstrained optimal FSC policy.

(b) A constrained FSC policy where a less optimal but less resource consuming action (Transmit RF) was introduced into the controller.
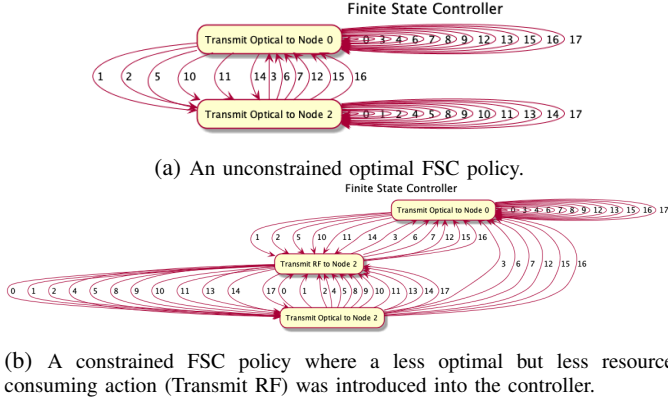
Fig. 7: An example of (a) an unconstrained and (b) an constrained controller of one of the drones (agents) performing Intelligent Communications.
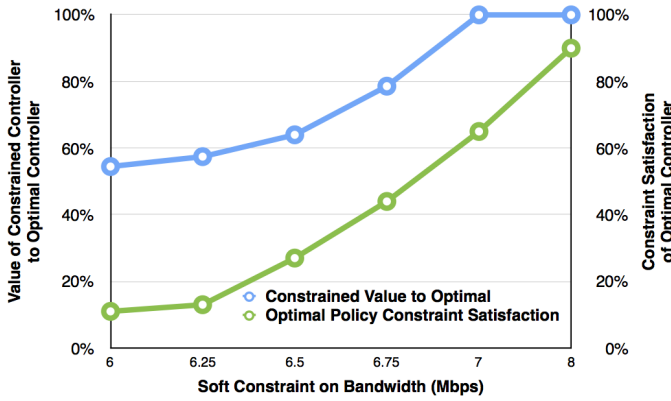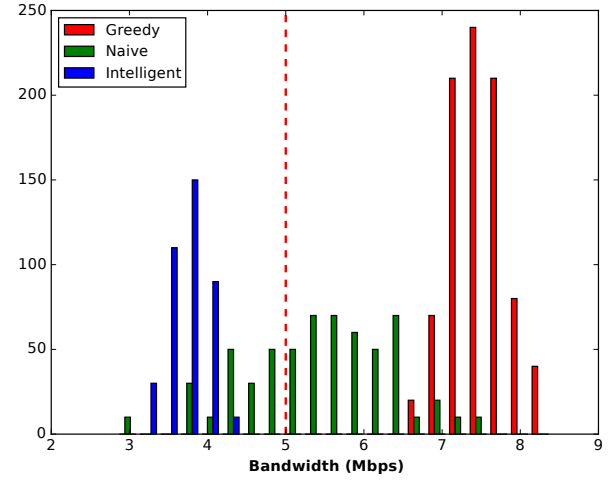

Fig. 8: The impact to the value function and probabilistic constraint satisfaction when satisfying soft limits for Bandwidth. The green line shows the probabilistic constraint satisfaction of an Optimal FSC policy that only satisfies 12% at 6 MBps. The blue line shows the value impact of a Constrained Controller that is 55% of optimal at 6 MBps.
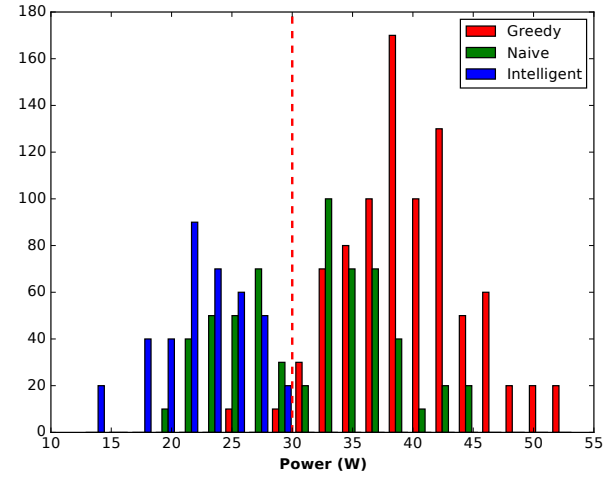
start location for the ground vehicle, a goal location for the ground vehicle, placement of hazards in the environment, and an initial belief state for the controller. The UAV monitoring drones had a predefined search pattern that was consistent between all simulations. During the IKD simulations, the CA-POMDP model was allowed to recalculate a new constrained policy during the execution of a simulation when the variation of information exceeded a threshold to ensure continued compliance to constraints.

The utilization of resources per second by each agent during all these simulations was graphed as histograms in Figure 9 for the two limiting resources along with their soft constraints (dotted red line). The probabilistic constraint satisfaction to these soft limits of both bandwidth and power were substantially improved for intelligent knowledge distribution compared to those of both naive and greedy communication models. This confirms that the CA-POMDP substantiation for *Intelligent Knowledge Distribution* was able to find a controller that complied with probabilistic constraints against soft limits and adapt the controller as needed. During execution, the constraint state "Transmit RF to Node 2" of Figure 7b would be taken at appropriate times to reduce resource usage.

As expected, the naive communication model with its prob-

(a) Bandwidth

(b) Power

Fig. 9: Comparison of constraint satisfaction from three Monte Carlo simulation scenarios involving drones performing either Greedy, Naive, or Intelligent Communications.
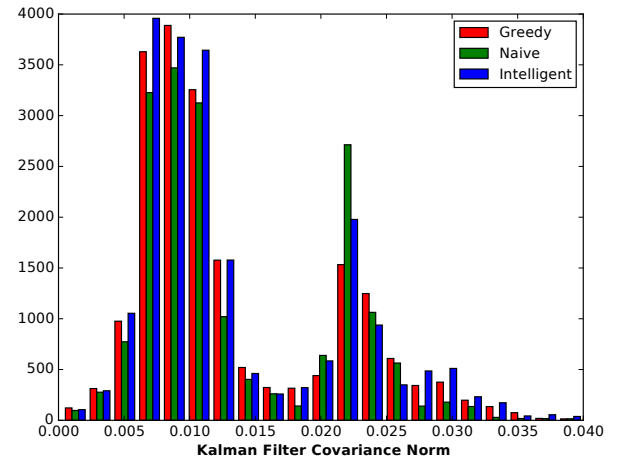

Fig. 10: Histogram of the norm of the Unscented Kalman Filter covariance matrix showing that despite the constrained actions of IKD, the intelligent controller was still able to maintain accurate estimates of the ground vehicle.

abilistic approach to making communication decisions on data relevance alone utilized less resources than the purely greedy communication model. Despite the improvement shown with the naive approach, our IKD approach was compliant to the soft constraints while not negatively impacting the state estimation, determined by the norm of the UKF covariance in Figure 10. The two distributions in the graph are indicative of the times when the filter has full observational data and when the filters have missing observations and measurements.

The first two columns of Table I further validates our results showing the probabilistic constraint satisfaction, or CDF below soft constraints, to the bandwidth and power limits. The accuracy of the state estimations are shown in the last column and it is of interest to note that IKD maintained better estimations than naive. It is hypothesized that the improvement results from IKD sending the most relevant data to ensure bounded performance of the Kalman filter, whereas naive does not have that level of contextual understanding.

TABLE I: Comparison of Simulation Metrics between Greedy, Naive, and Intelligent Communications.

Model	BW	Power	Cov
Greedy	0%	2%	0.0132
Naive	32%	40%	0.0135
Intelligent	97%	97%	0.0133

Remark 5. *It is interesting to note that during the construction and analysis of the system that if the optimal controller meets probabilistic constraint satisfaction less than 15% of the time, then there was an unacceptable level of probability that the the resulting constrained controller would not meet mission objectives. Future work can investigate the driving mechanism behind mission success and techniques to compensate like adaptive soft constraints to ensure mission objectives.*

VIII. FUTURE WORK & CONCLUSION

The Monte Carlo simulations indicate that the incorporation of Intelligent Knowledge Distribution (IKD) that determines what information is needed to whom and when under soft constraints was successful. The approach provides an *agnostic* "plug-and-play" framework for IKD by constraining the actions of a POMDP to only transmit information to a collaborative agent when the value of that information warrants the communication. Constrained-Action POMDPs provide a level of guarantee of probabilistic constraint satisfaction to a desired operational behavior while still allowing short-term bursts of critical information. It also validates the concept of using Markov chain Monte Carlo analysis to evaluate an infinite-horizon policy represented as a finite state controller for its probability of satisfying soft constraints and a combinatorial discrete optimization for achieving desired constraint behavior of a policy while minimizing impact on the controller's value.

As the number of agents grow to respond to operational needs, communication between agents grows according to $\frac{1}{2}n(n-1)$. To reduce the number of communication channels and improve the feasibility of solving for the IKD communication model in larger multi-agent systems, learning coordination graphs seen in Network Distributed POMDPs (ND-POMDP) implementations will be addressed in future work, which could be different for each taxonomy of information being tracked by the agents. For IKD in real-time operational scenarios, the agents available to collaborate and coordinate may not be known until they are actively in the field, therefore learning heterogeneous interactions and constructing a coordination graph online is a necessary expansion. Finally, the above approach is actively being expanded into a fully collaborative multi-agent target tracking simulation to evaluate its effectiveness in a Concurrent Constrained Decentralized POMDP, where the IKD CA-POMDP is one POMDP working concurrently with tasking, motion planning, and target tracking MDPs.

REFERENCES

[1] G. Tuna, B. Nefzi, and G. Conte, "Unmanned aerial vehicle-aided communications system for disaster recovery," *Journal of Network and Computer Applications*, vol. 41, pp. 27–36, 2014.

[2] S. Omidshafiei, A.-a. Agha-mohammadi, C. Amato, and J. P. How, "Decentralized control of partially observable Markov decision processes using belief space macro-actions," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5962–5969.

[3] F. A. Oliehoek, C. Amato *et al.*, *A concise introduction to decentralized POMDPs*. Springer, 2016, vol. 1.

[4] S. Seuken and S. Zilberstein, "Improved memory-bounded dynamic programming for decentralized POMDPs," *arXiv preprint arXiv:1206.5295*, 2012.

[5] J. Capitan, M. T. Spaan, L. Merino, and A. Ollero, "Decentralized multi-robot cooperation with auctioned POMDPs," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 650–671, 2013. [Online]. Available: http://ijr.sagepub.com/content/32/6/650.abstract

[6] J. Girard and M. R. Emami, "Concurrent Markov decision processes for robot team learning," *Engineering Applications of Artificial Intelligence*, vol. 39, pp. 223 – 234, 2015.

[7] O. Sigaud and O. Buffet, *Markov decision processes in artificial intelligence: MDPs, beyond MDPs and applications*. London: John Wiley & Sons, 2010.

[8] Q. Hu and W. Yue, *Markov decision processes with their applications*. Springer Science & Business Media, 2007, vol. 14.

[9] M. Fowler, P. Tokekar, T. C. Clancy, and R. K. Williams, "Constrained-action POMDPs for multi-agent intelligent knowledge distribution," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018 [Accepted].

[10] E. Altman, *Constrained Markov decision processes*. CRC Press, 1999, vol. 7.

[11] A. Beynier and A.-I. Mouaddib, "An iterative algorithm for solving constrained decentralized Markov decision processes," in *AAAI*, vol. 6, 2006, pp. 1089–1094.

[12] M. El Chamie and B. Acikmese, "Finite-Horizon markov decision processes with state constraints," *arXiv preprint arXiv:1507.01585*, 6 Jul. 2015.

[13] S. Feyzabadi and S. Carpin, "HCMDP: A hierarchical solution to constrained Markov decision processes," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 3971–3978.

[14] ——, "Risk-aware path planning using hirerachical constrained Markov decision processes," in *Automation Science and Engineering (CASE), 2014 IEEE International Conference on*. IEEE, 2014, pp. 297–303.

[15] E. A. Hansen, "Finite-memory control of partially observable systems," Ph.D. dissertation, University of Massachusets Amherst, 1998.

[16] J. D. Isom, S. P. Meyn, and R. D. Braatz, "Piecewise linear dynamic programming for constrained POMDPs," in *AAAI*, 2008, pp. 291–296.

[17] A. Undurti and J. P. How, "An online algorithm for constrained POMDPs," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 3966–3973.

[18] E. A. Hansen, "Solving POMDPs by searching in policy space," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1998, pp. 211–219.

[19] M. J. Kochenderfer, C. Amato, and H. J. D. Reynolds, *Decision making under uncertainty: theory and application*. MIT press, 2015.

[20] S. A. Williamson, E. H. Gerding, and N. R. Jennings, "Reward shaping for valuing communications during multi-agent coordination," in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 641–648.

[21] M. Araya, O. Buffet, V. Thomas, and F. Charpillet, "A pomdp extension with belief-dependent rewards," in *Advances in neural information processing systems*, 2010, pp. 64–72.

[22] S. Williamson, E. Gerding, and N. Jennings, "A principled information valuation for communications during multi-agent coordination," in *Proc AAMAS Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains*, 2008, pp. 137–151.

[23] M. T. Spaan, T. S. Veiga, and P. U. Lima, "Decision-theoretic planning under uncertainty with information rewards for active cooperative perception," *Autonomous Agents and Multi-Agent Systems*, vol. 29, no. 6, pp. 1157–1185, 2015.

[24] J. Pineau, G. Gordon, S. Thrun *et al.*, "Point-based value iteration: An anytime algorithm for POMDPs," in *IJCAI*, vol. 3, 2003, pp. 1025–1032.

[25] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[26] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010, vol. 33.

[27] D. S. Bernstein, E. A. Hansen, and S. Zilberstein, "Bounded policy iteration for decentralized POMDPs," in *Proceedings of the nineteenth international joint conference on artificial intelligence (IJCAI)*, 2005, pp. 52–57.

[28] J. Clausen, "Branch and bound algorithms –principles and examples," *Department of Computer Science, University of Copenhagen*, pp. 1–30, 1999.

[29] B. Taylor, "Integer programming: The branch and bound method," *Introduction to Management Science*, 2009.

[30] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[31] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[32] S. Koenig and M. Likhachev, "Dˆ* lite," *Aaai/iaai*, vol. 15, 2002.

[33] H. Musoff and P. Zarchan, *Fundamentals of Kalman filtering: a practical approach*. American Institute of Aeronautics and Astronautics, 2009.

[34] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068. International Society for Optics and Photonics, 1997, pp. 182–194.

[35] D. J. Thomas, K. M. Nastasi, K. Schroeder, and J. T. Black, "Autonomous multi-phenomenology space domain sensor tasking and adaptive estimation," in *2018 21st International Conference on Information Fusion (FUSION)*, July 2018, pp. 1331–1338.

[36] K. M. Nastasi and J. Black, "An autonomous sensor management strategy for monitoring a dynamic space domain with diverse sensors," in *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*, 2018, p. 0890.

T. Charles Clancy Dr. Charles Clancy is an Associate Professor of Electrical and Computer Engineering at Virginia Tech and directs of the Hume Center for National Security and Technology. Prior to joining Virginia Tech in 2010, he served as a senior researcher at the Laboratory for Telecommunications Sciences, a defense research lab at the University of Maryland, where he led research programs in software-defined and cognitive radio. Dr. Clancy received his B.S. in Computer Engineering from the Rose-Hulman Institute of Technology, M.S. in Electrical Engineering from the University of Illinois, and his Ph.D. in Computer Science from the University of Maryland. He is a Senior Member of the IEEE and has over 150 peer-reviewed technical publications. His current research interests include cognitive communications and spectrum security.

Michael Fowler Michael Fowler is a research scientist for the Ted and Karyn Hume Center for National Security and Technology at Virginia Tech responsible for driving and providing thought leadership into the centers research on autonomous systems, mission orchestration, distributed intelligence, and security for wireless and unmanned systems with currently over $4M in active research programs. His research focus is on the convergence of distributed intelligence and machine learning for decision making under uncertainty for embedded applications including drones, satellites, IoT, and wireless communications. He has accumulated over 10 years of experience managing and performing research in security, wireless systems and artificial intelligence at Harris Corporation and as faculty at Virginia Tech. He has received a Masters in Engineering Management from Old Dominion University and is currently a Ph.D. candidate in Computer Engineering at Virginia Tech.

Ryan Williams Ryan K. Williams received the B.S. degree in computer engineering from Virginia Polytechnic Institute and State University in 2005, and the Ph.D. degree from the University of Southern California in 2014. He is currently an Assistant Professor in the Electrical and Computer Engineering Department at Virginia Polytechnic Institute and State University where he runs the Virginia Tech Laboratory for Coordination at Scale (CAS Lab). His current research interests include control, cooperation, and intelligence in distributed multi-agent systems, topological methods in cooperative phenomena, and distributed algorithms for optimization, estimation, inference, and learning. Williams is a Viterbi Fellowship recipient, has been awarded the NSF CISE Research Initiation Initiative grant for young investigators, is a best multi-robot paper finalist at the 2017 IEEE International Conference on Robotics and Automation, and has been featured by various news outlets, including the L.A. Times.