Preprints of the 22nd IFAC World Congress
Yokohama, Japan, July 9-14, 2023

Entropy Rate Maximization of Markov Decision Processes for Surveillance Tasks ⋆

Yu Chen * **Shaoyuan Li** * **Xiang Yin** *

* *Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China. e-mail: {yuchen26,syli,yinxiang}@sjtu.edu.cn)*

Abstract: We consider the problem of synthesizing optimal policies for Markov decision processes (MDP) for both utility objective and security constraint. Specifically, our goal is to maximize the *entropy rate* of the MDP while achieving a surveillance task in the sense that a given region of interest is visited infinitely often with probability one (w.p.1). Such a policy is of our interest since it guarantees both the completion of tasks and maximizes the *unpredictability* of limit behavior of the system. Existing works either focus on the total entropy or do not consider the surveillance tasks which are not suitable for surveillance tasks with infinite horizon. We provide a complete solution to this problem. Specifically, we present an algorithm for synthesizing entropy rate maximizing policies for communicating MDPs. Then based on a new state classification method, we show the entropy rate maximization problem under surveillance task can be effectively solved in polynomial-time. We illustrate the proposed algorithm based on a case study of robot planning scenario.

Keywords: Markov Decision Processes, Entropy Rate, Surveillance Task, Security.

1. INTRODUCTION

Markov decision processes are one of the most widely used mathematical frameworks for decision making with uncertainty. In the classical setting, an MDP usually aims to minimize a cost (or maximize a reward) for finite or infinite horizons Puterman (1994). Recently, motivated by the growing interest in decision-making for complex tasks in autonomous systems, the optimal control for MDPs with temporal logic tasks has also been drawn consideration attention in the literature Ding et al. (2014); Xie et al. (2021).

Our work is motivated by the surveillance tasks in robotic applications Duan and Bullo (2021). In this setting, the robot needs to persistently visit some regions of interest, for example to search for enemies or to collect/upload data. Therefore, the robot needs to ensure that some surveillance tasks can be achieved *infinitely often* w.p.1. This problem has drawn considerable attention recently in the context of formal synthesis, e.g., Smith et al. (2011); Yu et al. (2022).

When there is no non-determinism in the system model, a direct way to achieve a surveillance task is to follow a pre-design cyclic path along which desired regions are visited infinitely often for sure. However, such a plan is completely predictable and has several disadvantages in an adversarial environment. Therefore, for the purpose of security, the surveillance strategies are desired to be as *unpredictable* as possible Li et al. (2020); Zheng et al. (2022); Yang and Yin (2022); Liu et al. (2022); Chen et al. (2023).

Among different notions of unpredictability of the system's behavior, *entropy* is a widely used and very fundamental information-theoretic measure for quantifying how uncertain a stochastic process is. Recently, in Savas et al. (2019), the authors consider the problem of finding a policy for an MDP in order to maximize the entropy while satisfying a linear temporal logic constraint. However, unless the behavior of the system eventually becomes deterministic, the entropy of a stochastic process may diverge when the time horizon goes to infinite. Therefore, for surveillance tasks with infinite horizons, a more meaningful way is to consider the *entropy rate* of the system; see, e.g., Chen and Han (2014); George et al. (2018). However, existing methods for entropy rate maximization already assume that MDP induces an irreducible Markov chain under all policies.

In this paper, we formulate and solve entropy rate maximization problem of MDP for surveillance tasks. Specifically, the robot is required to visit a given region of interest *infinitely often* w.p.1, while at the same time, maximize its *entropy rate* as much as possible. Our contributions are as follows. First, we provide a direct approach for synthesizing policies that maximize entropy rates for *communicating* MDPs without considering the surveillance task. Our approach is based on a new structural property of maxentropic policies and a convex nonlinear program. Second, we provide a polynomial-time algorithm for synthesizing policies that both satisfy the surveillance tasks and maximize the entropy rate, for general MDPs without structural assumptions. By classifying states in MDP into different levels, we solve the synthesis problem inductively via a set of expected reward optimization problems. The constructed policy is stationary. Finally, we illustrate the proposed method by a case study of robot surveillance.

⋆ This work was supported by the National Natural Science Foundation of China (62061136004, 62173226, 61833012).

Notations: We denote by $\mathbb{R}$ and $\mathbb{N}$ the set of real numbers and the set of nature numbers, respectively. For any set A, 2^A and $|A|$ denote its power-set and its cardinality, respectively. For a matrix $\mathbb{P} \in \mathbb{R}^{n\times n}$, $\mathbb{P}^k$ and $\mathbb{P}_{i,j}$ denote its k-th power and the (i,j)-th component of $\mathbb{P}$, respectively. All logarithms in this work are considered with base 2.

2. PRELIMINARY

2.1 Markov Decision Processes

A (finite) Markov decision process (MDP) is a 3-tuple

$$\mathcal{M} = (S, A, P),$$

where $S = \{1, \ldots, n\}$ is a finite set of states, A is finite set of actions and $P : S \times A \times S \to [0,1]$ is a transition function such that for any $s \in S, a \in A$, we have $\sum_{s' \in S} P(s' \mid s, a) \in \{0,1\}$. We also write $P(s' \mid s,a)$ as $P_{s,a,s'}$. We denote by $\mathtt{succ}(s,a) = \{s' \mid P_{s,a,s'} > 0\}$ the set of successor states of s under action a. For each state $s \in S$, we denote by $A(s) = \{a \in A : \mathtt{succ}(s,a) \neq \emptyset\}$ as the set of available actions at s. We assume there always exists at least one available action at each state, i.e., $\forall s \in S : A(s) \neq \emptyset$. An MDP also induces an underlying directed graph (digraph), where each vertex is a state and edge $\langle s, s' \rangle$ is defined if $P_{s,a,s'} > 0$ for some $a \in A$. An MDP is usually assigned with an *initial distribution* of states $\pi_0 : S \to [0,1]$ such that $\sum_{s\in S} \pi_0(s) = 1$.

A Markov chain (MC) $\mathcal{C}$ is an MDP such that $|A(s)| = 1$ for all $s \in S$. The transition matrix of MC is denoted by $\mathbb{P}$, i.e., $\mathbb{P}_{s,s'} = P_{s,a,s'}$, where $a \in A(s)$ is the unique action at state $s \in S$. Therefore, we can omit actions in MC and write it as $\mathcal{C} = (S, \mathbb{P})$. The limit transition matrix is defined by $\mathbb{P}^* = \lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n} \mathbb{P}^k$. Note that the limit matrix always exists for finite MC.

A *policy* for an MDP $\mathcal{M}$ is a sequence $\mu = (\mu_0, \mu_1, ...)$ where $\mu_k : S \times A \to [0,1]$ is a function such that $\forall s \in S, k \geq 0 : \sum_{a \in A(s)} \mu_k(s,a) = 1$. A policy is said to be *stationary* if $\mu_i = \mu_j$ for all i,j and we write a stationary policy by $\mu = (\mu, \mu, \ldots)$ for simplicity. Given an MDP $\mathcal{M}$, the sets of all policies and all stationary policies are denoted by $\Pi_{\mathcal{M}}$ and $\Pi^S_{\mathcal{M}}$, respectively. Given an MDP $\mathcal{M}$, a stationary policy $\mu \in \Pi^S_{\mathcal{M}}$ induces an MC denoted by $\mathcal{M}^\mu = (S, \mathbb{P}^\mu)$, where $\mathbb{P}^\mu_{i,j} = \sum_{a\in A(s)} \mu(i,a) P_{i,a,j}$. We denote by $\mathtt{reach}(s)$ the set of states reachable from state $s \in S$, i.e.,

$$\mathtt{reach}(s) = \{s' \in S \mid \exists \mu \in \Pi_{\mathcal{M}}, \exists n \in \mathbb{N} \text{ s.t. } (\mathbb{P}^\mu)^n_{s,s'} > 0\}.$$

Let $\mathcal{M} = (S, A, P)$ be an MDP with initial distribution π_0 and $\mu = (\mu_0, \mu_1, ...)$ be a policy. An infinite sequence $\rho = s_0 s_1 \cdots$ of states is said to be a *path* in $\mathcal{M}$ under μ if (i) $\pi_0(s_0) > 0$; and (ii) $\forall k \geq 0 : \sum_{a \in A(s_k)} \mu_k(s_k, a) P_{s_k, a, s_{k+1}} > 0$. We denote by $\mathsf{Path}^\mu(\mathcal{M})$ the set of all paths in $\mathcal{M}$ under μ. We use the standard probability measure in Baier and Katoen (2008) and denote it by $\mathsf{Pr}^\mu_{\mathcal{M}} : 2^{S^\omega} \to [0,1]$.

Given MDP $\mathcal{M} = (S, A, P)$, let $(\mathcal{S}, \mathcal{A})$ be a tuple, where $\mathcal{S} \subseteq S$ is a non-empty set of states and $\mathcal{A} : \mathcal{S} \to 2^A \setminus \emptyset$ is a function such that (i) $\forall s \in \mathcal{S} : \mathcal{A}(s) \subseteq A(s)$; and (ii) $\forall s \in \mathcal{S}, a \in \mathcal{A}(s) : \mathtt{succ}(s,a) \subseteq \mathcal{S}$. Essentially, state-action pair $(\mathcal{S}, \mathcal{A})$ induces a new MDP called the *sub-MDP* of $\mathcal{M}$, denoted by $\mathcal{M}(\mathcal{S}, \mathcal{A})$ (or $(\mathcal{S}, \mathcal{A})$), by restricting the state space to $\mathcal{S}$ and available actions to $\mathcal{A}(s)$ for each $s \in \mathcal{S}$.

Definition 1. (Maximal End Components). Let $(\mathcal{S}, \mathcal{A})$ be a sub-MDP of $\mathcal{M} = (S, A, P)$. We say $(\mathcal{S}, \mathcal{A})$ is an *end component* if its underlying digraph is strongly connected. Furthermore, we say $(\mathcal{S}, \mathcal{A})$ is an *maximal end component* if it is an end component and there is no other end component $(\mathcal{S}', \mathcal{A}')$ such that (i) $\mathcal{S} \subsetneq \mathcal{S}'$; and (ii) $\forall s \in \mathcal{S} : \mathcal{A}(s) \subseteq \mathcal{A}'(s)$. We denote by $\mathtt{MEC}(\mathcal{M})$ the set of all MECs in $\mathcal{M}$.

2.2 Entropy Rate of Stochastic Processes

Let X be a discrete random variable with support $\mathcal{X}$ and probability mass function $p(x) := \mathsf{Pr}(X = x), x \in \mathcal{X}$. The entropy of random variable X is defined as:

$$H(X) := -\sum_{x \in \mathcal{X}} p(x) \log p(x). \tag{1}$$

We define $0 \log(0) = 0$. For two random variables X_0 and X_1 with joint probability mass function $p(x_0, x_1)$, the *joint entropy* of X_0 and X_1 is defined by

$$H(X_0, X_1) := -\sum_{x_0 \in \mathcal{X}} \sum_{x_1 \in \mathcal{X}} p(x_0, x_1) \log p(x_0, x_1). \tag{2}$$

The joint entropy can also be directly extended to a discrete time stochastic process $\{X_k\}$. Intuitively, it provides a measure for how *unpredictable* the process is. However, joint entropy $H(X_0, X_1, \ldots, X_n)$ usually diverge when n goes to infinitely. Therefore, for infinite processes, one usually use the *entropy rate* instead of the joint entropy.

Definition 2. (Entropy Rate). The entropy rate of a stochastic process $\{X_k\}$ is defined as

$$\nabla H(\{X_k\}) := \lim_{k\to\infty} \frac{1}{k} H(X_0, \ldots, X_k). \tag{3}$$

Given an MC $\mathcal{C} = (S, \mathbb{P})$ with initial distribution π_0, it also induces a discrete stochastic process $\{X_k \in S : k \in \mathbb{N}\}$ where X_k is a random variable over state space S. We denote by $\nabla H(\mathcal{C})$ the entropy rate of MC $\mathcal{C}$, which is the entropy rate of its induced process. It has been shown in Chen and Han (2014) that this entropy rate is:

$$\nabla H(\mathcal{C}) = \sum_{s \in S} \pi(s) L(s), \tag{4}$$

where $\pi(s) = \pi_0 \mathbb{P}^*$ is the limit distribution and $L(s)$ is *local entropy* defined by $L(s) = \sum_{s' \in S} -\mathbb{P}_{s,s'} \log \mathbb{P}_{s,s'}$. For an MDP $\mathcal{M}$, we define $\nabla H(\mathcal{M}) := \sup_{\mu \in \Pi_{\mathcal{M}}} \nabla H(\mathcal{M}^\mu)$ as its entropy rate.

3. PROBLEM FORMULATION

Formally, let $\mathcal{M}$ be an MDP, $\mu \in \Pi_{\mathcal{M}}$ be a policy and $B \subseteq S$ be a set of *target states* representing states that need to be visited infinitely. For any path $\tau \in \mathsf{Path}^\mu(\mathcal{M})$, we denote by $\mathtt{inf}(\tau)$ the set of states that occur *infinite number of times* in τ. Then we define

$$\mathsf{Pr}^\mu_{\mathcal{M}}(\Box\Diamond B) = \mathsf{Pr}^\mu_{\mathcal{M}}(\{\tau \in \mathsf{Path}^\mu(\mathcal{M}) \mid \mathtt{inf}(\tau) \cap B \neq \emptyset\})$$

as the probability of visiting B infinitely often in $\mathcal{M}$ under μ. We denote by $\Pi^B_{\mathcal{M}}$ the set of all policies under which B is visited infinitely often w.p.1, i.e.,

$$\Pi^B_{\mathcal{M}} = \{\mu \in \Pi_{\mathcal{M}} \mid \mathsf{Pr}^\mu_{\mathcal{M}}(\Box\Diamond B) = 1\}.$$

Then we formulate the problem in this paper.

Problem 1. (Entropy Rate Maximization for Surveillance Tasks). Given MDP $\mathcal{M} = (S, A, P)$ with initial distribution π_0, find a policy $\mu^\star \in \Pi^B_{\mathcal{M}}$ such that

$$\forall \mu \in \Pi^B_{\mathcal{M}} : \nabla H(\mathcal{M}^{\mu^\star}) \geq \nabla H(\mathcal{M}^{\mu}).$$

In order to fulfill the surveillance task, the MDP needs to eventually stay in an MEC in which there exists at least one target state in B. We say $(\mathcal{S}, \mathcal{A}) \in \mathtt{MEC}(\mathcal{M})$ is *accepting* if $\mathcal{S} \cap B \neq \emptyset$. We denote by $\mathtt{AMEC}(\mathcal{M})$ set of accepting MECs.

4. ENTROPY RATE MAXIMIZATION FOR COMMUNICATING MDP

Before solving Problem 1, we first consider an unconstrained case without requiring that B is visited infinitely often. This problem has been considered in Chen and Han (2014), where it shows that $\nabla H(\mathcal{M})$ can be effectively computed and can be achieved by a stationary policy, i.e.,

$$\nabla H(\mathcal{M}) = \sup_{\mu \in \Pi^S_{\mathcal{M}}} \nabla H(\mathcal{M}^\mu) = \sup_{\mu \in \Pi_{\mathcal{M}}} \nabla H(\mathcal{M}^\mu) \tag{5}$$

The approach of Chen and Han (2014) is based on a nonlinear program to determine maximum entropy rate to the MDP, which does not directly yield a policy to achieve this value. Here, we consider a special case of MDP, where all states can be visited from one to another under some policy. We show that, under this assumption, the stationary policy achieving entropy maximization rate can be synthesized directly based on a different nonlinear program.

Formally, an MDP $\mathcal{M}$ is said to be *communicating* if

$$\forall s, s' \in S, \exists \mu \in \Pi_{\mathcal{M}}, \exists n \geq 0 : (\mathbb{P}^\mu)^n_{s,s'} > 0.$$

In fact, if $\mathcal{M}$ is communicating, then the above condition can be achieved by a stationary policy $\mu \in \Pi^S_{\mathcal{M}}$. Therefore, the resulting MC $\mathbb{P}^\mu$ is *irreducible*, i.e., each pair of states can be visited from one to the other via some path.

The following result shows that, for a communicating MDP, if a stationary policy maximizes the entropy rate, then its induced MC must be irreducible.

Lemma 1. Let $\mathcal{M}$ be a communicating MDP and $\mu \in \Pi^S_{\mathcal{M}}$ be a stationary policy such that $\nabla H(\mathcal{M}^\mu) = \nabla H(\mathcal{M})$. Then the induced MC $\mathcal{M}^\mu$ is irreducible.

With the above structural property and Equation (4), for the case of communicating MDP, we can transform the policy synthesis problem for entropy rate maximization to a steady-state parameter synthesis problem described by the following nonlinear program.

The intuition of the nonlinear program is as follows. The decision variables are $\gamma(s, a)$ for state-action pair $s \in S$ and $a \in A(s)$. They are used to represent the probability of occupying state s and choosing action a when the system goes to steady state, i.e., $\sum_{a \in A(s)} \gamma(s, a) = \pi(s)$ where π is limit distribution of induced MC. Variables $q(s, t)$ and $\lambda(s)$ in Equations (7) and (8) are functions of $\gamma(s, a)$, representing the probability of going from states s to t and the probability of occupying state s, respectively. Finally, the objective function is computation of entropy rate of an MC given in Equation (4).

Nonlinear Program for Communicating MDP

$$\max_{\gamma(s,a)} \quad \sum_{s \in S} \sum_{t \in S} -q(s,t) \log \left(\frac{q(s,t)}{\lambda(s)} \right) \tag{6}$$

$$\text{s.t.} \quad q(s,t) = \sum_{a \in A(s)} \gamma(s,a) P(t \mid s,a), \forall s,t \in S \tag{7}$$

$$\lambda(s) = \sum_{a \in A(s)} \gamma(s,a), \forall s \in S \tag{8}$$

$$\lambda(t) = \sum_{s \in S} q(s,t), \forall t \in S \tag{9}$$

$$\sum_{s \in S} \lambda(s) = 1 \tag{10}$$

$$\gamma(s,a) \geq 0, \forall s \in S, \forall a \in A(s) \tag{11}$$

The following result shows that, in fact, the proposed nonlinear program is convex. Hence, it can be computed by efficient algorithms.

Proposition 1. The nonlinear program in Equations (6)-(11) is convex and can be solved in polynomial-time.

Given a communicating MDP $\mathcal{M}$, suppose $\gamma^\star(s, a)$ is the solution to Equations (6)-(11). Then we can define a stationary policy by

$$\mu^\star(s, a) = \frac{\gamma^\star(s, a)}{\sum_{a \in A(s)} \gamma^\star(s, a)}. \tag{12}$$

The following result shows that this policy is indeed a maximum entropy rate policy.

Theorem 1. Let $\mathcal{M}$ be a communicating MDP. Then for policy $\mu^\star \in \Pi^S_{\mathcal{M}}$ defined in Equation (12), we have $\nabla H(\mathcal{M}^{\mu^\star}) = \nabla H(\mathcal{M})$.

5. STATE LEVEL CLASSIFICATIONS

Clearly, if the MDP is communicating, then the resulting maximum entropy rate policy for the unconstrained case also solves Problem 1. This is because the resulting MC under the optimal policy is ensured to be irreducible, i.e., all states can be visited infinitely often w.p.1. However, for the non-communicating case, the problem is more challenging. To resolve this issue, we propose an approach to classify MECs into different "levels" in terms of their connectivities.

Let $\mathtt{MEC}(\mathcal{M}) = \{(\mathcal{S}_1, \mathcal{A}_1), \ldots, (\mathcal{S}_n, \mathcal{A}_n)\}$ be the set of all MECs. Note that $\mathtt{MEC}(\mathcal{M})$ can be effectively computed; see, e.g., Algorithm 47 in Baier and Katoen (2008). Clearly, each state in an MDP can belong to at most one MEC. Furthermore, for two different MECs $(\mathcal{S}_i, \mathcal{A}_i)$ and $(\mathcal{S}_j, \mathcal{A}_j)$, if $\mathcal{S}_i$ is reachable from $\mathcal{S}_j$, then $\mathcal{S}_j$ is not reachable from $\mathcal{S}_i$; otherwise $(\mathcal{S}_i \cup \mathcal{S}_j, \mathcal{A}_i \cup \mathcal{A}_j)$ will be a larger MEC. We denote by $\mathcal{S}_M = \cup_{i=1}^n \mathcal{S}_i$ the set of all states in some MEC (called MEC states) and by $\mathcal{T} = S \setminus \mathcal{S}_M$ the set of states not in any MEC (called transtient states). Therefore, we have the following partition of the state space

$$S = \mathcal{S}_1 \dot\cup \mathcal{S}_2 \dot\cup \ldots \dot\cup \mathcal{S}_n \dot\cup \mathcal{T}.$$

For each state $s \in \mathcal{S}_M$, we denote by $\mathcal{S}(s) \subseteq S$, the corresponding set of states of the unique MEC it belongs to, i.e.,

$$(\mathcal{S}_i, \mathcal{A}_i) \in \mathtt{MEC}(\mathcal{M}) \wedge s \in \mathcal{S}_i \Rightarrow \mathcal{S}(s) = \mathcal{S}_i.$$

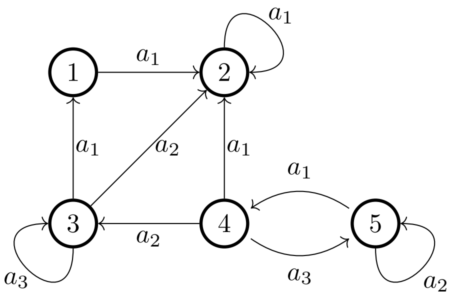


Fig. 1. In the figure, each transition has a non-zero probability and the specific value is omitted. For MDP $\mathcal{M}$, we have $L_0 = \{2\}$, $T_0 = \{1\}$, $L_1 = \{3\}$, $T_1 = \emptyset$, $L_2 = \{4, 5\}$ and $T_2 = \emptyset$. Therefore, We have $\texttt{level}(\mathcal{M}) = 2$.

Now we classify MECs into different "levels" as follows. We observe that there must exist states that cannot leave their MECs and we consider those states as the "lowest" level. Then an MEC state is said to be with level k if it can only reach MECs with lower levels or itself. This leads to the following definition.

Definition 3. (State Levels for MEC States). Let $\mathcal{M}$ be an MDP and S_M be the set of MECs states. Then for each $k \geq 0$, the set of k-level MEC states, denoted by L_k, is defined inductively as follows:

$$L_0=\{s\in\mathcal{S}_M \mid \texttt{reach}(s)\cap\mathcal{S}_M = \mathcal{S}(s)\} \tag{13}$$
$$L_k=\{s\in\mathcal{S}_M \mid \texttt{reach}(s)\cap\mathcal{S}_M\subseteq \bigcup_{m<k} L_m\cup\mathcal{S}(s)\}\setminus\bigcup_{m<k}L_m.$$

We define $\texttt{level}(\mathcal{M}) = \max\{k \mid L_k \neq \emptyset\}$ as the highest level of MECs in MDP $\mathcal{M}$.

Similarly, for transient states $\mathcal{T} = S \setminus \mathcal{S}_M$, we also define the set of k-level states as those states who can only reach MECs with levels smaller than or equal to k.

Definition 4. (State Levels for Transient States). Let $\mathcal{M}$ be an MDP and $\mathcal{T} = S \setminus \mathcal{S}_M$ be the set of transient states. Then for each $k \geq 0$, the set of k-level transient states, denoted by T_k, is defined by:

$$T_k = \{s \in \mathcal{T} \mid \texttt{reach}(s) \cap \mathcal{S}_M \subseteq \bigcup_{m\leq k} L_m\} \setminus \bigcup_{m<k} T_m. \tag{14}$$

Clearly, sets $L_0, L_1, \ldots, L_{\texttt{level}(\mathcal{M})}$ forms a partition of $\mathcal{S}_M$; similarly, we also have $\dot{\bigcup}_{k=1}^{\texttt{level}(\mathcal{M})} T_k = \mathcal{T}$. An example of the state classification is provided in Fig. 1.

The above state classes can be computed as follows. First, we compute all MECs $\texttt{MEC}(\mathcal{M}) = \{(\mathcal{S}_1, \mathcal{A}_1), \ldots, (\mathcal{S}_n, \mathcal{A}_n)\}$ and $\mathcal{T} = S \setminus (\bigcup_{k=1}^{n} \mathcal{S}_k)$ by Algorithm 47 in Baier and Katoen (2008). Then we aggregate each MECs as a single state and obtain an abstracted MDP $\tilde{\mathcal{M}}$; see, e.g., De Alfaro (1998). We denote by $\tilde{G}(\mathcal{M}) = (\tilde{V}, \tilde{E})$ the underlying digraph of the abstracted MDP $\tilde{\mathcal{M}}$. We can classify state levels by following steps:

- L_0 are those states only have self-loops;
- For each $k = 1, \ldots, \texttt{level}(\mathcal{M})$:
  - · T_{i-1} is set of transient states such that they cannot reach MEC states whose level have not yet been determined;
  - · L_i is set of MEC states that can only reach itself or states whose level have been determined;
- Those states left are in $T_{\texttt{level}(\mathcal{M})}$.

6. MAXIMIZATION OF ENTROPY RATE FOR SURVEILLANCE TASKS

Now, we tackle the entropy rate maximization problem for the constrained case, where the surveillance task also needs to be fulfilled. Specifically, we define

$$\nabla H_B(\mathcal{M}) = \sup_{\mu\in\Pi^B_{\mathcal{M}}} \nabla H(\mathcal{M}^\mu)$$

as the maximum entropy rate that can be achieved by a policy satisfying the surveillance task.

To handle non-communicating MDP $\mathcal{M}$, we define

$$R_k = \bigcup_{m=0}^{k} (L_m \cup T_m)$$

as the set of all MEC states and transient states with levels smaller than or equal to k. We also define

$$\hat{R}_k = R_k \setminus T_k$$

as the set of all MEC states with levels smaller than or equal to k and transient states with levels strictly smaller than k. We define $A_k = \cup_{s\in R_k} A(s)$ and $\hat{A}_k = \cup_{s\in \hat{R}_k} A(s)$ the sets of all available actions in R_k and $\hat{R}_k$, respectively.

Now we make the following observations for the above defined R_k and $\hat{R}_k$. First, we observe that, for any $k = 0, 1, \ldots, \texttt{level}(\mathcal{M})$, (R_k, A_k) and $(\hat{R}_k, \hat{A}_k)$ are both sub-MDPs. This is because, by the definition of state levels, states with level k can only go to states with lower levels. Second, for sub-MDPs $(\hat{R}_k, \hat{A}_k)$ and $(\hat{R}_m, \hat{A}_m)$, where $k < m$, suppose that μ and $\hat{\mu}$ are policies that solves Problem 1 for $(\hat{R}_k, \hat{A}_k)$ and $(\hat{R}_m, \hat{A}_m)$, respectively. Then by modifying $\hat{\mu}$ to $\hat{\mu}'$ such that (i) for all $s \in \hat{R}_k$, $\hat{\mu}(s)$ is changed to $\mu(s)$ and (ii) unchanged otherwise, we know that the modified $\hat{\mu}'$ also solves Problem 1 w.r.t. $(\hat{R}_m, \hat{A}_m)$.

The above observations suggest that we can find a solution to Problem 1 in a backwards manner from states with the lowest level as follows:

- Initially, we start from those AMECs $(\mathcal{S}_i, \mathcal{A}_i)$ with level 0 and compute the maximum entropy rate we can achieve for this sub-MDP. Since each MEC is communicating, we can use the method in Section 4. For those MECs that are not accepting, we directly define its entropy rate as minus infinity since staying in such MEC implies the violation of the task.
- Once all MECs in L_0 have been processed, we move to include transient states in T_0. This essentially provides an instance of Problem 1 w.r.t. sub-MDP (R_0, A_0). Since states in T_0 are transient no matter what action we take, the only factor that determines the total entropy rate is what MEC with level 0 they choose to go. Therefore, it suffices to solve an *expected total reward* maximization problem, where the reward of reaching each MEC with level 0 is the computed maximum entropy rate.

- Then we proceed to further consider MECs in L_1 in addition to (R_0, A_0), which gives an instance of Problem 1 w.r.t. sub-MDP $(\hat{R}_1, \hat{A}_1)$. Then for each $(\mathcal{S}_i, \mathcal{A}_i)$ in L_1, we still compute its maximum entropy rate if it is accepting and set it as minus infinity otherwise. Here we have two possible treatments for $(\mathcal{S}_i, \mathcal{A}_i)$: (i) consider it as a transient part as the case of T_0 by solving an expected total reward maximization problem; or (ii) choose to stay in the current MEC. Therefore, we need to compare these two alternatives and choose the one with larger reward (entropy rate).
- Once $(\hat{R}_1, \hat{A}_1)$ is processed, we further include T_1 to consider the instance of (R_1, A_1), and so forth, until the instance of $(R_{\texttt{level}(\mathcal{M})}, A_{\texttt{level}(\mathcal{M})}) = \mathcal{M}$ is solved.

Now, we formalize the implementation details of the above idea. In Section 4, we have shown how to compute the maximum entropy rate for each MEC. Here, we consider how to process transient states T_i and MEC states L_i together when we decide to move to MECs with lower levels by a single linear program.

Specifically, given MDP $\mathcal{M}$, at decision stage $k = 0, \dots, \texttt{level}(\mathcal{M})$, suppose we have the following information available:

- the set of states have been processed: $\hat{R}_k \subseteq S$;
- the solution $\hat{\mu}_k \in \Pi^S_{\hat{\mathcal{M}}_k}$ of Problem 1 w.r.t. current sub-MDP $\hat{\mathcal{M}}_k = (\hat{R}_k, \hat{A}_k)$;
- the maximum entropy rate one can achieve from each state while satisfying the surveillance task, which is specified by a function $\texttt{val}_k : \hat{R}_k \to \mathbb{R} \cup \{-\infty\}$.

Note that, for $k = 0$ and each $s \in \hat{R}_0$, $\texttt{val}_0(s) = -\infty$ when s is not in any AMEC and otherwise $\texttt{val}_0(s) = \nabla H((\mathcal{S}, \mathcal{A}))$ where $(\mathcal{S}, \mathcal{A}) \in \texttt{AMEC}(\mathcal{M})$ and $s \in \mathcal{S}$. The method to compute $\texttt{val}_k$ for $k \geq 1$ will be presented later. Then our objective is to find the optimal policy $\hat{\mu}_{k+1} \in \Pi^S_{\hat{\mathcal{M}}_{k+1}}$ for a larger sub-MDP $\hat{\mathcal{M}}_{k+1} = (\hat{R}_{k+1}, \hat{A}_{k+1})$, where $\hat{R}_{k+1} = \hat{R}_k \cup \{T_k, L_{k+1}\}$. To this end, given $(\hat{\mathcal{M}}_{k+1}, \hat{R}_k, \texttt{val}_k)$ as an instance, let $Q = T_k \cup L_{k+1}$ and α be an arbitrary positive vector such that $\sum_{s \in Q} \alpha(s) = 1$ and $\forall s \in Q : \alpha(s) > 0$. We define the following linear program (LP).

Linear Program for Each Level

$$\max_{\gamma(s,a)} \quad \sum_{s \in Q} \sum_{t \in \hat{R}_k} \texttt{val}(t) \lambda(s,t) \tag{15}$$

$$\text{s.t.} \quad \mu(s) - \sum_{t \in Q} \lambda(t,s) \leq \alpha(s), \forall s \in Q \tag{16}$$

$$\mu(s) = \sum_{a \in A(s)} \gamma(s,a), \forall s \in Q \tag{17}$$

$$\lambda(s,t) = \sum_{a \in A(s)} \gamma(s,a) P_{s,a,t}, \forall s \in Q, t \in \hat{R}_k \tag{18}$$

$$\gamma(s,a) \geq 0, \forall s \in Q, \forall a \in A(s) \tag{19}$$

The LP comes follows the standard framework of solving expected total reward problem as provided in Puterman (1994). The decision variables are $\gamma(s,a)$ in Equation (19) for $s \in Q$ and $a \in A(s)$. Intuitively, $\gamma(s,a)$ is the expected number of visits to state s and choose action a when initial distribution of MDP is α. The requirement that $\alpha(s) > 0$ is for technical consideration to ensure that in LP all states are visited with non-zero probability. Variables $\mu(s)$ and $\lambda(s,t)$ in Equations (17) and (18) are functions of $\gamma(s,a)$ representing the expected number of visits to state s and the expected number of transitions from s to t, respectively. Equation (16) is the constraint of the probability flow. Finally, objective function in Equation (15) multiplies the probability of reaching $\hat{R}_k$ and the reward (maximum entropy rate) in $\hat{R}_k$ and sums over $s \in Q$.

Based on the solution of the linear program (15)-(19) for $(\hat{\mathcal{M}}_{k+1}, \hat{R}_k, \texttt{val}_k)$, we can decode a policy $\hat{\mu}'_{k+1}$ for sub-MDP $\hat{\mathcal{M}}_{k+1} = (\hat{R}_{k+1}, \hat{A}_{k+1})$ as follows:

$$\hat{\mu}'_{k+1}(s,a) = \frac{\gamma(s,a)}{\sum_{a \in A(s)} \gamma(s,a)} \text{ for } s \in Q_* \tag{20}$$

$$\hat{\mu}'_{k+1}(s,a) = 1 \text{ for } s \in Q \setminus Q_*, \text{ arbitrary } a \in A(s) \tag{21}$$

where $Q_* = \{s \in Q \mid \sum_{a \in A(s)} \gamma(s,a) > 0\}$.

We now compute the maximum total reward (entropy rate) initial from each $s \in Q$ on condition that s is transient. Specifically, we denote by $T \subseteq Q$ the set of all transient states in MC $\hat{\mathcal{M}}^{\hat{\mu}'_{k+1}}_{k+1}$. We denote by $\mathbb{P}^{\hat{\mu}'_{k+1}}_T$ and $\mathbb{P}^{\hat{\mu}'_{k+1}}_{\hat{R}_k}$ the submatrix of $\mathbb{P}^{\hat{\mu}'_{k+1}}$ restricted rows on T and columns on T and $\hat{R}_k$, respectively. We denote by $v \in \mathbb{R}^{|T|}$ the solution of following equation

$$v = \mathbb{P}^{\hat{\mu}'_{k+1}}_T v + \mathbb{P}^{\hat{\mu}'_{k+1}}_{\hat{R}_k} \texttt{val}_k$$

Since $I - \mathbb{P}^{\hat{\mu}'_{k+1}}_T$ is invertible, above equation has unique solution denoted by v'. Then we can define maximum total reward by $\mathtt{v}'$ as follows:

$$\mathtt{v}'(s) = \begin{cases} v'(s) & \text{if} \quad s \in T \\ -\infty & \text{if} \quad s \in Q \setminus T \end{cases} \tag{22}$$

Intuitively, Equation (22) computes the maximum entropy rate assuming that the policy forces $s \in Q$ to be transient and the surveillance task is satisfied. If $s \in Q$ is recurrent in $\hat{\mathcal{M}}^{\hat{\mu}'_{k+1}}_{k+1}$, then the total reward initial from s is zero. It means that under any policy such that s is transient, the total reward initial from s is $-\infty$, i.e., the surveillance task cannot be achieved. Therefore, $\mathtt{v}'(s) = -\infty$ for $s \in Q \setminus T$.

Note that $\hat{\mu}'_{k+1}$ may not be the optimal policy for sub-MDP $\hat{\mathcal{M}}_{k+1} = (\hat{R}_{k+1}, \hat{A}_{k+1})$ since the linear program is designed by assuming that each state in L_{k+1} has to go to MEC with lower levels. However, states in L_{k+1} can also choose to stay at level $k+1$. To capture this issue, for each MEC $(\mathcal{S}_i, \mathcal{A}_i) \in \texttt{MEC}(\mathcal{M})$ with level $k+1$, i.e., $\mathcal{S}_i \subseteq L_{k+1}$, we denote $\mu_{\text{stay},i}$ be the optimal (stationary) policy maximizing the entropy rate for $(\mathcal{S}_i, \mathcal{A}_i)$. This policy can be computed by the approach presented in Section 4. Then for $s \in (\mathcal{S}_i, \mathcal{A}_i)$, we define the *stay value* of s as the maximum entropy rate of sub-MDP $(\mathcal{S}_i, \mathcal{A}_i)$ if it is an accepting MEC, and minus infinity otherwise, i.e.,

$$\texttt{stay}(s) = \begin{cases} \nabla H((\mathcal{S}_i, \mathcal{A}_i)) & \text{if} \quad (\mathcal{S}_i, \mathcal{A}_i) \in \texttt{AMEC}(\mathcal{M}) \\ -\infty & \text{if} \quad (\mathcal{S}_i, \mathcal{A}_i) \notin \texttt{AMEC}(\mathcal{M}) \end{cases} \tag{23}$$

Note that, since all MECs are disjoint, we can use a single policy, denoted by μ_{stay}, as the optimal staying policy for each MEC.

Then we fuse policies $\hat{\mu}'_{k+1}$ and μ_{stay}, to obtain a new stationary policy $\hat{\mu}_{k+1}$ as follows: for each $s \in \hat{R}_{k+1}$, we have

$$\hat{\mu}_{k+1}(s) = \begin{cases} \hat{\mu}_{k+1}(s) & \text{if} \quad \begin{matrix} s \in T_k \text{ or} \\ s \in L_{k+1} \wedge \texttt{v}'(s) > \texttt{stay}(s) \end{matrix} \\ \mu_{\text{stay}}(s) & \text{if} \quad s \in L_{k+1} \wedge \texttt{stay}(s) \geq \texttt{v}'(s) \\ \hat{\mu}_k(s) & \text{if} \quad s \in \hat{R}_k \end{cases} \tag{24}$$

We can compute $\texttt{val}_{k+1}$ by selecting larger value between $\texttt{stay}$ and $\texttt{v}'$. Formally. we have

$$\texttt{val}_{k+1}(s) = \begin{cases} \texttt{v}'(s) & \text{if} \quad \begin{matrix} s \in T_k \text{ or} \\ s \in L_{k+1} \wedge \texttt{v}'(s) > \texttt{stay}(s) \end{matrix} \\ \texttt{stay}(s) & \text{if} \quad s \in L_{k+1} \wedge \texttt{stay}(s) \geq \texttt{v}'(s) \\ \texttt{val}_k(s) & \text{if} \quad s \in \hat{R}_k \end{cases} \tag{25}$$

The following result formally establishes that, the above fused policy $\hat{\mu}_{k+1}$ indeed solves Problems 1 for $\hat{\mathcal{M}}_{k+1} = (\hat{R}_{k+1}, \hat{A}_{k+1})$.

Proposition 2. Suppose that $\hat{\mu}_k$ is an optimal solution to Problem 1 for instant sub-MDP $\hat{\mathcal{M}}_k = (\hat{R}_k, \hat{A}_k)$. Then policy $\hat{\mu}_{k+1}$ as defined in Equation (24) is an optimal solution to Problem 1 for instant sub-MDP $\hat{\mathcal{M}}_{k+1} = (\hat{R}_{k+1}, \hat{A}_{k+1})$.

Based on Proposition 2, now we can solve Problem 1 for any MDP $\mathcal{M}$ by following steps:

- Find all MECs $\texttt{MEC}(\mathcal{M})$ and AMECs $\texttt{AMEC}(\mathcal{M})$;
- For each AMEC, compute the maximum entropy rate policy by the method in Section 4;
- Classify states into $L_0, T_0, \ldots, L_{\texttt{level}(\mathcal{M})}, T_{\texttt{level}(\mathcal{M})}$;
- Synthesize policy $\hat{\mu}_0$ by: for each state $s \in L_0$, if s is in some AMEC, then adopt the maximum entropy rate policy; otherwise assign an arbitrary action. Compute $\texttt{val}_0 = \texttt{stay}$ by Equation (23);
- For each $k = 0, \ldots \texttt{level}(\mathcal{M}) - 1$:
    - · Solve LP $(\hat{\mathcal{M}}_{k+1}, \hat{R}_k, \texttt{val}_k)$ and obtain $\gamma^*(s, a)$;
    - · Induce policy $\hat{\mu}'_{k+1}$ by Equations (20) and (21);
    - · Compute $\texttt{v}'$ by Equation (22);
    - · Fuse policy $\hat{\mu}_{k+1}$ by Equation (24) and compute $\texttt{val}_{k+1}$ by Equation (25).
- Then if $\texttt{val}_{\texttt{level}(\mathcal{M})}(s) = -\infty$ for some $s \in S$ such that $\pi_0(s) > 0$, there exists no solution for Problem 1 w.r.t $\mathcal{M}$. Otherwise $\hat{\mu}_{\texttt{level}(\mathcal{M})}$ solves Problem 1.

The correctness of the above procedure is summarized by the following result.

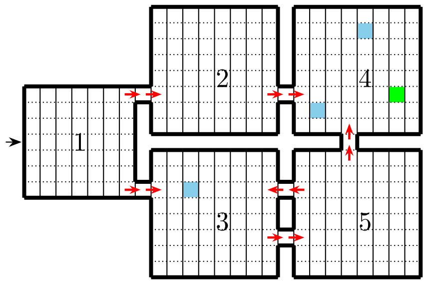


Fig. 2. Workspace of the robot.

Theorem 2. Given MDP $\mathcal{M} = (S, A, P)$, the procedure above generates a solution of Problem 1 for $\mathcal{M}$.

7. CASE STUDY

In this section we present a case study of surveillance robot to illustrate the proposed method. We use the splitting conic solver (SCS) Odonoghue et al. (2016) in CVXPY Diamond and Boyd (2016) to solve convex optimization problems.

Let us consider a robot moving in a workspace shown in Fig. 2. The entire workspace consists of five regions, where Region 1 consists of 7×7 grids and each of Regions 2-5 consists of 8×8 grids. The regions are connected by some one-way path grids whose feasible directions are depicted in the figure. The mobility of the robot is as follows. Inside of each region, the robot has five actions, left/right/up/down/stay. By choosing each action, the robot will move to the target grid w.p.1. Furthermore, if the robot choose an action but the target grid is a wall (the boundary of the region), then it will stay in the current grid. Between two regions, the robot can only move through the one-way path grids following the given direction. Therefore, the mobility of the robot can be modeled as an MDP $\mathcal{M}$ (in fact, deterministic) with 310 states and 1379 edges. Clearly, there are four MECs in $\mathcal{M}$, where Regions 3 and 5 belongs to the same MEC. The robot needs to visit some specific grid infinitely often to get resource, while making its behavior as unpredictable as possible.

Case 1 of Blue Tasks: In this case, we assume that blue grids in the workspace represent the task region needed to be visited infinitely often. For this case, there are two AMECs in $\mathcal{M}$: the Region 4, and the union of Regions 3 and 5. Our algorithm spends 143s and 626s to solve (6)-(11) for the first and the second AMEC, respectively. We denote by μ the solution of Problem 1 and by π^{μ} the limit distribution of $\mathcal{M}^{\mu}$. The value of π^{μ} is shown in Fig. 3, where the value of grid s is equal to $100 \cdot \pi^{\mu}(s)$. The robot will eventually stay in Regions 3 and 5 w.p.1.

Case 2 of Green Tasks: Now we consider the case, where the surveillance task is capture by the green grid in Region 4. Clearly, Region 4 becomes the only AMEC in $\mathcal{M}$, and the robot will eventually stay in Region 4 forever. We denote by μ' the solution to Problem 1 for this case and the limit distribution $\pi^{\mu'}$ is shown as Fig. 4(a); still, it suffices to show Region 4 only.

One may think that the maximum entropy rate policy is uniformly randomized, which is not the case. Specifically,

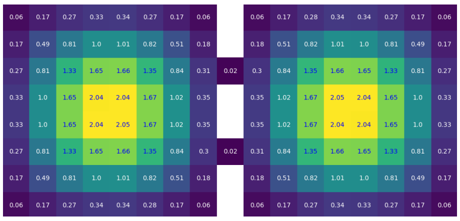
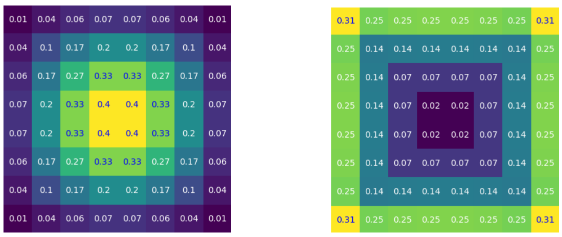


Fig. 3. Limit distribution (multiplied by 100) of the optimal policy $\mathcal{M}^{\mu}$ for the case of blue tasks.



(a) Limit distribution multiplied by 10.

(b) Largest difference of the probabilities of picking two different actions at each state.

Fig. 4. The optimal policy $\mathcal{M}^{\mu}$ for the case of green tasks.

for each state in Region 4, we compute the difference between highest probability of an action and the lowest probability of an action. This difference value is shown in Fig. 4(b). Clearly, only when the robot is at the center of the Region, it will follow a purely randomized strategy. For the remaining states, the optimal policy maximizing the entropy rate is not uniformly randomized.

In the context of information-theoretical foundation of security, a useful measure for quantifying the unpredictability of an agent is to use the the weight of the Huffman tree of the distribution; see, e.g., Paruchuri et al. (2006). If we adopt the algorithm in Savas et al. (2019) to synthesize a policy μ_1 that maximizes the total entropy of MDP and finish the task w.p.1, then the corresponding value is $O_a^{\mu_1} = 1$ for both blue tasks and green tasks since μ_1 will choose a deterministic action in the steady state. However, for our algorithm, we have $O_a^{\mu} = 2.56$ for blue tasks and $O_a^{\mu'} = 2.55$ for green tasks. Therefore, method proposed makes the limit behavior of agent more unpredictable.

8. CONCLUSION

In this paper, we solved a new entropy rate maximization problem for MDPs under the requirement that some specific region of interest needs to be visited infinitely often. We showed that this problem can be effectively solved by decomposing it as a finite set of sub-problems. Our results extended the existing result in entropy rate maximization by taking logic constraint into account. Note that, although we only consider the case of visiting state, our result is equivalent to achieve an omega-regular objective accepted by Büchi conditions. We demonstrated the proposed algorithm by a case study of robot task planning. In the future, we plan to investigate the trade-off between the entropy rate and the satisfaction probability of the task.

REFERENCES

Baier, C. and Katoen, J.P. (2008). *Principles of model checking*. MIT press.

Chen, T. and Han, T. (2014). On the Complexity of Computing Maximum Entropy for Markovian Models. In *34th International Conference on Foundation of Software Technology and Theoretical Computer Science*, volume 29, 571–583.

Chen, Y., Yang, S., Mangharam, R., and Yin, X. (2023). You don't know when i will arrive: Unpredictable controller synthesis for temporal logic tasks. In *22nd IFAC World Congress*.

De Alfaro, L. (1998). *Formal verification of probabilistic systems*. stanford university.

Diamond, S. and Boyd, S. (2016). CVXPY: A Python-embedded modeling language for convex optimization. *The J. Machine Learning Research*, 17(1), 2909–2913.

Ding, X., Smith, S.L., Belta, C., and Rus, D. (2014). Optimal control of Markov decision processes with linear temporal logic constraints. *IEEE Trans. Automatic Control*, 59(5), 1244–1257.

Duan, X. and Bullo, F. (2021). Markov chain–based stochastic strategies for robotic surveillance. *Annual Rev. Control, Rob. Autonomous Systems*, 4, 243–264.

George, M., Jafarpour, S., and Bullo, F. (2018). Markov chains with maximum entropy for robotic surveillance. *IEEE Trans. Automatic Control*, 64(4), 1566–1580.

Li, N., Kolmanovsky, I., and Girard, A. (2020). Detection-averse optimal and receding-horizon control for Markov decision processes. *Automatica*, 122, 109278.

Liu, S., Trivedi, A., Yin, X., and Zamani, M. (2022). Secure-by-construction synthesis of cyber-physical systems. *Annual Reviews in Control*.

Odonoghue, B., Chu, E., Parikh, N., and Boyd, S. (2016). Conic optimization via operator splitting and homogeneous self-dual embedding. *J. Optimization Theory and Applications*, 169(3), 1042–1068.

Paruchuri, P., Tambe, M., Ordónez, F., and Kraus, S. (2006). Security in multiagent systems by policy randomization. In *International Joint Conf. Autonomous Agents and Multiagent Systems*, 273–280.

Puterman, M.L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., USA, 1st edition.

Savas, Y., Ornik, M., Cubuktepe, M., Karabag, M.O., and Topcu, U. (2019). Entropy maximization for Markov decision processes under temporal logic constraints. *IEEE Trans. Automatic Control*, 65(4), 1552–1567.

Smith, S.L., Tůmová, J., Belta, C., and Rus, D. (2011). Optimal path planning for surveillance with temporal-logic constraints. *The International Journal of Robotics Research*, 30(14), 1695–1708.

Xie, Y., Yin, X., Li, S., and Zamani, M. (2021). Secure-by-construction controller synthesis for stochastic systems under linear temporal logic specifications. In *IEEE Conference on Decision and Control*, 7015–7021.

Yang, S. and Yin, X. (2022). Secure your intention: On notions of pre-opacity in discrete-event systems. *IEEE Trans. Automatic Control*.

Yu, X., Yin, X., Li, S., and Li, Z. (2022). Security-preserving multi-agent coordination for complex temporal logic tasks. *Control Engineering Practice*, 123, 105130.

Zheng, W., Jung, T., and Lin, H. (2022). Privacy-Preserving POMDP Planning via Belief Manipulation. *IEEE Control Systems Letters*, 6, 3415–3420.