Entropy Maximization for Partially Observable Markov Decision Processes

Yagiz Savas, Michael Hibbard, Bo Wu, Takashi Tanaka, and Ufuk Topcu

***Abstract*—We study the problem of synthesizing a controller that maximizes the entropy of a partially observable Markov decision process (POMDP) subject to a constraint on the expected total reward. Such a controller minimizes the predictability of an agent's trajectories to an outside observer while guaranteeing the completion of a task expressed by a reward function. Focusing on finite-state controllers (FSCs) with deterministic memory transitions, we show that the maximum entropy of a POMDP is lower bounded by the maximum entropy of the parameteric Markov chain (pMC) induced by such FSCs. This relationship allows us to recast the entropy maximization problem as a so-called parameter synthesis problem for the induced pMC. We then present an algorithm to synthesize an FSC that locally maximizes the entropy of a POMDP over FSCs with the same number of memory states. In a numerical example, we highlight the benefit of using an entropy-maximizing FSC compared with an FSC that simply finds a feasible policy for accomplishing a task.**

***Index Terms*—Autonomous systems, entropy, stochastic processes.**

I. Introduction

The information-theoretic concept of entropy [1] quantifies the uncertainty of outcomes in a random variable. We consider a sequential decision-making framework of partially observable Markov decision processes (POMDPs) in which an entropy-based reward is introduced in addition to the classical state-dependent reward. Specifically, we seek an entropy-maximizing controller that ensures the expected state-dependent reward remains above a given threshold. Intuitively, the entropy reward promotes the unpredictability of the controlled process to an observer. Therefore, the POMDP formulation considered provides a framework for sequential decision-making in stochastic environments with imperfect information and nondeterministic choices, where a task should be accomplished unpredictably.

A POMDP controller resolves the nondeterminism and induces a stochastic process whose unpredictability we quantify by defining the entropy as the joint entropy of a sequence of random variables [2], [3]. Our main objective is to synthesize a controller that induces a process whose realizations accumulate rewards most unpredictably to an outside observer. Controller synthesis problems for POMDPs are notoriously hard to solve. Optimal controllers must often use the full observation history, which makes searching for them undecidable in the infinite-horizon case and PSPACE-complete in the finite-horizon case [4], [5]. For computational tractability, controllers are often restricted to finite states representing finite observation memory [6]. Furthermore, in contrast to classical POMDP problems with deterministic optimal controllers, problems adopting information-theoretic performance criteria typically admit randomized controllers specifying probability distributions for action selection.

We synthesize a randomized finite-state controller (FSC) for a POMDP that specifies a probability distribution over actions for each memory state [7]. Particularly, we consider the entropy maximization problem over all FSCs with a fixed number of memory states. A key observation is that one can use a parameteric Markov chain (pMC) to succinctly represent the product between a POMDP and the set of all FSCs with a fixed number of memory states [8], [9]. By restricting our attention to FSCs with deterministic memory transitions, we recast the POMDP controller synthesis problem as a so-called parameter synthesis problem for a pMC whose entropy we aim to maximize. We first derive a system of recursive equations for entropy maximization and prove that the maximum entropy of a pMC induced from a POMDP by FSCs with deterministic memory transitions lower bounds the POMDP's maximum entropy. Furthermore, we introduce a specific FSC memory transition function using which one can monotonically increase the entropy of the induced stochastic processes by increasing the number of memory states in the FSC. Finally, we present an algorithm, based on a nonlinear optimization problem (NLP), to synthesize FSCs that maximize the entropy of a pMC subject to expected reward constraints.

Related work: A preliminary version of this article appeared in [10], where we present solutions for entropy maximization over FSCs with a *specific memory transition function* and the same number of memory states. This extended version includes detailed proofs for all theoretical results, a NLP formulating the entropy maximization over all deterministic FSCs with the same number of memory states, and a new numerical example. We refer the interested reader to [10] for further numerical examples.

Recently, we showed in [3] that an entropy-maximizing controller for a fully observable MDP can be synthesized efficiently by solving a convex optimization problem. Moreover, we established that for an MDP with finite maximum entropy, it is sufficient to focus only on memoryless controllers to induce a process with maximum entropy. It is known [11] that synthesizing a controller accumulating a desired level of total reward in a POMDP is, in general, intractable. Therefore, partial observability in the system model dramatically changes the complexity of the problem. As a result, in this article, we focus on FSCs and present a NLP with bilinear constraints to synthesize entropy-maximizing controllers for POMDPs.

In POMDPs, entropy has often been used for active sensing applications [12]–[14], where an agent seeks to select actions that decrease its uncertainty by taking actions that minimize the entropy of a probability distribution. Such a distribution typically expresses the agent's belief on the task-relevant aspects of the environment. Here, we consider an agent that aims to maximize the entropy of its *true state trajectories* instead of

Manuscript received 16 May 2021; revised 31 December 2021; accepted 7 June 2022. Date of publication 16 June 2022; date of current version 5 December 2022. This work was supported in part by the AFRL under Grant FA9550-19-1-0169, in part by the DARPA under Grant D19AP00004, Grant D19AP00078, and Grant FOA-AFRL-AFOSR-2019-0003, and in part by the NSF under Grant 1944318. Recommended by Associate Editor R. Jain. *(Yagiz Savas and Michael Hibbard contributed equally to this work.) (Corresponding author: Yagiz Savas.)*

The authors are with the Department of Aerospace Engineering and Engineering Mechanics, Oden Institute for Computational Engineering and Sciences, University of Texas at Austin, Austin, TX 78712 USA (e-mail: yagiz.savas@utexas.edu; mwhibbard@utexas.edu; ttanaka@utexas.edu; utopcu@utexas.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAC.2022.3183564.

Digital Object Identifier 10.1109/TAC.2022.3183564

minimizing the entropy of its *final belief distribution*. Therefore, despite the similarity of the information-theoretic measures considered, the problem studied in this article and the developed solution approach significantly differ from the ones investigated in active sensing literature.

In the reinforcement learning literature, the entropy of a controller has been used as a regularization term in an agent's objective to balance the tradeoff between exploration and exploitation [15]. As discussed in [16], using a controller with high entropy, an agent can learn various ways of completing a task, leading to a greater robustness when subsequently fine-tuned to specific scenarios. The aforementioned work concerns the synthesis of a controller that balances the accumulated reward and the entropy of the induced process in a fully observable setting. Here, we aim to synthesize a controller that maximizes the entropy in a partially observable setting while ensuring the accumulation of a desired level of total reward.

A range of solution techniques exist for POMDP controller synthesis using FSCs. For deterministic FSCs, existing approaches include branch-and-bound method [6], automaton learning-based method [17], and expectation–maximization [18], which all focus on finding an optimal transition structure for the FSC. As for randomized FSCs, in addition to the transition structure, one also needs to optimize the probabilistic transition probabilities between FSC states and the action selection probabilities. To this end, researchers propose solutions using policy iteration [7], [19], [20], gradient descent [21], and nonlinear optimization [22], [23]. However, these results only consider state-dependent reward optimization or the satisfaction of a given specification. In contrast, we consider the synthesis of FSCs for entropy maximization, which is a nonlinear objective that requires a new optimization formulation as well as solution techniques.

Contribution: This article has four main contributions. First, we derive a system of recursive equations, the fixed-point of which corresponds to the maximum entropy of a POMDP. Second, by restricting attention to FSCs with deterministic memory transitions, we prove that the maximum entropy of the induced pMC is a lower bound on the maximum entropy of the POMDP. Third, we present a NLP whose solution provides a controller maximizing the entropy of the POMDP over all deterministic FSCs with the same number of memory states. Finally, for deterministic FSCs, we propose a specific memory transition function that increases the entropy of the induced stochastic process with respect to an increasing number of memory states.

II. Preliminaries

We denote the power set and cardinality of a set $\mathcal{S}$ by $2^{\mathcal{S}}$ and $|\mathcal{S}|$, respectively. The set of all probability distributions on a finite set $\mathcal{S}$, i.e., all functions $f : \mathcal{S} \to [0,1]$ such that $\sum_{s\in\mathcal{S}} f(s) = 1$, is denoted by $\Delta(\mathcal{S})$. For a sequence $\{X_t, t \in \mathbb{N}\}$, a subsequence $(X_k, X_{k+1}, \ldots, X_l)$ is denoted by X_k^l. The subsequence $(X_1, X_2, \ldots, X_l)$ is simply denoted by X^l.

A. Partially Observable Markov Decision Processes

Definition 1: A POMDP is a tuple $\mathbf{M} = (\mathcal{S}, s_I, \mathcal{A}, P, \mathcal{Z}, O, R)$ where $\mathcal{S}$ is a finite set of states, $s_I \in \mathcal{S}$ is a unique initial state, $\mathcal{A}$ is a finite set of actions, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is a transition function, $\mathcal{Z}$ is a finite set of observations, $O : \mathcal{S} \to \Delta(\mathcal{Z})$ is an observation function, and $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function.

For simplicity, we assume that all actions $a \in \mathcal{A}$ are available in all states $s \in \mathcal{S}$. For the ease of notation, we denote the transition probability $P(s'|s, a)$ and the observation probability $O(z|s)$ by $P_{s,a,s'}$ and $O_{s,z}$, respectively.

For a POMDP $\mathbf{M}$, the *corresponding fully observable* MDP $\mathbf{M}_{fo}$ is obtained by setting $\mathcal{Z} = \mathcal{S}$ and $O_{s,s} = 1$ for all $s \in \mathcal{S}$.

A *system history* of length $t \in \mathbb{N}$ for a POMDP $\mathbf{M}$ is a sequence $h^t = (s_I, a_1, s_2, a_2, s_3, \ldots, s_t)$ of states and actions such that $P_{s_k,a_k,s_{k+1}} > 0$ for all $k \in \mathbb{N}$. We denote the set of all system histories of length t by $\mathcal{H}^t$. For any system history $h^t = (s_I, a_1, s_2, \ldots, s_t)$ of length t, there is an associated *observation history* $o^t = (z_1, a_1, z_2, \ldots, z_t)$ of length $t \in \mathbb{N}$ where $O_{s_k,z_k} > 0$ for all $k \in \mathbb{N}$. Note that there are, in general, multiple observation histories that are admissible for a given system history h^t. Finally, we denote the set of all observation histories of length t by $\mathcal{O}^t$.

Definition 2: A *controller* $\pi : \cup_{t\in\mathbb{N}} \mathcal{O}^t \to \Delta(\mathcal{A})$ is a mapping from observation histories to distributions over actions. For a POMDP $\mathbf{M}$, we denote the set of all controllers by $\Pi(\mathbf{M})$.

The probability with which the controller π takes the action $a \in \mathcal{A}$ upon receiving the history $o^t \in \mathcal{O}^t$ is denoted by $\pi(a|o^t)$.

B. Entropy of Stochastic Processes

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

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

We use the convention that $0 \log 0 = 0$. Let (X_1, X_2) be a pair of random variables with the joint pmf $p(x_1, x_2)$ and the support $\mathcal{X} \times \mathcal{X}$. The *joint entropy* of (X_1, X_2) is

$$H(X_1, X_2) := -\sum_{x_1\in\mathcal{X}} \sum_{x_2\in\mathcal{X}} p(x_1, x_2) \log p(x_1, x_2) \tag{2}$$

and the *conditional entropy* of X_2 given X_1 is

$$H(X_2|X_1) := -\sum_{x_1\in\mathcal{X}} \sum_{x_2\in\mathcal{X}} p(x_1, x_2) \log p(x_2|x_1). \tag{3}$$

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

Definition 3 ([24]): The *entropy of a stochastic process* $\mathbb{X}$ is

$$H(\mathbb{X}) := \lim_{t\to\infty} H(X_1, X_2, \ldots, X_t). \tag{4}$$

The above-mentioned definition is different from the *entropy rate* of a stochastic process, which is defined as $\lim_{t\to\infty} \frac{1}{t} H(X^t)$ when the limit exists [1]. The limit in (4) either converges to a nonnegative number or diverges to positive infinity [24].

For a POMDP $\mathbf{M}$, a controller $\pi \in \Pi(\mathbf{M})$ induces a discrete stochastic process $\{S_t \in \mathcal{S} : t \in \mathbb{N}\}$ in which each S_t is a random variable over the state space $\mathcal{S}$. We denote the entropy of a POMDP $\mathbf{M}$ under a controller $\pi \in \Pi(\mathbf{M})$ by $H^\pi(\mathbf{M})$.

III. Problem Statement

We consider an *agent* whose behavior is modeled as a POMDP and an *outside observer* whose objective is to infer the states occupied by the agent in the future from the states occupied in the past. Being aware of the observer's objective, the agent aims to minimize the predictability of its future states while ensuring that the expected total reward it collects exceeds a specified threshold.

We measure the predictability of the agent's future states by the entropy of the underlying stochastic process. The rationale behind this choice can be better understood by recalling (see, e.g., Th. 2.5.1 in [1]) that, for an arbitrary controller $\pi \in \Pi(\mathbf{M})$, the identity

$$H^\pi(S_1, S_2, \ldots, S_N) = H^\pi(S_t^N|S^{t-1}) + H^\pi(S^{t-1}) \tag{5}$$

holds for any $N \in \mathbb{N}$ and $t \leq N$. Therefore, by maximizing the value of the left-hand side of (5), one maximizes the entropy of all future sequences $(S_t, \ldots, S_N)$ for any given sequence $(S_1, \ldots, S_{t-1})$.

We consider an agent with infinite decision horizon whose aim is to randomize its infinite length state trajectories. When the decision horizon is infinite, the total reward collected by the agent, as well as the entropy of the underlying stochastic process, may be infinite [3],

[25]. A common approach to ensure the finiteness of the solution in infinite horizon models is to discount the collected rewards and the gained entropy in the future [26]–[28]. Accordingly, note that

$$\lim_{t\to\infty} H^\pi(S_1, S_2, \dots, S_t) = H^\pi(S_1) + \sum_{t=2}^{\infty} H^\pi(S_t|S^{t-1}) \quad (6)$$

we treat each term $H^\pi(S_t|S^{t-1})$ as a virtual entropy reward for the agent. Note that $H^\pi(S_1) = H^{\pi'}(S_1)$ for any $\pi, \pi' \in \Pi(\mathbf{M})$ since the initial state distribution in a POMDP is fixed. By discounting the future rewards $R(S_t, A_t)$ as well as the virtual entropy reward $H^\pi(S_t|S^{t-1})$, we define the main problem as follows.

Problem 1 (Entropy maximization): For a POMDP $\mathbf{M}$, a discount factor $\beta \in$[0,1), and a reward threshold $\Gamma \in \mathbb{R}$, synthesize a controller $\pi^\star \in \Pi(\mathbf{M})$ that solves the following problem:

$$\underset{\pi\in\Pi(\mathbf{M})}{\text{maximize}} \ \sum_{t=2}^{\infty} \beta^{t-2} H^\pi(S_t|S^{t-1}) \quad (7a)$$

$$\text{subject to: } \mathbb{E}^\pi\left[\sum_{t=1}^{\infty} \beta^{t-1} R(S_t, A_t)\right] \geq \Gamma. \quad (7b)$$

For $\beta \in [0, 1)$, the existence of a policy that satisfies the constraint (7b) is, in general, undecidable [4]. Hence, the synthesis of globally optimal controllers that solve the entropy maximization problem is, in general, intractable. In what follows, we restrict our attention to FSCs and present a method to synthesize FSCs that are local optimal solutions to the entropy maximization problem among all FSCs with fixed number of memory states and fixed memory transition functions.

IV. Recursive Equations for Entropy Maximization

In this section, we derive the system of recursive equations for the entropy maximization objective in (7a).

For a given system history $h^t = (s_I, a_1, s_2, a_2, s_3, \dots, s_t)$, let the sequences $s^t = (s_I, s_2, \dots, s_t)$ and $a^t = (a_1, a_2, \dots, a_t)$ be the corresponding state and action histories of length t, respectively. We denote the set of all state and action histories of length t by $\mathcal{SH}^t$ and $\mathcal{AH}^t$, respectively. It can be shown that, for a POMDP $\mathbf{M}$ under a controller $\pi \in \Pi(\mathbf{M})$, the realization probability $\Pr^\pi(s^{t+1}|s^t)$ of the state history $s^{t+1} \in \mathcal{SH}^{t+1}$ for a given $s^t \in \mathcal{SH}^t$ is

$$\Pr^\pi(s^{t+1}|s^t) = \sum_{a^t\in\mathcal{AH}^t} \prod_{k=1}^{t} \mu_k(a_k|h^k) P_{s_t,a_t,s_{t+1}}. \quad (8)$$

In the previous equation, h^k are prefixes of h^t from which the state sequence s^t is obtained, and $\mu_t : \mathcal{H}^t \to \Delta(\mathcal{A})$ is a mapping such that

$$\mu_t(a|h^t) := \sum_{o^t\in\mathcal{O}^t} \pi(a|o^t)\Pr(o^t|h^t). \quad (9)$$

We note that, for t =1, we have $\Pr(o^1 = z_1|h^1) = O_{s_I,z_1}$, and for all $t \geq$2, $\Pr(o^t|h^t)$ can be recursively written as

$$\Pr(o^t|h^t) = O_{s_t,z_t} P_{s_{t-1},a_{t-1},s_t} \Pr(o^{t-1}|h^{t-1}). \quad (10)$$

For a given controller $\pi \in \Pi(\mathbf{M})$ and a constant $N \in \mathbb{N}$, let $V^\pi_{t,N} : \mathcal{SH}^t \to \mathbb{R}$ be the *value function* such that, for all $t < N$

$$V^\pi_{t,N}(s^t) := \sum_{k=t}^{N-1} \beta^{k-t} H^\pi(S_{k+1}|S^k_t, S^t = s^t). \quad (11)$$

Lemma 1: For a POMDP $\mathbf{M}$, a controller $\pi \in \Pi(\mathbf{M})$, and a constant $N \in \mathbb{N}$, the value function $V^\pi_{t,N}$, defined in (11), satisfies the equality

$$V^\pi_{t,N}(s^t) = H^\pi(S_{t+1}|S^t = s^t) + \beta \sum_{s^{t+1}\in\mathcal{SH}^{t+1}} \Pr^\pi(s^{t+1}|s^t) V^\pi_{t+1,N}(s^{t+1}) \quad (12)$$

for all $t < N$ and $s^t \in \mathcal{SH}^t$.

Proof of all technical results are provided in Appendix A. For $t < N$, let $V^\star_{t,N} : \mathcal{SH}^t \to \mathbb{R}$ be a function such that

$$V^\star_{t,N}(s^t) := \sup_{\pi\in\Pi(\mathbf{M})} V^\pi_{t,N}(s^t). \quad (13)$$

Using Lemma 1, together with *the principle of optimality* [25, Ch. 4], we conclude that, for all $t < N$ and $s^t \in \mathcal{SH}^t$

$$V^\star_{t,N}(s^t) = \sup_{\pi\in\Pi(\mathbf{M})} \Big[H^\pi(S_{t+1}|S^t = s^t) + \beta \sum_{s^{t+1}\in\mathcal{SH}^{t+1}} \Pr^\pi(s^{t+1}|s^t) V^\star_{t+1,N}(s^{t+1})\Big]. \quad (14)$$

Then, the summation in (6), together with the definition of the value function in (11), implies that, for any $N \in \mathbb{N}$, we have

$$\sup_{\pi\in\Pi(\mathbf{M})} \sum_{t=2}^{N} \beta^{t-2} H^\pi(S_t|S^{t-1}) = V^\star_{1,N}(s_I). \quad (15)$$

Since $V^\pi_{t,N}$, defined in (11), is monotonically nondecreasing in N for all $\pi \in \Pi(\mathbf{M})$, i.e., $V^\pi_{t,N+1} \geq V^\pi_{t,N}$, we have

$$\sup_{\pi\in\Pi(\mathbf{M})} \lim_{N\to\infty} V^\pi_{t,N}(s^t) = \lim_{N\to\infty} \sup_{\pi\in\Pi(\mathbf{M})} V^\pi_{t,N}(s^t) \quad (16)$$

for all $s^t \in \mathcal{SH}^t$. Therefore, by taking the limits of both sides in (15), we conclude that

$$\sup_{\pi\in\Pi(\mathbf{M})} \sum_{t=2}^{\infty} \beta^{t-2} H^\pi(S_t|S^{t-1}) = \lim_{N\to\infty} V^\star_{1,N}(s_I). \quad (17)$$

The derivations mentioned previously show that an agent having access to state histories s^t can synthesize an entropy-maximizing controller by recursively computing the values $V^\star_{t,N}(s^t)$ via dynamic programming. In a POMDP, only observation histories are available to the agent; hence, the previous derivations cannot be directly used for controller synthesis. In the next section, we consider FSCs and present a tractable controller synthesis method by utilizing the results of Lemma 1.

V. Entropy Maximization Over FSCs

Optimal controllers solving the entropy maximization problems may, in general, use the complete system history to determine the next action to perform. A common approach to overcome intractability is to restrict attention to FSCs whose memory states represent (potentially insufficient) statistics of the system histories [6], [7]. Accordingly, we focus on FSCs with a fixed number of memory states and develop methods to synthesize locally optimal controllers within this restricted domain.

Definition 4: For a POMDP $\mathbf{M}$, a *k-finite-state controller* (k-FSC) is a tuple $\mathbf{C} = (\mathcal{Q}, q_1, \gamma, \delta)$, where $\mathcal{Q} = \{q_1, q_2, \dots, q_k\}$ is a finite set of memory states, $q_1 \in \mathcal{Q}$ is the initial memory state, $\gamma : \mathcal{Q} \times \mathcal{Z} \to \Delta(\mathcal{A})$ is a decision function, and $\delta : \mathcal{Q} \times \mathcal{Z} \times \mathcal{A} \to 3\Delta(\mathcal{Q})$ is a memory transition function. We denote the collection of all k-FSCs by $\mathcal{F}_k(\mathbf{M})$.

In Fig. 1, we present an illustration of k-FSCs. For a POMDP, a k-FSC induces a Markov chain (MC), which is an MDP with a single available action, i.e., $|\mathcal{A}|$ =1. It is shown in [23] that the set of all MCs that can be induced by a k-FSC is the set of all well-defined instantiations of a parameteric MC (pMC). Therefore, without loss of generality, one can work on that pMC to synthesize an instantiation, which corresponds to the MC induced by an entropy-maximizing FSC. In the following sections, we reformulate the entropy maximization problem over k-FSCs as another optimization problem over pMCs.

A. Parameteric MCs

We develop solutions to entropy maximization problems through the use of pMCs.

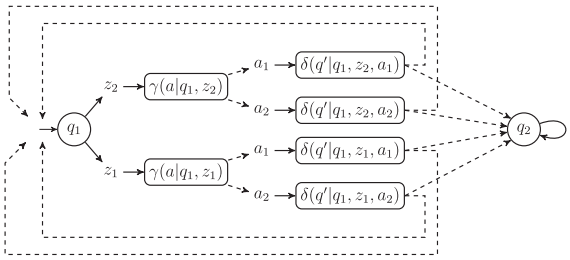

Fig. 1. Illustration of FSCs. In a memory state q, the agent receives an observation z, chooses an action a based on the decision function $\gamma(a|q,z)$, and transitions to a memory state q' based on the transition function $\delta(q'|q,z,a)$. The functions γ and δ are design variables, and their outcomes are indicated with dashed lines. Solid lines represent deterministic transitions.

Definition 5: For a POMDP $\mathbf{M}$ and a constant $k \in \mathbb{N}$, the *induced* pMC is a tuple $\mathbf{D}_{\mathbf{M},k} = (\mathcal{S}_{\mathbf{M},k}, s_{I,\mathbf{M},k}, \mathcal{V}_{\mathbf{M},k}, P_{\mathbf{M},k}, R_{\mathbf{M},k})$ where

1) $\mathcal{S}_{\mathbf{M},k} = \mathcal{S} \times \{1, 2, \ldots, k\}$ is the finite set of states,
2) $s_{I,\mathbf{M},k} = \langle s_I, 1 \rangle$ is the initial state,
3) $\mathcal{V}_{\mathbf{M},k} = \{\gamma_a^{q,z} | z \in \mathcal{Z}, q \in \mathcal{Q}, a \in \mathcal{A}\} \cup \{\delta_{q'}^{q,z,a} | z \in \mathcal{Z}, q, q' \in \mathcal{Q}, a \in \mathcal{A}\}$ is the finite set of parameters,
4) $P_{\mathbf{M},k} : \mathcal{S}_{\mathbf{M},k} \to \Delta(\mathcal{S}_{\mathbf{M},k})$ is a transition function such that $P_{\mathbf{M},k}(\langle s,' q' \rangle \mid \langle s, q \rangle) := \sum_{a \in \mathcal{A}} \overline{P}(\langle s,' q' \rangle \mid \langle s, q \rangle, a)$ for all $\langle s, q \rangle, \langle s,' q' \rangle \in \mathcal{S}_{\mathbf{M},k}$, where $\overline{P} : \mathcal{S}_{\mathbf{M},k} \times \mathcal{A} \to \Delta(\mathcal{S}_{\mathbf{M},k})$ is a mapping such that
$$\overline{P}(\langle s,' q' \rangle \mid \langle s, q \rangle, a) := \sum_{z \in \mathcal{Z}} O_{s,z}\, P_{s,a,s'}\, \gamma_a^{q,z}\, \delta_{q'}^{q,z,a} \tag{18}$$
5) $R_{\mathbf{M},k}(\langle s, q \rangle, a) := R(s, a)$ for all $s \in \mathcal{S}$, $q \in \mathcal{Q}$, and $a \in \mathcal{A}$.

An MC can be obtained from an induced pMC by instantiating the parameters $\mathcal{V}_{\mathbf{M},k}$ in a way that the resulting transition function is well defined. Formally, a *well-defined instantiation* for $\mathcal{V}_{\mathbf{M},k}$ is a function $u : \mathcal{V}_{\mathbf{M},k} \to$[0,1] such that, for all $a \in \mathcal{A}$, $q \in \mathcal{Q}$, and $z \in \mathcal{Z}$
$$\sum_{a \in \mathcal{A}} u(\gamma_a^{q,z}) = 1 \quad \text{and} \quad \sum_{q' \in \mathcal{Q}} u(\delta_{q'}^{q,z,a}) = 1.$$

Applying a well-defined instantiation u to the induced pMC $\mathbf{D}_{\mathbf{M},k}$, denoted $\mathbf{D}_{\mathbf{M},k}[u]$, replaces each parameteric transition probability $P_{\mathbf{M},k}$ by $P^u_{\mathbf{M},k}$. Let $\Upsilon_{\mathbf{M},k}$ denote the set of all well-defined instantiations for a pMC $\mathbf{D}_{\mathbf{M},k}$. For an induced pMC $\mathbf{D}_{\mathbf{M},k}$, every instantiation $u \in \Upsilon_{\mathbf{M},k}$ describes a k-FSC $\mathbf{C}_u \in \mathcal{F}_k(\mathbf{M})$ [23]. Thus, we can synthesize all admissible MCs that can be induced from a POMDP $\mathbf{M}$ by a k-FSC $\mathbf{C} \in \mathcal{F}_k(\mathbf{M})$ through well-defined instantiations u over $\mathcal{V}_{\mathbf{M},k}$. In Fig. 2, we provide an example to illustrate the derivation of $\mathbf{D}_{\mathbf{M},k}[u]$ from a given $\mathbf{M}$ and $\mathbf{C}_u$.

B. Reformulation Over pMCs

Recall that we are interested in maximizing the entropy of the state sequence of a given POMDP $\mathbf{M}$. As can be seen from Fig. 2, the number of states that are reachable from the initial state of a pMC $\mathbf{D}_{\mathbf{M},k}$ is, in general, larger than that of the POMDP $\mathbf{M}$. It is known [1] that the maximum entropy of a random variable increases as the cardinality of its support increases. Hence, by appropriately choosing the transition probabilities in the example given in Fig. 2, it is possible to construct a well-defined instantiation $\mathbf{D}_{\mathbf{M},k}[u]$ whose entropy of state sequences is higher than the maximum entropy of the state sequences of the POMDP $\mathbf{M}$. This observation implies that, in general, the maximum entropy of a POMDP $\mathbf{M}$ is *not* an upper bound on the maximum entropy of the induced pMC $\mathbf{D}_{\mathbf{M},k}$.

The maximum entropy of $\mathbf{D}_{\mathbf{M},k}$ is, in general, higher than that of $\mathbf{M}$ due to the stochasticity introduced to the process by the parameters $\delta_{q'}^{q,z,a}$. To synthesize an entropy-maximizing k-FSC for a POMDP $\mathbf{M}$ using the induced pMC $\mathbf{D}_{\mathbf{M},k}$, we impose restrictions on the memory transition function of the k-FSCs. For each memory state $q \in \mathcal{Q}$ in a given k-FSC, let
$$\text{Succ}(q) := \{q' \in \mathcal{Q} : \delta(q'|q, z, a) > 0, z \in \mathcal{Z}, a \in \mathcal{A}\}.$$

Definition 6: A *deterministic k-FSC* $\mathbf{C} = (\mathcal{Q}, q_1, \gamma, \delta)$ is a k-FSC such that for all $q \in \mathcal{Q}$, $|\text{Succ}(q)| = 1$. We denote the set of all deterministic k-FSCs by $\mathcal{F}_k^{\text{det}}(\mathbf{M})$.

For a k-FSC $\mathbf{C} \in \mathcal{F}_k^{\text{det}}(\mathbf{M})$, let $u_{\mathbf{C}} : V_{\mathbf{M},k} \to \mathbb{R}$ be the corresponding instantiation of the induced pMC $\mathbf{D}_{\mathbf{M},k}$ such that
$$u_{\mathbf{C}}(\gamma_a^{q,z}) := \gamma(a|q,z) \;\text{ and }\; u_{\mathbf{C}}(\delta_{q'}^{q,z,a}) := \delta(q'|q,z,a).$$
Moreover, let $\Upsilon^{\text{det}}_{\mathbf{M},k}$ denote the set of all well-defined instantiations $u_{\mathbf{C}}$ that corresponds to a deterministic k-FSC $\mathbf{C}$. Noting that $\mathbf{D}_{\mathbf{M},k}[u_{\mathbf{C}}]$ is a stochastic process, we denote its sequence of states by $(S_{\mathbf{M},k,1}, S_{\mathbf{M},k,2}, \ldots)$. Moreover, for a given state $S_{\mathbf{M},k,t-1}$, we denote the one-step entropy of $\mathbf{D}_{\mathbf{M},k}[u_{\mathbf{C}}]$ by $H^{u_{\mathbf{C}}}(S_{\mathbf{M},k,t}|S_{\mathbf{M},k,t-1})$.

Proposition 1: For a given POMDP $\mathbf{M}$, a controller $\mathbf{C} \in \mathcal{F}_k^{\text{det}}(\mathbf{M})$, and constants $t, k \in \mathbb{N}$, we have
$$H^{\mathbf{C}}(S_t|S^{t-1}) = H^{u_{\mathbf{C}}}(S_{\mathbf{M},k,t}|S_{\mathbf{M},k,t-1}). \tag{19}$$

Proposition 1 shows that the local entropy gained in a POMDP $\mathbf{M}$ under a deterministic k-FSC $\mathbf{C}$ is equal to the local entropy gained in $\mathbf{D}_{\mathbf{M},k}[u_{\mathbf{C}}]$. Note that, since the memory states q are explicitly represented in the states $\langle s, q \rangle$ of $\mathbf{D}_{\mathbf{M},k}[u_{\mathbf{C}}]$, local entropy in $\mathbf{D}_{\mathbf{M},k}[u_{\mathbf{C}}]$ depends only on the state occupied in the previous step.

Proposition 1, together with the definition of the induced pMC, implies that, for any $\mathbf{C} \in \mathcal{F}_k^{\text{det}}(\mathbf{M})$
$$\mathbf{C} \in \arg \max_{\mathbf{C}' \in \mathcal{F}_k^{\text{det}}(\mathbf{M})} \sum_{t=2}^{\infty} \beta^{t-2} H^{\mathbf{C}'}(S_t|S^{t-1}) \tag{20a}$$
$$\text{subject to: } \mathbb{E}^{\mathbf{C}'}\left[\sum_{t=1}^{\infty} \beta^{t-1} R(S_t, A_t)\right] \geq \Gamma$$
$$\text{if and only if} \tag{20b}$$
$$u_{\mathbf{C}} \in \arg \max_{u \in \Upsilon^{\text{det}}_{\mathbf{M},k}} \sum_{t=2}^{\infty} \beta^{t-2} H^{u}(S_{\mathbf{M},k,t}|S_{\mathbf{M},k,t-1}) \tag{21a}$$
$$\text{subject to: } \mathbb{E}^{u}\left[\sum_{t=1}^{\infty} \beta^{t-1} R(S_{\mathbf{M},k,t}, A_t)\right] \geq \Gamma. \tag{21b}$$

The following result is due to the fact that $\mathcal{F}_k^{\text{det}}(\mathbf{M}) \subseteq \Pi(\mathbf{M})$ and shows that the maximum entropy of the pMC induced from FSCs with deterministic memory transitions is upper bounded by that of the corresponding POMDP.

Corollary 1: Let $G_1^\star$ and $G_2^\star$ be the optimal values of the problems given in (7a), (7b), (21a), and (21b), respectively. We have $G_1^\star \geq G_2^\star$.

C. FSC Synthesis: Optimization Problem

We now present a NLP to synthesize a deterministic k-FSC that maximizes the entropy of a POMDP over all deterministic k-FSCs.

Recall that for a POMDP $\mathbf{M}$ and a constant $k >$0, the induced pMC $\mathbf{D}_{\mathbf{M},k}$ represents all possible MCs that can be induced from $\mathbf{M}$ by a k-FSC. Moreover, Proposition 1 implies that the maximum entropy of $\mathbf{D}_{\mathbf{M},k}$ is equal to the maximum entropy of $\mathbf{M}$ *if one restricts attention to k-FSCs with deterministic memory transitions*. In what follows, we formulate an optimization problem to synthesize a well-defined instantiation u for the pMC $\mathbf{D}_{\mathbf{M},k}$ such that the entropy of the MC $\mathbf{D}_{\mathbf{M},k}[u]$ is maximized over all MCs $\mathbf{D}_{\mathbf{M},k}[u']$ for which $P^{u'}_{\mathbf{M},k}(\langle s,' q' \rangle \mid \langle s, q \rangle) >$0 for a single $q' \in \mathcal{Q}$.

To restrict the search space to FSCs with deterministic memory transitions, we introduce the following constraints:
$$u(\delta_{q'}^{q,z,a}) \in \{0, 1\} \;\text{ and }\; \sum_{z \in \mathcal{Z}} \sum_{a \in \mathcal{A}} u(\delta_{q'}^{q,z,a}) \in \{0, |\mathcal{Z}||\mathcal{A}|\}.$$

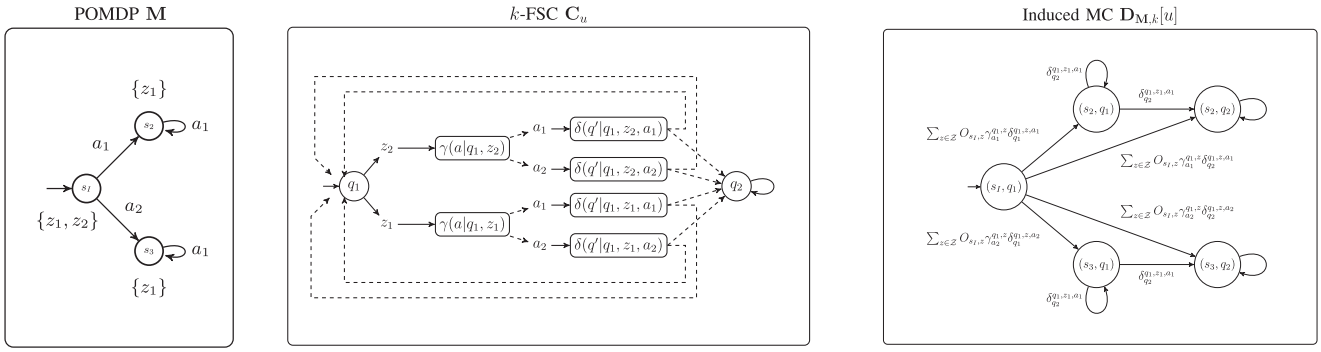

Fig. 2. Illustration of an MC $\mathbf{D}_{\mathbf{M},k}[u]$ (right-hand side) induced from a POMDP $\mathbf{M}$ (left-hand side) by a k-FSC $\mathbf{C}_u$ (middle). Note that the number of states in $\mathbf{D}_{\mathbf{M},k}[u]$ is larger than $\mathbf{M}$ due to the stochasticity in the memory transition function δ.

Intuitively, the above-mentioned constraints ensure the transition to a single successor memory state regardless of the received observation and the taken action. We note that the second integer constraint can be implemented as $|\mathcal{Z}||\mathcal{A}|$ equality constraints. Finally, the abovementioned constraints do not prevent the agent from randomizing its actions. The agent can still randomize its actions at a given state $s \in \mathcal{S}$ by instantiating the parameters $\gamma_a^{q,z}$ appropriately.

For notational simplicity, let $\mathbf{s}$ denote an arbitrary state $\langle s, t\rangle \in \mathcal{S}_{\mathbf{M},k}$. Let $L^u : \mathcal{S}_{\mathbf{M},k} \to \mathbb{R}$ be the *local entropy* function such that, for all $\mathbf{s} \in \mathcal{S}_{\mathbf{M},k}$

$$L^u(\mathbf{s}) := -\sum_{\mathbf{s}' \in \mathcal{S}_{\mathbf{M},k}} P^u_{\mathbf{M},k}(\mathbf{s}'|\mathbf{s}) \log P^u_{\mathbf{M},k}(\mathbf{s}'|\mathbf{s}). \tag{22}$$

Note that we have $L^u(\mathbf{s}) = H^u(S_{\mathbf{M},k,t}|S_{\mathbf{M},k,t-1} = \mathbf{s})$ for any $t \in \mathbb{N}$. Hence, $L^u(\mathbf{s})$ corresponds to the local entropy reward gained in the MC $\mathbf{D}_{\mathbf{M},k}[u]$ from the state $\mathbf{s}$. Recalling the equivalence given in (19), the local entropy function L^u allows us to transfer the results of Section IV, which are obtained for a POMDP $\mathbf{M}$, to the pMC $\mathbf{D}_{\mathbf{M},k}$. Specifically, by defining variables $\nu \in \mathbb{R}^{|\mathcal{S}_{\mathbf{M},k}|}$, it can be shown that the maximum entropy (21a) of $\mathbf{D}_{\mathbf{M},k}$ is the unique fixed point of the system of equations

$$\nu(\mathbf{s}) = \max_{u \in \Upsilon_{\mathbf{M},k}} \left\{ L^u(\mathbf{s}) + \beta \sum_{\mathbf{s}' \in \mathcal{S}_{\mathbf{M},k}} P^u_{\mathbf{M},k}(\mathbf{s}'|\mathbf{s})\nu(\mathbf{s}') \right\} \tag{23}$$

and equal to $\nu(\mathbf{s}_I) := \nu(s_{I,\mathbf{M},k})$. Hence, the maximum entropy (21a) of $\mathbf{D}_{\mathbf{M},k}$ can be computed by finding the maximum $\nu(\mathbf{s}_I)$ that satisfies

$$\nu(\mathbf{s}) \leq L^u(\mathbf{s}) + \beta \sum_{\mathbf{s}' \in \mathcal{S}_{\mathbf{M},k}} P^u_{\mathbf{M},k}(\mathbf{s}'|\mathbf{s})\nu(\mathbf{s}') \quad \forall \mathbf{s} \in \mathcal{S}_{\mathbf{M},k}.$$

In the abovementioned inequality, both $P^u_{\mathbf{M},k}(\mathbf{s}'|\mathbf{s})$ and $\nu(\mathbf{s}')$ are variables. Hence, standard methods, e.g., value iteration, cannot be used to compute $\nu(\mathbf{s}_I)$; instead, one needs to solve a NLP, which we present shortly, for the computation of $\nu(\mathbf{s}_I)$.

Let $R^u : \mathcal{S}_{\mathbf{M},k} \to \mathbb{R}$ be the expected immediate rewards on $\mathbf{D}_{\mathbf{M},k}$ such that, for all $\mathbf{s} \in \mathcal{S}_{\mathbf{M},k}$

$$R^u(\mathbf{s}) := \sum_{\mathbf{s}' \in \mathcal{S}_{\mathbf{M},k}} \sum_{a \in \mathcal{A}} \overline{P}^u(\mathbf{s}'|\mathbf{s}, a) R(\mathbf{s}', a) \tag{24}$$

where $\overline{P}^u : S_{\mathbf{M},k} \times \mathcal{A} \to \Delta(\mathcal{S}_{\mathbf{M},k})$ is defined by replacing parameters $\gamma_a^{q,z}$ and $\delta_{q'}^{q,z,a}$ in (18) with their corresponding instantiations $u(\gamma_a^{q,z})$ and $u(\delta_{q'}^{q,z,a})$. Then, the problem in (21a) and (21b) can be formulated as a NLP as follows:

$$\underset{\nu,u,\eta}{\text{maximize}} \quad \nu(\mathbf{s}_I) \tag{25a}$$

subject to:

$$\nu(\mathbf{s}) \leq L^u(\mathbf{s}) + \beta \sum_{\mathbf{s}' \in \mathcal{S}_{\mathbf{M},k}} P^u_{\mathbf{M},k}(\mathbf{s}'|\mathbf{s})\nu(\mathbf{s}') \quad \forall \mathbf{s} \in \mathcal{S}_{\mathbf{M},k} \tag{25b}$$

$$\eta(\mathbf{s}) \leq R^u(\mathbf{s}) + \beta \sum_{\mathbf{s}' \in \mathcal{S}_{\mathbf{M},k}} P^u_{\mathbf{M},k}(\mathbf{s}'|\mathbf{s})\eta(\mathbf{s}') \quad \forall \mathbf{s} \in \mathcal{S}_{\mathbf{M},k} \tag{25c}$$

$$\eta(\mathbf{s}_I) \geq \Gamma \tag{25d}$$

$$u(\delta_{q'}^{q,z,a}) \in \{0,1\}, \ \sum_{q' \in Q} u(\delta_{q'}^{q,z,a}) = 1 \tag{25e}$$

$$0 \leq u(\gamma_a^{q,z}) \leq 1, \quad \sum_{a \in \mathcal{A}} u(\gamma_a^{q,z}) = 1 \tag{25f}$$

$$\sum_{z \in \mathcal{Z}} \sum_{a \in \mathcal{A}} u(\delta_{q'}^{q,z,a}) \in \{0, |\mathcal{Z}||\mathcal{A}|\}. \tag{25g}$$

In the above-mentioned optimization problem, the variable $\eta(\mathbf{s})$ denotes the expected reward collected by starting from the state $\mathbf{s} \in \mathcal{S}_{\mathbf{M},k}$. It follows from [25] that the constraint (25d) ensures that a solution $u^\star$ to the above problem collects an expected total reward exceeding the threshold Γ.

Recall that the transition function $P^u_{\mathbf{M},k}$ of the MC $\mathbf{D}_{\mathbf{M},k}[u]$, which results from the instantiation u of the pMC $\mathbf{D}_{\mathbf{M},k}$, is given by $P^u_{\mathbf{M},k}(\mathbf{s}'|\mathbf{s}) = \sum_{a \in \mathcal{A}} \overline{P}^u(\mathbf{s}'|\mathbf{s}, a)$, where

$$\overline{P}^u(\mathbf{s}'|\mathbf{s}, a) := \sum_{z \in \mathcal{Z}} O_{s,z}\, P_{s,a,s'}\, u(\gamma_a^{q,z})\, u(\delta_{q'}^{q,z,a}) \tag{26}$$

$\mathbf{s} = \langle s, q\rangle$, and $\mathbf{s}' = \langle s,' q'\rangle$. Therefore, the problem in (25a)–(25g) involves nonlinear constraints in (25b) where three variables are multiplied with each other. Even though certain relaxation techniques, e.g., McCormick envelopes [29], can be used to replace the constraints in (25b) with specific bilinear constraints, finding an optimal solution to the resulting NLP remains as a challenge due to binary constraints in (25e).

For practical purposes, instead of computing a globally optimal solution, one can aim to obtain a locally optimal solution to the problem in (25a)–(25g) *after setting the instantiation* $u(\delta_{q'}^{q,z,a})$ *of memory transitions to a constant*. In the next section, we provide a method to obtain a local optimal solution to the problem in (7a) and (7b) over all k-FSCs with a specific deterministic transition function.

D. FSC Synthesis: A Solution Approach

In this section, we consider the entropy maximization problems over k-FSCs with a specific deterministic transition function and present an algorithm to synthesize a controller which locally maximizes the entropy of a given POMDP.

We first set the variables $u(\delta_{q'}^{q,z,a})$ in the problem (25a)–(25g) to constants such that they satisfy the constraint in (25e). This operation is equivalent to restricting the search space in (7a) and (7b) to k-FSCs with a specific deterministic transition function, where the transition function satisfies $\delta(q'|q,z,a) = u(\delta_{q'}^{q,z,a})$. The resulting optimization problem has decision variables $\nu(\mathbf{s})$, $\eta(\mathbf{s})$, and $u(\gamma_a^{q,z})$, i.e., $u(\delta_{q'}^{q,z,a})$ is not a variable anymore. We can obtain a locally optimal solution to the resulting problem using a variant of the convex–concave procedure (CCP) [30]. In particular, we employ *penalty CCP*, which is introduced in [31] and used in the context of pMCs in [8].

The penalty CCP algorithm takes five inputs: a threshold constant $\epsilon > 0$, initial penalty constant $\tau_0 > 0$, multiplication factor $\mu > 1$, maximum penalty constant $\tau_{\max}$, and initial estimates $\hat{\nu}_0(\mathbf{s})$, $\hat{\eta}_0(\mathbf{s})$, and $\hat{u}_0(\gamma_a^{q,z})$ for the variables $\nu(\mathbf{s})$, $\eta(\mathbf{s})$, and $u(\gamma_a^{q,z})$, respectively. Moreover, for each iteration $k \in \mathbb{Z}_+$ of the algorithm, we recursively define $\tau_{k+1} := \min\{\mu\tau_k, \tau_{\max}\}$.

Let $\mathbf{v}$ denote an arbitrary tuple $(\mathbf{s}', q, z, a) \in \mathcal{S}_{\mathbf{M},k} \times \mathcal{Q} \times \mathcal{Z} \times \mathcal{A}$. For each $\mathbf{v}$, we introduce two new variables $\Phi_{\nu,\mathbf{v}} \geq 0$ and $\Phi_{\eta,\mathbf{v}} \geq 0$. The introduced variables are typically referred to as *slack variables* and quantify the infeasibility of the constraints in (25b) and (25c) [31]. In particular, when $\sum_{\mathbf{v}}(\Phi_{\eta,\mathbf{v}} + \Phi_{\nu,\mathbf{v}}) = 0$, the output of the penalty CCP algorithm becomes feasible for the problem in (25a)–(25g).

At each iteration $k \in \mathbb{Z}_+$, we first convexify the constraints in (25b) and (25c) (explained in the following). We then solve the resulting convex optimization problem by replacing the objective function (25a) with

$$\underset{\nu,u,\eta}{\text{maximize}} \quad \nu(\mathbf{s}_I) - \tau_k \sum_{\mathbf{v}} (\Phi_{\eta,\mathbf{v}} + \Phi_{\nu,\mathbf{v}}).$$

Intuitively, the second term in the above-mentioned objective function is a penalty term, which encourages the algorithm to output feasible solutions for the original problem in (25a)–(25g).

Let Val_k be the optimal value of the problem described previously. We terminate the algorithm if $|Val_k - Val_{k-1}| < \epsilon$ and the optimal solution satisfies $\sum_{\mathbf{v}}(\Phi_{\eta,\mathbf{v}} + \Phi_{\nu,\mathbf{v}}) = 0$; otherwise, we set the optimal decision variables $\nu^\star(\mathbf{s})$, $\eta^\star(\mathbf{s})$, and $u^\star(\gamma_a^{q,z})$ for the current iteration as the estimates $\hat{\nu}_{k+1}(\mathbf{s})$, $\hat{\eta}_{k+1}(\mathbf{s})$, and $\hat{u}_{k+1}(\gamma_a^{q,z})$ for the successive iteration, and solve the resulting optimization problem. The procedure explained previously has no theoretical convergence guarantees to a feasible solution [31], i.e., a solution that satisfies $\sum_{\mathbf{v}}(\Phi_{\eta,\mathbf{v}} + \Phi_{\nu,\mathbf{v}}) = 0$. However, any feasible solution that is obtained through the above procedure is guaranteed to be locally optimal for the problem in (25a)–(25g). In practice, we observe that the penalty CCP usually converges to a feasible solution.

We now explain the convexification procedure for the constraint in (25b); the convexification of (25c) is performed by following the same procedure. Note that the last term on the right-hand side of (25b) is the summation of bilinear terms $c(s,s',a,z,u)\nu(\mathbf{s}')u(\gamma_a^{q,z})$ where $c(s,s',a,z,u)$ is a constant such that

$$c(s,s',a,z,u) := O_{s,z}\, P_{s,a,s'}\, u(\delta_{q'}^{q,z,a}).$$

With an abuse of notation, we denote $c(s,s',a,z,u)$ by c. As explained in [8], a bilinear function $f(x,y) = 2Cxy$, where C is a constant, can be written as a difference of convex functions $f(x,y) = f_1(x,y) - f_2(x,y)$, where $f_1(x,y) = C(x+y)^2$ and $f_2(x,y) = C(x^2+y^2)$. Since we have a constraint of the form $0 \leq L^u(\mathbf{s}) + f(x,y)$ in (25b), we linearize the function $f_1(x,y)$ around the point $\hat{\nu}_k(\mathbf{s})$ and $\hat{u}_k(\gamma_a^{q,z})$. The resulting expression then becomes concave in the variables $\nu(\mathbf{s}')$ and $u(\gamma_a^{q,z})$. Therefore, the resulting problem becomes a convex optimization problem.

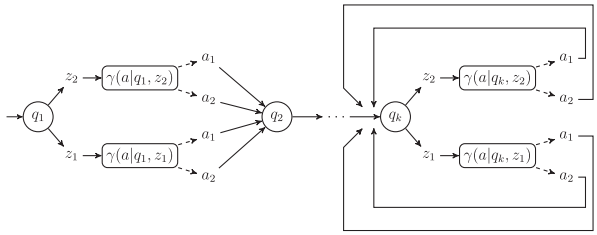

Fig. 3. Illustration of the proposed deterministic k-FSC structure. Regardless of the received observations and taken actions, the controller transitions from the memory state q_i to q_{i+1} for all $i < k$, and, finally, stays in the state q_k indefinitely.

E. FSC Synthesis: A Monotonocity Result

In the previous section, we presented an algorithm to solve the problem in (25a)–(25g), which requires one to set the variables $u(\delta_{q'}^{q,z,a})$ to constants that satisfy the constraint in (25e). In this section, we present a particular memory transition function which has a monotonocity property. That is, under this memory transition function, by increasing the number of memory states, one can only increase the optimal value of the optimization problem in (7a)–(7b).

For a POMDP $\mathbf{M}$, consider a k-FSC $\mathbf{C} = (\mathcal{Q}, q_1, \gamma, \overline{\delta})$ with the memory transition function $\overline{\delta} : \mathcal{Q} \times \mathcal{Z} \times \mathcal{A} \to \Delta(\mathcal{Q})$ such that

$$\begin{cases} \overline{\delta}(q_{i+1}|q_i,z,a) = 1 & \forall z \in \mathcal{Z}, a \in \mathcal{A}, 1 \leq i < k \\ \overline{\delta}(q_k|q_k,z,a) = 1 & \forall z \in \mathcal{Z}, a \in \mathcal{A} \\ \overline{\delta}(q_i|q_j,z,a) = 0 & \text{otherwise.} \end{cases} \tag{27}$$

We present an illustration of the k-FSC described previously in Fig. 3. Intuitively, the transition function $\overline{\delta}$ represents a finite horizon memory. In the first $k-1$ steps, the agent summarizes the set $\mathcal{H}^i$ of system histories using the memory state q_i and makes a decision based on the decision function $\gamma(a|q_i,z)$. For the rest of the process, the agent stays in the memory state q_k and follows a memoryless strategy by making stationary decisions based on $\gamma(a|q_k,z)$.

Let $\overline{\mathcal{F}}_k(\mathbf{M}) \subset \mathcal{F}_k^{\text{det}}(\mathbf{M})$ be the set of k-FSCs whose memory transition function is given in (27). In addition, let

$$E_{k,\max} := \max_{\mathbf{C} \in \overline{\mathcal{F}}_k(\mathbf{M})} \sum_{t=2}^{\infty} \beta^{t-2} H^{\mathbf{C}}(S_t|S^{t-1}).$$

Lemma 2: For all $j \leq k$, we have $E_{j,\max} \leq E_{k,\max}$.

The abovementioned monotonocity result establishes that, by *fixing* the memory transition function to $\overline{\delta}$ given in (27), one can obtain nondecreasing maximum entropy values by increasing the number of memory states in the k-FSC. We note that such a monotonocity result hold due to the specific structure of the transition function $\overline{\delta}$ and may not hold if one considers transition functions other than $\overline{\delta}$.

Using the result of Lemma 2, we can obtain a practical algorithm to synthesize an entropy-maximizing controller as follows. First, fix the number of memory states to an initial value k. Then, by setting $u(\delta_{q'}^{q,z,a}) = \overline{\delta}(q'|q,z,a)$, find a local optimal solution to the problem in (25a)–(25g). Next, set the number of memory states to $k+1$, solve the resulting problem, and compare the optimal value of the problem with the previous result. Repeat this procedure until the percent increase in the optimal value is below a predetermined threshold.

VI. A Numerical Example

We now provide a numerical example to demonstrate an application of the proposed method to motion planning. We use the MOSEK [32] solver with the CVX [33] interface to solve the convex optimization problems. To improve the approximation of exponential cone constraints, we use the CVXQUAD [34] package.

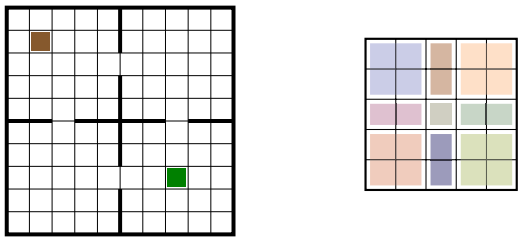

Fig. 4. Motion planning example. (Left-hand side) Grid world with 100 states and 36 observations. The agent starts from the brown state and aims to reach the green state. (Right-hand side) Partition of a room with respect to the agent's observation function.

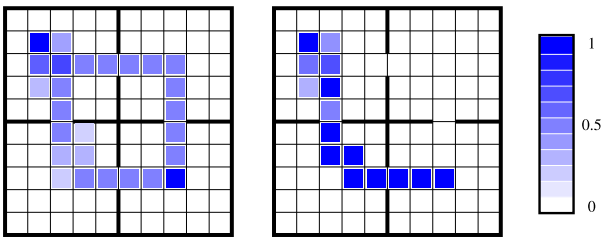

Fig. 5. Expected number of times the agent visits each state under the synthesized controllers. (Left-hand side) Entropy-maximizing controller. (Right-hand side) Controller synthesized by the approach given in [8].

We consider an agent that aims to reach a target in an adversarial environment. We model the environment as a 10×10 grid world, as shown in Fig. 4 (left-hand side), that consists of four *rooms* and four *doors* using which the agent can transition between the rooms. The rooms are numbered clockwise starting from the bottom left corner, and the doors are numbered clockwise starting from the door between room 1 and room 4. Finally, the thick black lines represent the walls.

The agent observes its current room and its relative position to the doors (36 total observations, 9 in each room). We illustrate the partition of a room with respect to the agent's observation function in Fig. 4 (right-hand side). For example, if the agent is at the bottom left corner of the environment, its observation is *room 1, below door 1, left of door 2*. In Fig. 4 (left-hand side), the brown and green states are the agent's initial and target state, respectively. We set the discount factor to $\beta = 0.9$, and the expected total reward threshold to $\Gamma = \beta^{12}$, i.e., the agent must reach the target state in at most 12 steps, which is the minimum number of steps to reach the target from the initial state. Hence, the agent can follow only the shortest trajectory to the target.

We focus on 1-FSCs and synthesize two controllers for the agent. We synthesize the first controller using the proposed approach based on the convex–concave procedure. For comparison, we also synthesize a controller by solving a feasibility problem given in [8]. In Fig. 5, we demonstrate the expected number of times the agent visits each state under the synthesized controllers. As can be seen from Fig. 5, under the entropy-maximizing controller, the agent reaches the target state by passing through room 1 and room 3 with equal probability, which minimizes the predictability of the room it visits to an outside observer by maximizing the entropy of its trajectories. On the other hand, under the controller synthesized by the feasibility approach, the agent always passes through room 1. Hence, it becomes trivial for an outside observer to predict the agent's trajectory.

VII. Conclusion

We studied the synthesis of a controller that, from a given POMDP, induces a stochastic process with maximum entropy among the ones whose realizations accumulate a certain level of expected reward. By restricting our attention to FSCs with deterministic memory transitions, we recast the entropy maximization problem as a so-called parameter synthesis problem for pMCs. We present a NLP for the synthesis of an FSC that maximizes the entropy of a POMDP over all FSCs with the same number of memory states and deterministic memory transitions. Considering the intractability of finding a global optimal solution to the presented optimization problem, we proposed a convex–concave procedure approach to obtain a local optimal solution after setting the memory transition of FSCs to a fixed structure.

Appendix

Proof of Lemma 1: For any $t < N$, we have

$$V^{\pi}_{t,N}(s^t) = H^{\pi}(S_{t+1}|S^t = s^t) + \sum_{k=t+1}^{N-1} \beta^{k-t} H^{\pi}(S_{k+1}|S^k_t, S^t = s^t)$$

since the random variable S_t is a part of the sequence S^t. The last term in the previous equation satisfies

$$H^{\pi}(S_{k+1}|S^k_t, S^t = H^{\pi}(S_{k+1}|S^k_{t+2}, S_{t+1}, S^t = s^t)$$

since $S^k_t = (S_t, S_{t+1}, S^k_{t+2})$. Moreover, the term on the left-hand side of the previous equality satisfies

$$H^{\pi}(S_{k+1}|S^k_{t+2}, S_{t+1}, S^t = s^t) = H^{\pi}(S_{k+1}|S^k_t, S_{t+1}, S^t = s^t)$$

since the introduced conditioning on (S_t, S_{t+1}) does not change the entropy as the random variable S_{t+1} is already included in the conditioning, and the value of the random variable S_t is already fixed to s_t. Using the definition of conditional entropy [1, Ch. 2]

$$H^{\pi}(S_{k+1}|S^k_t, S_{t+1}, S^t = s^t) = \sum_{s_{t+1} \in \mathcal{S}} \Pr(S_{t+1} = s_{t+1}|S^t = s^t) H^{\pi}(S_{k+1}|S^k_t, S^{t+1} = s^{t+1}).$$

Note that, under the policy π, $\Pr(S_{t+1} = s_{t+1}|S^t = s^t)$ is equal to the realization probability $\Pr^{\pi}(s^{t+1}|s^t)$. Furthermore, $\Pr^{\pi}(s^{t+1}|s^t) > 0$ for a given state history $s^{t+1} \in \mathcal{SH}^{t+1}$ if and only if $s^{t+1} = (s^t, s_{t+1})$ where $s_{t+1} \in \mathcal{S}$. As a result, we have

$$\begin{aligned} V^{\pi}_{t,N}(s^t) &= H^{\pi}(S_{t+1}|S^t = s^t) \\ &\quad + \sum_{k=t+1}^{N-1} \beta^{k-t} \sum_{s^{t+1} \in \mathcal{SH}^{t+1}} \Pr^{\pi}(s^{t+1}|s^t) \\ &\quad \times H^{\pi}(S_{k+1}|S^k_t, S^{t+1} = s^{t+1}) \qquad (28a) \\ &= H^{\pi}(S_{t+1}|S^t = s^t) \\ &\quad + \sum_{s^{t+1} \in \mathcal{SH}^{t+1}} \Pr^{\pi}(s^{t+1}|s^t) \sum_{k=t+1}^{N-1} \beta^{k-t} \\ &\quad \times H^{\pi}(S_{k+1}|S^k_t, S^{t+1} = s^{t+1}) \qquad (28b) \\ &= H^{\pi}(S_{t+1}|S^t = s^t) \\ &\quad + \beta \sum_{s^{t+1} \in \mathcal{SH}^{t+1}} \Pr^{\pi}(s^{t+1}|s^t) V^{\pi}_{t+1,N}(s^{t+1}) \qquad (28c) \end{aligned}$$

where (28b) follows from the fact that the expression $\Pr^{\pi}(s^{t+1}|s^t)$ does not depend on k, and (28c) follows from the definition of the value function $V^{\pi}_{t,N}(s^t)$. ■

Proof of Proposition 1: The result follows from the fact that the controller $\mathbf{C}$ only allows deterministic memory transitions and that $\mathbf{D}_{\mathbf{M},k}[u_{\mathbf{C}}]$ is an MC. By the definition of conditional entropy [1]

$$H^{\mathbf{C}}(S_t|S^{t-1}) = \sum_{s^t \in \mathcal{SH}^t} \Pr^{\mathbf{C}}(s_t, s^{t-1}) \log \Pr^{\mathbf{C}}(s_t|s^{t-1}). \qquad (29)$$

Note that the summation on the right-hand side of the abovementioned equation is over state histories. For any given state history s^t, there is a corresponding *memory history* $(q_1, q_2, \ldots, q_t)$, where $q_k \in \mathcal{Q}$, for the controller $\mathbf{C}$. Let $\mathcal{MH}^t$ denote the set of all possible memory histories of length $t \in \mathbb{N}$. Then, by the law of total probability

$$\Pr^{\mathbf{C}}(s_t|s^{t-1}) = \sum_{q^t \in \mathcal{MH}^t} \Pr^{\mathbf{C}}(s_t, q_t|q^{t-1}, s^{t-1}) \Pr^{\mathbf{C}}(q^{t-1}|s^{t-1}).$$

Since the memory transitions are deterministic under $\mathbf{C} \in \mathcal{F}_k^{det}(\mathbf{M})$, by recursively expanding the right-hand side of the previous equality, it can be observed that $\Pr^{\mathbf{C}}(q^{t-1}|s^{t-1}) = 1$ for a given state history realization s^{t-1}. Since for each state history realization s^t on the POMDP $\mathbf{M}$ under the controller $\mathbf{C}$, there is a unique state history realization $(\langle s_1, q_1\rangle, \langle s_2, q_2\rangle, \ldots, \langle s_t, q_t\rangle)$ on the instantiation $\mathbf{D}_{\mathbf{M},k}[u_{\mathbf{C}}]$ of the induced pMC $\mathbf{D}_{\mathbf{M},k}$, we have $H^{\mathbf{C}}(S_t|S^{t-1}) = H^{u_{\mathbf{C}}}(S_{\mathbf{M},k,t}|S^{t-1}_{\mathbf{M},k})$. Finally, since the instantiation $\mathbf{D}_{\mathbf{M},k}[u_{\mathbf{C}}]$ constitutes an MC, as a result of the Markov property [1], we have $H^{u_{\mathbf{C}}}(S_{\mathbf{M},k,t}|S^{t-1}_{\mathbf{M},k}) = H^{u_{\mathbf{C}}}(S_{\mathbf{M},k,t}|S_{\mathbf{M},k,t-1})$. ■

Proof of Lemma 2: We prove the claim by showing that, for any $k \in \mathbb{N}$, we have $E_{k-1,\max} \leq E_{k,\max}$.

Consider an arbitrary $(k-1)$-FSC $\mathbf{C} \in \overline{\mathcal{F}}_{k-1}(\mathbf{M})$ with the decision function γ. Let k-FSC $\mathbf{C}' \in \overline{\mathcal{F}}_k(\mathbf{M})$ be such that its decision function γ' satisfies $\gamma'(a|q_i, z) = \gamma(a|q_i, z)$ for $i = 1, \ldots, k-1$, and $\gamma'(a|q_k, z) = \gamma(a|q_{k-1}, z)$. Note that the state sequences in $\mathbf{M}$ under the controllers $\mathbf{C}$ and $\mathbf{C}'$ are the same. This is true since we can explicitly write down the decisions taken by the agent under the controllers $\mathbf{C}$ and $\mathbf{C}'$, thanks to the specific transition function given in (27). In particular, the sequence of decisions under the controller $\mathbf{C}$ is $(\gamma(a|q_1, z), \gamma(a|q_2, z), \ldots, \gamma(a|q_{k-1}, z), \gamma(a|q_{k-1}, z), \ldots)$, and the sequence of decisions under the controller $\mathbf{C}'$ is $(\gamma'(a|q_1, z), \gamma'(a|q_2, z), \ldots, \gamma'(a|q_k, z), \gamma'(a|q_k, z), \ldots)$, which are the same by construction. Hence, the state sequences induced by these decision sequences are the same. Consequently, we have

$$\sum_{t=2}^{\infty} \beta^{t-2} H^{\mathbf{C}}(S_t|S^{t-1}) = \sum_{t=2}^{\infty} \beta^{t-2} H^{\mathbf{C}'}(S_t|S^{t-1}).$$

Since, for an arbitrary $(k-1)$-FSC $\mathbf{C} \in \overline{\mathcal{F}}_{k-1}(\mathbf{M})$, there exists a k-FSC $\mathbf{C}' \in \overline{\mathcal{F}}_k(\mathbf{M})$ that achieves the same entropy of state sequences in $\mathbf{M}$, we conclude that $E_{k-1,\max} \leq E_{k,\max}$. ■

REFERENCES

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 1991.

[2] F. Biondi, A. Legay, B. F. Nielsen, and A. Wasowski, "Maximizing entropy over Markov processes," *J. Log. Algebr. Methods Program.*, vol. 83, no. 5, pp. 384–399, 2014.

[3] Y. Savas, M. Ornik, M. Cubuktepe, M. O. Karabag, and U. Topcu, "Entropy maximization for Markov decision processes under temporal logic constraints," *IEEE Trans. Autom. Control*, vol. 65, no. 4, pp. 1552–1567, Apr. 2020.

[4] O. Madani, S. Hanks, and A. Condon, "On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems," in *Proc. 16th Nat. Conf. Artif. Intell. 11th Innov. Appl. Artif. Intell. Conf. Innov. Appl. Artif. Intell.*, 1999, pp. 541–548.

[5] K. Chatterjee, M. Chmelík, and M. Tracol, "What is decidable about partially observable Markov decision processes with ω-regular objectives," *J. Comput. Syst. Sci.*, vol. 82, no. 5, pp. 878–911, 2016.

[6] N. Meuleau, K.-E. Kim, L. P. Kaelbling, and A. R. Cassandra, "Solving POMDPs by searching the space of finite policies," in *Proc. Conf. Uncertainty Artif. Intell.*, 1999, pp. 417–426.

[7] P. Poupart and C. Boutilier, "Bounded finite-state controllers," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2004, pp. 823–830.

[8] M. Cubuktepe, N. Jansen, S. Junges, J.-P. Katoen, and U. Topcu, "Synthesis in pMDPs: A tale of 1001 parameters," in *Proc. Int. Symp. Automated Technol. Verification Anal.*, 2018, pp. 160–176.

[9] C. Baier, C. Hensel, L. Hutschenreiter, S. Junges, J.-P. Katoen, and J. Klein, "Parametric Markov chains: PCTL complexity and fraction-free Gaussian elimination," *Inf. Comput.*, vol. 272, 2020, Art. no. 104504.

[10] M. Hibbard, Y. Savas, B. Wu, T. Tanaka, and U. Topcu, "Unpredictable planning under partial observability," in *Proc. IEEE 58th Conf. Decis. Control*, 2019, pp. 2271–2277.

[11] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of Markov decision processes," *Math. Operations Res.*, vol. 12, no. 3, pp. 441–450, 1987.

[12] M. Lauri and R. Ritala, "Stochastic control for maximizing mutual information in active sensing," in *Proc. Int. Conf. Robot. Automat.*, 2014, pp. 1–6.

[13] A. Ryan and J. K. Hedrick, "Particle filter based information-theoretic active sensing," *Robot. Auton. Syst.*, vol. 58, no. 5, pp. 574–584, 2010.

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

[15] T. Haarnoja, V. Pong, A. Zhou, M. Dalal, P. Abbeel, and S. Levine, "Composable deep reinforcement learning for robotic manipulation," in *Proc. Int. Conf. Robot. Automat.*, 2018, pp. 6244–6251.

[16] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1352–1361.

[17] X. Zhang, B. Wu, and H. Lin, "Learning based supervisor synthesis of POMDP for PCTL specifications," in *Proc. Conf. Decis. Control*, 2015, pp. 7470–7475.

[18] J. K. Pajarinen and J. Peltonen, "Periodic finite state controllers for efficient POMDP and DEC-POMDP planning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 2636–2644.

[19] E. A. Hansen, "Solving POMDPs by searching in policy space," in *Proc. Conf. Uncertainty Artif. Intell.*, 1998, pp. 211–219.

[20] M. Ahmadi, R. Sharan, and J. W. Burdick, "Stochastic finite state control of POMDPs with LTL specifications," 2020, *arXiv:2001.07679*.

[21] N. Meuleau, L. Peshkin, K.-E. Kim, and L. P. Kaelbling, "Learning finite-state controllers for partially observable environments," in *Proc. Conf. Uncertainty Artif. Intell.*, 1999, pp. 427–436.

[22] C. Amato, D. S. Bernstein, and S. Zilberstein, "Optimizing fixed-size stochastic controllers for POMDPs and decentralized POMDPs," *Auton. Agents Multi-Agent Syst.*, vol. 21, pp. 293–320, 2010.

[23] S. Junges et al., "Finite-state controllers of POMDPs using parameter synthesis," in *Proc. Conf. Uncertainty Artif. Intell.*, 2018, pp. 519–529.

[24] F. Biondi, "Markovian processes for quantitative information leakage," Ph.D. thesis, Dept. Comput. Sci., IT Univ. Copenhagen, Copenhagen, Denmark, 2014.

[25] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2014.

[26] Z. Zhou, M. Bloem, and N. Bambos, "Infinite time horizon maximum causal entropy inverse reinforcement learning," *IEEE Trans. Autom. Control*, vol. 63, no. 9, pp. 2787–2802, Sep. 2018.

[27] J. Mei, C. Xiao, C. Szepesvari, and D. Schuurmans, "On the global convergence rates of softmax policy gradient methods," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 6820–6829.

[28] L. P. Hansen, T. J. Sargent, G. Turmuhambetova, and N. Williams, "Robust control and model misspecification," *J. Econ. Theory*, vol. 128, no. 1, pp. 45–90, 2006.

[29] G. P. McCormick, "Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems," *Math. Program.*, vol. 10, pp. 147–175, 1976.

[30] A. L. Yuille and A. Rangarajan, "The concave-convex procedure," *Neural Comput.*, vol. 15, no. 4, pp. 915–936, 2003.

[31] T. Lipp and S. Boyd, "Variations and extension of the convex-concave procedure," *Optim. Eng.*, vol. 17, pp. 263–287, 2016.

[32] M. ApS, *MOSEK Optimizer API for Python. Version 8.1.*, 2019. [Online]. Available: https://docs.mosek.com/8.1/pythonapi/index.html

[33] M. Grant and S. Boyd, "CVX: MATLAB software for disciplined convex programming, version 2.1," Mar. 2014. [Online]. Available: http://cvxr.com/cvx

[34] H. Fawzi, J. Saunderson, and P. A. Parrilo, "Semidefinite approximations of the matrix logarithm," *Found. Comput. Math.*, vol. 19, no. 2, pp. 259–296, 2019.