Entropy Maximization for Constrained Markov Decision Processes

Yagiz Savas, Melkior Ornik, Murat Cubuktepe, Ufuk Topcu

***Abstract*—We study the problem of synthesizing a policy that maximizes the entropy of a Markov decision process (MDP) subject to expected reward constraints. Such a policy minimizes the predictability of the paths it generates in an MDP while attaining certain reward thresholds. We first show that the maximum entropy of an MDP can be finite, infinite or unbounded. We provide necessary and sufficient conditions under which the maximum entropy of an MDP is finite, infinite or unbounded. We then present an algorithm to synthesize a policy that maximizes the entropy of an MDP. The proposed algorithm is based on a convex optimization problem and runs in time polynomial in the size of the MDP. Finally, we extend the algorithm to an MDP subject to expected total reward constraints. In numerical examples, we demonstrate the proposed method on different motion planning scenarios and illustrate the trade-off between the predictability of paths and the level of the collected reward.**

I. Introduction

Markov decision processes (MDPs) model sequential decision-making in stochastic systems with nondeterministic choices. A policy, i.e., a decision strategy, resolves the nondeterminism in an MDP and induces a stochastic process. In this regard, an MDP represents an (infinite) family of stochastic processes. In this paper, for a given MDP, we synthesize a policy that induces a process with the maximum entropy, among the ones that attain a certain expected reward threshold.

Entropy, as an information-theoretic quantity, measures the unpredictability of outcomes in a random variable [1]. Considering a stochastic process as an infinite sequence of (dependent) random variables, we define the entropy of a stochastic process as the joint entropy of these random variables by following [2], [3]. Therefore, intuitively, our aim is to obtain a process whose paths accumulate rewards in the most unpredictable way to an observer.

For an MDP, desired behaviors are commonly encoded by reward functions, and the quality of a policy is measured by the expected reward collected by an agent following that policy [4], [5]. Various approaches including value iteration and linear programming have been proposed to synthesize a policy that maximizes the expected reward [6]. However, such approaches are known to yield deterministic policies [7], and paths generated by these policies are usually easy to predict. The approach proposed in this paper, on the other hand, enables the synthesis of a randomized policy with a desired level of quality that minimizes the predictability of the paths it generates.

We observe that, for an MDP, a process with maximum entropy may not exists, i.e., for any given level of entropy, one can find a process whose entropy is greater than that level. In this case, we say that the maximum entropy of the MDP is unbounded. Additionally, if there exists a process with maximum entropy, the entropy of such process can be finite or infinite. Hence, before attempting to synthesize a policy that maximizes the entropy of an MDP, we first verify whether there exists a policy that attains the maximum entropy.

The contributions of this paper are threefold. First, we provide necessary and sufficient conditions on the structure of the MDP under which the maximum entropy of the MDP is finite, infinite or unbounded. We also present a polynomial-time algorithm to check whether the maximum entropy of an MDP is finite, infinite or unbounded. Second, we present a polynomial-time algorithm based on a convex optimization problem to synthesize a policy that maximizes the entropy of an MDP. Lastly, we provide a procedure to obtain a policy that maximizes the entropy of an MDP subject to expected reward constraints.

The applications of this theoretical framework range from motion planning to software security. In a motion planning scenario, for security purposes, an autonomous robot might need to randomize the paths it follows while carrying out a mission which is encoded by certain reward functions [8]. In such a scenario, a policy synthesized by the proposed methods both provides guarantees on the level of the collected rewards and minimizes the predictability of the robot's paths. Additionally, such a policy allows the robot to explore different parts of the environment [9], and behave robustly against uncertainties in the environment [10]. Finally, as it is shown in [2], the maximum information that an adversary can leak from a (deterministic) software, which is modeled as an MDP, can be quantified by computing the maximum entropy of the MDP.

Related Work. The computation of the maximum entropy of an MDP is first considered in [3], where the authors present a robust optimization problem to compute the maximum entropy for an MDP with finite maximum entropy.

Y. Savas, M. Cubuktepe and U. Topcu are with the Department of Aerospace Engineering, University of Texas at Austin, TX, USA. E-mail: {yagiz.savas, mcubuktepe, utopcu}@utexas.edu

M. Ornik is with the Institute for Computational Engineering and Sciences, University of Texas at Austin, TX, USA. E-mail: mornik@ices.utexas.edu

This work was supported in part by the grant W911NF-16-1-0001 from the Defense Advanced Research Projects Agency.

However, their approach does not allow to incorporate additional constraints due to the formulation of the problem. References [2] and [11] computes the maximum entropy of an MDP for special cases without providing a general algorithm. Additionally, the work [2] provides the necessary and sufficient conditions for an interval Markov chain to have a finite maximum entropy.

In [8], the authors maximize the entropy of a *policy* while keeping the expected reward above a threshold. They claim that the entropy maximization problem is not convex. However, their formulation is a special case of the convex optimization problem that we provide in this paper. Therefore, here, we also establish the convexity of their formulation.

We also note that none of the above work discusses the unbounded and infinite maximum entropy for an MDP.

Organization. We provide the preliminary definitions and formal problem statement in Sections II and III, respectively. We analyze the properties of the maximum entropy of an MDP and present an algorithm to synthesize a policy that maximizes the entropy of an MDP in Section IV. We present a procedure to synthesize a policy that maximizes the entropy of an MDP subject to expected reward constraints in Section V. We provide numerical examples in Section VI and conclude the paper in Section VII.

II. Preliminaries

Notation: For a set S, we denote its power set and cardinality by 2^S and $|S|$, respectively. The set of all probability distributions on a finite set S, i.e., all functions $f{:}S{\to}[0,1]$ such that $\sum_{s\in S} f(s){=}1$, is denoted by $\Delta(S)$. For a matrix $P{\in}\mathbb{R}^{n\times n}$, we use P^k and $P^k_{i,j}$ to denote the k-th power of P and the (i,j)-th component of the k-th power of P, respectively. For vectors $a,b{\in}\mathbb{R}^k$, $a{\geqslant}b$ denotes $a_i{\geq}b_i$ for all $i{=}1,\dots,k$. All logarithms are to the base 2 and the set $\mathbb{N}$ denotes $\{0,1,2,\dots\}$.

A. Markov decision processes

Definition 1: A *Markov decision process* (MDP) is a tuple $\mathcal{M}{=}(S,s_0,\mathcal{A},\mathbb{P},\mathcal{R})$ where S is a finite set of states, $s_0{\in}S$ is the initial state, $\mathcal{A}$ is a finite set of actions, $\mathbb{P}{:}S{\times}\mathcal{A}{\times}S{\to}[0,1]$ is a transition function, and $\mathcal{R}{:}S{\times}A{\to}\mathbb{R}^k$ is a k-dimensional immediate reward vector.

We denote the transition probability $\mathbb{P}(s,a,t)$ by $\mathbb{P}_{s,a,t}$, all available actions in a state $s{\in}S$ by $\mathcal{A}(s)$, and i-th coordinate of the reward vector by r_i. The set of successor states for a state action pair (s,a) is defined as $Succ(s,a){:=}\{t{\in}S|\mathbb{P}_{s,a,t}{>}0, a{\in}\mathcal{A}(s)\}$. The *size of an MDP* is the number of triples $(s,a,t){\in}S{\times}\mathcal{A}{\times}S$ such that $\mathbb{P}_{s,a,t}{>}0$.

A *Markov chain* (MC) $\mathcal{C}$ is an MDP such that $|\mathcal{A}|{=}1$. We denote the transition function (matrix) for an MC by $\mathcal{P}$, and the set of successor states for a state $s{\in}S$ by $Succ(s){=}\{t{\in}S|\mathcal{P}_{s,t}{>}0\}$. The *expected residence time* in a state $s{\in}S$ for an MC $\mathcal{C}$ is defined as

$$\xi_s := \sum_{k=0}^{\infty} \mathcal{P}^k_{s_0,s}. \tag{1}$$

The expected residence time ξ_s represents the expected number of visits to state s starting from the initial state [12]. A state $s{\in}S$ is *recurrent* for an MC if and only if $\xi_s{=}\infty$, and is *transient* otherwise; it is *stochastic* if and only if it satisfies $|Succ(s)|{>}1$, and is *deterministic* otherwise; and it is *reachable* if and only if $\xi_s{>}0$, and is *unreachable* otherwise.

Definition 2: A *policy* for an MDP $\mathcal{M}$ is a sequence $\pi{=}\{\mu_0,\mu_1,\dots\}$ where each $\mu_k{:}S{\to}\Delta(\mathcal{A})$ is a function such that $\mu_k(s){\in}\Delta(\mathcal{A}(s))$. A *stationary* policy is a policy of the form $\pi{=}\{\mu,\mu,\dots\}$. For an MDP $\mathcal{M}$, we define the set of all policies and all stationary policies by $\Pi(\mathcal{M})$ and $\Pi^S(\mathcal{M})$, respectively.

We denote the probability of choosing an action $a{\in}\mathcal{A}(s)$ in a state $s{\in}S$ under a stationary policy π by $\pi_s(a)$. For an MDP $\mathcal{M}$, a stationary policy $\pi{\in}\Pi^S(\mathcal{M})$ induces an MC denoted by $\mathcal{M}^\pi$. We refer to $\mathcal{M}^\pi$ as *induced MC* and specify the transition matrix for $\mathcal{M}^\pi$ by $\mathcal{P}^\pi$, whose (s,t)-th component is given by

$$\mathcal{P}^\pi_{s,t} = \sum_{a\in\mathcal{A}(s)} \pi_s(a)\mathbb{P}_{s,a,t}. \tag{2}$$

Throughout the paper, we assume that for a given MDP $\mathcal{M}$, for any state $s{\in}S$ there exists an induced MC $\mathcal{M}^\pi$ for which the state s is reachable. This is a standard assumption for MDPs [13], which ensures that each state in the MDP is reachable under some policy.

An infinite sequence $\varrho^\pi{=}s_0s_1s_2\dots$ of states generated in $\mathcal{M}$ under a policy $\pi{\in}\Pi(\mathcal{M})$ is called a *path* starting from the initial state s_0 and satisfies $\sum_{a_k\in\mathcal{A}(s_k)}\mu_k(s_k)(a_k)\mathbb{P}_{s_k,a_k,s_{k+1}}{>}0$ for all $k{\geq}0$. Any finite prefix of ϱ^π that ends in a state is a finite path fragment. We define the set of all paths and finite path fragments of $\mathcal{M}$ under the policy π by $Paths^\pi(\mathcal{M})$ and $Paths^\pi_{fin}(\mathcal{M})$, respectively.

We use the standard probability measure for an MDP $\mathcal{M}$ under a policy $\pi{\in}\Pi(\mathcal{M})$ as is in [14]. Using this measure, for given $\mathcal{M}$ and $\pi{\in}\Pi(\mathcal{M})$, we define the *expected reward* as the expected sum of the immediate rewards collected by following the policy π, i.e., $\mathbb{E}^\pi[\sum_{t=0}^{\infty}\mathcal{R}(s_t,a_t)]$. Note that the i-th coordinate of the expected reward is given by $\mathbb{E}^\pi[\sum_{t=0}^{\infty}\mathcal{R}_i(s_t,a_t)]$.

B. The entropy of stochastic processes

The *entropy* of a random variable X with countable support $\mathcal{X}$ and probability mass function (pmf) $p(x)$ is defined as

$$H(X) := -\sum_{x\in\mathcal{X}} p(x)\log p(x). \tag{3}$$

We use the convention that $0\log 0{=}0$. Let (X_0,X_1) be a pair of random variables with the joint pmf $p(x_0,x_1)$ and the support $\mathcal{X}\times\mathcal{X}$. The *joint entropy* of (X_0,X_1) is

$$H(X_0,X_1) := -\sum_{x_0\in\mathcal{X}}\sum_{x_1\in\mathcal{X}} p(x_0,x_1)\log p(x_0,x_1), \tag{4}$$

and the *conditional entropy* of X_1 given X_0 is

$$H(X_1|X_0) := -\sum_{x_0\in\mathcal{X}}\sum_{x_1\in\mathcal{X}} p(x_0,x_1)\log p(x_1|x_0). \tag{5}$$

The definitions of the joint and conditional entropies extend to collection of k random variables as it is shown in [1]. A discrete *stochastic process* $\mathbb{X}$ is a discrete time-indexed sequence of random variables, i.e., $\mathbb{X}=\{X_k\in\mathcal{X} : k\in\mathbb{N}\}$.

Definition 3: (Entropy of a stochastic process) [15] The *entropy of a stochastic process* $\mathbb{X}$ is defined as

$$H(\mathbb{X}) := \lim_{k\to\infty} H(X_0, X_1 \ldots, X_k). \tag{6}$$

Note that this definition is different from the *entropy rate* of a stochastic process, which is defined as $\lim_{k\to\infty}\frac{1}{k}H(X_0, X_1 \ldots, X_k)$ when the limit exists [1]. The limit in (6) either converges to a non-negative real number or diverges to positive infinity [15].

An MC $\mathcal{C}$ is equipped with a discrete stochastic process $\{X_k\in S : k\in\mathbb{N}\}$ where each X_k is a random variable over the state space S. For a given k-dimensional pmf $p(s_0, s_1, \ldots, s_k)$, this process respects the *Markov property*, i.e., $p(s_k|s_{k-1}, \ldots, s_0)=p(s_k|s_{k-1})$ for all $k\in\mathbb{N}$. Then, the *entropy of a Markov Chain* $\mathcal{C}$ is given by

$$H(\mathcal{C}) = H(X_0) + \sum_{i=1}^{\infty} H(X_i|X_{i-1}) \tag{7}$$

using (4), (5) and (6). Note that $H(X_0)=0$, since we define an MC with a unique initial state.

For an MDP $\mathcal{M}$, a policy $\pi\in\Pi(\mathcal{M})$ induces a discrete stochastic process $\{X_k\in S : k\in\mathbb{N}\}$. We denote the entropy of an MDP $\mathcal{M}$ under a policy $\pi\in\Pi(\mathcal{M})$ by $H(\mathcal{M},\pi)$.

Definition 4: (Maximum entropy of an MDP) [3] The *maximum entropy of an MDP* $\mathcal{M}$ is

$$H(\mathcal{M}) := \sup_{\pi\in\Pi(\mathcal{M})} H(\mathcal{M},\pi). \tag{8}$$

A policy $\pi^\star\in\Pi(\mathcal{M})$ *maximizes* the entropy of $\mathcal{M}$ if $H(\mathcal{M})=H(\mathcal{M},\pi^\star)$. Finally, we define the properties of the maximum entropy of an MDP as follows.

Definition 5: (The properties of the maximum entropy) The maximum entropy of an MDP $\mathcal{M}$ is

- *finite*, if and only if

$$H(\mathcal{M}) = \max_{\pi\in\Pi(\mathcal{M})} H(\mathcal{M},\pi) < \infty; \tag{9}$$

- *infinite*, if and only if

$$H(\mathcal{M}) = \max_{\pi\in\Pi(\mathcal{M})} H(\mathcal{M},\pi) = \infty; \tag{10}$$

- *unbounded*, if and only if the following two conditions hold.

$$(i) \qquad H(\mathcal{M}) = \sup_{\pi\in\Pi(\mathcal{M})} H(\mathcal{M},\pi) = \infty, \tag{11}$$

$$(ii) \qquad H(\mathcal{M},\pi) < \infty \ \text{ for all } \pi\in\Pi(\mathcal{M}). \tag{12}$$

Although it is not defined here, there is a fourth possible property which is unachievable finite maximum entropy, i.e., $\max_{\pi\in\Pi(\mathcal{M})} H(\mathcal{M},\pi)<H(\mathcal{M})<\infty$. In Theorem 1, we show that it is not possible for the maximum entropy of an MDP to have this property.

III. Problem Statement

The first problem we study concerns the synthesis of a policy that maximizes the entropy of an MDP.

Problem 1: (**Entropy Maximization**) For a given MDP $\mathcal{M}$, provide an algorithm to verify whether there exists a policy $\pi^\star\in\Pi(\mathcal{M})$ such that $H(\mathcal{M})=H(\mathcal{M},\pi^\star)$. If such a policy exists, provide an algorithm to synthesize it. If it does not exist, provide a procedure to synthesize a policy $\pi'\in\Pi(\mathcal{M})$ such that $H(\mathcal{M},\pi')\geq\ell$ for a given constant ℓ.

In the second problem, we introduce the expected reward constraint to the framework. While optimal policies to solve reward maximization problems in the presence of constraints are not generally stationary [5], for simplicity we focus on stationary policies.

Problem 2: (**Constrained Entropy Maximization**) Let $\mathcal{M}$ be an MDP and $\Gamma=[\Gamma_1, \ldots, \Gamma_k]$ be a vector of constants. Verify whether there exists a stationary policy $\pi^\star\in\Pi^S(\mathcal{M})$ that solves the following problem.

$$\underset{\pi\in\Pi^S(\mathcal{M})}{\text{maximize}} \qquad H(\mathcal{M},\pi) \tag{13a}$$

$$\text{subject to:} \qquad \mathbb{E}^\pi\left[\sum_{t=0}^{\infty}\mathcal{R}(s_t,a_t)\right] \succcurlyeq \Gamma. \tag{13b}$$

If such a policy exists, provide a procedure to synthesize it. If it does not exist, provide a procedure to synthesize a policy $\pi'\in\Pi^S(\mathcal{M})$ such that (i) $\mathbb{E}^{\pi'}[\sum_{t=0}^{\infty}\mathcal{R}(s_t,a_t)]\succcurlyeq\Gamma$ and (ii) $H(\mathcal{M},\pi')\geq\ell$ for a given constant ℓ.

IV. Entropy maximization for MDPs

In this section, we focus on the entropy maximization problem. We refer to a policy as an *optimal* policy for an MDP if it maximizes the entropy of the MDP.

A. The entropy of MCs versus MDPs

For an MC, the *local entropy* of a state $s\in S$ is defined as

$$L(s) := -\sum_{t\in S}\mathcal{P}_{s,t}\log\mathcal{P}_{s,t}. \tag{14}$$

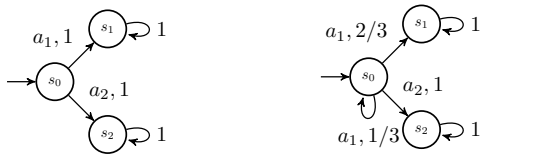
The following proposition characterizes the relationship between the local entropy of states and the entropy of an MC.

Proposition 1: (Theorem 1 in [2]) For an MC $\mathcal{C}$,

$$H(\mathcal{C}) = \sum_{s\in S} L(s)\xi_s. \quad \triangleleft \tag{15}$$

An MC $\mathcal{C}$ has a finite entropy if and only if all of its recurrent states have zero local entropy [2]. That is, $H(\mathcal{C})<\infty$ if and only if for all states $s\in S$, $\xi_s=\infty$ implies $L(s)=0$. If the entropy of an MC is finite, recurrent states have no contribution to the sum in (7). In this case, we take the sum in (15) only over the transient states.

For an MDP, different policies may induce stochastic processes with different entropies. For example, consider the MDP given in Fig. 1a and suppose that the action a_1 at state s_0 is taken with probability ε. If we let ε range over $[0, \frac{1}{2}]$, then the entropy of the resulting stochastic processes ranges over $[0, 1]$. The optimal policy for this MDP is



(a) (b)

Fig. 1: Randomizing actions uniformly at each state may or may not achieve the maximum entropy. The optimal policy for the MDP given in (a) is $\pi_{s_0}(a_1){=}\pi_{s_0}(a_2){=}1/2$, and for the MDP given in (b) is $\pi_{s_0}(a_1){=}2/3$, $\pi_{s_0}(a_2){=}1/3$.

$\pi_{s_0}(a_1){=}\pi_{s_0}(a_2){=}1/2$, which uniformly randomizes actions at each state.

Unlike the MDP given in Fig. 1a, the maximum entropy of an MDP is not generally achieved by a policy that chooses available actions at each state uniformly. For example, consider the MDP given in Fig. 1b. The optimal policy for this MDP is $\pi_{s_0}(a_1){=}2/3$, $\pi_{s_0}(a_2){=}1/3$.

Examples given in Fig. 1 show that finding an optimal policy for an MDP may not be trivial. In general, it might require a search over all randomized policies. The following result establishes the sufficiency of stationary policies to maximize the entropy of MDPs.

Proposition 2: (Proposition 36 in [3]) Stationary policies are sufficient to compute the maximum entropy of MDPs, i.e.,

$$\sup_{\pi\in\Pi(\mathcal{M})} H(\mathcal{M},\pi) = \sup_{\pi\in\Pi^S(\mathcal{M})} H(\mathcal{M},\pi). \quad \triangleleft \tag{16}$$

We can now express the expected residence time in, and the local entropy of, states in terms of stationary policies. For an MC $\mathcal{M}^\pi$ induced by a stationary policy $\pi{\in}\Pi^S(\mathcal{M})$, the expected residence time in a state $s{\in}S$ is given by $\xi_s^\pi{=}\sum_{k=0}^\infty (\mathcal{P}^\pi)^k_{s_0,s}$ due to (1) and (2). Additionally, the local entropy of a state $s{\in}S$ in $\mathcal{M}^\pi$ can be expressed as $L^\pi(s){=}-\sum_{t\in S}\mathcal{P}^\pi_{s,t}\log\mathcal{P}^\pi_{s,t}$ by using (2). Accordingly, the maximum entropy of an MDP $\mathcal{M}$ can be written as

$$H(\mathcal{M}) = \sup_{\pi\in\Pi^S(\mathcal{M})} \Big[\sum_{s\in S}\xi_s^\pi L^\pi(s)\Big]. \tag{17}$$

Note that the right hand side of (17) can still be infinite or unbounded. We analyze the properties of the maximum entropy of MDPs in the next section.

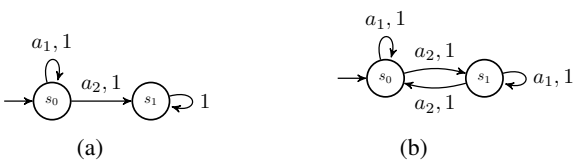
B. Properties of the maximum entropy of MDPs

The maximum entropy of an MDP can be infinite or unbounded even for simple cases. For example, consider MDPs given in Fig. 2. For the MDP shown in Fig. 2a, let the action a_2 be taken with probability $\delta{\in}(0,1]$ in state s_0.

Then, the expected residence time $\xi_{s_0}^\pi$ in state s_0 is equal to $\frac{1}{\delta}$, and the entropy of the induced MC $\mathcal{M}^\pi$ is given by

$$H(\mathcal{M},\pi) = -\frac{(1-\delta)\log(1-\delta)+\delta\log(\delta)}{\delta}, \tag{18}$$

which satisfies $H(\mathcal{M},\pi){\to}\infty$ as $\delta{\to}0$. Note also that if $\delta{=}0$, the entropy of the induced MC is zero due to (15). Hence, the maximum entropy is unbounded, and there is no optimal policy for this MDP.



(a) (b)

Fig. 2: Examples of MDPs with (a) unbounded maximum entropy and (b) infinite maximum entropy.

For the MDP given in Fig. 2b, choosing a policy such that $\pi_i(a_j){>}0$ for $i{=}1,2$, $j{=}1,2$ yields $\xi_{s_0}^\pi{=}\xi_{s_1}^\pi{=}\infty$ and $L^\pi(s_0){>}0$, $L^\pi(s_1){>}0$. Then, the maximum entropy of this MDP is infinite, and the maximum can be attained by any randomized policy.

Examples in Fig. 2 show that we should first verify the existence of optimal policies before attempting to synthesize them. We need the following definitions about the structure of MDPs to state the conditions that cause an MDP to have finite, infinite or unbounded maximum entropy.

A *sub-MDP* of an MDP is a pair (C,D) where $\emptyset{\neq}C{\subseteq}S$ and $D{:}C{\to}2^{\mathcal{A}}$ is a function such that (i) $D(s){\subseteq}\mathcal{A}(s)$ is non-empty for all $s{\in}C$, and (ii) $s{\in}C$ and $a{\in}D(s)$ imply that $Succ(s,a){\subseteq}C$. An *end component* is a sub-MDP (C,D) such that the digraph induced by (C,D) is strongly connected.

Definition 6: A *maximal end component* (MEC) (C,D) in an MDP is an end component such that there is no end component (C',D') with $(C,D){\neq}(C',D')$, and $C{\subseteq}C'$ and $D(s){\subseteq}D'(s)$ for all $s{\in}C$.

A MEC (C,D) in an MDP is *bottom strongly connected* (BSC) if for all $s{\in}C$, $\mathcal{A}(s)\backslash D(s){=}\emptyset$. Every MDP contains at least one BSC MEC, since it would otherwise be possible to induce an MC in which all states are transient. For a given state $s{\in}C$, we define the set of all actions under which the MDP can leave the MEC (C,D) as $D_0(s){:=}\{a{\in}\mathcal{A}(s)|Succ(s,a){\not\subseteq}C\}$. Note that in a BSC MEC (C,D), $D_0(s){=}\emptyset$ for all $s{\in}C$.

Lemma 1: For an MDP $\mathcal{M}$ and the union (C,D) of its MECs, there exists an induced MC $\mathcal{M}^\pi$ for which a state $s{\in}C$ is both stochastic and recurrent if and only if $|\cup_{a\in D(s)}Succ(s,a)|{>}1$. $\triangleleft$

Theorem 1: For an MDP $\mathcal{M}$ and the union (C,D) of its MECs, the following statements hold.

(i) $H(\mathcal{M})$ is infinite if and only if there exists an induced MC for which a state $s{\in}C$ is both stochastic and recurrent.

(ii) $H(\mathcal{M})$ is unbounded if and only if $|\cup_{a\in D(s)}Succ(s,a)|{=}1$ for all $s{\in}C$, and there exists a MEC that is not bottom strongly connected.

(iii) $H(\mathcal{M})$ is finite if and only if it is not infinite and not unbounded. $\triangleleft$

Proofs for above results can be found in [16]. We now provide Algorithm 1 which, for a given MDP $\mathcal{M}$, verifies whether $H(\mathcal{M})$ is finite, infinite or unbounded by checking the necessary conditions that are provided in Theorem 1. For $\mathcal{M}$, its MECs can be found in $\mathcal{O}(|S|^2)$ time [14], $Succ(s,a)$ can be found in $\mathcal{O}(|S|^2|\mathcal{A}|)$ time, and the necessary conditions can be verified in $\mathcal{O}(|S|)$ time. Hence, Algorithm 1 runs in

polynomial-time in the size of $\mathcal{M}$.

Algorithm 1 Verify the properties of the maximum entropy

Require: $\mathcal{M}=(S, s_0, \mathcal{A}, \mathbb{P}, \mathcal{R})$
Return: R
Find MECs (C_i, D_i), $i = 1, ..., k$
Find $Succ(s, a)$ for all $s\in S$, $a\in\mathcal{A}(s)$
R := $\emptyset$;
for i=1, 2, . . . , k **do**
  for s **in** C_i **do**
    if $|\cup_{a\in D_i(s)} Succ(s, a)|>1$ **then**
      R := R $\cup$ {infinite} ;
    if $\mathcal{A}(s)\backslash D_i(s)\neq\emptyset$ **then**
      R := R $\cup$ {unbounded} ;
if infinite$\in$R **then** R=infinite
else if unbounded$\in$R **then** R=unbounded
else R=finite

C. The policy synthesis

We now provide algorithms to synthesize policies that solve the entropy maximization problem.

1) Finite maximum entropy: We first modify a given MDP by making all states in its MECs absorbing.

Proposition 3: Let $\mathcal{M}$ be an MDP such that $H(\mathcal{M})<\infty$, (C, D) be the union of all MECs in $\mathcal{M}$, and $\mathcal{M}'$ be the modified MDP that is obtained from $\mathcal{M}$ by making all states $s\in C$ absorbing, i.e., if $s\in C$, then $\mathbb{P}_{s,a,s}=1$ for all $a\in\mathcal{A}(s)$ in $\mathcal{M}'$. Then, we have

$$H(\mathcal{M}) = H(\mathcal{M}'). \quad \triangleleft \tag{19}$$

We provide the proof for Proposition 3 in [16]. There is a one-to-one correspondence between the paths of $\mathcal{M}$ and $\mathcal{M}'$ since all states in MECs (C, D) must have a single successor state in an MDP with finite maximum entropy due to Theorem 1. Moreover, for a given policy $\pi'\in\Pi^S(\mathcal{M}')$ on $\mathcal{M}'$, the policy $\pi\in\Pi^S(\mathcal{M})$ induced by π' on $\mathcal{M}$ is the same policy with π', i.e. $\pi'=\pi$. Hence, we synthesize an optimal policy for $\mathcal{M}$ by synthesizing an optimal policy for $\mathcal{M}'$.

We use the nonlinear programming problem in (20a)-(20g) to synthesize an optimal policy for $\mathcal{M}'$.

$$\underset{\lambda(s,a),\lambda(s)}{\text{maximize}} \quad -\sum_{s\in S\backslash C}\sum_{t\in S}\eta(s,t)\log(\frac{\eta(s,t)}{\nu(s)}) \tag{20a}$$

subject to:

$$\nu(s) - \sum_{t\in S\backslash C}\eta(t,s) = \alpha(s) \quad \forall s\in S\backslash C \tag{20b}$$

$$\lambda(s) - \sum_{t\in S\backslash C}\eta(t,s) = \alpha(s) \quad \forall s\in C \tag{20c}$$

$$\eta(s,t) = \sum_{a\in\mathcal{A}(s)}\lambda(s,a)\mathbb{P}_{s,a,t} \quad \forall t\in S, \forall s\in S\backslash C \tag{20d}$$

$$\nu(s) = \sum_{a\in\mathcal{A}(s)}\lambda(s,a) \quad \forall s\in S\backslash C \tag{20e}$$

$$\lambda(s,a)\geq 0 \quad \forall a\in\mathcal{A}(s), \forall s\in S\backslash C \tag{20f}$$

$$\lambda(s)\geq 0 \quad \forall s\in C \tag{20g}$$

The decision variables in (20a)-(20c) are $\lambda(s)$ for each $s\in C$, and $\lambda(s, a)$ for each $s\in S\backslash C$ and each $a\in\mathcal{A}(s)$. The function $\alpha: S\to\{0, 1\}$ satisfies $\alpha(s_0)=1$ and $\alpha(s)=0$ for all $s\in S\backslash\{s_0\}$. Variables $\eta(s, t)$ and $\nu(s)$ are functions of $\lambda(s, a)$, and used just to simplify the notation.

The constraints (20b)-(20c) represent the balance between the "inflow" to and "outflow" from states. The constraints (20d) and (20e) are used to simplify the notation and define the variables $\eta(s, t)$ and $\nu(s)$, respectively. The constraints (20f) and (20g) ensure that the expected residence time in the state-action pair (s, a) and the probability of reaching the state s is non-negative, respectively. We refer the reader to [12], [7] for further details about the constraints.

Proposition 4: The nonlinear program in (20a)-(20g) is convex. $\triangleleft$

The above result indicates that a global maximum for the problem in (20a)-(20g) can be computed efficiently. We now introduce Algorithm 2 to synthesize an optimal policy for a given MDP with finite maximum entropy.

Theorem 2: Let $\mathcal{M}$ be an MDP such that $H(\mathcal{M})<\infty$ and (C, D) be the union of all MECs in $\mathcal{M}$. For the input $(\mathcal{M}, C)$, Algorithm 2 returns an optimal policy $\pi^\star\in\Pi^S(\mathcal{M})$ for $\mathcal{M}$, i.e. $H(\mathcal{M}, \pi^\star)=H(\mathcal{M})$. $\triangleleft$

Proofs for Proposition 4 and Theorem 2 are provided in [16]. Computationally, the most expensive step of Algorithm 2 is to solve the problem in (20a)-(20g). A solution whose objective value is arbitrarily close to the optimal value of (20a) can be computed in time polynomial in the size of $\mathcal{M}$ via interior-point methods [17]. Hence, the time complexity of Algorithm 2 is polynomial in the size of $\mathcal{M}$.

Algorithm 2 Synthesize the maximum entropy policy

Require: $\mathcal{M}=(S, s_0, \mathcal{A}, \mathbb{P}, \mathcal{R})$ and C.
Return: An optimal policy $\pi^\star$ for $\mathcal{M}$
1: Form the modified MDP $\mathcal{M}'$.
2: Solve (20a)-(20g) for ($\mathcal{M}'$, C), and obtain $\lambda^\star(s, a)$.
3: **for** $s\in S$ **do**
  if $s\notin C$ **then**
    if $\sum_{a\in\mathcal{A}(s)}\lambda^\star(s,a)>0$ **then**
      $\pi^\star_s(a):=\frac{\lambda^\star(s,a)}{\sum_{a\in\mathcal{A}(s)}\lambda^\star(s,a)}$
    else
      $\pi^\star_s(a):=1$ for an arbitrary $a\in\mathcal{A}(s)$,
  else
    $\pi^\star_s(a):=1$ for an arbitrary $a\in\mathcal{A}(s)$.

2) Unbounded maximum entropy: There is no optimal policy for this case due to (11)-(12). Therefore, for a given MDP $\mathcal{M}$ and a constant ℓ, we aim to synthesize a policy $\pi'\in\Pi(\mathcal{M})$ such that $H(\mathcal{M}, \pi')\geq\ell$. Let S_B be the union of all bottom strongly connected MECs of $\mathcal{M}$, which can be found by slightly modifying Algorithm 1. We form the modified MDP $\mathcal{M}'$ by making all states $s\in S_B$ absorbing. It can be easily shown that $H(\mathcal{M}')=H(\mathcal{M})$. We now introduce a feasibility problem by removing the objective (20a) and

adding

$$-\sum_{s\in S\setminus S_B}\sum_{t\in S}\eta(s,t)\log(\frac{\eta(s,t)}{\nu(s)})\geq\ell \tag{21}$$

to the constraints in (20b)-(20g). By solving the resulting convex feasibility problem for ($\mathcal{M}'$, S_B, ℓ) and using the step 3 of Algorithm 2, we obtain the desired policy π'.

3) Infinite maximum entropy: In this case, for a given MDP $\mathcal{M}$ with the union (C,D) of its MECs, there exists at least one state $s^\star\in C$ such that $|\cup_{a\in D(s^\star)}Succ(s^\star,a)|>1$ due to Theorem 1. We aim to synthesize a policy that induces an MC for which the state $s^\star$ is both stochastic and recurrent. For simplicity, we assume that all MECs in $\mathcal{M}$ are BSC, and there exists only one state $s^\star$ such that $|\cup_{a\in D(s^\star)}Succ(s^\star,a)|>1$. We form the modified MDP $\mathcal{M}'$ by replacing each MEC with an absorbing state. Let C' be the union of all those absorbing states, and $C_\star$ be the absorbing state that is replaced with the MEC that $s^\star$ is contained in. We solve the problem in (20a)-(20g) for ($\mathcal{M}'$, C') together with the constraint $\lambda(C_\star)>0$. We use step 3 of Algorithm 2 to obtain a policy for states $s\notin C$ in $\mathcal{M}$, and choose actions in state $s^\star$ such that $|Succ(s^\star)|>1$ in the induced MC. By construction, the state $s^\star$ is both stochastic and recurrent in the induced MC, and due to Proposition 1, the entropy of the induced MC is infinite.

V. Constrained Entropy Maximization for MDPs

In this section, we focus on the constrained entropy maximization problem. We refer to a policy as an *optimal* policy for an MDP if it solves the problem in (13a)-(13b).

The expected reward constraint (13b) can be written [5] as

$$\sum_{s\in S}\sum_{a\in\mathcal{A}(s)}\lambda(s,a)\mathcal{R}_i(s,a)\geq\Gamma_i, \tag{22}$$

for all $i=1,\dots,k$ where $\lambda(s,a)$ is the expected residence time in the state-action pair (s,a). For an MDP $\mathcal{M}$, let S_B be the union of all states in its BSC MECs. We assume that, for all $s\in S_B$, the reward function satisfies $\mathcal{R}_i(s,a)=0$ for all $i=1,\dots,k$. Similar assumptions are standard in reward maximization problems [6] to avoid infinite maximum reward. We also assume that the expected reward threshold Γ is achievable, i.e., $\max_\pi\mathbb{E}^\pi[\sum_{t=0}^\infty\mathcal{R}(s_t,a_t)]\succcurlyeq\Gamma$, which can be verified in time polynomial in the size of $\mathcal{M}$ by solving a linear optimization problem [7].

We consider three cases to verify the existence, and the synthesis, of an optimal policy.

1) Finite maximum entropy: For a given MDP $\mathcal{M}$ with the union (C,D) of its MECs, we form the modified MDP $\mathcal{M}'$ by making all states $s\in C$ absorbing, as explained in Section IV. Since all MECs are BSC in an MDP with finite maximum entropy due to Theorem 1, the expected reward in $\mathcal{M}$ and $\mathcal{M}'$ are the same. We then solve the problem in (20a)-(20g) for $(\mathcal{M}',C,\Gamma)$ together with the constraint (22). There always exists an optimal policy for $\mathcal{M}'$, under the assumption that the expected reward threshold is achievable. Finally, we synthesize an optimal policy by using step 3 of Algorithm 2.

2) Unbounded maximum entropy: In this case, depending on the expected reward constraint, the maximum constrained entropy can be either finite or unbounded. For a given MDP $\mathcal{M}$, let S_B be the union of all states in its BSC MECs. We form the modified MDP $\mathcal{M}'$ by making all states $s\in S_B$ absorbing. Then, we solve the problem in (20a)-(20g) for $(\mathcal{M}',S_B,\Gamma)$ together with the constraint (22). If the optimal value of this problem is bounded, we synthesize an optimal policy through step 3 of Algorithm 2. Otherwise, there is no optimal policy for $\mathcal{M}$, in which case we synthesize a policy π' such that the induced MC satisfies (i) $\mathbb{E}^{\pi'}[\sum_{t=0}^\infty\mathcal{R}(s_t,a_t)]\succcurlyeq\Gamma$ and (ii) $H(\mathcal{M},\pi')\geq\ell$ for a given constant ℓ. We obtain such a policy by solving a convex feasibility problem as it is explained in Section IV. Note that, in this case, we also need to add the constraint (22) to the feasibility problem to ensure that the synthesized policy satisfies the expected reward constraint.

3) Infinite maximum entropy: In this case, depending on the expected reward constraint, the maximum constrained entropy can be either finite, infinite, or unbounded. For a given MDP $\mathcal{M}$ with the union (C,D) of its MECs, there exists at least one state $s^\star\in C$ such that $|\cup_{a\in D(s^\star)}Succ(s^\star,a)|>1$ due to Theorem 1. For simplicity, we assume there exists only one state $s^\star$ with this property. We form the modified MDP $\mathcal{M}'$ by replacing each BSC MEC in $\mathcal{M}$ with an absorbing state. Let C' be the union of all those absorbing states, and $C_\star$ be the absorbing state that is replaced with the MEC that $s^\star$ is contained in. We solve the problem in (20a)-(20g) for ($\mathcal{M}'$, C', Γ) together with the constraints (22) and $\lambda(C_\star)>0$. If the problem is infeasible, then the maximum constrained entropy is not infinite, in which case we remove the constraint $\lambda(C_\star)>0$ and solve the problem again. If the maximum value of the resulting problem is arbitrarily large, then we follow the procedure explained in the unbounded case; if it is bounded, then we synthesize the optimal policy through step 3 of Algorithm 2.

VI. Examples

In this section, we illustrate the proposed methods on different motion planning scenarios. All computations are run on an 2.2 GHz dual core desktop with 8 GB RAM. All optimization problems are solved by using the splitting conic solver (SCS) [18] in CVXPY [19].

A. Relation between entropy and predictability

In this example, we consider an agent whose aim is to complete a task while leaking minimum information about its paths to an observer.

The agent moves in the grid world shown in Fig. 3 and starts from the brown state. The red and green states are absorbing, i.e., once entered those states cannot be left. The agent has four actions in all other states, namely left, right, up and down. A transition to the chosen direction occurs with probability (w.p.) 1 if the successor state in that direction is not a wall. If it is a wall, e.g., left in brown state, the agent stays in the same state w.p. 1.

The agent's task is to surveil blue states until reaching the green state in a limited time. Such a task can be encountered

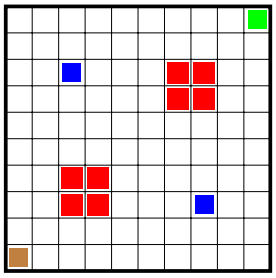


Fig. 3: Grid world environment. The brown and green states are the initial and target states, respectively. The red states are absorbing. The agent surveils blue states.

in a scenario where an unmanned aerial vehicle is required to perform a surveillance mission over designated areas, and reach a charging station before running out of energy. We describe the task using a three dimensional immediate reward vector as follows. Let B, G, R be the sets of blue, green and red states, respectively. The first reward function $\mathcal{R}_1$ is for surveilling blue states and such that $\mathcal{R}_1(s,a){=}\sum_{t\in B}\mathbb{P}_{s,a,t}$ for all $s{\in}S$. The second function $\mathcal{R}_2$ is for reaching the green state and such that $\mathcal{R}_2(s,a){=}\sum_{t\in G}\mathbb{P}_{s,a,t}$ for all $s{\in}S$. Finally, the third reward function $\mathcal{R}_3$ is for limiting the time spent in the environment and such that $\mathcal{R}_3(s,a){=}{-}1$ for all $s{\in}S\backslash\{G\cup R\}$ and $\mathcal{R}_3(s,a){=}0$ for all $s{\in}G\cup R$. We require the agent to surveil blue states at least γ times, i.e., $\mathbb{E}^\pi[\sum_{t=0}^{\infty}\mathcal{R}_1(s_t,a_t)]{\geq}\gamma$, while reaching the green state w.p. 1, i.e., $\mathbb{E}^\pi[\sum_{t=0}^{\infty}\mathcal{R}_2(s_t,a_t)]{\geq}1$, in at most 150 steps, i.e., $\mathbb{E}^\pi[\sum_{t=0}^{\infty}\mathcal{R}_3(s_t,a_t)]{\geq}{-}150$. More specifically, we parameterize the number of visits to blue states by γ and investigate the effect of having different γ values on the predictability of the agent's paths.

We consider an observer that is aware of the agent's task, knows the transition probabilities exactly, and runs yes-no probes in each state to determine the successor state of the agent, i.e., probes that return an answer yes if the agent moves to the predicted successor state and no otherwise. The average number of yes-no probes run in a state is the expected number of observations needed by the observer to determine the correct successor state in that state [8]. The observer uses the Huffman procedure [20] to minimize the required number of probes. Let $\mathcal{P}_s{=}(\mathcal{P}_{s,1},\mathcal{P}_{s,2},\ldots,\mathcal{P}_{s,n})$ be the transition probabilities from state s to successor states sorted in decreasing order. The number of yes-no probes run in state s is denoted by $\Upsilon_s{=}\mathcal{P}_{s,1}{+}\ldots{+}(n{-}1)\mathcal{P}_{s,n-1}{+}(n{-}1)\mathcal{P}_{s,n}$. The expected number of observations required to determine the agent's path is given by $O_{avg}{=}\sum_s\xi_s\Upsilon_s$, which weighs the required number of probes in each state with the expected residence time in the state. We refer the reader to [8] for further details about the observer model.

We use six increasingly restrictive γ values which are $\gamma{=}10,20,\ldots,60$. The maximum entropy of the MDP subject to reward constraints is finite for all γ values. Through the procedure explained in Section V, we synthesize a policy for the agent for each γ value. The MDP has 100 states and 370 transitions, and the longest computation time, which is for $\gamma{=}60$, is 11.3 seconds.

The entropy of Markov chains induced by the synthesized policies and the average number of observations required to predict the agent's paths are shown in Fig. 4. As the agent is required to visit blue states more often with increasing γ values, the entropy of the induced MC decreases and the prediction requires fewer observations on average. Therefore, the agent faces with a trade-off between collecting rewards and following unpredictable paths to reach the green state.

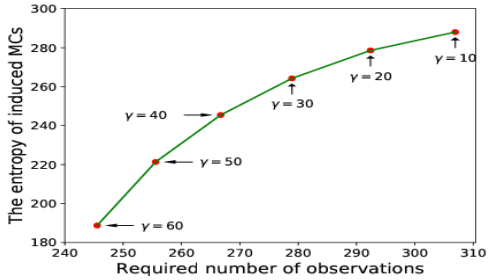


Fig. 4: The relation between the maximum entropy of an MDP subject to reward constraints and the required number of observations to predict the agent's paths.

B. Predictability in a randomly generated MDP

In [8], the authors propose to maximize the entropy of an agent's *policy* that minimizes the information leaked to an observer about the movements of the agent. In this example, we illustrate that a policy that maximizes the entropy of an MDP yields less predictable paths than a policy that maximizes the policy of an agent.

We randomly generate an MDP to show the importance of utilizing the stochasticity in the environment. The MDP has 200 states, where each state has 8 randomly selected successor states. We choose four states, make them absorbing, and label three of them as "unsafe" states and the remaining one as the "target" state. The agent has 5 actions at each state, for which the transition probabilities to successor states are assigned randomly.

The agent's task is to reach the target state in a limited time while avoiding unsafe states in the MDP. We describe the task using a two dimensional immediate reward vector as follows. Let U and T be the set of unsafe and target states, respectively. The first reward function $\mathcal{R}_1$ is for reaching the target state and such that $\mathcal{R}_1(s,a){=}\sum_{t\in T}\mathbb{P}_{s,a,t}$ for all $s{\in}S$. The second reward function $\mathcal{R}_2$ is for limiting the time spent in the environment and such that $\mathcal{R}_2(s,a){=}{-}1$ for all $s{\in}S\backslash\{U\cup T\}$ and $\mathcal{R}_2(s,a){=}0$ for all $s{\in}U{\cup}T$. We require the agent to reach the target state w.p. γ, i.e., $\mathbb{E}^\pi[\sum_{t=0}^{\infty}\mathcal{R}_1(s_t,a_t)]{\geq}\gamma$, in at most 200 steps, i.e., $\mathbb{E}^\pi[\sum_{t=0}^{\infty}\mathcal{R}_2(s_t,a_t)]{\geq}{-}200$. Similar to the previous example, we investigate the effect of reaching the target state, i.e., completing the task, with different probabilities by using γ as a parameter.

We assume the existence of an observer in the environment, which aims to predict the agent's paths by using a procedure that is explained in the previous example.

We compare the proposed method with weighted maximum entropy (WME) and binary search for randomization

linear programming ($BRLP$) algorithms which are introduced in [8] for randomizing an agent's policy to minimize predictability. We remark that in [8], the authors claim that WME algorithm is non-convex and cannot be solved in polynomial-time. However, its convexity can be proven by Proposition 4 since it solves a special case of the convex optimization problem given in (20), i.e., it is equivalent to problem in (20) when transition probabilities are either 0 or 1. We refer the reader to [8] for further details about the WME and $BRLP$ algorithms.

The maximum probability γ of completing the task is obtained as γ=0.75 by solving a linear programming problem introduced in [12]. We use six different γ values which are γ=$0.5, 0.55, \ldots, 0.75$. The maximum entropy of the MDP subject to reward constraints is finite for all γ values. The MDP has 200 states and 1572 transitions. Solving the optimization problems takes at most 52.7, 15.6, and 17 seconds for the proposed method, WME and $BRLP$ algorithms, respectively.

The required number of observations to predict the agent's paths for different γ values is shown in Fig. 5. As the probability of completing the task decreases, the randomness of the agent's paths increases and the prediction requires more observations on average. The proposed method (green) requires twice as many observations as the $BRLP$ algorithm (red) when γ=0.5. Note also that the WME algorithm (blue) cannot achieve better predictability results than the proposed method because it does not exploit the inherent stochasticity in the environment and rely solely on the randomization of the agent's actions to generate unpredictable paths.

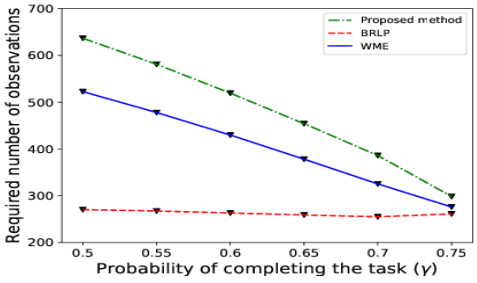


Fig. 5: The trade-off between the probability of completing the task and predictability of paths. $BRLP$ and WME are algorithms proposed in [8] to randomize the agent's actions.

VII. Conclusions

In this paper, we study the problem of synthesizing a policy that maximizes the entropy of an MDP subject to expected reward constraints. We show that the maximum entropy of an MDP can be either finite, infinite or unbounded, and present an algorithm to verify the property of the maximum entropy for a given MDP. We present an algorithm, which is based on a convex optimization problem and runs in time-polynomial in the size of the MDP, to synthesize a policy that maximizes the entropy of an MDP. Finally, we provide a procedure to obtain a policy that maximizes the entropy of an MDP subject to expected reward constraints, and demonstrate the proposed procedure on motion planning scenarios. It is observed that as the level of the expected reward increases, the maximum entropy decreases, which in turn, results in more predictable paths.

References

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, Inc., 2006.

[2] F. Biondi, A. Legay, B. F. Nielsen, and A. Wsowski, "Maximizing entropy over Markov processes," *Journal of Logical and Algebraic Methods in Programming*, vol. 83, no. 5, pp. 384 – 399, 2014.

[3] T. Chen and T. Han, "On the complexity of computing maximum entropy for Markovian models," *Technical Report, Middlesex University London*, 2014. [Online]. Available: http://www.cs.mdx.ac.uk/staffpages/taoluechen/pub-papers/fsttcs14-full.pdf

[4] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[5] E. Altman, *Constrained Markov Decision Processes*. Chapman and Hall/CRC, 1999.

[6] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[7] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1994.

[8] P. Paruchuri, M. Tambe, F. Ordóñez, and S. Kraus, "Security in multiagent systems by policy randomization," in *Joint Conference on Autonomous Agents and Multiagent Systems*, 2006, pp. 273–280.

[9] M. Saerens, Y. Achbany, F. Fouss, and L. Yen, "Randomized shortest-path problems: Two related models," *Neural Computation*, vol. 21, no. 8, pp. 2363–2404, 2009.

[10] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in *International Conference on Machine Learning*, vol. 70, 2017, pp. 1352–1361.

[11] F. Biondi, A. Legay, P. Malacaria, and A. Wasowski, "Quantifying information leakage of randomized protocols," *Theoretical Computer Science*, vol. 597, no. C, pp. 62–87, 2015.

[12] K. Etessami, M. Kwiatkowska, M. Y. Vardi, and M. Yannakakis, "Multi-objective model checking of Markov decision processes," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 2007, pp. 50–65.

[13] X. Ding, S. L. Smith, C. Belta, and D. Rus, "Optimal control of Markov decision processes with linear temporal logic constraints," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1244–1257, 2014.

[14] C. Baier and J.-P. Katoen, *Principles of Model Checking*. MIT Press, 2008.

[15] F. Biondi, *Markovian Processes for Quantitative Information Leakage*. PhD thesis, IT University of Copenhagen, 2014.

[16] Y. Savas, M. Ornik, M. Cubuktepe, and U. Topcu, "Entropy maximization for Markov decision processes under temporal logic constraints," *arXiv:1807.03223 [math.OC]*, 2018.

[17] Y. Nesterov and A. Nemirovski, *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.

[18] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd, "Conic optimization via operator splitting and homogeneous self-dual embedding," *Journal of Optimization Theory and Applications*, vol. 169, no. 3, pp. 1042–1068, June 2016.

[19] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[20] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, 1952.