Entropy Rate Maximization of Markov Decision Processes under Linear Temporal Logic Tasks

Yu Chen, Shaoyuan Li and Xiang Yin

Abstract—We investigate the problem of synthesizing optimal control policies for Markov decision processes (MDPs) with both qualitative and quantitative objectives. Specifically, our goal is to achieve a given linear temporal logic (LTL) task with probability one, while maximizing the entropy rate of the system. The notion of entropy rate characterizes the long-run average (un)predictability of a stochastic process. Such an optimal policy is of our interest, in particular, from the security point of view, as it not only ensures the completion of tasks, but also maximizes the unpredictability of the system. However, existing works only focus on maximizing the total entropy which may diverge to infinity for infinite horizon. In this paper, we provide a complete solution to the entropy rate maximization problem under LTL constraints. Specifically, we first 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 LTL task can be effectively solved in polynomial-time. We illustrate the proposed algorithm based on two case studies of robot task planning scenario.

Index Terms—Markov Decision Process, Entropy Rate, Linear Temporal Logic, Security.

I. INTRODUCTION

A. Motivations

ASK planning and decision-making are central problems in autonomous systems. For example, a field mobile robot needs to navigate in intricate and uncertain environments while achieving complex tasks associated with its spatiotemporal behavior. Markov Decision Processes (MDPs) provide a foundational mathematical framework for decision-making under uncertainty in this context. By abstracting system dynamics as well as uncertainties as transition probabilities, MDPs offer a suitable tool for modeling and optimizing the interactions between the decision-maker and the environment, enabling adaptability and efficient response to ever-changing conditions.

In the context of MDPs, numerous works have focused on synthesizing optimal control policies that maximize or minimize various quantitative performance metrics, such as total reward, long-run average reward (or mean payoff), and discounted total reward [30]. More recently, motivated by the growing needs in decision-making for complex requirements in autonomous systems, optimal control for MDPs with high-level tasks described by temporal logic formulae has also drawn considerable attention in the literature; see, e.g., the

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

Yu Chen, Shaoyuan Li and Xiang Yin are with Department of Automation and Key Laboratory of System Control and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China. {yuchen26, syli, yinxiang}@sjtu.edu.cn. (Corresponding author: Xiang Yin)

recent survey papers [2], [22], [39]. Among many formal specification languages, linear temporal logic (LTL) [1] offers a rich and user-friendly way for designers to describe desired high-level specifications, such as "visiting region A infinitely often while never reaching region B". For example, when the structural information of MDP models is known, algorithms have been developed to synthesize optimal policies achieving LTL tasks with probabilistic guarantees [14], [17], [27]. When transition probabilities in MDPs are unknown a priori, reinforcement learning techniques have also been used to learn optimal policies for LTL tasks [6], [18], [36].

Despite the randomized nature of MDPs, the control policy applied can also be randomized, i.e., at each time instant, the decision-maker will choose a specific action according to some probability distribution. To accomplish an LTL task, it is known that considering deterministic policies is already sufficient [1]. Yet, using randomized policies still have many advantages beyond satisfaction consideration. One important reason is the security consideration in adversarial environments. For example, when the transition probability of each action in an MDP is one, applying a deterministic policy results in a purely deterministic trajectory. In this case, a malicious attacker could easily hack the information of the system or even physically destroy the agent by predicting its trajectory. Therefore, synthesizing randomized policies that not only achieve the desired task but also make the system's behavior as *unpredictable* as possible is of great importance and has drawn considerable attention from researchers recently [15], [19], [24], [40]. For example, several different measures for unpredictability have been used such as pre-opacity [12], [38], Bayes risk [9] and entropy [25], [29].

B. Our Results

In this work, we investigate the problem of unpredictable control policy synthesis under LTL task constraints for stochastic systems modeled as finite MDPs. Among many 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 random variable or a stochastic process is [35]. Here, we adopt the notion of *entropy rate* as a measure of the unpredictability for the infinite horizon behavior of the system. Specifically, the entropy rate characterizes the long-run average total entropy of a stochastic process; hence, a higher entropy rate implies greater unpredictability of the stochastic process. Our objective is to synthesize an optimal policy in the sense that (i) the given LTL task is satisfied with probability one (w.p.1); and (ii) the entropy rate of the induced stochastic process is maximized.

Our approach for solving the entropy rate maximization problem under LTL constraints consists of two stages. First, we restrict our attention to the special case of communicating MDPs. For this case, we reveal a new structural property that shows the optimal policy is stationary and its resulting Markov chain is irreducible. Based on this structural property, a convex program is formulated to find the optimal policy. Then, we tackle the general case where the MDPs may not be communicating. To this end, we introduce a novel state classification technique that characterizes system states into different levels. The overall synthesis algorithm iteratively solves convex programs for communicating MDPs within each level as well as a new linear program that determines how high-level states transition to lower levels. We prove that our synthesis algorithm is both sound and complete. Furthermore, its computational complexity is polynomial with respect to the size of the MDP. Moreover, we have implemented our algorithm, and two case studies on robot task planning are provided to demonstrate the effectiveness of our approach.

C. Related Works

In the past years, formal control synthesis for MDPs under LTL tasks has drawn much attention in the literature. For example, in [8], [17], the authors investigate how to synthesize optimal policies that achieve LTL tasks while maximizing the long-run average reward (or mean payoff). In [14], the authors further consider the optimal control problem for a new metric called average reward per cycle. In [37], the authors consider the robust control problem for uncertain MDPs. However, these metrics only characterize the optimality of the policy rather than the unpredictability. Therefore, the optimization problems considered therein are very different from the entropy rate optimization problem.

In the context of unpredictable policy synthesis, [3] first shows how to maximize the total entropy of MDPs. This result is further extended to the partial observation setting in [32]. However, in general, the total entropy of a stochastic process diverges unless the process becomes deterministic eventually. Therefore, for MDPs with infinite horizon, recent works have considered the maximization of entropy rate [5], [10], [16], which is the long-run average of the total entropy. For example, [10] computes the value of the maximum entropy rate for MDPs without any task constraint. In [20], the authors consider the maximization of entropy rate under moments constraints. In [16], the authors further consider stationary distributions as constraints in addition to entropy rate maximization. However, none of the above-mentioned works consider the temporal logic requirements when maximizing the unpredictability of the MDP. Moreover, [5], [16], [20] only consider communicating MDPs, which is a special case of the general MDPs we consider in this work.

Our work is mostly related to [33], which solves the problem of maximizing the total entropy under LTL constraints. However, as we mentioned, the total entropy generally goes to infinity for systems operating over infinite horizon. Therefore, the approach in [33] essentially yields a solution in which the steady state of the system is deterministic, which is somewhat restrictive. That is, only unpredictability in the transient part is taken into account, and the steady part is, in fact, completely predictable. In fact, since LTL formulae are evaluated over infinite traces, it seems more natural to consider entropy rate rather than total entropy as the metric for unpredictability when dealing with LTL tasks.

Finally, we note that entropy maximization has also been widely adopted in the context of reinforcement learning as a regularization term in the objective function; see, e.g., [7], [23], [26]. Specifically, entropy maximization can be used to promote exploration and accelerate the convergence of the learning agent. However, our purpose for entropy rate maximization here is to enhance the unpredictability of the agent's behavior. Moreover, these works consider model-free learning problems, while here we consider a model-based optimal control problem.

D. Organization

The remaining parts of the paper are organized as follows. We first introduce some necessary preliminaries in Section II and then formulate the problem of entropy rate maximization under LTL constraints in Section III. In Section IV, we solve the problem by focusing on a special class of MDPs called communicating MDPs. Next, in Section V, we introduce the technique of state-level classification and show how to leverage this technique to solve the problem for the general case in Section VI. Two case studies on robot task planning are provided in Section VII to illustrate our results. Finally, we conclude the paper in Section VIII.

The preliminary version of some results in this paper is presented in [11]. Compared with [11], the present work has the following differences. First and foremost, this paper considers the general LTL tasks while [11] only considers the so called surveillance tasks. Furthermore, this paper contains detailed proof compared to the sketchy analysis in [11]. Finally, additional experimental results are provided to illustrate the effectiveness of our approach.

II. PRELIMINARY

A. Markov Decision Processes

A (finite and labeled) Markov decision process (MDP) is a 6-tuple

$$\mathcal{M} = (S, s_0, A, P, \mathcal{AP}, \ell), \tag{1}$$

where $S = \{1, \dots, n\}$ is a finite set of states, $s_0 \in S$ is the initial state, A is a finite set of actions, $P: S \times A \times S \to [0,1]$ is a transition function such that: for any $s \in S, a \in A$, we have $\sum_{s' \in S} P(s' \mid s, a) \in \{0,1\}$, \mathcal{AP} is a set of atomic propositions, and $\ell: S \to 2^{\mathcal{AP}}$ is a labeling function that assigns each state a set of atomic propositions. For simplicity, we also write $P(s' \mid s, a)$ as $P_{s,a,s'}$. We denote by $\operatorname{SUCC}(s,a) = \{s' \mid P(s' \mid s,a) > 0\}$ the set of successor states of state $s \in S$ under action $a \in A$. For each state $s \in S$, we denote by $A(s) = \{a \in A \mid \operatorname{SUCC}(s,a) \neq \emptyset\}$ the set of available actions at s. We assume that, for each state, there exists at least one available action, 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 an edge of form $\langle s, s' \rangle$ is defined whenever $P(s' \mid s, a) > 0$ for some $a \in A$. We denote by π_0 the *initial distribution* such that $\pi_0(s) = 1$ if $s = s_0$ is the initial state and $\pi_0(s) = 0$ otherwise.

A Markov chain (MC) $\mathcal C$ is an MDP such that |A(s)|=1 for all $s\in S$. We denote by $\mathbb P$ the *transition matrix* of an MC, i.e., $\mathbb P_{s,s'}=P(s'\mid s,a)$, where $a\in A(s)$ is the unique action at state $s\in S$. Therefore, we can omit action set and write an MC as $\mathcal C=(S,s_0,\mathbb P)$. The *limit transition matrix* of an MC is defined by $\mathbb P^\star=\lim_{n\to\infty}\frac{1}{n}\sum_{k=0}^n\mathbb P^k$. Note that this limit matrix always exists for any finite MC [30].

A policy for an MDP \mathcal{M} is a sequence $\mu=(\mu_0,\mu_1,\ldots)$, where each $\mu_k:S\times A\to [0,1]$ is a function such that $\forall s\in S:\sum_{a\in A(s)}\mu_k(s,a)=1$. 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 $\Pi_{\mathcal{M}}^S$, respectively. For policy $\mu\in\Pi_{\mathcal{M}}$, at time k,μ induces a transition matrix \mathbb{P}^{μ_k} such that $\mathbb{P}^{\mu_k}_{i,j}=\sum_{a\in A(i)}\mu_k(i,a)P_{i,a,j}$. We denote by reach(s) the set of all states reachable from state $s\in S$ in MDP, i.e.,

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

Let $\mathcal{M}=(S,s_0,A,P,\mathcal{AP},\ell)$ be an MDP and $\mu\in\Pi_{\mathcal{M}}$ be a policy. An infinite sequence $\rho=s_0s_1\cdots$ of states is said to be a path in \mathcal{M} under μ if (i) s_0 is the initial state of the MDP, and (ii) $\forall k\geq 0: \sum_{a\in A(s_k)}\mu_k(s_k,a)P(s_{k+1}\mid s_k,a)>0.$ We denote by $\operatorname{Path}^{\mu}(\mathcal{M})\subseteq S^{\omega}$ the set of all paths in \mathcal{M} under μ , where S^{ω} denotes the set of all infinite sequences of symbols over set S. We use the standard probability measure $\operatorname{Pr}^{\mu}_{\mathcal{M}}: 2^{S^{\omega}} \to [0,1]$ for the sample space over infinite paths, which satisfies: for any finite sequence $s_0\cdots s_n$, we have

$$\mathsf{Pr}^{\mu}_{\mathcal{M}}(\mathsf{Cly}(s_0 \dots s_n)) = \pi_0(s_0) \prod_{k=0}^{n-1} \sum_{a \in A(s_k)} \mu_k(s_k, a) P_{s_k, a, s_{k+1}},$$

where $\mathsf{Cly}(s_0 \dots s_n) \subseteq \mathsf{Path}^{\mu}(\mathcal{M})$ is the *cylinder* set, which is the set of all paths having prefix $s_0 \dots s_n$. The reader is referred to [1] for details on this probability measure on infinite paths.

Given an MDP, let (S, A) be a state-action pair, where $S \subseteq S$ is a non-empty set of states and $A: S \to 2^A \setminus \emptyset$ is a function such that (i) $\forall s \in S: A(s) \subseteq A(s)$; and (ii) $\forall s \in S, a \in A(s): SUCC(s, a) \subseteq S$. Essentially, state-action pair (S, A) induces a new MDP called the *sub-MDP* of A, denoted by A (A) (or A) directly), by restricting the state space to A and available actions to A(A) for each state A0.

Definition 1 (Maximal End Components). Let (S, A) be a sub-MDP of $\mathcal{M} = (S, s_0, A, P, \mathcal{AP}, \ell)$. We say (S, \mathcal{A}) is an *end component* if its underlying digraph is strongly connected. We say (S, \mathcal{A}) is a *maximal end component* (MEC) if it is an end component and there is no other end component (S', \mathcal{A}') such that (i) $S \subseteq S'$; and (ii) $\forall s \in S, \mathcal{A}(s) \subseteq \mathcal{A}'(s)$. We denote by $\text{MEC}(\mathcal{M})$ the set of all MECs in \mathcal{M} .

Intuitively, if (S, A) is an MEC, then we can find a policy such that, once a state in S is reached, we will stay in the

MEC forever and all states in it will be visited infinitely w.p.1 thereafter.

B. Entropy Rate of Stochastic Processes

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

$$H(X) := -\sum_{x \in \mathcal{X}} p(x) \log p(x). \tag{2}$$

All logarithms in this work are with base 2 and 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

$$H(X_0, X_1) := -\sum_{x_0 \in \mathcal{X}} \sum_{x_1 \in \mathcal{X}} p(x_0, x_1) \log p(x_0, x_1).$$
 (3)

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 diverges when n goes to infinity. Therefore, for infinite processes, one usually uses the *entropy rate* instead of the joint entropy.

Definition 2 (Entropy Rate [35]). The entropy rate of a discrete-time stochastic process $\{X_k\}$ is defined as

$$\nabla H(\lbrace X_k \rbrace) := \lim_{k \to \infty} \frac{1}{k} H(X_0, \dots, X_k) \tag{4}$$

when the limit exists.

Given an MC $\mathcal{C} = (S, s_0, \mathbb{P})$, it also induces a discretetime stochastic process $\{X_k : k \in \mathbb{N}\}$, 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 is known that this entropy rate can be computed by [10]:

$$\nabla H(\mathcal{C}) = \sum_{s \in S} \pi(s) L(s), \tag{5}$$

where $\pi = \pi_0 \mathbb{P}^*$ is the limit distribution, and L(s) is the so called *local entropy* defined by

$$L(s) = \sum_{s' \in S} -\mathbb{P}_{s,s'} \log \mathbb{P}_{s,s'}. \tag{6}$$

For MDP \mathcal{M} , we only consider polices under which (4) is well-defined and directly denote $\Pi_{\mathcal{M}}$ as such policies set. We define $\nabla H(\mathcal{M}) := \sup_{\mu \in \Pi_{\mathcal{M}}} \nabla H(\mathcal{M}^{\mu})$ as its entropy rate.

C. Linear Temporal Logic

We employ Linear Temporal Logic (LTL) to express formal tasks. Let \mathcal{AP} be the set of atomic propositions. An LTL formula is constructed based on atomic propositions, Boolean operators and temporal operators. Specifically, the syntax of LTL formulae is defined recursively as follows:

$$\varphi ::= true \mid a \mid \varphi_1 \wedge \varphi_2 \mid \neg \varphi \mid \bigcirc \varphi \mid \varphi_1 U \varphi_2,$$

where $a \in \mathcal{AP}$ is an atomic proposition; \neg and \land are Boolean operators "negation" and "conjunction", respectively; \bigcirc and U are temporal operators "next" and "until", respectively.

Note that one can further induce temporal operators such as "eventually" $\Diamond \varphi := trueU\varphi$ and "always" $\Box \varphi := \neg \Diamond \neg \varphi$.

An LTL formula φ is interpreted over infinite words on $2^{\mathcal{AP}}$. Readers can find detailed information about the semantics of LTL formulae in [1]. For any infinite word $\sigma \in (2^{\mathcal{AP}})^{\omega}$, we denote by $\sigma \models \varphi$ if it satisfies LTL formula φ . We denote by $\mathcal{L}_{\varphi} = \{\sigma \in (2^{\mathcal{AP}})^{\omega} \mid \sigma \models \varphi\}$ the set of all infinite words satisfying φ .

Definition 3 (**Deterministic Rabin Automata**). A deterministic Rabin automata (DRA) is a tuple $R = (Q, \Sigma, \delta, q_0, Acc)$, where Q is a finite set of states, Σ is a finite set of alphabet, $\delta: Q \times \Sigma \to Q$ is the transition function, $q_0 \in Q$ is the initial state, and $Acc = \{(B_1, G_1), \ldots, (B_n, G_n)\}$ is a finite set of Rabin pairs such that $B_i, G_i \subseteq Q$ for all $i = 1, 2, \ldots, n$.

Given an infinite word $\sigma = \sigma_1 \sigma_2 \cdots \in \Sigma^\omega$, its induced infinite run in DRA R is the sequence of states $\rho = q_0 q_1 \cdots \in Q^\omega$ such that $q_i = \delta(q_{i-1}, \sigma_i)$ for all $i \geq 1$. An infinite run $\rho \in Q^\omega$ is said to be accepted if there exists a Rabin pair $(B_i, G_i) \in \operatorname{Acc}$ such that $\inf(\rho) \cap G_i \neq \emptyset$ and $\inf(\rho) \cap B_i = \emptyset$, where $\inf(\rho)$ is set of states that occur infinitely many times in ρ . An infinite word σ is said to be $\operatorname{accepted}$ if its induced infinite run is accepted. We denote by $\mathcal{L}(R) \subseteq \Sigma^\omega$ the set of all accepted words of DRA R. Given an arbitrary LTL formula φ over \mathcal{AP} , it is well-known that [1], there exists a DRA with $\Sigma = 2^{\mathcal{AP}}$ that accepts all infinite words satisfying φ , i.e., $\mathcal{L}_\varphi = \mathcal{L}(R)$.

Given an MDP \mathcal{M} under policy μ , a path $\rho = s_0 s_1 \cdots \in \mathsf{Path}^{\mu}(\mathcal{M})$ generates a word $\ell(\rho) = \ell(s_0)\ell(s_1) \cdots \in (2^{\mathcal{AP}})^{\omega}$. Given an LTL formula φ , we define

$$\mathsf{Pr}^{\mu}_{\mathcal{M}}(s_0 \models \varphi) := \mathsf{Pr}^{\mu}_{\mathcal{M}}(\{\rho \in \mathsf{Path}^{\mu}(\mathcal{M}) \mid \ell(\rho) \models \varphi\})$$

as the probability of satisfying LTL formula φ for MDP \mathcal{M} starting from s_0 under policy $\mu \in \Pi_{\mathcal{M}}$. We denote by $\Pi_{\mathcal{M}}^{\varphi}$ as the set of policies under which the LTL task can be satisfied with probability 1, i.e.,

$$\Pi_{\mathcal{M}}^{\varphi} = \{ \mu \in \Pi_{\mathcal{M}} \mid \mathsf{Pr}_{\mathcal{M}}^{\mu}(s_0 \models \varphi) = 1 \}.$$

D. Product MDPs

To integrate the task information into the MDP model, it is necessary to construct the product system between the DRA representing the LTL task and the original MDP.

Definition 4 (Product MDPs). Let $\mathcal{M}=(S,s_0,A,P,\mathcal{AP},\ell)$ be an MDP and $R=(Q,2^{\mathcal{AP}},\delta,q_0,Acc)$ be the DRA accepting LTL formula φ . The *product MDP* $\mathcal{M}_{\otimes}=(S_{\otimes},s_{0,\otimes},A,P_{\otimes},\mathcal{AP},\ell_{\otimes},Acc_{\otimes})$ is a 7-tuples, where $S_{\otimes}=S\times Q$ is the product space, $s_{0,\otimes}=(s_0,q)$ is the initial state such that $q=\delta(q_0,\ell(s_0)),\,P_{\otimes}:S_{\otimes}\times A\times S_{\otimes}\to[0,1]$ is transition function defined by

$$P_{\otimes}((s,q),a,(s',q')) = \begin{cases} P_{s,a,s'} & \text{if } q' = \delta(q,\ell(s')) \\ 0 & \text{otherwise} \end{cases}, \quad (7)$$

 ℓ_{\otimes} is the labeling function such that $\ell_{\otimes}((s,q)) = \ell(s)$ and $Acc_{\otimes} = \{(B_1^{\otimes}, G_1^{\otimes}), \dots, (B_n^{\otimes}, G_n^{\otimes})\}$ such that $B_i^{\otimes} = S \times B_i$ and $G_i^{\otimes} = S \times G_i$ for all $i = 1, \dots n$.

Note that, since R is deterministic, there exists a one-to-one correspondence between paths in \mathcal{M} and \mathcal{M}_{\otimes} [1]. Specifically, let $s=s_0\dots s_n$ be a path executed in \mathcal{M} . Then there exists a unique path $s'=s'_0\dots s'_n$ in \mathcal{M}_{\otimes} , where $s'_i=(s_i,q_i)$, such that $s'_0=s_{0,\otimes}$ and $q_i=\delta(q_{i-1},\ell(s_i)), i\geq 1$. Since the action spaces of \mathcal{M} and \mathcal{M}_{\otimes} are same, there also exists a one-to-one correspondence between policies in \mathcal{M} and \mathcal{M}_{\otimes} [17].

For the sake of simplicity, hereafter in this paper, we will omit the subscript and directly denote by $\mathcal{M}=(S,s_0,A,P,\mathcal{AP},\ell,Acc)$ the product MDP. The control synthesis problem is solved based on the product MDP. Specifically, for any path in (product) MDP $\rho=s_0s_1\ldots$, it satisfies the LTL formula if and only if there exists an accepting pair $(B_k,G_k)\in Acc$ such that $\inf(\rho)\cap G_k\neq\emptyset$ and $\inf(\rho)\cap B_k=\emptyset$. This accepting condition can be captured by the notion of accepting maximal end component.

Definition 5 (Accepting Maximal End Components). Given a (product) MDP $\mathcal{M}=(S,s_0,A,P,\mathcal{AP},\ell,Acc)$, an accepting end component (AEC) of product MDP is an end component $(\mathcal{S},\mathcal{A})$ such that for some accepting pair $(B_k,G_k)\in Acc$, we have $\mathcal{S}\cap B_k=\emptyset$ and $\mathcal{S}\cap G_k\neq\emptyset$. Moreover $(\mathcal{S},\mathcal{A})$ is said to be an accepting maximal end component (AMEC) if there exists no other AEC $(\mathcal{S}',\mathcal{A}')$ such that (i) $\mathcal{S}\subseteq\mathcal{S}'$; and (ii) $\forall s\in\mathcal{S},\mathcal{A}(s)\subseteq\mathcal{A}'(s)$. We denote by $\mathsf{AEC}(\mathcal{M})$ and $\mathsf{AMEC}(\mathcal{M})$ the set of AECs and AMECs of product MDP \mathcal{M} , respectively.

Intuitively, given policy $\mu \in \Pi_{\mathcal{M}}$, the probability of satisfying a given LTL formula is equal to the probability of reaching AMEC and staying there forever. Note that both the set of all MECs and the set of all AMECs can be found effectively via graph search over the product space; see, e.g., [1], [17]. Although states in different MECs are always disjoint, this is not true for different AMECs. This is because different AMECs may be accepted by different accepting pairs and their union may violate both acceptance conditions.

III. PROBLEM FORMULATION

Now, we are ready to formulate the problem that we solve in this paper. Our objective is to synthesize a control policy such that

- 1) The given LTL task is satisfied with probability one; and
- The behavior of the agent needs to be as unpredictable as possible in the sense that the entropy rate of its induced stochastic process is maximized.

This problem is formally stated as follow.

Problem 1 (Entropy Rate Maximization for Linear Temporal Logic Tasks). Given MDP \mathcal{M} and LTL formula φ , which is equivalent to given the product MDP, find an optimal policy $\mu^* \in \Pi^{\varphi}_{\mathcal{M}}$ such that

$$\nabla H(\mathcal{M}^{\mu^{\star}}) = \nabla H_{\varphi}(\mathcal{M}),$$

where
$$\nabla H_{\varphi}(\mathcal{M}) = \sup_{\mu \in \Pi_{\mathcal{M}}^{\varphi}} \nabla H(\mathcal{M}^{\mu}).$$

Without loss of generality, we assume that, starting from each state in the product MDP, there exists a policy such that the LTL task can be satisfied w.p.1. Otherwise, we can use Algorithm 45 in [1] to eliminate such undesired states in polynomial time.

The following result reveals a key structural property of Problem 1, which shows that focusing only on stationary policies is sufficient.

Proposition 1. Let $\mathcal{M} = (S, s_0, A, P, \mathcal{AP}, \ell, Acc)$ be a product MDP. Then we have

$$\nabla H_{\varphi}(\mathcal{M}) = \sup_{\mu \in \Pi_{\mathcal{M}}^{\varphi} \cap \Pi_{\mathcal{M}}^{S}} \nabla H(\mathcal{M}^{\mu}).$$

Proof. The proof is provided in the Appendix.

IV. SOLUTION FOR COMMUNICATING MDP

In this section, instead of tackling the general case, we focus on solving Problem 1 for *communicating MDPs*, i.e., MDPs in which all states can visit each other under some policy. Formally, an MDP \mathcal{M} is said to be *communicating* if

$$\forall s, s' \in S, \exists \mu \in \Pi^S_{\mathcal{M}}, \exists n \ge 0 : (\mathbb{P}^{\mu})_{s,s'}^n > 0.$$

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 \mathcal{M}^{μ} is *irreducible*, i.e., each pair of states can be visited from one to the other via some path.

Note that, without considering the LTL task, the computation of the entropy rate for a general MDP has been addressed in [10]. However, the approach in [10] does not directly yield a policy to achieve this value. Here, instead of considering the general MDPs, we first focus only on communicating MDPs, and show that, for this class of MDPs, the stationary policy achieving entropy rate maximization can be synthesized more efficiently based on a different nonlinear program.

First, we show the following structural property for communicating MDPs. It reveals that, for a communicating MDP, if a stationary policy maximizes the entropy rate, then its induced MC must be also irreducible.

Lemma 1. Let \mathcal{M} be a communicating MDP. From [10] there exists stationary policy $\mu \in \Pi^S_{\mathcal{M}}$ such that $\nabla H(\mathcal{M}) = \nabla H(\mathcal{M}^{\mu})$. Then the induced MC \mathcal{M}^{μ} is irreducible.

Proof. The proof is provided in the Appendix.
$$\Box$$

Recall that, if an MC is irreducible, then the limit distribution is initial-distribution independent [30]. Therefore, the above structural property implies that $\nabla H(\mathcal{M})$ is also initial-distribution independent. Note that a sub-MDP $(\mathcal{S}, \mathcal{A}) \in \text{MEC}(\mathcal{M})$ is always communicating. In rest of paper, we will write the maximum entropy rate of sub-MDP $(\mathcal{S}, \mathcal{A})$ by $\nabla H(\mathcal{S}, \mathcal{A})$ directly by ignoring the initial distribution.

 the induced MC. Variables q(s,t) and $\lambda(s)$ in Equations (9) and (10) are functions of $\gamma(s,a)$, representing the probability of going from states s to t and the probability of occupying state s, respectively. Equations (11) and (12) are constraints for stationary distribution that the decision variables should satisfy. Finally, the objective function is determined according to the computation of entropy rate given in Equation (5).

Nonlinear Program for Communicating MDP

$$\max_{\gamma(s,a)} \quad \sum_{s \in S} \sum_{t \in S} -q(s,t) \log \left(\frac{q(s,t)}{\lambda(s)} \right) \tag{8}$$

s.t.
$$q(s,t) = \sum_{a \in A(s)} \gamma(s,a) P(t \mid s,a), \forall s,t \in S$$
 (9)

$$\lambda(s) = \sum_{a \in A(s)} \gamma(s, a), \forall s \in S$$
 (10)

$$\lambda(t) = \sum_{s \in S} q(s, t), \forall t \in S$$
 (11)

$$\sum_{s \in S} \lambda(s) = 1 \tag{12}$$

$$\gamma(s, a) \ge 0, \forall s \in S, \forall a \in A(s)$$
(13)

Now, given a communicating MDP \mathcal{M} , let $\gamma^*(s, a)$ be the solution to Equations (8)-(13). Then the maximum entropy rate is achieved by the stationary policy defined as

$$\mu^{\star}(s,a) = \frac{\gamma^{\star}(s,a)}{\sum_{a \in A(s)} \gamma^{\star}(s,a)}, \quad \forall s \in S, \forall a \in A(s). \quad (14)$$

The optimality of this policy is intuitive according to the meaning of the nonlinear program (8)-(13). First, each $\mu \in \Pi^S_{\mathcal{M}}$ corresponds to a feasible solution of the program. Then according to Lemma 1, for a communicating MDP, the maximum entropy rate policy must induce an irreducible MC for which a stationary distribution exists. This guarantees that among all feasible solutions, the one that maximizes the entropy rate is optimal for any initial distribution. Therefore, Equation (14) simply decodes a policy that achieves the best stationary distribution of the desired MC.

The above intuition is formally proved as follows.

Theorem 1. Let \mathcal{M} be a communicating MDP. Then for policy $\mu^* \in \Pi^S_{\mathcal{M}}$ defined by (14), we have $\nabla H(\mathcal{M}^{\mu^*}) = \nabla H(\mathcal{M})$.

Proof. By Proposition 1 we only need to prove that μ^* achieves maximum entropy rate over stationary policies. Let $\mu \in \Pi^S_{\mathcal{M}}$ be an arbitrary policy and let π^μ be the limit distribution under μ . We define $\gamma(s,a) = \pi^\mu(s)\mu(s,a)$, $\lambda(s) = \pi^\mu(s)$ and $q(s,t) = \pi^\mu(s)\mathbb{P}^\mu_{s,t}$. Clearly, these variables satisfy constraints (9), (10), (12) and (13). Since $\pi^\mu\mathbb{P}^\mu = \pi^\mu$, we know that constraint (11) is also satisfied. Hence, $\{\gamma(s,a)\}_{s\in S,a\in A(s)}$ is a feasible solution to program (8)-(13). Moreover, we have

$$\begin{split} \nabla H(\mathcal{M}^{\mu}) &= \sum_{s \in S} \pi^{\mu}(s) L^{\mu}(s) = \sum_{s \in S} \sum_{t \in S} -\pi^{\mu}(s) \mathbb{P}^{\mu}_{s,t} \log(\mathbb{P}^{\mu}_{s,t}) \\ &= \sum_{s \in S} \sum_{t \in S} -q(s,t) \log\left(\frac{q(s,t)}{\lambda(s)}\right), \end{split}$$

i.e., the entropy rate of MC \mathcal{M}^{μ} is equal to the value of objective function (8). Since no restriction on initial distribution is made, the optimal value of (8)-(13), denoted by V^* , satisfies that for any initial distribution, $V^* \geq \nabla H(\mathcal{M})$.

On the other hand, for any $\gamma(s,a)$ satisfying constraints (9)-(13), we construct a policy μ as follows:

$$\mu(s,a) = \frac{\gamma(s,a)}{\sum_{a \in A(s)} \gamma(s,a)} \text{ for } s \in Rc,$$
 (15)

where $Rc = \{s \in S \mid \sum_{a \in A(s)} \gamma(s,a) > 0\}$ and $\mu(s,a)$ is assigned arbitrarily for $s \in S \setminus Rc$. By Proposition 9.3.2 of [30], we know that Rc consists of K recurrent classes in MC \mathcal{M}^{μ} . For each $k = 1, \ldots, K$, we denote by \mathbb{P}^k the submatrix of \mathbb{P}^{μ} restricted on recurrent class R_k . For any states $s,t \in R_k$, we have

$$\mathbb{P}_{s,t}^{k} = \sum_{a \in A(s)} \frac{\gamma(s,a)}{\sum_{a' \in A(s)} \gamma(s,a')} P(t \mid s,a) = \frac{q(s,t)}{\lambda(s)}. \quad (16)$$

Then the limit distribution of R_k is $(\lambda(s)/\zeta_k)_{s\in R_k}$, where $\zeta_k = \sum_{s\in R_k} \lambda(s)$, since for $t\in R_k$,

$$\lambda(t) = \sum_{s \in R_k} q(s,t) = \sum_{s \in R_k} \lambda(s) \frac{q(s,t)}{\lambda(s)} = \sum_{s \in R_k} \lambda(s) \mathbb{P}^k_{s,t}.$$

The first equality holds since (11) and q(s,t) = 0 for $s \in S \setminus R_k$. The third equality comes from (16). Then the objective function (8) with solution $\gamma(s,a)$ can be written as

$$\sum_{k=1}^{K} \sum_{s \in R_k} \sum_{t \in R_k} -\lambda(s) \mathbb{P}_{s,t}^{\mu} \log(\mathbb{P}_{s,t}^{\mu}). \tag{17}$$

The value in (17) is also equal to the entropy rate of MC \mathcal{M}^{μ} with initial distribution π'_0 satisfying $\sum_{s \in R_k} \pi'_0(s) = \sum_{s \in R_k} \lambda(s)$ for all $k = 1, \ldots, K$. Let $\gamma^{\star}(s, a)$ be the optimal solution of (8)-(13) and $\hat{\mu}^{\star}$ be the policy constructed by (15) based on $\gamma^{\star}(s, a)$. Therefore, $\nabla H(\mathcal{M}^{\hat{\mu}^{\star}}) = V^{\star}$ when the initial distribution is π'_0 . Since $V^{\star} \geq \nabla H(\mathcal{M})$ regardless of initial distribution, we have $\nabla H(\mathcal{M}^{\hat{\mu}^{\star}}) = \nabla H(\mathcal{M})$ when initial distribution is π'_0 . Also, by Lemma 1, $\mathcal{M}^{\hat{\mu}^{\star}}$ is irreducible, i.e., under any initial distribution, values $\nabla H(\mathcal{M}^{\hat{\mu}^{\star}})$ are same. Thus $V^{\star} = \nabla H(\mathcal{M}^{\hat{\mu}^{\star}}) = \nabla H(\mathcal{M})$ when the initial state is s_0 . Since $\hat{\mu}^{\star}$ is irreducible and $\sum_{a \in A(s)} \gamma(s, a)$ is equal to the limit distribution for state s, $\sum_{a \in A(s)} \gamma(s, a) > 0$ for all $s \in S$, which means that Rc = S. This further implies that $\hat{\mu}^{\star} = \mu^{\star}$, which completes the proof.

Remark 1 (Complexity of the Nonlinear Program). Here we would like to remark that the proposed nonlinear program in Equations (8)-(13) is convex. Therefore, it can be solved in polynomial-time. To see this, first, we note that constraints (9)-(13) are affine. Furthermore, we note that the objective function (8) is concave when $\lambda(s,a) \geq 0$. To see this, for $s \in S$, term $\ell(s) = \sum_{t \in S} q(s,t) \log(\frac{q(s,t)}{\lambda(s)})$ is the relative entropy of vectors $(q(s,t))_{t \in S}$ and $\lambda(s)\mathbf{1} \in \mathbb{R}^{|S|}$, where $\mathbf{1} \in \mathbb{R}^{|S|}$ is the vector of ones. Since relative entropy is convex [4] and finite sum of convex functions is convex, the negative of the objective function is still convex, i.e., objective function is concave. Therefore, a solution of (9)-(13) whose objective value (8)

is arbitrarily close to the optimal value can be computed in polynomial-time via interior-point method in the size of \mathcal{M} .

Note that, the above nonlinear program does not take the satisfaction of the LTL task into account. In order to satisfy the LTL task, one needs to stay in some AMEC forever. Therefore, our approach is to first compute all AMECs and for each AMEC, which is a communicating MDP, we compute the optimal policy that maximizes the entropy rate within this AMEC. Then the overall strategy is to ensure that one can reach the AMEC with maximum entropy rate w.p.1. The idea above is summarized by Algorithm 1, where lines 12-16 ensure that the AMEC with maximum entropy rate can be reached eventually w.p.1.

Algorithm 1: Solution for Communicating MDP

```
Input: communicating MDP
                \mathcal{M} = (S, s_0, A, P, \mathcal{AP}, \ell, Acc)
    Output: optimal policy \mu^* \in \Pi^S_{\mathcal{M}} and its associated
                  maximum entropy rate v^* of MC \mathcal{M}^{\mu^*}
 1 compute AMEC(\mathcal{M}) = \{(\mathcal{S}_1, \mathcal{A}_1), \dots, (\mathcal{S}_n, \mathcal{A}_n)\}
 2 if AMEC(\mathcal{M}) = \emptyset then
          \mu^{\star} \leftarrow arbitrary policy, and v^{\star} \leftarrow -\infty
 4 end
5 else
          for each sub-MDP (S_i, A_i), i = 1, ..., n do
 6
                compute maximum entropy rate policy \mu_i by
                  Equation (14) and the associated value
                  v_i = \nabla H(\mathcal{S}_i, \mathcal{A}_i)
          i^{\star} \leftarrow \arg\max_{i} \{v_1, v_2, \dots, v_n\}
          for s \in \mathcal{S}_{i^*}, a \in \mathcal{A}_{i^*}(s), \mu^*(s, a) \leftarrow \mu_{i^*}(s, a)
          T \leftarrow S \setminus \mathcal{S}_{i^*}, and G \leftarrow \mathcal{S}_{i^*}
11
          while T \neq \emptyset do
12
               pick s \in T, a \in A(s) s.t. \sum_{t \in G} P_{s,a,t} > 0
                \mu^{\star}(s,a) \leftarrow 1
                T \leftarrow T \setminus \{s\}, and G \leftarrow G \cup \{s\}
16
          end
17 end
```

The following result shows that, for a communicating MDP, the policy computed by Algorithm 1 indeed (i) satisfies the LTL task; and (ii) maximizes the entropy rate.

Theorem 2. Given communicating MDP $\mathcal{M} = (S, s_0, A, P, \mathcal{AP}, \ell, Acc)$, the output policy of Algorithm 1 is a solution to Problem 1 for \mathcal{M} .

Proof. According to Lemma 1, for each AMEC (S_i, A_i) , policy μ_i ensures that all states in it will be visited infinitely often w.p.1. Based on the action assignment procedure in lines 12-16, MC \mathcal{M}^{μ^*} has only one recurrent class S_i . By the definition of AMEC, we have $\mu^* \in \Pi^{\varphi}_{\mathcal{M}}$, i.e., $\nabla H(\mathcal{M}^{\mu^*}) \leq \nabla H_{\varphi}(\mathcal{M})$.

We now consider arbitrary $\mu \in \Pi^S_{\mathcal{M}} \cap \Pi^{\varphi}_{\mathcal{M}}$. Let $R_1, R_2, \ldots, R_K \subseteq S$ be the recurrent classes in \mathcal{M}^{μ} . Since $\mu \in \Pi^{\varphi}_{\mathcal{M}}$, for any R_k , we can find $(\mathcal{S}, \mathcal{A}) \in \mathtt{AMEC}(\mathcal{M})$ such that $R_k \subseteq \mathcal{S}$. Let r_k be the entropy rate restricted on recurrent class R_k . We denote by r_k^{\star} maximum entropy rate of the AMEC that R_k belongs to. Then by Claim 1 in the

Appendix, we can find a set of values $\beta_1, \dots \beta_K \in [0, 1]$ with $\sum_{k=1}^K \beta_k = 1$ such that

$$\nabla H(\mathcal{M}^{\mu}) = \sum_{k=1}^{K} \beta_k r_k \le \sum_{k=1}^{K} \beta_k r_k^{\star} \le v_{i^{\star}} = \nabla H(\mathcal{M}^{\mu^{\star}}),$$

where the first equality and the last inequality come from Claim 1 and line 9 of Algorithm 1, respectively. Thus

$$\nabla H(\mathcal{M}^{\mu^{\star}}) \ge \sup_{\mu \in \Pi^{S}_{\mathcal{M}} \cap \Pi^{\varphi}_{\mathcal{M}}} \nabla H(\mathcal{M}^{\mu}) = \nabla H_{\varphi}(\mathcal{M}),$$

where the last equality comes from Proposition 1. Hence, we have $\nabla H(\mathcal{M}^{\mu^*}) = \nabla H_{\varphi}(\mathcal{M})$.

V. STATE LEVEL CLASSIFICATION

In the previous section, we have solved Problem 1 for the case of communicating MDPs. However, for the general case of non-communicating MDPs, this problem is much more challenging. Particularly, to enforce the satisfaction of the LTL task, the MDP should eventually stay in some MECs and different MECs may result in different entropy rates. To resolve this issue, this section proposes an approach to classify MECs into different "levels" in terms of their connectivities.

A. Definitions of State Levels

Let $MEC(\mathcal{M}) = \{(\mathcal{S}_1, \mathcal{A}_1), \dots, (\mathcal{S}_n, \mathcal{A}_n)\}$ be the set of all MECs of MDP \mathcal{M} . For each state $s \in S$, we say it is

- an *MEC state* if it belongs to some MEC, and we denote by $S_M = \bigcup_{i=1}^n S_i$ the set of MEC states; and
- a transient state if it does not belong to any MEC, and we denote by $\mathcal{T} = S \setminus \mathcal{S}_M$ the set of all transient states.

Note that each state can only belong to at most one MEC. Then for each MEC state $s \in \mathcal{S}_M$, we denote by $(\mathcal{S}_{[s]}, \mathcal{A}_{[s]})$ the unique MEC it belongs to. Also, we note that, 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 must be 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. Based on this observation, we can classify MECs into different "levels" as follows.

- First, there must exist states that cannot leave their MECs and we consider those states as the "lowest" level.
- Then, for any MEC state, it is said to be with level k if it can only reach MECs with lower levels or itself.

The above discussion leads to the following definition.

Definition 6 (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:

$$L_0 = \{ s \in \mathcal{S}_M \mid \mathit{reach}(s) \cap \mathcal{S}_M = \mathcal{S}_{[s]} \}, \tag{18}$$

$$L_k = \{ s \in \mathcal{S}_M \mid \mathit{reach}(s) \cap \mathcal{S}_M \subseteq \bigcup_{m < k} L_m \cup \mathcal{S}_{[s]} \} \setminus \bigcup_{m < k} L_m.$$

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

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

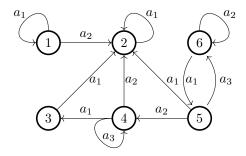


Fig. 1. Illustrative example of state level classification.

Definition 7 (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:

$$T_k = \{ s \in \mathcal{T} \mid \mathit{reach}(s) \cap \mathcal{S}_M \subseteq \bigcup_{m \le k} L_m \} \setminus \bigcup_{m < k} T_m.$$
 (19)

We use the following example to illustrate the definitions.

Example 1. Let us consider an MDP \mathcal{M} shown in Figure 1. For each action, the transition probability is one and the value is omitted in the figure. This MDP has four MECs $\{(S_1, A_1), (S_2, A_2), (S_3, A_3), (S_4, A_4)\}$, where $S_1 = \{1\}$ with $A_1(1) = \{a_1\}$, $S_2 = \{2\}$ with $A_2(2) = \{a_1\}$, $S_3 = \{4\}$ with $A_3(4) = \{a_3\}$, and $S_4 = \{5,6\}$ with $A_4(5) = \{a_3\}$, $A_4(6) = \{a_1, a_2\}$. Then we have $S_M = \{1, 2, 4, 5, 6\}$ and $T = \{3\}$. Clearly, $L_0 = \{2\}$ since $\operatorname{reach}(2) \cap S_M = S_{[2]}$. For states i = 1, 2, 4, since $\operatorname{reach}(i) \cap S_M \subseteq L_0 \cup S_{[i]}$, we have $L_1 = \{1, 2, 4\} \setminus L_0 = \{1, 4\}$. Similarly, we have $L_2 = \{5, 6\}$. Since $\operatorname{reach}(3) \cap S_M = \{2\} \cap S_M \subseteq L_0$, we have $T_0 = \{3\}$ and $T_1 = T_2 = \emptyset$. The level of the MDP is $1 \in V \in I(\mathcal{M}) = 2$.

B. Computations of State Levels

In order to classify the level of each state, we first define a relation $\leq_L\subseteq \text{MEC}(\mathcal{M})\times \text{MEC}(\mathcal{M})$ by: for any $U_1=(\mathcal{S}_1,\mathcal{A}_1), U_2=(\mathcal{S}_2,\mathcal{A}_2)\in \text{MEC}(\mathcal{M}),$ we have

$$U_1 \leq_L U_2 \iff \forall s \in \mathcal{S}_2 : \mathcal{S}_1 \subseteq \operatorname{reach}(s).$$
 (20)

Proposition 2. Relation $\leq_L \subseteq MEC(\mathcal{M}) \times MEC(\mathcal{M})$ defined in Equation (20) is a partial order relation.

Proof. We show that \leq_L is reflexive, anti-symmetric and transitive. First, by the definition of MEC, we have $\mathcal{S}_{[s]} \subseteq \operatorname{reach}(s)$. Thus $(\mathcal{S},\mathcal{A}) \leq_L (\mathcal{S},\mathcal{A})$, i.e., \leq_L is reflexive. Second, for two MECs $(\mathcal{S}_1,\mathcal{A}_1) \neq (\mathcal{S}_2,\mathcal{A}_2)$, if $(\mathcal{S}_1,\mathcal{A}_1) \leq_L (\mathcal{S}_2,\mathcal{A}_2)$ and $(\mathcal{S}_2,\mathcal{A}_2) \leq_L (\mathcal{S}_1,\mathcal{A}_1)$, then $(\mathcal{S}_1 \cup \mathcal{S}_2,\mathcal{A}_1 \cup \mathcal{A}_2)$ is also an MEC. However, this will contradict to the fact that $(\mathcal{S}_1,\mathcal{A}_1)$ and $(\mathcal{S}_2,\mathcal{A}_2)$ are two different MECs. Thus, \leq_L is anti-symmetric. Finally, If $(\mathcal{S}_1,\mathcal{A}_1) \leq_L (\mathcal{S}_2,\mathcal{A}_2)$ and $(\mathcal{S}_2,\mathcal{A}_2) \leq_L (\mathcal{S}_3,\mathcal{A}_3)$, then for any $s \in \mathcal{S}_3$, $\mathcal{S}_2 \subseteq \operatorname{reach}(s)$ and for any $t \in \mathcal{S}_2$, $\mathcal{S}_1 \subseteq \operatorname{reach}(t)$. Thus $\mathcal{S}_1 \subseteq \operatorname{reach}(s)$, i.e., $(\mathcal{S}_1,\mathcal{A}_1) \leq_L (\mathcal{S}_3,\mathcal{A}_3)$, which means that \leq_L is transitive. \square

Note that partial order \leq_L can be computed directly by checking the reachability between MECs. With partial order

 \leq_L , we can compute state levels for MEC states by Algorithm 2. Specifically, since $\text{MEC}(\mathcal{M})$ is a finite set, there must exist minimal elements in relation \leq_L , and L_0 contains all states in MECs that are minimal elements in \leq_L . Then we eliminate these elements in \leq_L and the minimal elements for remaining relation consist the L_1 . We repeat this procedure and find MECs of each level. The level for each transient state $s \in \mathcal{T}$ is computed in lines 11-14, which is simply the highest level of MEC states it can reach.

Algorithm 2: State Level Classification

```
Input: MDP \mathcal{M} = (S, s_0, A, P, \mathcal{AP}, \ell, Acc)

Output: L'_0, T'_0, \dots, L'_{\mathtt{level}(\mathcal{M})}, T'_{\mathtt{level}(\mathcal{M})}

1 compute \mathtt{MEC}(\mathcal{M}) = \{(\mathcal{S}_1, \mathcal{A}_1), \dots, (\mathcal{S}_n, \mathcal{A}_n)\}

2 compute partial order \leq_L in (20)

3 num \leftarrow 0

4 while \leq_L \neq \emptyset do

5 \mid \mathcal{I} \leftarrow \{i \mid (\mathcal{S}_i, \mathcal{A}_i) \text{ is minimal element in } \leq_L\}

6 \mid L'_{num} \leftarrow \bigcup_{i \in \mathcal{I}} \mathcal{S}_i

7 \mid \leq_L \leftarrow \leq_L \setminus \{((\mathcal{S}_i, \mathcal{A}_i), (\mathcal{S}_j, \mathcal{A}_j)) \mid i \in \mathcal{I}\}

8 \mid num \leftarrow num + 1

9 end

10 T'_0 = T'_1 = \cdots = T'_{num-1} \leftarrow \emptyset

11 for s \in S \setminus \bigcup_{i=1}^k \mathcal{S}_i do

12 \mid cnt \leftarrow \max\{j \mid \operatorname{reach}(s) \cap L'_j \neq \emptyset\}

13 \mid T'_{cnt} \leftarrow T'_{cnt} \cup \{s\}

14 end
```

Proposition 3. Given MDP $\mathcal{M} = (S, s_0, A, P, \mathcal{AP}, \ell, Acc)$, the outputs of Algorithm 2 are indeed $L_0, L_1, \ldots, L_{1 \text{evel}(\mathcal{M})}$ defined in Equation (18) and $T_0, T_1, \ldots, T_{1 \text{evel}(\mathcal{M})}$ defined in Equation (19).

Proof. We first prove by induction that, at n-th time the algorithm executes the "while"-loop in lines 4-9, we have $L'_{n-1} = L_{n-1}$. When n = 0, since \leq_L is finite, we know that there exists minimum elements in \leq_L . If $i \in \mathcal{I}$, we know that for $s \in \mathcal{S}_i$, reach $(s) \cap \mathcal{S}_M = \mathcal{S}_i$. Thus $L'_0 = L_0$. Assume that $L'_0 = L_0, \ldots, L'_k = L_k$ for some $k \geq 0$ and $\leq_L \neq \emptyset$ after eliminating the minimum elements in each "while"-loop. For n = k + 1, if $i \in \mathcal{I}$, it means that for all $s \in \mathcal{S}_i$,

$$\operatorname{reach}(s)\cap\mathcal{S}_M\subseteq\bigcup_{m< k+1}L_m'\cup\mathcal{S}_{[s]}=\bigcup_{m< k+1}L_m\cup\mathcal{S}_{[s]},$$

where last equality comes from induction hypothesis. Since $L'_{k+1} \cap (\bigcup_{m < k+1} L'_m) = \emptyset$, from (18) it holds that $L'_{k+1} = L_{k+1}$. The proof of induction is completed.

Next, we show that $T_0', T_1', \ldots, T_{num-1}'$ are indeed $T_0, T_1, \ldots, T_{\text{level}(\mathcal{M})}$. In above, we have $\text{level}(\mathcal{M}) = num - 1$. For T_k defined in (19), $s \in T_k$ if and only if $s \in \mathcal{T}$ and

$$[L_k \cap \operatorname{reach}(s) \neq \emptyset] \land (\forall i > k)[L_i \cap \operatorname{reach}(s) = \emptyset].$$

Therefore, T'_k exactly matches the definition of T_k .

VI. SOLUTION FOR GENERAL CASE

In this section, we tackle the Problem 1 for general MDPs based on the solutions for communicating MDPs and the state level classification technique in the previous two sections.

A. General Ideas

For non-communicating MDP \mathcal{M} , we define

$$R_k = \bigcup_{m=0}^k (L_m \cup T_m)$$
 and $A_k : R_k \to 2^A$ s.t. $A_k(s) = A(s)$

as the set of all MEC states and transient states whose levels are smaller than or equal to k, and the associated available actions, respectively. Similarly, we also define

$$\hat{R}_k = R_k \setminus T_k$$
 and $\hat{A}_k : \hat{R}_k \to 2^A$ s.t. $\hat{A}_k(s) = A(s)$

as the set of all MEC states with levels smaller than or equal to k and transient states with levels strictly smaller than k, and the associated available actions, 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_k and μ_m are policies that solve Problem 1 for (\hat{R}_k,\hat{A}_k) and (\hat{R}_m,\hat{A}_m) , respectively. By modifying μ_m to μ'_m such that $\mu_m(s,a)$ is (i) changed to $\mu_k(s,a)$, for all $s\in \hat{R}_k$, and (ii) unchanged otherwise, we know that the modified μ'_m 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:

Step 1: Initially, we start from those MECs with level 0 and compute solutions for these sub-MDPs. Since each MEC is communicating, we can use Algorithm 1 in Section IV.

Step 2: Once all MECs in L_0 have been processed, we move to include transient states in T_0 . This 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 actions we take, the only factor that determines the total entropy rate is what MECs in L_0 they choose to go. Therefore, it suffices to solve an *expected total reward* maximization problem, where the reward of reaching each MEC in L_0 is the computed maximum entropy rate.

Step 3: 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) . Still, for each MEC in L_1 , we use Algorithm 1 to compute the maximum entropy rate within it. However, here we have two alternatives for each MEC in L_1 :

- (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 forever.

Therefore, we need to compare these two alternatives and choose the one with larger reward (entropy rate).

Step 4: Once (R_1, 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 $(R_{1\text{evel}(\mathcal{M})}, A_{1\text{evel}(\mathcal{M})}) = \mathcal{M}$ is solved.

B. Synthesis Algorithm

Now, we formalize the implementation details of the above idea. Suppose that, at decision stage $k=0,\ldots, \mathtt{level}(\mathcal{M})$, we have the following information available:

- the set of states have been processed: $R_k \subseteq S$;
- the solution $\hat{\mu}_k \in \Pi^S_{\hat{\mathcal{M}}_k}$ to 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 LTL task w.p.1, which is specified by a function $\operatorname{val}_k: \hat{R}_k \to \mathbb{R}$.

Then our objective is to find the optimal policy, denoted by $\hat{\mu}_{k+1} \in \Pi^S_{\hat{\mathcal{M}}_{k+1}}$, for a larger sub-MDP

$$\hat{\mathcal{M}}_{k+1} = (\hat{R}_{k+1} = \hat{R}_k \cup T_k \cup L_{k+1}, \hat{A}_{k+1}).$$

Our approach consists of the following four steps.

Optimal Transient-Enforcing Policy: Note that, although states in T_k are always transient, the system may choose to stay in L_{k+1} . Here, we first synthesize a transient-enforcing policy, denoted by $\hat{\mu}'_{k+1}$, such that all states in L_{k+1} are enforced to be transient. This policy only applies to states in $T_k \cup L_{k+1}$. To this end, we first solve a new linear program (LP) (21)-(25), where α is an arbitrary distribution vector over $T_k \cup L_{k+1}$ such that all elements are non-zero, and $val = val_k + \epsilon$ is reward vector over R_k with $\epsilon > 0$ be an arbitrary value ensuring that val(s) > 0 for any $s \in \hat{R}_k$.

Linear Program for Transient-Enforcing Policy

$$\max_{\gamma(s,a)} \sum_{s \in T_k \cup L_{k+1}} \sum_{t \in \hat{R}_k} \operatorname{val}(t) \lambda(s,t)$$
 (21)

s.t.
$$\eta(s) - \sum_{t \in T_k \cup L_{k+1}} \lambda(t, s) \le \alpha(s), \forall s \in T_k \cup L_{k+1}$$

$$(22)$$

$$\eta(s) = \sum_{a \in A(s)} \gamma(s, a), \forall s \in T_k \cup L_{k+1}$$
 (23)

$$\lambda(s,t) = \sum_{a \in A(s)} \gamma(s,a) P_{s,a,t}, \forall s \in T_k \cup L_{k+1}, t \in \hat{R}_{k+1}$$

(24)

$$\gamma(s, a) \ge 0, \forall s \in T_k \cup L_{k+1}, \forall a \in A(s)$$
 (25)

Let γ^* be the optimal solution to LP (21)-(25). Similar to the LP to the standard expected total reward maximization problem [30], the optimal solution satisfies that, for each $s \in$ $T_k \cup L_{k+1}$, there exists at most one action, denoted as $a^*(s) \in$ A(s) such that $\gamma^{\star}(s, a^{\star}(s)) > 0$. Then we define deterministic transient-enforcing policy $\hat{\mu}'_{k+1}$ by

$$\hat{\mu}'_{k+1}(s) = \begin{cases} a^{\star}(s) & \text{if } \sum_{a \in A(s)} \gamma^{\star}(s, a) > 0\\ \text{arbitrary } & \text{if } \sum_{a \in A(s)} \gamma^{\star}(s, a) = 0 \end{cases}$$
 (26)

where $\hat{\mu}'_{k+1}(s)$ denotes the unique action chosen w.p.1 at s. Intuitively, decision variable $\gamma(s,a)$ in LP (21)-(25) is the the expected number of visits to state s and choosing action a when the initial distribution is α . Here, we require that all elements in α are non-zero in order to ensure that all states can be visited with non-zero probability. Variables $\eta(s)$ and $\lambda(s,t)$ in Equations (23) and (24) 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 (22) is the constraint of the probability flow. Finally, objective function in Equation (21) multiplies the probability of reaching R_k and the reward (maximum entropy rate) in R_k and sums over $s \in T_k \cup L_{k+1}$, representing the entropy rate of corresponding policy under initial distribution α . Note that, here we add ϵ uniformly to the original reward val_k in order to force the optimal policy to visit states in R_k .

Optimal Staying Policy: Note that the transient-enforcing policy $\hat{\mu}'_{k+1}$ may not be the optimal policy for states in $T_k \cup L_{k+1}$ since we force each state in L_{k+1} to go to MECs with lower levels. However, states in L_{k+1} can also choose to stay at level k + 1. To capture this situation, for each MEC $(S_i, A_i) \in MEC(\mathcal{M})$ with level k+1, i.e., $S_i \subseteq L_{k+1}$, we denote by $\mu_{\text{stay},i}$ and v_i^{\star} the outputs of Algorithm 1 when considering (S_i, A_i) as the input communicating MDP. 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.

Value Evaluations: To fuse the transient-enforcing policy and staying policy, we need to compute their value functions. Then for each state in $s \in S_i$, where (S_i, A_i) is an MEC with level k+1, the stay value of s is the maximum entropy rate when s stays in the MEC forever, i.e.,

$$v_{\text{stay}}(s) = v_i^{\star}. \tag{27}$$

To compute the value function under transient-enforcing policy $\hat{\mu}'_{k+1}$, we define matrix $\mathbb{P}_{\hat{R}_k}^{\hat{\mu}'_{k+1}} \in \mathbb{R}^{|T_k \cup L_{k+1}| \times |\hat{R}_k|}$ such that $\mathbb{P}_{\hat{R}_k}^{\hat{\mu}'_{k+1}}(s,t)=\sum_{a\in A(s)}\hat{\mu}'_{k+1}(s,a)P_{s,a,t}$ is the transition probability from $s \in T_k \cup L_{k+1}$ to $t \in \hat{R}_k$ under partial policy $\hat{\mu}'_{k+1}$. We can similarly define matrix $\mathbb{P}^{\hat{\mu}'_{k+1}}_{T_k \cup L_{k+1}}$ computing transition probabilities between states in $T_k \cup L_{k+1}$. The transient value function $\mathbf{v}_{\text{trans}} \in \mathbb{R}^{|T_k \cup L_{k+1}|}$ is the solution of following equation

$$\mathbf{v}_{\text{trans}} = \mathbb{P}_{T_k \cup L_{k+1}}^{\hat{\mu}'_{k+1}} \mathbf{v}_{\text{trans}} + \mathbb{P}_{\hat{R}_k}^{\hat{\mu}'_{k+1}} \mathbf{val}_k. \tag{28}$$

Since $I - \mathbb{P}_{T_k \cup L_{k+1}}^{\hat{\mu}'_{k+1}}$ is invertible [30, page 595], we have

$$\mathbf{v}_{\text{trans}} = (I - \mathbb{P}_{T_k \cup L_{k+1}}^{\hat{\mu}'_{k+1}})^{-1} \mathbb{P}_{\hat{R}_k}^{\hat{\mu}'_{k+1}} \text{val}_k. \tag{29}$$

Policy Fusion: To fuse policies $\hat{\mu}'_{k+1}$, μ_{stay} , and the low level policy $\hat{\mu}_k$, we consider three cases:

- Case 1 : $[s \in T_k]$ or $[s \in L_{k+1} \land v_{trans}(s) > v_{stay}(s)];$
- Case 2 : $[s \in L_{k+1} \land v_{trans}(s) \le v_{stay}(s)];$
- Case $3:s\in \hat{R}_k$.

Then the new stationary policy $\hat{\mu}_{k+1}$ is fused as follows: for each $s \in \hat{R}_{k+1}$, we have

$$\hat{\mu}_{k+1}(s,a) = \begin{cases} \hat{\mu}'_{k+1}(s,a) & \text{if } \text{Case 1} \\ \mu_{\text{stay}}(s,a) & \text{if } \text{Case 2} \\ \hat{\mu}_{k}(s,a) & \text{if } \text{Case 3} \end{cases}$$
(30)

The value function for the fused policy $\hat{\mu}_{k+1}$ is updated as

$$val_{k+1}(s) = \begin{cases} v_{trans}(s) & \text{if } Case 1 \\ v_{stay}(s) & \text{if } Case 2 \\ val_{k}(s) & \text{if } Case 3 \end{cases}$$
 (31)

Note that, for k = 0 and each $s \in \hat{R}_0$, we have

$$\mathrm{val}_0(s) = v_{[s]}^{\star} \text{ and } \hat{\mu}_0(s, a) = \mu_{\mathrm{stay}, [s]}(s, a)$$
 (32)

where $v_{[s]}^{\star}$ and $\mu_{\text{stay},[s]}$ are maximum entropy rate and the corresponding optimal policy for staying within MEC $(\mathcal{S}_{[s]}, \mathcal{A}_{[s]})$.

Algorithm 3: Solution for General MDP

```
Input: MDP \mathcal{M} = (S, s_0, A, P, \mathcal{AP}, \ell, Acc)
    Output: optimal policy \mu^* \in \Pi^S_{\mathcal{M}}
 1 compute MEC(\mathcal{M}) = \{(\mathcal{S}_1, \mathcal{A}_1), \dots, (\mathcal{S}_n, \mathcal{A}_n)\}
 2 compute \mu_i^* and v_i^* for each (S_i, A_i) \in MEC(\mathcal{M})
3 classify states into L_0, T_0, \dots, L_{\mathtt{level}(\mathcal{M})}, T_{\mathtt{level}(\mathcal{M})}
4 compute \hat{\mu}_0 and val<sub>0</sub> according to Eq. (32)
5 for k = 0, 1, \dots, level(\mathcal{M}) do
         solve LP (21)-(25) for (\hat{\mathcal{M}}_{k+1}, \hat{R}_k, \text{val}_k)
 6
         compute policy \hat{\mu}'_{k+1} according to Eq. (26)
 7
         compute value functions v_{\text{stay}} and v_{\text{trans}}
           according to Eq. (27) and (29), respectively
         fuse policy \hat{\mu}_{k+1} according to Eq. (30) and
           compute val_{k+1} according to Eq. (31)
10 end
11 Return \hat{\mu}_{\texttt{level}(\mathcal{M})+1}
```

The overall synthesis procedure is summarized in Algorithm 3. First, we initialize the optimal policy as well as the value function for level 0 in line 4. In lines 5-10, we iteratively compute the optimal policy $\hat{\mu}_k$ for each level k. When $k = \texttt{level}(\mathcal{M}) + 1$, $\hat{\mu}_k$ is already the optimal policy for the entire MDP. Note that sub-MDP $\hat{\mathcal{M}}_{\texttt{level}(\mathcal{M})}$ has not included states in $T_{\texttt{level}(\mathcal{M})}$ yet.

C. Correctness proof

We conclude this section by analyzing the correctness of the synthesis algorithm. We start by proving that $\hat{\mu}'_{k+1}$ is indeed the optimal transient-enforcing policy. Let

$$\begin{split} &\Pi^T_{\hat{\mathcal{M}}_{k+1}} = \\ &\{\mu \in \Pi^\varphi_{\hat{\mathcal{M}}_{k+1}} \cap \Pi^S_{\hat{\mathcal{M}}_{k+1}} \mid T_k \cup L_{k+1} \text{ is transient in } \hat{\mathcal{M}}^\mu_{k+1} \} \end{split}$$

be the set of stationary policies under which all states in $T_k \cup L_{k+1}$ are transient and the LTL task is satisfied w.p.1. Then the following result shows that policy $\hat{\mu}_{k+1}$ achieves maximum entropy rate among $\Pi^T_{\hat{\mathcal{M}}_{k+1}}$.

Proposition 4. Let $\hat{\mu}_k$ be a solution to Problem 1 w.r.t. MDP $\hat{\mathcal{M}}_k = (\hat{R}_k, \hat{A}_k)$ regardless of the initial state and $\hat{\mu}'_{k+1}$ be the policy defined in (26). We define policy $\tilde{\mu}_{k+1}$ by: $\tilde{\mu}_{k+1}(s,a) = \hat{\mu}'_{k+1}(s,a)$ for $s \in T_k \cup L_{k+1}$ and $\tilde{\mu}_{k+1}(s,a) = \hat{\mu}_k(s,a)$ for $s \in \hat{R}_k$. Then regardless of the initial state of MDP $\hat{\mathcal{M}}_{k+1} = (\hat{R}_{k+1}, \hat{A}_{k+1})$, we have

$$\nabla H(\hat{\mathcal{M}}_{k+1}^{\tilde{\mu}_{k+1}}) = \sup_{\mu \in \Pi_{\hat{\mathcal{M}}}^T} \nabla H(\hat{\mathcal{M}}_{k+1}^{\mu}).$$

Moreover, for $s \in T_k \cup L_{k+1}$, value $v_{trans}(s)$ in (29) is the entropy rate $\nabla H(\hat{\mathcal{M}}_{k+1}^{\tilde{\mu}_{k+1}})$ when the initial state is s.

Proof. The proof is provided in the Appendix.
$$\Box$$

Next, we prove that, if the initial state of MDP is an MEC state, then the maximum entropy rate is the maximum of stay value and transient value.

Proposition 5. For MDP $\hat{\mathcal{M}}_{k+1} = (\hat{R}_k \cup T_k \cup L_{k+1}, \hat{A}_{k+1})$, suppose that the initial state satisfies $s_0 \in L_{k+1}$. Then

$$\nabla H_{\varphi}(\hat{\mathcal{M}}_{k+1}) = \max\{v_{stav}(s_0), v_{trans}(s_0)\}, \quad (33)$$

where v_{stay} and v_{trans} are defined in (27) and (29), respectively.

Proof. By [30, Thm 8.3.2], for any two different initial states in $\mathcal{S}_{[s_0]}$, $\nabla H_{\varphi}(\hat{\mathcal{M}}_{k+1})$ are same. Thus, if $s_0 \in L_{k+1}$, we can find a solution $\tilde{\mu}_{k+1} \in \Pi^S_{\hat{\mathcal{M}}_{k+1}}$ of Problem 1 w.r.t. $\hat{\mathcal{M}}_{k+1}$ such that w.p.1, MC $\hat{\mathcal{M}}_{k+1}^{\tilde{\mu}_{k+1}}$ either 1) stays in MEC $(\mathcal{S}_{[s_0]}, \mathcal{A}_{[s_0]})$ forever or 2) leaves $\mathcal{S}_{[s_0]}$ eventually. Note that $v_{\text{stay}}(s_0)$ and $v_{\text{trans}}(s_0)$ record the maximum entropy rates under situations 1) and 2), respectively. Thus (33) holds.

By combining Propositions 4 and 5, we have the following result immediately for the fused policy $\hat{\mu}_{k+1}$.

Proposition 6. 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 $\hat{\mu}_{k+1}$ defined in Equation (30) is an optimal solution to Problem 1 for instant sub-MDP $\hat{\mathcal{M}}_{k+1} = (\hat{R}_{k+1}, \hat{A}_{k+1})$.

Finally, we establish the correctness of Algorithm 3.

Theorem 3. Given MDP $\mathcal{M} = (S, s_0, A, P, \mathcal{AP}, \ell, Acc)$, the output of Algorithm 3 is a solution of Problem 1 for \mathcal{M} .

Proof. Note that we assume that initial from any $s \in S$, the system can satisfy LTL task w.p.1. Thus for any $s \in L_0$, MEC $(\mathcal{S}_{[s]}, \mathcal{A}_{[s]})$ contains some AMECs. By Theorem 2, we know that $\hat{\mu}_0$ in line 4 of Algorithm 3 is a solution of Problem 1 w.r.t. $\hat{\mathcal{M}}_0$. By Proposition 6, we know that at each step k we find solution of Problem 1 w.r.t. $\hat{\mathcal{M}}_{k+1}$. Since $\hat{\mathcal{M}}_{\text{level}(\mathcal{M})+1} = \mathcal{M}$, we know the return policy solves Problem 1 for \mathcal{M} .

Remark 2. Note that, for the sake of simplicity, in Equation (26), we construct the transient-enforcing policy $\hat{\mu}'_{k+1}$ as a deterministic policy. This is because we are concerned with maximizing the entropy rate, and transient behaviors will not affect this result. If one wants to further maximize the unpredictability of the transient behaviors, then one can adopt the approach in [33] for the transient part.

VII. CASE STUDIES OF ROBOT TASK PLANNING

In this section, we present two case studies of robot task planning to illustrate the proposed method. All computations are performed on a desktop with 16 GB RAM. Specifically, we use the splitting conic solver (SCS) [28] in CVXPY [13] to solve convex optimization problems. Also, we use the tool in [21] to transform the LTL task to DRA.

A. Case Study 1

System Model: We consider a robot moving in a workspace shown in Figure 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 initial location of the robot is pointed by the black arrow. The five regions are connected by some one-way path grids whose feasible directions are depicted in the figure, e.g., robot can reach regions 2 and 3 from region 1 but cannot reach region 1 from

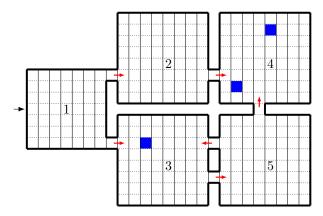


Fig. 2. Workspace of the robot.

regions 2 or 3. 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 chooses 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 directions. 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 belong to the same MEC.

LTL Task: We assume that the robot needs to visit some specific grids, which are marked by blue color with label b, infinitely often in order to communicate with the center station. Then the task is simply $\varphi = \Box \Diamond b$.

Results: Note that, the product MDP for this task is isomorphic to the original MDP, and there are two AMECs in \mathcal{M} : Region 4, and the union of Regions 3 and 5. We denote by μ the solution to Problem 1. Clearly, the robot will choose to eventually stay in Regions 3 and 5 rather than stay in Region 4 only. This is because staying a larger region will increase its entropy rate. The limit distribution (multiplied by 100) of each state under μ is shown in Figure 3. Note that, it suffices to show Regions 3 and 5 as the limit distribution of states in other regions are zero. One may think that the maximum entropy rate policy is uniformly randomized. However, it is not the case. Specifically, for each state in Regions 3 and 5, we compute the difference between highest probability of an action and the lowest probability of an action in the optimal policy; the difference values are shown in Figure 4. 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 is not uniformly randomized if it wants to maximize the entropy rate.

Comparison: In the context of information-theoretical foundation of security, a useful measure for quantifying the unpredictability of an agent is the weight of the Huffman tree of the distribution; see, e.g., [29]. This value has the following physical meaning. Suppose that there is an observer knowing the structure of the MDP and the policy of the agent. In each state, it runs yes-no probes to know the successor state of the agent. If agent moves to state that probe predicts, the probe



Fig. 3. Limit distribution (multiplied by 100) of the optimal policy \mathcal{M}^{μ} .

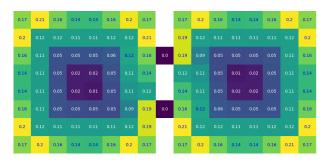


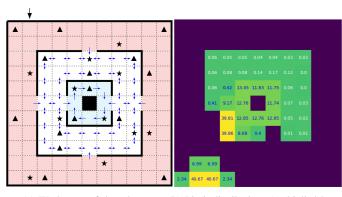
Fig. 4. Largest difference of the probabilities of picking two different actions at each state.

will return yes; otherwise it returns no. Specifically, let Υ_s be the weight of the Huffman tree for the next-step transition distribution at state s. Then this value is actually the running time of yes-no probes in state s. Therefore, the *average number of observations* needed to determine the path of agent under policy μ can be characterized by $O_a^\mu = \sum_s \pi^\mu(s) \Upsilon_s$ where π^μ is limit distribution of MC \mathcal{M}^μ . If we adopt the algorithm in [33] to synthesize a policy μ_1 that finishes the task w.p.1 and maximizes the total entropy of MDP, then it holds that $O_a^{\mu_1} = 0$ since μ_1 will choose a deterministic action in the steady state. However, for our algorithm, we have $O_a^\mu = 2.56$. Therefore, method proposed makes the limit behavior of agent more unpredictable.

B. Case Study 2

System Model: We consider a data collecting and uploading scenario where the robot works in an 11×11 grids as shown in Figure 5(a). Specifically, the overall workspace consists of three regions segmented by walls, denoted by solid black lines in the figure. Region 1 is the light blue area in the center, Region 2 is the white area in the middle, and Region 3 is the light red area on the outside. The *initial* location of the robot is the upper left grid, indicated by the black arrow.

We assume that the robot will have more mobility constraints in the internal regions. Specifically, it can move to adjacent grids freely in Region 3. However, in Regions 1 and 2, the robot can only choose to stay in its current grid or move in the directions indicated by the blue arrows. To move from Region 3 to Region 2, the robot can only use one of the four one-way openings in the wall. Similarly, to move from Region 2 to Region 1, the robot can only use one of the two one-way openings in the wall. Therefore, eventually, the robot will stay



- (a) Workspace of the robot
- (b) Limit distributions (multiplied by 100) of three AMECs.

Fig. 5. Workspace and optimal policy for Case Study 2.

within one region forever and cannot travel between regions interchangeably.

LTL Task: We assume that in the workspace, there are two types of data stations, denoted by \bigstar and \blacktriangle , respectively, where data is collected and uploaded. Specifically, information b can only be collected at station \star and uploaded at station \triangle . Conversely, data r can only be collected at station \triangle and uploaded at station \bigstar . For each region i = 1, 2, 3, there exists a time bound t_i such that, once data is collected within this region, it must be uploaded within this time interval. We denote by b_i and r_i as the atomic propositions such that the grid is a \star station and a \blacktriangle station in Region i, respectively. This communication constraint can be described by the following LTL formula

$$\varphi_i(t_i) = \Box(b_i \to \bigvee_{j=0}^{t_i} \bigcirc^j r_i) \land \Box(r_i \to \bigvee_{j=0}^{t_i} \bigcirc^j b_i),$$

where \bigcirc^j means $\bigcirc \cdots \bigcirc$. We consider the following time

bounds for each region: $t_1=8,\ t_2=5$ and $t_3=3$. The overall objective is to collect and upload each type of data infinitely often while satisfying the time bound constraints, i.e.,

$$\phi = \bigvee_{i=1,2,3} (\Box \Diamond b_i \wedge \Box \Diamond r_i \wedge \varphi_i(t_i)). \tag{34}$$

Clearly, the robot will choose to stay within one region forever. There are also three AMECs in product MDP corresponding to each grid region. We compute the optimal staying policy for each AMEC, and the maximum entropy rates for AMECs corresponding to Regions 1, 2 and 3 are 0.68, 0.86 and 0.81, respectively. Clearly, the optimal overall policy of the robot is to move to Region 2 and stay there forever. We show in Figure 5(b) the limit distributions (projected from product states to grid states, and multiplied by 100) of each AMEC under each optimal staying policy. Note that, for Region 3, to fulfill the LTL task, the robot can only move around the lower left corner, since $t_3 = 3$ and stations \bigstar and ▲ are too far away from each other in other places.

To better demonstrate how the LTL task affects the optimal policy, we further consider the LTL task $\Box \Diamond b_i \land \Box \Diamond r_i \land \varphi_i(t_i)$

TABLE I MER VALUES FOR DIFFERENT PARAMETERS. Ri is Region i. TB is time bound.

TB	1	2	3	4	5	6	7	8	∞
R2	0	0.48	0.69	0.80	0.86	0.90	0.67 0.92 1.07	0.94	1.18

for different regions i = 1, 2, 3 and different time bounds t_i . The maximum entropy rate of the policy achieving the LTL task for each pair of parameters is shown in Table I, where TB represents the time bound between collection and upload, and MER represents the maximum entropy rate. When the time bound goes to ∞ , the LTL task simplifies to $\Box \Diamond b_i \wedge \Box \Diamond r_i$. Clearly, for each region, as the time bound increases, the maximum entropy rate also increases because the robot has more flexibility to be unpredictable. Therefore, if one considers the LTL task in the form of Equation (34), the region in which the robot eventually chooses to stay depends on the time bound value t_i for each region.

VIII. CONCLUSION

In this paper, we solved a new entropy rate maximization problem for MDPs under the requirement that a given linear temporal logic task needs to be achieved with probability one. We first solved this problem for special case of communicating MDPs by an efficient convex optimization problem. For general MDPs, we showed that this problem can be effectively solved by decomposing it as a finite set of sub-problems by proposed state level classification method. Our results extended existing results in entropy rate maximization by taking temporal logic constraints into account. We demonstrated the proposed algorithm by two case studies of robot task planning. In the future, we would like to further investigate how to solve this problem under the partial observation setting.

APPENDIX

Our proof latter needs to leverage the results from MDPs with expected total reward. Specifically, let $\mathcal{R}: S \times A \to \mathbb{R}$ be a reward function that maps each state-action pair to a real number. Given policy $\mu \in \Pi_{\mathcal{M}}$, the *expected total* reward starting from $s \in S$ is defined by

$$v_{\mathcal{M}}^{\mu}(s) = E_s^{\mu} \left[\sum_{t=0}^{\infty} \mathcal{R}(S_t, A_t) \right]. \tag{35}$$

The reward vector for all states is denoted by $v_{\mathcal{M}}^{\mu}$. For $s \in S$, we define $v_{\mathcal{M}}^{\star}(s) = \sup_{\mu \in \Pi_{\mathcal{M}}} v_{\mathcal{M}}^{\mu}(s)$. A policy $\mu^{\star} \in \Pi_{\mathcal{M}}$ is optimal if $v_{\mathcal{M}}^{\star}(s) = v_{\mathcal{M}}^{\mu^{\star}}(s)$ for all $s \in S$. We omit subscript \mathcal{M} if it is clear from context.

Note that, for each AMEC $(\hat{S}, \hat{A}) \in AMEC(\mathcal{M})$, we can find an MEC $(S, A) \in MEC(M)$ such that $\hat{S} \subseteq S$ and $\hat{A} \subseteq A$. We denote by $MEC_{\varphi}(\mathcal{M})$ the set of MECs containing at least one AMEC. For a set of MECs $M \subseteq MEC_{\varphi}(\mathcal{M})$, we denote by $\mathcal{S}_{\mathbb{M}}\subseteq S$ the set of all states in M. Also, for each MEC $(\mathcal{S}, \mathcal{A}) \in MEC_{\varphi}(\mathcal{M})$, we define

$$V^{\star}(\mathcal{S}, \mathcal{A}) = \max_{(\hat{\mathcal{S}}, \hat{\mathcal{A}}) \in \mathtt{AMEC}(\mathcal{S}, \mathcal{A})} \nabla H(\mathcal{M}(\hat{\mathcal{S}}, \hat{\mathcal{A}}))$$

as the maximum entropy rate one can achieve among all AMECs in $(\mathcal{S},\mathcal{A})$. Let $(\hat{\mathcal{S}},\hat{\mathcal{A}}) \in \mathtt{AMEC}(\mathcal{S},\mathcal{A})$ be the AMEC achieving $V^\star(\mathcal{S},\mathcal{A})$. Define $\mu_{(\mathcal{S},\mathcal{A})} \in \Pi^S_{\mathcal{M}(\mathcal{S},\mathcal{A})}$ s.t. for $s \in \hat{\mathcal{S}}$, $\mu_{(\mathcal{S},\mathcal{A})}(s,a) = \mu^\star_{(\hat{\mathcal{S}},\hat{\mathcal{A}})}(s,a)$ with $\mu^\star_{(\hat{\mathcal{S}},\hat{\mathcal{A}})}$ the maximum entropy rate policy of sub-MDP $(\hat{\mathcal{S}},\hat{\mathcal{A}})$, and for states set $\mathcal{S} \setminus \hat{\mathcal{S}}$, $\mu_{(\mathcal{S},\mathcal{A})}$ ensures that $\mathcal{S} \setminus \hat{\mathcal{S}}$ can reach $\hat{\mathcal{S}}$ eventually w.p.1 [30, Page 480]. Then $\mu_{(\mathcal{S},\mathcal{A})}$ achieves $V^\star(\mathcal{S},\mathcal{A})$ over sub-MDP $(\mathcal{S},\mathcal{A})$.

Given MDP \mathcal{M} with state space S and $\hat{S} \subseteq S$, let

$$\mathcal{M}_{\hat{S}} = (\bar{S}, \bar{A}, \bar{P}) \tag{36}$$

be an MDP such that $\bar{S} = S \cup \{b\}$ with b be a new state, $\bar{A}(s) = A(s)$ for $s \in S \setminus \hat{S}$ and $\bar{A}(s) = \{\bar{a}\}$ for $s \in \hat{S} \cup \{b\}$ with \bar{a} be a new action. For $s \in S \setminus \hat{S}$, we define $\bar{P}_{s,a,t} = P_{s,a,t}$ and for $s \in \hat{S} \cup \{b\}$, we define $\bar{P}_{s,\bar{a},b} = 1$. Given MECs set $\mathbb{M} \subseteq \text{MEC}_{\varphi}(\mathcal{M})$, we define a reward function $\mathcal{R}_{\mathbb{M}}$ s.t.

$$\mathcal{R}_{\mathbb{M}}(s,a) = \begin{cases} V^{\star}(\mathcal{S}_{[s]}, \mathcal{A}_{[s]}) & \text{if } s \in \mathcal{S}_{\mathbb{M}} \\ 0 & \text{otherwise} \end{cases}$$
 (37)

From [30] there exists a stationary policy, denoted by $\mu_{\mathbb{M}}^{\star} \in \Pi_{\mathcal{M}_{S_{\mathbb{M}}}}^{S}$, for total expected reward maximization w.r.t. $\mathcal{M}_{S_{\mathbb{M}}}$ and $\mathcal{R}_{\mathbb{M}}$ under any initial distribution. We denote by $\tilde{\mathcal{S}}_{\mathbb{M}} \subseteq S$ the set of states which can reach states in $\mathcal{S}_{\mathbb{M}}$ w.p.1 under some policy. Since $V^{\star}(\mathcal{S},\mathcal{A}) \geq 0$ for any $(\mathcal{S},\mathcal{A}) \in \mathrm{MEC}_{\varphi}(\mathcal{M})$, we can assume without loss of generality that states in $\tilde{\mathcal{S}}_{\mathbb{M}}$ will reach $\mathcal{S}_{\mathbb{M}}$ w.p.1 in MC $\mathcal{M}_{S_{\mathbb{M}}}^{\mu_{\mathbb{M}}}$, because it can only get zero reward when reaching MECs not in M under reward $\mathcal{R}_{\mathbb{M}}$. For $\mathbb{M} \subseteq \mathrm{MEC}_{\varphi}(\mathcal{M})$, we denote by $\mu_{\mathbb{M}} \in \Pi_{\mathcal{M}}^{S}$ the policy such that $\mu_{\mathbb{M}}(s,a) = \mu_{\mathbb{M}}^{\star}(s,a)$ if $s \in S \setminus S_{\mathbb{M}}$ and $\mu_{\mathbb{M}}(s,a) = \mu_{(\mathcal{S}_{[s]},\mathcal{A}_{[s]})}(s,a)$ if $s \in S_{\mathbb{M}}$. We define by

$$\mathrm{col} = \{ \mu_{\mathrm{M}} \in \Pi^{S}_{\mathcal{M}} \mid \mathrm{M} \subseteq \mathrm{MEC}_{\varphi}(\mathcal{M}), s_{0} \in \tilde{\mathcal{S}}_{\mathrm{M}} \} \subseteq \Pi^{\varphi}_{\mathcal{M}} \quad (38)$$

the set of policies $\mu_{\mathbb{M}}$ such that MC $\mathcal{M}^{\mu_{\mathbb{M}}}$ will reach $\mathcal{S}_{\mathbb{M}}$ w.p.1 from initial state s_0 .

We start by proving Lemma 1. To this end, we need the following auxiliary result showing that the entropy rate for MC is a linear combination of that for each recurrent class.

Claim 1. Given MDP \mathcal{M} and policy $\mu \in \Pi^S_{\mathcal{M}}$, suppose that \mathcal{M}^{μ} has K recurrent classes $R_1, \ldots, R_K \subseteq S$. Let E_k be the entropy rate for the MC restricted on R_k . Then there exists a set of values $\beta(1), \ldots, \beta(K) \in [0,1]$ such that $\sum_{i=1}^K \beta(k) = 1$ and

$$\nabla H(\mathcal{M}^{\mu}) = \sum_{k=1}^{K} \beta(k) E_k.$$

Proof. Let $T = S \setminus \bigcup_{k=1}^K R_k$ be the set of transient states. Let Q_0 and Q_k be the transition matrices from T to T and R_k , respectively. Let π_0 be the row vector of the initial distribution. According to [30, page 593], we have $\sum_{s \in R_k} \pi(s) L(s) = \beta(k) E_k$ and

$$\beta(k) = (\pi_0^T (I - Q_0)^{-1} Q_k + \pi_0^k) \mathbf{e},$$

where e denotes the one-vector with suitable dimension, π_0^T and π_0^k are the initial distributions restricted on T and R_k , respectively. The probability of reaching R_k from $s \in T$ is

$$(I - Q_0)^{-1}Q_k \mathbf{e}.$$
 (39)

Therefore, we have $\sum_{k=1}^{K} (I - Q_0)^{-1} Q_k \mathbf{e} = \mathbf{e}$, which means that $\sum_{k=1}^{K} \beta(k) = 1$. Note that \mathbf{e} in left and right hand side have different dimensions. This completes the proof.

Proof of Lemma 1. By Claim 1, the entropy rate is a linear combination of each recurrent class entropy rate. If there are several recurrent classes, then they must have the same entropy rate. Since MDP is communicating, we can select one recurrent class and make other states reach it w.p.1 by procedure in [30, Page 480]. This new policy has same entropy rate as original policy. Thus, without loss of generality, we assume that \mathcal{M}^{μ} only has one recurrent class R.

Now we prove our result by contradiction. Assume that MC \mathcal{M}^{μ} is not irreducible, i.e., state set $S\setminus R$ is not empty. Since \mathcal{M} is communicating, there exist $r\in R$ and $\hat{a}\in A(r)$ such that $P_{r,\hat{a},\hat{t}}>0$ for some $\hat{t}\notin R$. We define policy $\hat{\mu}$ such that $\hat{\mu}(s,a)=\mu(s,a)$ for $s\neq r$ and $\hat{\mu}(r,\hat{a})=1$. Also, we define

$$\mu_{\epsilon} = (1 - \epsilon)\mu + \epsilon \hat{\mu}$$
, where $\epsilon \in [0, 1)$.

Define $\mathbf{d} \in \mathbb{R}^{|S|}$ such that $\mathbf{d}(s) = \mathbb{P}_{r,s}^{\hat{\mu}} - \mathbb{P}_{r,s}^{\mu}$. Since MC \mathcal{M}^{μ} only contains one recurrent class and some transient states, by (8) of [34], we have

$$\pi^{\mu_{\epsilon}}(s) = \pi^{\mu}(s) + \pi^{\mu}(r) \frac{\epsilon \mathbf{u}(s)}{1 - \epsilon \mathbf{u}(r)},\tag{40}$$

where $\mathbf{u} \in \mathbb{R}^{|S|}$ is the vector such that $\mathbf{u} = \mathbf{d}\mathbf{Z}^{\mu}$ and

$$\mathbf{Z}^{\mu} = (I - \mathbb{P}^{\mu} + (\mathbb{P}^{\mu})^{\star})^{-1},$$

where $(\mathbb{P}^{\mu})^*$ is limit matrix of transition matrix \mathbb{P}^{μ} . Then we define $G(\epsilon) = \nabla H(\mathcal{M}^{\mu_{\epsilon}})$ as a function of ϵ . By (6), we have

$$G(\epsilon) = \pi^{\mu_{\epsilon}}(r)L^{\mu_{\epsilon}}(r) + \sum_{s \in S, s \neq r} \pi^{\mu_{\epsilon}}(s)L^{\mu_{\epsilon}}(s).$$

Let $G_2(\epsilon) = \pi^{\mu_{\epsilon}}(r)L^{\mu_{\epsilon}}(r)$ and $G_1(\epsilon) = G(\epsilon) - G_2(\epsilon)$. Since $L^{\mu_{\epsilon}}(s) = L^{\mu}(s)$ for $s \in S$ such that $s \neq r$, the derivative of $G_1(\epsilon)$ is

$$G_1'(\epsilon) = \sum_{s \in S, s \neq r} \frac{\pi^{\mu}(r)\mathbf{u}(s)}{(1 - \epsilon \mathbf{u}(r))^2} L^{\mu}(s). \tag{41}$$

Therefore, there exists $\epsilon_1 \in [0,1)$ such that for any $\epsilon \in [0,\epsilon_1]$, $G_1'(\epsilon)$ is bounded.

Similarly, for $G_2(\epsilon)$, its derivative is

$$G_2'(\epsilon) = (\pi^{\mu_{\epsilon}})'(r)L^{\mu_{\epsilon}}(r) + \pi^{\mu_{\epsilon}}(r)(L^{\mu_{\epsilon}})'(r).$$

Let $G_4(\epsilon) = \pi^{\mu_{\epsilon}}(r)(L^{\mu_{\epsilon}})'(r)$ and $G_3(\epsilon) = G_2'(\epsilon) - G_4(\epsilon)$. Similarly to (41), we know that, there exists $\epsilon_2 \in [0,1)$ such that for any $\epsilon \in [0,\epsilon_2]$, $G_3(\epsilon)$ is bounded.

Now we focus on $G_4(\epsilon)$. We define

$$\mathbb{P}_s(\epsilon) = \mathbb{P}_{r,s}^{\mu} - \mathbb{P}_{r,s}^{\hat{\mu}} + (\mathbb{P}_{r,s}^{\mu} - \mathbb{P}_{r,s}^{\hat{\mu}}) \log(((1 - \epsilon)\mathbb{P}_{r,s}^{\mu} + \epsilon\mathbb{P}_{r,s}^{\hat{\mu}}))$$

and obtain $G_4(\epsilon) = \pi^{\mu_\epsilon}(r)(\sum_{s \in S} \mathbb{P}_s(\epsilon))$. We define $T = \{s \in S \mid \mathbb{P}_{r,s}^{\hat{\mu}} > 0 \land \mathbb{P}_{r,s}^{\mu} = 0\}$ which is the set of states that can reach from r under $\hat{\mu}$ but cannot reach from r under μ by one step. Note that $\hat{t} \in T$, i.e., T is non-empty. For $s \in S \setminus T$, $\mathbb{P}_s(\epsilon)$ is bounded for sufficiently small ϵ . For $s \in T$, we have

$$\mathbb{P}_s(\epsilon) = -\mathbb{P}_{r,s}^{\hat{\mu}} - \mathbb{P}_{r,s}^{\hat{\mu}} \log(\epsilon \mathbb{P}_{r,s}^{\hat{\mu}}).$$

Therefore, $\mathbb{P}_s(\epsilon) \to +\infty$ as $\epsilon \to 0$. By (40), we know that $\lim_{\epsilon \to 0} G_4(\epsilon) = \pi^{\mu}(r) \lim_{\epsilon \to 0} \mathbb{P}_s(\epsilon) = +\infty$.

In summary, we have

$$G'(\epsilon) = G'_1(\epsilon) + G_3(\epsilon) + G_4(\epsilon).$$

Note we have shown that $G_1'(\epsilon)$ and $G_3(\epsilon)$ are both bounded and $G_4(\epsilon) \to +\infty$ as $\epsilon \to 0$. Therefore, $G'(\epsilon) \to +\infty$ as $\epsilon \to 0$. By Mean Value Theorem [31, Thm. 5.10] there exists $\epsilon_0 > 0$ such that $\nabla H(\mathcal{M}^{\mu_{\epsilon_0}}) = G(\epsilon_0) > G(0) = \nabla H(\mathcal{M}^{\mu})$. However, it contradicts to the fact that μ is the maximum entropy rate policy.

Now, we proceed to prove Proposition 1. To this end, we still need two auxiliary results, listed as Claims 2 and 3.

For any policy $\mu=(\mu_0,\mu_1,\dots)\in \Pi^{\varphi}_{\mathcal{M}}$ and integer n, we define a new truncated policy $\mu^n=(\mu^n_0,\mu^n_1,\dots)\in \Pi_{\mathcal{M}}$ such that: (i) for i< n, we have $\mu^n_i=\mu_i$; and (ii) for $i\geq n$, we have $\mu^n_i(s,a)=\mu_{(\mathcal{S}_{[s]},\mathcal{A}_{[s]})}(s,a)$ if $s\in \mathcal{S}_{\text{MEC}_{\varphi}(\mathcal{M})}$ and $\mu^n_i(s,a)=\mu_i(s,a)$ if $s\in S\setminus \mathcal{S}_{\text{MEC}_{\varphi}(\mathcal{M})}$. The following result shows that it is without loss of generality to consider such truncated policies.

Claim 2. Let $\hat{\Pi}_{\mathcal{M}}^{\varphi} = \bigcup_{\mu \in \Pi_{\mathcal{M}}^{\varphi}} \{\mu^{i}\}_{i=1}^{\infty}$ be the set of truncated policies. Then we have $\hat{\Pi}_{\mathcal{M}}^{\varphi} \subseteq \Pi_{\mathcal{M}}^{\varphi}$ and

$$\sup_{\mu \in \Pi_{\mathcal{M}}^{\varphi}} \nabla H(\mathcal{M}^{\mu}) = \sup_{\mu \in \hat{\Pi}_{\mathcal{M}}^{\varphi}} \nabla H(\mathcal{M}^{\mu}). \tag{42}$$

Proof. For any policy $\mu \in \Pi_{\mathcal{M}}^{\varphi}$, AEC $(\hat{\mathcal{S}}, \hat{\mathcal{A}}) \in \text{AEC}(\mathcal{M})$ and MEC $(\mathcal{S}, \mathcal{A}) \in \text{MEC}_{\varphi}(\mathcal{M})$, let

$$\begin{aligned} & \mathsf{Pr}^{\mu}(\hat{\mathcal{S}}, \hat{\mathcal{A}}) = & \mathsf{Pr}^{\mu}_{\mathcal{M}}(\{\rho \in \mathsf{Path}^{\mu}_{\mathcal{M}} \mid \mathsf{inf}(\rho) = \hat{\mathcal{S}}\}), \\ & \mathsf{Pr}^{\mu}_{M}(\mathcal{S}, \mathcal{A}) = \sum_{(\hat{\mathcal{S}}', \hat{\mathcal{A}}') \in \mathtt{AEC}(\mathcal{S}, \mathcal{A})} \mathsf{Pr}^{\mu}(\hat{\mathcal{S}}', \hat{\mathcal{A}}') \end{aligned} \tag{43}$$

be the probability of staying forever in AEC (\hat{S}, \hat{A}) and the probability of staying forever in AECs in MEC (S, A), respectively. Then entropy rate under μ can be expressed as

$$\nabla H(\mathcal{M}^{\mu}) = \sum_{(\hat{\mathcal{S}}, \hat{\mathcal{A}}) \in AEC(\mathcal{M})} \mathsf{Pr}^{\mu}(\hat{\mathcal{S}}, \hat{\mathcal{A}}) V^{\mu}(\hat{\mathcal{S}}, \hat{\mathcal{A}}), \tag{44}$$

where $V^{\mu}(\hat{\mathcal{S}},\hat{\mathcal{A}})$ is the entropy rate under μ restricted on AEC $(\hat{\mathcal{S}},\hat{\mathcal{A}})$. Note that we have $\sum_{(\hat{\mathcal{S}},\hat{\mathcal{A}})\in \mathtt{AEC}(\mathcal{M})} \mathsf{Pr}^{\mu}(\hat{\mathcal{S}},\hat{\mathcal{A}}) = 1$ since $\mu \in \Pi^{\varphi}_{\mathcal{M}}$. From (44), we further have

$$\nabla H(\mathcal{M}^{\mu}) \leq \sum_{(\mathcal{S}, \mathcal{A}) \in \mathtt{MEC}_{\varphi}(\mathcal{M})} \mathsf{Pr}_{M}^{\mu}(\mathcal{S}, \mathcal{A}) V^{\star}(\mathcal{S}, \mathcal{A}). \tag{45}$$

By Lemma 1, the MC induced by $\mu_{(\mathcal{S},\mathcal{A})}$ will eventually reach some AMEC and visit all states in the AMEC infinitely often. From definition of AMEC, under policy $\mu_{(\mathcal{S},\mathcal{A})}$ LTL task φ can finish w.p.1 over MEC $(\mathcal{S},\mathcal{A})$. Thus $\mu^n \in \Pi^{\varphi}_{\mathcal{M}}$ for any n, i.e., $\hat{\Pi}^{\varphi}_{\mathcal{M}} \subseteq \Pi^{\varphi}_{\mathcal{M}}$. Since $\Pr^{\mu}_{M}(\mathcal{S},\mathcal{A}) = \lim_{n \to \infty} \Pr^{\mu^n}_{M}(\mathcal{S},\mathcal{A})$, according to (45), we know that (42) holds.

For $\mu^n \in \hat{\Pi}^{\varphi}_{\mathcal{M}}$, let $\mathbb{M}(\mu^n)$ be the set of MECs in which it stays forever with non-zero probability. Under policy μ^n , it may happen that: (i) $(\mathcal{S},\mathcal{A}),(\mathcal{S}',\mathcal{A}')\in\mathbb{M}(\mu^n)$ and, (ii) it stays forever in $(\mathcal{S}',\mathcal{A}')$ and stays temporarily at $(\mathcal{S},\mathcal{A})$ with non-zero probability. We now further prove that situation (ii) can be prevented without loss of generality.

Claim 3. For $\mu^n \in \hat{\Pi}^{\varphi}_{\mathcal{M}}$, if (i) and (ii) hold, we can find $\hat{\mu}^n \in \hat{\Pi}^{\varphi}_{\mathcal{M}}$ such that $\nabla H(\mathcal{M}^{\mu^n}) \leq \nabla H(\mathcal{M}^{\hat{\mu}^n})$ and (ii) is false.

Proof. From definition of μ^n , we know that under policy μ^n , once reaching $(S, A) \in MEC_{\varphi}(M)$ at time i with $i \geq n$, it will stay in (S, A) forever and execute policy $\mu_{(S,A)}$ which achieves entropy rate $V^*(S, A)$. From (44), it holds that

$$\nabla H(\mathcal{M}^{\mu^n}) = \sum_{(\mathcal{S}, \mathcal{A}) \in \mathtt{MEC}_{\varphi}(\mathcal{M})} \mathsf{Pr}_M^{\mu^n}(\mathcal{S}, \mathcal{A}) V^{\star}(\mathcal{S}, \mathcal{A}). \tag{46}$$

Then we can select $\hat{\mu}^{\hat{n}} \in \hat{\Pi}^{\varphi}_{\mathcal{M}}$ such that for path satisfying (ii), it only stays forever in either (a) $(\mathcal{S}, \mathcal{A})$ or (b) $(\mathcal{S}', \mathcal{A}')$ and chooses higher entropy rate one between (a) and (b). Then $\nabla H(\mathcal{M}^{\mu^n}) \leq \nabla H(\mathcal{M}^{\hat{\mu}^{\hat{n}}})$ and (ii) no longer holds for $\hat{\mu}^{\hat{n}}$. \square

Now, we proceed to prove Proposition 1 based on the above two claims.

Proof of Proposition 1. For $\mu^n \in \hat{\Pi}^{\varphi}_{\mathcal{M}}$, we can repeatedly use Claim 3 and get a policy $\bar{\mu} \in \hat{\Pi}^{\varphi}_{\mathcal{M}}$ such that for each $(\mathcal{S},\mathcal{A}) \in M(\bar{\mu})$, if from initial state s_0 , it will stay forever in $(\mathcal{S},\mathcal{A})$ and stay temporarily at $(\mathcal{S}',\mathcal{A}') \in \mathrm{MEC}_{\varphi}(\mathcal{M})$ with non-zero probability, then $(\mathcal{S}',\mathcal{A}') \notin M(\bar{\mu})$. Note that $M(\bar{\mu})$ is set of MECs it will stay in forever with non-zero probability under $\bar{\mu}$. Let $\bar{\mu}' \in \Pi_{\mathcal{M}_{\mathcal{S}_{M}(\bar{\mu})}}$ be a policy such that $\bar{\mu}'(s,a) = \bar{\mu}(s,a)$ for $S \setminus \mathcal{S}_{M(\bar{\mu})}$ and $\bar{\mu}'(s,\bar{a}) = 1$ for $s \in \mathcal{S}_{M(\bar{\mu})} \cup \{b\}$. Let reaching probability of $(\mathcal{S},\mathcal{A}) \in \mathrm{MEC}_{\varphi}(\mathcal{M})$ under $\bar{\mu}'$ be

$$\operatorname{Pr}_{R}^{\bar{\mu}'}(\mathcal{S},\mathcal{A}) = \operatorname{Pr}^{\bar{\mu}'}(\{\rho \in \operatorname{Path}^{\bar{\mu}'} \mid \exists s \in \mathcal{S}, s \text{ is in } \rho\}). \tag{47}$$

Then it holds that

$$v^{\bar{\mu}'}(s_0) = \sum_{(\mathcal{S}, \mathcal{A}) \in \mathbb{M}(\bar{\mu})} \mathsf{Pr}_R^{\bar{\mu}'}(\mathcal{S}, \mathcal{A}) V^{\star}(\mathcal{S}, \mathcal{A}) \tag{48}$$

where $v^{\bar{\mu}'}$ is total expected reward vector w.r.t. MDP $\mathcal{M}_{\mathcal{S}_{\mathbb{M}(\bar{\mu})}}$ in (36) and reward $\mathcal{R}_{\mathbb{M}(\bar{\mu})}$ in (37). For $(\mathcal{S},\mathcal{A})\in\mathbb{M}(\bar{\mu})$, if (43) and (47) satisfy $\mathsf{Pr}_{M}^{\bar{\mu}}(\mathcal{S},\mathcal{A})\neq\mathsf{Pr}_{R}^{\bar{\mu}'}(\mathcal{S},\mathcal{A})$, since $\bar{\mu}$ and $\bar{\mu}'$ are same over $S\setminus\mathcal{S}_{\mathbb{M}(\bar{\mu})}$, it means that there exists another $(\mathcal{S}',\mathcal{A}')\in\mathbb{M}(\bar{\mu})$ such that it will stay forever in $(\mathcal{S},\mathcal{A})$ and stay temporarily at $(\mathcal{S}',\mathcal{A}')$ with non-zero probability. This violates property of $\bar{\mu}$. Thus $\mathsf{Pr}_{M}^{\bar{\mu}}(\mathcal{S},\mathcal{A})=\mathsf{Pr}_{R}^{\bar{\mu}'}(\mathcal{S},\mathcal{A})$ for $(\mathcal{S},\mathcal{A})\in\mathbb{M}(\bar{\mu})$. Then combining (46) and (48), it holds that $\nabla H(\mathcal{M}^{\bar{\mu}})=v^{\bar{\mu}'}(s_0)$. Therefore, we have

$$\nabla H(\mathcal{M}^{\mu^{n}}) \leq \nabla H(\mathcal{M}^{\bar{\mu}}) \leq \nabla H(\mathcal{M}^{\mu_{\mathbb{M}(\bar{\mu})}}) \leq \max_{\mu_{\mathbb{M}} \in \text{col}} \nabla H(\mathcal{M}^{\mu_{\mathbb{M}}})$$
(49)

where col and $\mu_{\mathbb{M}(\bar{\mu})}$ are defined in Equation (38). Since μ^n is selected arbitrary, we know that

$$\sup_{\mu \in \hat{\Pi}_{\mathcal{M}}^{\varphi}} \nabla H(\mathcal{M}^{\mu}) \le \max_{\mu \in \text{col}} \nabla H(\mathcal{M}^{\mu}). \tag{50}$$

Furthermore, it is easy to have $col \subseteq \Pi_{\mathcal{M}}^{\varphi}$. Thus

$$\max_{\mu \in \text{col}} \nabla H(\mathcal{M}^{\mu}) \le \sup_{\mu \in \Pi_{\mathcal{M}}^{\varphi}} \nabla H(\mathcal{M}^{\mu}). \tag{51}$$

Finally, combining (42), (50), (51), we know that one of stationary policies in col is a solution of Problem 1 w.r.t. \mathcal{M} . This completes the proof.

Finally, we proceed to prove Proposition 4. The idea is to first translate the entropy rate problem to a total expected

reward problem and then prove that the optimal solution of program (21)-(25) maximizes the total expect reward.

Proof of Proposition 4. Let $Q = T_k \cup L_{k+1} = \hat{R}_{k+1} \setminus \hat{R}_k$. Consider $\tilde{\mathcal{M}}_{k+1} = \hat{\mathcal{M}}_{k+1,\hat{R}_k}$ such that $\hat{\mathcal{M}}_{k+1,\hat{R}_k}$ is defined in (36). Moreover, a reward function $\tilde{\mathcal{R}}$ is equipped with $\tilde{\mathcal{M}}_{k+1}$ satisfying $\tilde{\mathcal{R}}(s,a) = \mathrm{val}(s)$ for $s \in \hat{R}_k$ and $\tilde{\mathcal{R}}(s,a) = 0$ for $s \in Q \cup \{b\}$.

For any policy $\mu \in \Pi^S_{\tilde{\mathcal{M}}_{k+1}}$, let $\mu' \in \Pi^S_{\hat{\mathcal{M}}_{k+1}}$ be the policy such that $\mu'(s,a) = \hat{\mu}_k(s,a)$ for $s \in \hat{R}_k$ and $\mu'(s,a) = \mu(s,a)$ for $s \in Q$. We denote by $\mathcal{H}^{\mu'} \in \mathbb{R}^{|Q|}$ the entropy rate vector such that $\mathcal{H}^{\mu'}(s)$ is the entropy rate of MC $\hat{\mathcal{M}}_{k+1}^{\mu'}$ when the initial state is s. Now we prove that if every state of Q is transient in $\hat{\mathcal{M}}_{k+1}^{\mu'}$, then for $\epsilon > 0$ such that $\operatorname{val} = \operatorname{val}_k + \epsilon$, we have

$$\mathcal{H}^{\mu'}(s) + \epsilon = v^{\mu}_{\tilde{\mathcal{M}}_{k+1}}(s), \forall s \in Q.$$
 (52)

The entropy rate vector can be expressed equivalently as

$$\mathcal{H}^{\mu'} = \mathbb{P}_Q^{\mu'} \mathcal{H}^{\mu'} + \mathbb{P}_{\hat{R}_k}^{\mu'} (\text{val} - \epsilon \mathbf{e}),$$

where $\mathbb{P}_Q^{\mu'}$ and $\mathbb{P}_{\hat{R}_k}^{\mu'}$ are the transition matrices from Q to Q and \hat{R}_k of MC $\hat{\mathcal{M}}_{k+1}^{\mu'}$, respectively. When states in Q are transient, $I - \mathbb{P}_Q^{\mu'}$ is invertible and we get

$$\mathcal{H}^{\mu'} = (I - \mathbb{P}_Q^{\mu'})^{-1} \mathbb{P}_{\hat{R}_k}^{\mu'} (\text{val} - \epsilon \mathbf{e}). \tag{53}$$

Let \mathbb{P}_Q^{μ} and $\mathbb{P}_{\hat{R}_k}^{\mu}$ be transition matrices from Q to Q and \hat{R}_k of MC $\tilde{\mathcal{M}}_{k+1}^{\mu}$, respectively. From [30, (7.1.6)], we have

$$v_{\tilde{\mathcal{M}}_{k+1}}^{\mu,Q} = \mathbb{P}_Q^\mu v_{\tilde{\mathcal{M}}_{k+1}}^{\mu,Q} + \mathbb{P}_{\hat{R}_k}^\mu \mathrm{val}.$$

By $\mathbb{P}_Q^\mu = \mathbb{P}_Q^{\mu'}$, $\mathbb{P}_{\hat{R}_k}^\mu = \mathbb{P}_{\hat{R}_k}^{\mu'}$ and (39), we know that (52) holds. We now prove that the optimal solution $\gamma^\star(s,a)$ of LP (21)-(25) can generate policy that maximizes total expected reward $\tilde{\mathcal{R}}$ w.r.t. $\tilde{\mathcal{M}}_{k+1}$ by equation (26). To this end, for $s \in \hat{R}_k \cup \{b\}$, we add variables $\gamma(s,\bar{a})$ and constraints

$$\gamma(s,\bar{a}) - \sum_{t \in Q} \lambda(t,s) \le 0, s \in \hat{R}_k, \tag{54}$$

$$\gamma(b,\bar{a}) - \sum_{s \in \hat{R}_k} \gamma(s,\bar{a}) \le 0 \tag{55}$$

to (22)-(25) and the objective function is changed to

$$\sum_{s \in \hat{R}_k} \operatorname{val}(s) \gamma(s, \bar{a}). \tag{56}$$

From (7.2.18) in [30], we know that the total expected reward maximization problem of $\tilde{\mathcal{M}}_{k+1}$ with $\tilde{\mathcal{R}}$ is equivalent to the new linear program (22)-(25), (54), (55), (56). Theorem 7.2.18 in [30] proves the existence of optimal basic feasible solution of new-LP and the policy constructed by (26) is stationary policy that maximizes total expected reward $\tilde{\mathcal{R}}$ for $\tilde{\mathcal{M}}_{k+1}$.

For any optimal solution $\gamma^{\star}(s,a)$ of the new-LP, one can argue by contradiction that $\gamma^{\star}(s,\bar{a}) = \sum_{t \in Q} \lambda^{\star}(t,s)$. Therefore, the optimal value of (56) is equal to

$$\sum_{s \in \hat{R}_k} \mathrm{val}(s) \gamma^\star(s, \bar{a}) = \sum_{s \in \hat{R}_k} \sum_{t \in Q} \mathrm{val}(s) \lambda^\star(t, s).$$

Therefore, the optimal values of LP (21)-(25) and new-LP (22)-(25), (54), (55), (56) are equivalent. It means that optimal solution of LP (21)-(25) can generate policy that maximizes total expected reward $\tilde{\mathcal{R}}$ w.r.t. $\tilde{\mathcal{M}}_{k+1}$ by Equations (26). Since $\mathrm{val}(t)>0$ for any $t\in \hat{R}_k$, $s\in Q$ can get positive total expected reward only when s is transient in the induce MC. Thus every state in Q is transient in MC $\hat{\mathcal{M}}_{k+1}^{\tilde{\mu}_{k+1}}$. Then from Equation (52) we know regardless of initial state,

$$\nabla H(\hat{\mathcal{M}}_{k+1}^{\tilde{\mu}_{k+1}}) = \sup_{\mu \in \Pi_{\tilde{\mathcal{M}}_{k+1}}^T} \nabla H(\hat{\mathcal{M}}_{k+1}^{\mu}).$$

From (53), we know in (29), we have $v_{\text{trans}}(s) = \nabla H(\hat{\mathcal{M}}_{k+1}^{\tilde{\mu}_{k+1}})$ when the initial state is s. This completes the proof.

REFERENCES

- Christel Baier and Joost-Pieter Katoen. Principles of Model Checking. MIT press, 2008.
- [2] Calin Belta and Sadra Sadraddini. Formal Methods for Control Synthesis: An Optimization Perspective. Annual Review of Control, Robotics, and Autonomous Systems, 2:115–140, 2019.
- [3] Fabrizio Biondi, Axel Legay, Bo Friis Nielsen, and Andrzej Wasowski. Maximizing entropy over Markov processes. *Journal of Logical and Algebraic Methods in Programming*, 83(5):384–399, 2014.
- [4] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.
- [5] Zdzislaw Burda, Jarek Duda, Jean-Marc Luck, and Bartek Waclaw. Localization of the Maximal Entropy Random Walk. *Physical review letters*, 102(16):160602, 2009.
- [6] Mingyu Cai, Mohammadhosein Hasanbeig, Shaoping Xiao, Alessandro Abate, and Zhen Kan. Modular Deep Reinforcement Learning for Continuous Motion Planning With Temporal Logic. *IEEE Robotics and Automation Letters*, 6(4):7973–7980, 2021.
- [7] Shicong Cen, Chen Cheng, Yuxin Chen, Yuting Wei, and Yuejie Chi. Fast Global Convergence of Natural Policy Gradient Methods with Entropy Regularization. *Operations Research*, 70(4):2563–2578, 2022.
- [8] Krishnendu Chatterjee, Thomas A Henzinger, Barbara Jobstmann, and Rohit Singh. Measuring and Synthesizing Systems in Probabilistic Environments. *Journal of the ACM*, 62(1):1–34, 2015.
- [9] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Prakash Panangaden. Anonymity protocols as noisy channels. *Information and Computation*, 206(2):378–401, 2008.
- [10] T. Chen and T. Han. 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, pages 571–583, 2014.
- [11] Yu Chen, Shaoyuan Li, and Xiang Yin. Entropy Rate Maximization of Markov Decision Processes for Surveillance Tasks. In 22nd IFAC World Congress, pages 5012–5018, 2023.
- [12] Yu Chen, Shuo Yang, Rahul Mangharam, and Xiang Yin. You Don't Know When I Will Arrive: Unpredictable Controller Synthesis for Temporal Logic Tasks. In 22nd IFAC World Congress, pages 3967– 3973, 2023.
- [13] Steven Diamond and Stephen Boyd. CVXPY: A Python-Embedded Modeling Language for Convex Optimization. The J. Machine Learning Research, 17(1):2909–2913, 2016.
- [14] Xuchu Ding, Stephen L. Smith, Calin Belta, and Daniela Rus. Optimal Control of Markov Decision Processes With Linear Temporal Logic Constraints. *IEEE Trans. Automatic Control*, 59(5):1244–1257, 2014.
- [15] Xiaoming Duan and Francesco Bullo. Markov Chain–Based Stochastic Strategies for Robotic Surveillance. Annual Review of Control, Robotics, and Autonomous Systems, 4(1):243–264, 2021.
- [16] Mishel George, Saber Jafarpour, and Francesco Bullo. Markov Chains With Maximum Entropy for Robotic Surveillance. *IEEE Trans. Automatic Control*, 64(4):1566–1580, 2018.
- [17] Meng Guo and Michael M Zavlanos. Probabilistic Motion Planning Under Temporal Tasks and Soft Constraints. *IEEE Trans. Automatic Control*, 63(12):4051–4066, 2018.
- [18] Ernst Moritz Hahn, Mateo Perez, Sven Schewe, Fabio Somenzi, Ashutosh Trivedi, and Dominik Wojtczak. Omega-Regular Objectives in Model-Free Reinforcement Learning. In *International Conference* on Tools and Algorithms for the Construction and Analysis of Systems, pages 395–412. Springer, 2019.

- [19] Michael Hibbard, Yagiz Savas, Bo Wu, Takashi Tanaka, and Ufuk Topcu. Unpredictable Planning Under Partial Observability. In 58th IEEE Conf. on Decision and Control, pages 2271–2277, 2019.
- [20] J. Justesen and T. Hoholdt. Maxentropic Markov chains. IEEE Transactions on Information Theory, 30(4):665–667, 1984.
- [21] Joachim Klein. ltl2dstar-LTL to deterministic Streett and Rabin automata, 2007.
- [22] Hadas Kress-Gazit, Morteza Lahijanian, and Vasumathi Raman. Synthesis for Robots: Guarantees and Feedback for Robot Behavior. Annual Review of Control, Robotics, and Autonomous Systems, 1:211–236, 2018
- [23] Nevena Lazic, Dong Yin, Mehrdad Farajtabar, Nir Levine, Dilan Gorur, Chris Harris, and Dale Schuurmans. A Maximum-Entropy Approach to Off-Policy Evaluation in Average-Reward MDPs. Advances in Neural Information Processing Systems, 33:12461–12471, 2020.
- [24] Nan Li, Ilya Kolmanovsky, and Anouck Girard. Detection-averse optimal and receding-horizon control for Markov decision processes. *Automatica*, 122:109278, 2020.
- [25] Ehsan Nekouei, Takashi Tanaka, Mikael Skoglund, and Karl H. Johansson. Information-theoretic approaches to privacy in estimation and control. *Annual Reviews in Control*, 47:412–422, 2019.
- [26] Gergely Neu, Anders Jonsson, and Vicenç Gómez. A Unified View of Entropy-Regularized Markov Decision Processes. arXiv preprint arXiv:1705.07798, 2017.
- [27] Luyao Niu and Andrew Clark. Optimal Secure Control With Linear Temporal Logic Constraints. *IEEE Trans. on Automatic Control*, 65(6):2434–2449, 2019.
- [28] Brendan O'donoghue, Eric Chu, Neal Parikh, and Stephen Boyd. Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding. J. Optimization Theory and Applications, 169(3):1042–1068, 2016
- [29] Praveen Paruchuri, Milind Tambe, Fernando Ordónez, and Sarit Kraus. Security in Multiagent Systems by Policy Randomization. In *Interna-*

- tional Joint Conf. Autonomous Agents and Multiagent Systems, pages 273–280, 2006.
- [30] Martin L. Puterman. Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc., USA, 1st edition, 1994.
- [31] Walter Rudin et al. *Principles of Mathematical Analysis*, volume 3. McGraw-hill New York, 1976.
- [32] Yagiz Savas, Michael Hibbard, Bo Wu, Takashi Tanaka, and Ufuk Topcu. Entropy Maximization for Partially Observable Markov Decision Processes. *IEEE Trans. on Automatic Control*, 67(12):6948–6955, 2022.
- [33] Yagiz Savas, Melkior Ornik, Murat Cubuktepe, Mustafa O Karabag, and Ufuk Topcu. Entropy Maximization for Markov Decision Processes Under Temporal Logic Constraints. *IEEE Trans. Automatic Control*, 65(4):1552–1567, 2019.
- [34] Paul J Schweitzer. Perturbation theory and finite Markov chains. *Journal of Applied Probability*, 5(2):401–413, 1968.
- [35] MT Thomas and A Thomas Joy. Elements of information theory. Wiley-Interscience, 2006.
- [36] Cameron Voloshin, Hoang Le, Swarat Chaudhuri, and Yisong Yue. Policy Optimization with Linear Temporal Logic Constraints. Advances in Neural Information Processing Systems, 35:17690–17702, 2022.
- [37] Eric M Wolff, Ufuk Topcu, and Richard M Murray. Robust control of uncertain Markov Decision Processes with temporal logic specifications. In 51st IEEE Conf. on decision and control, pages 3372–3379, 2012.
- [38] Shuo Yang and Xiang Yin. Secure Your Intention: On Notions of Pre-Opacity in Discrete-Event Systems. *IEEE Trans. on Automatic Control*, 68(8):4754–4766, 2023.
- [39] Xiang Yin, Bingzhao Gao, and Xiao Yu. Formal synthesis of controllers for safety-critical autonomous systems: Developments and challenges. *Annual Reviews in Control*, page 100940, 2024.
- [40] Wei Zheng, Taeho Jung, and Hai Lin. Privacy-Preserving POMDP Planning via Belief Manipulation. IEEE Control Systems Letters, 6:3415–3420, 2022.