Controller Synthesis for Omega-Regular and Steady-State Specifications

Alvaro Velasquez
Air Force Research Laboratory
Rome, New York, USA
alvaro.velasquez.1@us.af.mil

Ismail Alkhouri
University of Central Florida
Orlando, Florida, USA
ialkhouri@knights.ucf.edu

Andre Beckus
Air Force Research Laboratory
Rome, New York, USA
andre.beckus@us.af.mil

Ashutosh Trivedi
University of Colorado Boulder
Boulder, Colorado, USA
ashutosh.trivedi@colorado.edu

George Atia
University of Central Florida
Orlando, Florida, USA
george.atia@ucf.edu

ABSTRACT

Given a Markov decision process (MDP) and a linear-time (ω-regular or Linear Temporal Logic) specification which reasons about the infinite-trace behavior of a system, the controller synthesis problem aims to compute the optimal policy that satisfies said specification. Recently, problems that reason over the complementary infinite-frequency behavior of systems have been proposed through the lens of steady-state planning or steady-state policy synthesis. This entails finding a control policy for an MDP such that the Markov chain induced by the solution policy satisfies a given set of constraints on its steady-state distribution. This paper studies a generalization of the controller synthesis problem for a linear-time specification under steady-state constraints on the asymptotic behavior of the agent. We present an algorithm to find a deterministic policy satisfying ω-regular and steady-state constraints by characterizing the solutions as an integer linear program, and experimentally evaluate our approach.

KEYWORDS

Planning for Deterministic Actions; Constrained MDPs; Omega-Regular; Multichain MDPs; Steady-State; Controller Synthesis; Linear Temporal Logic; Expected Reward; Average Reward; Correct-by-Construction

ACM Reference Format:
Alvaro Velasquez, Ismail Alkhouri, Andre Beckus, Ashutosh Trivedi, and George Atia. 2022. Controller Synthesis for Omega-Regular and Steady-State Specifications. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 12 pages.

1 INTRODUCTION

The controller synthesis problem is often used to establish safety and performance guarantees of stochastic systems such as Markov decision processes (MDPs) by inducing Markov chains exhibiting some desirable behavior. The ω-regular languages [1, 2] provide an expressive formalism to unambiguously express such safety and progress properties of MDPs, while Linear Temporal Logic (LTL) provides a convenient and interpretable way to encode such ω-regular properties. For the verification or synthesis of systems subject to these properties, an ω-regular objective is usually translated into a finite-state machine that monitors the traces of the MDP [3]. Successful executions cause the finite-state machine to take certain accepting transitions infinitely often, and ultimately avoid certain rejecting transitions. That is, ω-regular objectives reason about the asymptotic trace behavior of an MDP. The related notion of asymptotic frequency of states visited is not accounted for in such objectives. To illustrate the utility of being able to reason about both of these types of behavior, consider a simple robot tasked to explore terrain on Mars. For such a mission, one may come up with ω-regular specifications that its traces of behavior should satisfy. For example, we may impose that, whenever a state labeled "ice" is encountered, the robot must collect a sample and drop it off at a state labeled "base". Furthermore, the robot may also need to spend a certain proportion of its time – but not too much time so as not to conflict with gathering ice samples – exploring certain regions of the Martian landscape. Indeed, the robot may be requested to spend at least 25% of its time in regions of interest, but no more than 50% of its time. This is easily encoded as a steady-state specification. Such specifications cannot be directly expressed in LTL.

LTL controller synthesis through probabilistic model checking approaches [3, 4] generally begins by computing the product MDP from the original MDP and the finite-state machine representation of the given objective. Then, the union of accepting maximal end components (AMECs) are computed and a policy is found such that the agent reaches some such component. Once there, actions can be chosen arbitrarily such that all states within the AMEC are visited infinitely often, thereby ensuring that the acceptance condition of the automaton is met and the objective is therefore satisfied by said policy. Generally, this choice of actions within the AMECs is arbitrary. *However, it is evident that these choices are critical in the situations with constraints on the steady-state distribution.* This distribution characterizes the asymptotic frequency behavior of a Markov chain induced by some policy in an MDP.

The controller synthesis problem subject to steady-state specifications has been explored recently [5–8] and the integration of LTL constraints has been considered for stochastic policy settings [9]. In this paper, we seek to complement and unify much of the preceding work by reasoning about both ω-regular properties as well as steady-state distributions simultaneously and without making

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

common assumptions of ergodicity on the underlying MDP. The proposed approach finds an optimal expected-reward deterministic policy that satisfies given ω-regular and steady-state specifications. The computation of deterministic policies is an important avenue of inquiry when guarantees or predictable behavior are desired [10].

2 PRELIMINARIES

We recall classical definitions and introduce notation for the paper.

Markov Decision Processes. A *probability distribution* over a finite set S is a function $d : S \to [0, 1]$ such that $\sum_{s \in S} d(s) = 1$. Let $D(S)$ denote the set of all discrete distributions over S. A Labeled Markov Decision Process (LMDP) $\mathcal{M}$ is a tuple $(S, \beta, A, T, R, AP, L)$, where S is a finite set of states, $\beta \in D(S)$ is the initial state distribution, A is a finite set of actions, $T : S \times A \to D(S)$ is the transition function, $R : S \times A \times S \to \mathbb{R}$ is the reward signal, AP is the set of atomic propositions, and $L : S \to 2^{AP}$ is the *labeling function*.

For any state $s \in S$, we let $A(s)$ denote the set of actions that can be selected in state s. For states $s, s' \in S$ and $a \in A(s)$, $T(s, a)(s')$ equals $p(s'|s, a)$. A *run* of $\mathcal{M}$ is a sequence $\langle s_0, a_1, s_1, \ldots \rangle \in S \times (A \times S)^*$ such that $p(s_{i+1}|s_i, a_{i+1}) > 0$ for all $i \geq 0$. A finite run is a finite such sequence. When convenient, runs are sometimes defined as sequences of states, without including actions. For a *run* $r = \langle s_0, a_1, s_1, \ldots \rangle$, we define the corresponding labeled run as $L(r) = \langle L(s_0), L(s_1), \ldots \rangle \in \left(2^{AP}\right)^+$. A policy (or a strategy) is a recipe for a decision-maker to resolve the non-determinism of the LMDP. A policy in $\mathcal{M}$ is a function $\pi : S^+ \to D(A)$ mapping finite runs to actions. A policy is *finite-memory* if it remembers a finite amount of information about the past and a finite-memory policy can be represented using a finite-state machine. In this paper, we are interested in finite-memory deterministic policies of the form $\pi : S \times Q \to A$, where Q is a set of memory modes. This memory is obtained from the finite-state machine representation of the given linear-time specification to be satisfied. We write $\pi(a|s, q) \in \{0, 1\}$ for the probability of choosing action a in the state s when the memory mode is q. For the remainder of this paper, we assume finite-memory deterministic policies π. For an LMDP $\mathcal{M} = (S, \beta, A, T, R, AP, L)$, a finite-memory deterministic policy π resolves its non-determinism and gives rise to a Labeled Markov Chain (LMC) $\mathcal{M}_\pi = (S_\pi, \beta_\pi, T_\pi, R_\pi, AP_\pi, L_\pi)$. Note that an LMC is an LMDP whose set of actions is a singleton and hence can be omitted. It is customary to represent the probabilistic transition function T of the LMC as a matrix such that $T_{i,j} = T(s_i)(s_j)$. When other information is not pertinent, we write an LMC as (S, T).

Given an LMDP $\mathcal{M} = (S, \beta, A, T, R, AP, L)$, we define its underlying directed graph $G_\mathcal{M} = (V, E)$, where $V = S$ and $E \subseteq S \times S$ is such that $(s, s') \in E$ if $T(s, a)(s') > 0$ for some $a \in A(s)$. A sub-MDP of $\mathcal{M}$ is an LMDP $\mathcal{M}' = (S', \beta', A', T', R', AP', L')$, where $S' \subseteq S$, $A' \subseteq A$ is such that $A'(s) \subseteq A(s)$ for every $s \in S'$, and β', T', R' and L' are analogous to β, T, R, and L when restricted to S' and A'. An *end component* [3] of an LMDP $\mathcal{M}$ is a sub-MDP $\mathcal{M}'$ of $\mathcal{M}$ such that $G_{\mathcal{M}'}$ is strongly connected. A *bottom strongly connected component* (BSCC) of an LMC is any of its maximal end components (MECs), where a MEC is an end component that is maximal under set inclusion.

Linear-Time Specifications. Given the set of atomic propositions AP of an LMDP $\mathcal{M}$, a linear-time property of $\mathcal{M}$ is characterized by an ω-language, i.e., a set of infinite sequences over the alphabet $\Sigma = 2^{AP}$. Formally, an *ω-word* w on an alphabet Σ is a function $w : \mathbb{N} \to \Sigma$. We abbreviate $w(i)$ by w_i. The set of ω-words on Σ is written Σ^ω and a subset of Σ^ω is an *ω-language*. We are interested in expressing properties using ω-regular languages given as a type of finite-state machine. In this context, we choose deterministic Rabin automata (DRA) as defined in the sequel.

A *deterministic Rabin automaton* (DRA) $\mathcal{A}$ is a tuple $(\Sigma, Q, q_0, \delta, F)$, where Σ is a finite *alphabet*, Q is a finite set of *states*, $q_0 \in Q$ is the *initial state*, $\delta : Q \times \Sigma \to Q$ is the *transition function*, and $F = \left\{(B_i, G_i) \in 2^Q \times 2^Q\right\}_{i \in [m]}$ is the *Rabin acceptance condition*. A *run* r of a DRA $\mathcal{A}$ on $w \in \Sigma^\omega$ is an ω-word $r_0, w_0, r_1, w_1, \ldots$ in $(Q \cup \Sigma)^\omega$ such that $r_0 = q_0$ and, for $i > 0$, $r_i = \delta(r_{i-1}, w_{i-1})$. We write $\inf(r) \subseteq Q$ for the set of states that appear infinitely often in the run r. A run r of a DRA $\mathcal{A}$ is *accepting* if there is some $(B, G) \in F$ such that $\inf(r) \cap B = \emptyset$ and $\inf(r) \cap G \neq \emptyset$. The *language* of $\mathcal{A}$ (or, *accepted* by $\mathcal{A}$) is the subset of words in Σ^ω that have accepting runs in $\mathcal{A}$. A language is *ω-regular* iff it is accepted by a DRA [4].

Given an LMDP $\mathcal{M}$ and an ω-regular objective φ given as a DRA $\mathcal{A} = (\Sigma, Q, q_0, \delta, F)$, the controller synthesis problem is to compute a policy that maximizes the probability of satisfaction of the ω-regular objective. This problem is typically reduced to solving a product LMDP as shown in Figure 1. Given an LMDP $\mathcal{M} = (S, \beta, A, T, R, AP, L)$ and a DRA $\mathcal{A} = (2^{AP}, Q, q_0, \delta, F)$, their *product* LMDP $\mathcal{M} \times \mathcal{A}$ is the tuple $(S^\times, \beta^\times, A^\times, T^\times, R^\times, Q, L^\times)$, where $S^\times = S \times Q$; $\beta^\times \in D(S^\times)$ is such that for all $(s, a) \in S^\times$, we have that $\beta^\times(s, q)$ equals $\beta(s)$ if $q = \delta(q_0, L(s))$ and is 0 otherwise; $A^\times = A$ and $A^\times(s, q) = A(s)$ for all $(s, q) \in S^\times$; $T^\times : S^\times \times A^\times \mapsto S^\times$ is such that for all $(s, q), (s', q') \in S^\times$ and $a \in A(s, q)$ we have $T^\times((s, q), a)(s', q')$ equals $T(s, a)(s')$ if $q' = \delta(q, L(s'))$ and is 0 otherwise; $R^\times((s, q), a) = R(s, a)$ for all $(s, q) \in S^\times$ and $a \in A(s, q)$; and $L^\times((s, q)) = \{q\}$ for all $(s, q) \in S^\times$ [4].

End components and runs are defined for products just like for LMDPs. The acceptance condition for the product LMDP can be lifted from the DRA and is used to define accepting MECs (AMECs). An AMEC of a product LMDP $\mathcal{M} \times \mathcal{A}$ is a MEC such that every run of the product LMDP that eventually dwells in it is accepting. Formally, a MEC $E = (S^E, A^E)$ of $\mathcal{M} \times \mathcal{A}$ is accepting if $S^E \cap (S \times B) = \emptyset$ and $S^E \cap (S \times G) \neq \emptyset$ for some $(B, G) \in F$. The satisfaction of an ω-regular objective φ by an LMDP $\mathcal{M}$ can be formulated in terms of AMECs of the product $\mathcal{M} \times \mathcal{A}_\varphi$, where $\mathcal{A}_\varphi$ is a DRA accepting φ. The maximum probability of satisfaction of φ by $\mathcal{M}$ is the maximum probability, over all policies, that a run of the product LMDP $\mathcal{M} \times \mathcal{A}_\varphi$ eventually dwells in one of its AMECs [3, 11]. Once an AMEC is reached, one must simply choose actions in the AMEC infinitely often in order to ensure that all states within it are visited infinitely often. It is worth noting that there always exists a stationary and deterministic policy over the product LMDP $\mathcal{M} \times \mathcal{A}$ to maximize the probability of visiting AMECs. This policy defines the optimal finite-memory policy over the original LMDP $\mathcal{M}$ to satisfy the ω-regular objective given by the DRA $\mathcal{A}$. The DRA states $q \in Q$ in $\pi : S \times Q \to A$ in the product LMDP naturally define the memory mode of the finite-memory policy in the original LMDP [12].

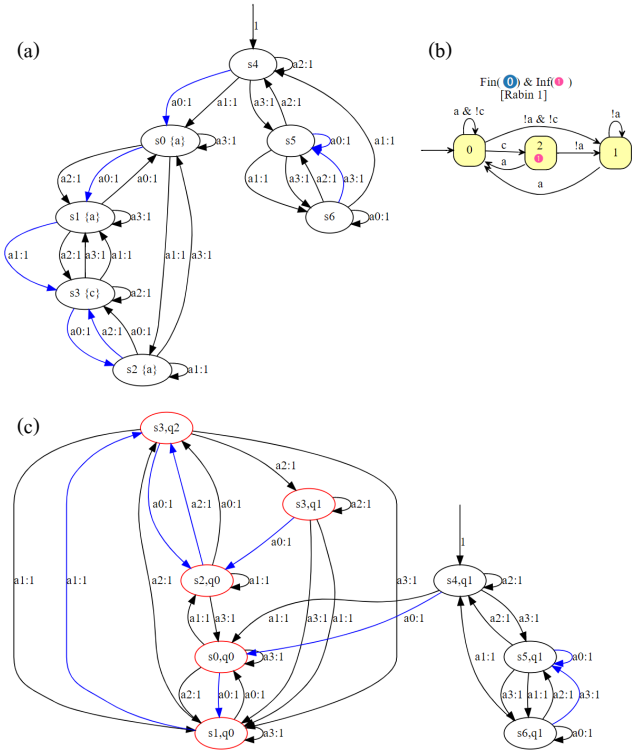

Figure 1: (a) LMDP $\mathcal{M} = (S, \beta, A, T, R = \emptyset, AP, L)$, where $S = \{s_0, \ldots, s_6\}$, $\beta(s_4) = 1$, $A = \{a_0, \ldots, a_3\}$, $AP = \{a, c\}$, and $L(s_0) = L(s_1) = L(s_2) = \{a\}, L(s_3) = \{c\}$. The transition function is deterministic and shown in the figure by the transitions $a_i : 1$ between states s, s' denoting that $T(s, a_i)(s') = 1$. The blue transitions define a policy π which induces a unichain LMC $\mathcal{M}_\pi$ (The isolated component consisting of states s_5 and s_6 is ignored since it is unreachable). (b) The DRA $\mathcal{A} = (Q, q_0, \Sigma, \delta, F = \{(B_i, G_i)\}_i)$ is shown, where $Q = \{q_0, q_1, q_2\}$, $\Sigma = 2^{AP}$, $F = \{(\emptyset, \{q_2\})\}$ and !, & denote logical negation and conjunction. (c) Product LMDP $\mathcal{M} \times \mathcal{A}$, where red nodes represent states in the accepting MEC of $\mathcal{M} \times \mathcal{A}$. The blue transitions define the product LMC $(\mathcal{M} \times \mathcal{A})_\pi$ induced by the policy π. Note that this policy induces an LMC that has probability 1 of being trapped in the accepting MEC. Therefore, the probability of satisfying the ω-regular property represented by $\mathcal{A}$ given that we start in state s_0 is 1.

Steady-State Constraints. Let $\mathcal{M} = (S, \beta, T, R, AP, L)$ be an LMC. A state $s' \in S$ in $\mathcal{M}$ is *reachable* from a state $s \in S$, denoted by $s \hookrightarrow s'$, if there exists a run $\langle s_i, s_j, \ldots, s_k \rangle \in S^+$ such that $s_i = s$, $s_k = s'$, and for all $0 \leq i < k$ we have that $T(s_i)(s_{i+1}) > 0$. We say that two states $s, s' \in S$ communicate if $s \hookrightarrow s'$ and $s' \hookrightarrow s$. A Markov chain is *irreducible* if every pair of states $s, s' \in S$ communicates. A state $s \in S$ is *recurrent* if for all states $s' \in S$ such that $s \hookrightarrow s'$, we have that $s' \hookrightarrow s$. A transient state is a state that is not recurrent.

A recurrent component $C \subseteq S$ of states is a nonempty set of states such that every state in C communicates with every other state in C, and does not communicate with the states not in C. A *unichain* is an LMC that contains a single recurrent component and possibly some transient states. Otherwise, it is called a *multichain*. Our proposed approach finds a unichain LMC $\mathcal{M}_\pi$ in the original LMDP $\mathcal{M}$ that satisfies a given set of linear-time and steady-state constraints, though its corresponding product LMC $(\mathcal{M} \times \mathcal{A})_\pi$ in the product LMDP $\mathcal{M} \times \mathcal{A}$ may be a multichain.

The steady-state distribution $\Pr^\infty \in D(S)$ of an LMC $\mathcal{M} = (S, \beta, T, R, AP, L)$ denotes the proportion of time spent in each state as the number of transitions within $\mathcal{M}$ approaches ∞. This distribution is characterized by the following system of steady-state equations:

$$(\Pr^\infty(s_1), ..., \Pr^\infty(s_{|S|})) \cdot T = (\Pr^\infty(s_1), ..., \Pr^\infty(s_{|S|}))$$
$$\sum_{s \in S} \Pr^\infty(s) = 1 \tag{1}$$

The unichain condition is sufficient for LMCs to yield solutions to the steady-state equations in system (1). In particular, solutions in such settings yield the unique stationary distribution corresponding to the true steady-state behavior of the agent. For multichain LMCs, however, solutions to these equations may not be unique and may not correspond to the true steady-state behavior of the agent [13]. Indeed, consider the following simple example. Let M be a Markov chain defined over states $S = \{s_0, s_1, s_2\}$ such that $T(s_0)(s_1) = 0.6, T(s_0)(s_2) = 0.4, T(s_1)(s_1) = T(s_2)(s_2) = 1$. That is, state s_0 connects to s_1 and s_2 whereas these states self-loop with probability 1. Solving the steady-state equations for this Markov chain yields the trivial identities $\Pr^\infty(s_0) = 0, \Pr^\infty(s_1) = \Pr^\infty(s_1), \Pr^\infty(s_2) = \Pr^\infty(s_2)$, and the equation $\Pr^\infty(s_1) + \Pr^\infty(s_2) = 1$. Note that there are an infinite number of solutions to this equation. This elucidates the challenge of reasoning about steady-state constraints in multichain settings. We circumvent these challenges by focusing our attention on multichain product LMCs whose BSCCs share some state of the original LMC. We show that this is a necessary and sufficient condition for the original LMC to be a unichain. Furthermore, this restricts the BSCCs of the product LMC to be identical to one another in that their transition matrices are the same (up to row ordering). This yields a one-to-one correspondence between the solutions to the steady-state equations in the product LMC and the solutions to the steady-state equations in the original LMC. Since the original LMC is a unichain, this implies that these solutions will reflect the true steady-state behavior of the agent.

Given an LMDP $\mathcal{M} = (S, \beta, A, T, R, AP, L)$, the inverse of the labeling function $L^{-1} : 2^{AP} \to 2^S$ returns the states where a given set of atomic propositions holds. More generally, given a Boolean formula over atomic propositions $\psi \triangleq \textbf{true} \mid p \in AP \mid \psi_1 \wedge \psi_2 \mid \neg\psi$, the function $L^{-1}(\psi) \subseteq S$ returns the set of states where ψ holds. We now formalize what a steady-state specification is.

Definition 2.1 (Steady-State Specification). Given an LMC $\mathcal{M} = (S, \beta, T, R, AP, L)$ and a Boolean formula ψ over AP, a steady-state specification is a constraint of the form $l \leq \sum_{s \in L^{-1}(\psi)} \Pr^\infty(s) \leq u$, where l and u are user-defined bounds. We let $\mathrm{SS}_{[l,u]}\psi$ denote such specifications.

3 RELATED WORK

The controller synthesis problem given ω-regular objectives has been studied at length in the literature, particularly under the name

of LTL controller synthesis [14–16]. Traditionally, such problems are solved by efficiently computing the set of AMECs [17] and finding a policy that reaches these and visits an accepting state therein infinitely often. The problem of deriving a control policy which satisfies constraints on the steady-state distribution of the resulting agent has been studied more recently [5–7]. However, the literature on solving expected-reward constrained MDPs has often studied similar problems given that the expected-reward objective leverages the steady-state distribution or occupation measures, which are analogous to the steady-state distribution over state-action pairs, in order to determine expected policy values [18–21]. However, the common assumption that all policies yield an irreducible Markov chain is adopted for these methods. Indeed, the stronger ergodic assumption is often made in average-reward reinforcement learning problems ([22], Sections 10.3, 13.6).

While various extensions to LTL have introduced average [23], discounted [24], mean-payoff [25], and frequency [26] modalities to the logic, to the best of the authors' knowledge, the two facets of asymptotic behavior given by the steady-state (SS) distribution and linear-time (LTL) behavior of the agent have not yet been incorporated for the deterministic controller synthesis problem. To reiterate one of the challenges in this SS+LTL controller synthesis, the choice of actions within AMECs is critical since it is the states within said AMECs that will contribute to the steady-state distribution of the Markov chain induced by the solution policy. All other states would be transient or not visited, yielding a steady-state probability measure of 0. While this challenge is not present in traditional controller synthesis problems, a restricted form of it is addressed in the problem of LTL controller synthesis subject to persistent surveillance costs [27]. The goal in these problems is to satisfy a given LTL formula, or some restricted logic fragment thereof, while minimizing the cost incurred between satisfactions of a given surveillance goal specified as the repeated observance of a goal state. Perhaps the work most relevant to the results established in this paper stems from [5] and [6]. In [5], the Steady-State Control (SSC) problem is introduced. This is then generalized as Steady-State Policy Synthesis (SSPS) in [6]. In particular, the SSC problem entails finding a policy whose induced Markov chain satisfies a given steady-state distribution. This problem assumes that the underlying MDP is ergodic in that every policy yields irreducible Markov chains. This ensures that steady-state distributions reflect the true asymptotic behavior of the Markov chain. This is a fairly common assumption as observed recently by Altman in [28] for average-reward or -cost problems in constrained MDPs. In [6], the SSPS problem is posed as a generalization of SSC by allowing steady-state constraints to contain inequalities as well as probability intervals. The solution proposed therein does not assume ergodic MDPs and instead finds an irreducible Markov chain within an arbitrary MDP, if one exists, such that steady-state constraints are satisfied. However, that approach cannot handle transient states nor multichain MDPs. These issues were addressed recently in [7] and [8], wherein a solution to the steady-state planning problem is proposed for multichain MDPs by focusing on a restricted class of policies, such as imposing that all actions be taken with some probability by the solution policy or that the long-term play is restricted to the bottom strongly connected (BSCCs) of the MDP. Indeed, the general problem of finding policies that satisfy arbitrary steady-state constraints in multichain MDPs remains open. This warrants an important distinction in our setting. Even though the product MDP over which we define our solution may be multichain, our setting is restricted in that we search for a policy that induces a product Markov chain whose BSCCs are isomorphic to one another in that their graph structures are identical. As we demonstrate, this is a necessary and sufficient condition for the original Markov chain (in the original MDP) to be a unichain, thereby ensuring that the steady-state equations admit a solution corresponding to the steady-state behavior of the agent.

Our solution to what we call the SS+LTL controller synthesis problem unifies much of the foregoing by reasoning about both linear-time ω-regular properties as well as steady-state distributions simultaneously. The proposed approach finds an optimal expected-reward control policy that is deterministic and satisfies the given steady-state (SS) and LTL specifications. We do not assume that the underlying MDP is ergodic nor communicating. Instead, our solution finds a unichain Markov chain satisfying the given specifications, if one exists. This complements the recent results in [9], where a stochastic history-dependent (possibly with unbounded memory) policy as in [29] is computed for the LTL-constrained steady-state policy synthesis problem. It is worth noting that, from a complexity perspective, these are fundamentally different problems due to the distinction between stochastic and deterministic policies. Indeed, finding a stochastic policy for this problem is in the complexity class **P** as demonstrated by the polynomial-time solution proposed in [9]. On the other hand, the problem of computing a deterministic policy in this setting is an **NP-complete** problem [6]. Therefore, a polynomial-time solution is not likely to exist.

4 SS+LTL CONTROLLER SYNTHESIS

We combine the linear-time and steady-state specifications and solve the corresponding controller synthesis problem. Given an LMC $\mathcal{M}$ and a steady-state specification $\mathbf{SS}_{[l,u]}(\psi)$, we say $\mathcal{M}$ satisfies $\mathbf{SS}_{[l,u]}(\psi)$, denoted by $\mathcal{M} \models \mathbf{SS}_{[l,u]}(\psi)$, iff $\Sigma_{s \in L^{-1}(\psi)} \Pr^{\infty}(s) \in [l,u]$ per Definition 2.1. Given an LTL formula ϕ defined inductively over a set of atomic propositions AP, Boolean connectives, and temporal modalities *next*, *until*, *eventually*, *always* ($\mathbf{X}$, $\mathbf{U}$, $\mathbf{F}$, $\mathbf{G}$), the satisfaction semantics $\mathcal{M} \models \phi$ are defined in the standard way [4]. We are interested in the combination of these LTL and SS specifications, henceforth referred to as SS+LTL specifications denoted by $\theta = (\phi_{\text{LTL}}, (\mathbf{SS}_{[l_i,u_i]}\psi_i)_i)$. We say that $\mathcal{M}$ satisfies θ, denoted by $\mathcal{M} \models \theta$, if $\mathcal{M} \models \phi_{\text{LTL}}$ and $\mathcal{M} \models \mathbf{SS}_{[l_i,u_i]}\psi_i$ for all i.

Definition 4.1 (Deterministic SS+LTL Controller Synthesis). Given an LMDP $\mathcal{M}$ and SS+LTL specification θ, compute a finite-memory deterministic policy π, if one exists, such that $\mathcal{M}_\pi \models \theta$ and π maximizes the expected reward among all such policies.

Let us fix an LMDP $\mathcal{M} = (S, \beta, A, T, R, AP, L)$ and an SS+LTL specification $\theta = (\phi_{\text{LTL}}, (\mathbf{SS}_{[l_i,u_i]}\psi_i)_i)$ for the rest of the paper. Recall that the LTL formula ϕ_{LTL} can be compiled into a DRA $\mathcal{A}$. In what follows, we work with the product LMDP $\mathcal{M} \times \mathcal{A} = (S^\times, \beta^\times, A^\times, T^\times, R^\times, Q, L^\times)$, sometimes referred to as $\mathcal{M}^\times$ for convenience. Our goal is to characterize the existence of a stationary and deterministic policy $\pi : S \times Q \to A$ over the product LMDP. This, in turn, is equivalent to a finite-memory deterministic policy over the original LMDP. See Figure 2 for an example.

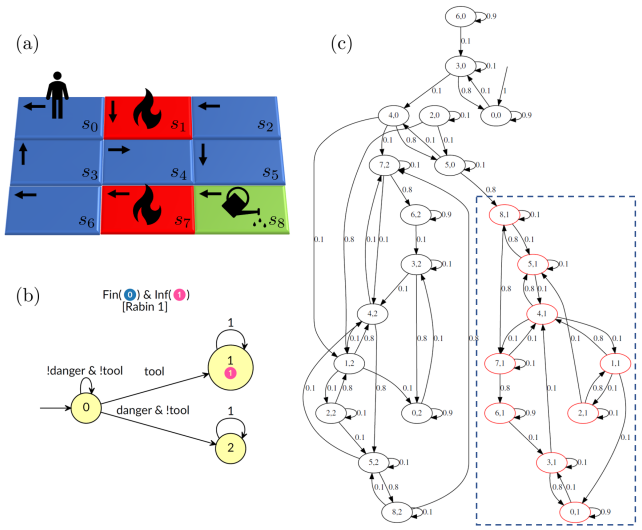

Figure 2: (a) LMDP $\mathcal{M} = (S, \beta, A, T, R = \emptyset, AP, L)$, where $S = \{s_0, \ldots, s_8\}$, $\beta(s_0) = 1$, $A = \{\leftarrow, \downarrow, \rightarrow, \uparrow\}$, $AP = \{home, danger, tool\}$, and $L(s_0) = \{home\}, L(s_1) = L(s_7) = \{danger\}, L(s_8) = \{tool\}$. the agent has a chance of slipping whenever it moves, causing a transition into one of three possible states. If the agent chooses to go right (left), there is an 80% chance that it will transition to the right (left), and the chance of transitioning to either of the states above or below it is 10% each. Similarly, if the agent chooses to go up (down), it will end up in the states above (below) it with 80% chance, and in the states to the right and left of it with probability 10% each. In the corners of the map, the agent may stay in place with 90% probability by choosing to move against the boundary of the map (e.g. $T(s_0, \leftarrow)(s_0) = 0.9$). (b) Given the SS+LTL specification $\theta = ((!danger \mathbf{U} tool), \mathbf{SS}_{[0.75,1]} home)$, the corresponding LTL DRA $\mathcal{A} = (Q, q_0, \Sigma, \delta, F = \{(B_i, G_i)\}_i)$ is defined, where $Q = \{q_0, q_1, q_2\}$, $\Sigma = 2^{AP}$, $F = \{(\emptyset, \{q_1\})\}$, and the transition function is given by $\delta(q_0, \emptyset) = q_0, \delta(q_0, \{tool\}) = q_1, \delta(q_0, \{danger\}) = q_2, \delta(q_1, \cdot) = q_1, \delta(q_2, \cdot) = q_2$. The symbols !, & denote logical negation and conjunction. Note that the steady-state specification in θ is not used in defining $\mathcal{A}$. (c) Product LMC $(\mathcal{M} \times \mathcal{A})_\pi$ induced by the policy π given by the black arrows in (a) for the product LMDP $\mathcal{M} \times \mathcal{A}$, where red nodes represent states in the accepting BSCC of $(\mathcal{M} \times \mathcal{A})_\pi$. Note that this policy has non-zero probability of being trapped in the accepting BSCC. Furthermore, note that $\sum_{s \in L^{-1}(home)} \mathbf{Pr}^\infty_\pi(s) = \mathbf{Pr}^\infty_\pi(s_0) = 0.76$, thereby satisfying the steady-state operator $\mathbf{SS}_{[0.75,1]} home$. In this example, the product LMC is a multichain; however, note that the original LMC over states $s \in S$ as given by the dashed component (ignoring the $q \in Q$ in each $(s, q) \in S^\times$) is a unichain. Furthermore, the two BSCCs of the multichain product LMC are identical with respect to their transition matrices due to the one-to-one correspondence of paths in the original LMC and the product LMC.

5 INTEGER LINEAR PROGRAM CHARACTERIZATION

Let us first consider an agent whose goal is to find a stationary stochastic policy $\pi : S \times Q \to D(A)$ to maximize the expected reward in a product LMDP $\mathcal{M} \times \mathcal{A}$. If $\mathcal{M} \times \mathcal{A}$ is a unichain LMDP, the program in system (2) suffices to compute the optimal policy such that the solution yields the identity $x_{sqa} = \pi(a|s,q)\Pr^\infty(s,q) = \pi(a|s,q) \sum_a x_{sqa}$ for $s \in S$, $q \in Q$, and $a \in A$ from which a stochastic policy can then be derived, where x_{sqa} denotes the occupation measure of taking action a in state $(s,q) \in S^\times$ [30].

$$\max \sum_{(s,q) \in S^\times} \sum_{a \in A(s)} x_{sqa} \sum_{s' \in S} T(s,a)(s') R(s,a,s') \text{ subject to}$$

$$(i) \sum_{(s,q) \in S^\times} \sum_{a \in A(s)} x_{sqa} T^\times((s,q),a)(s',q') = \sum_{a \in A(s')} x_{s'q'a} \quad \forall (s',q') \in S^\times \tag{2}$$

$$(ii) \sum_{(s,q) \in S^\times} \sum_{a \in A(s)} x_{sqa} = 1$$

Now, consider the more general case where the given product LMDP $\mathcal{M} \times \mathcal{A}$ may be multichain. Two key problems arise. First, the policy π derived from the solution to (2) may not yield a unichain original LMC $\mathcal{M}_\pi$ (i.e., one with a single BSCC and possibly some transient states). Second, we note the challenges of deriving the correct steady-state distributions for an agent using linear programming in the multichain setting. In particular, in his seminal work [18], Kallenberg demonstrated that there is not a one-to-one correspondence between the steady-state distribution derived from linear programming solutions to expected-reward MDPs and the true steady-state distribution of the agent enacting the resulting policy when the Markov chain is multichain (i.e. contains multiple BSCCs, and possibly some transient states). On the other hand, unichains yield a one-to-one correspondence between the solution of the steady-state equations and the true steady-state behavior of the agent [19]. Furthermore, the solution to these equations is unique in said setting. We thus focus on deriving an optimal solution policy $\pi : S \times Q \to A$ in a (potentially) multichain product LMDP $\mathcal{M} \times \mathcal{A}$ such that the induced original LMC $\mathcal{M}_\pi$ is a unichain and satisfies a given SS+LTL formula. The interplay with the product LMC $(\mathcal{M} \times \mathcal{A})_\pi$ introduces some challenges in deriving such a policy. In particular, it may be the case that the product LMC $(\mathcal{M} \times \mathcal{A})_\pi$ induced by the solution policy π is multichain and its corresponding original LMC $\mathcal{M}_\pi$ is unichain. We present a novel solution which accounts for such settings by ensuring that all BSCCs in the product multichain $(\mathcal{M} \times \mathcal{A})_\pi$ share some state of the original LMC $\mathcal{M}_\pi$. This establishes that the original LMC $\mathcal{M}_\pi$ is a unichain. We further prove that the steady-state probabilities derived over such product LMCs yield a one-to-one correspondence with the true steady-state behavior of the agent in the original LMC.

First, let us consider the simpler case where the product LMC is a unichain. Note that the single BSCC may contain the same state $s \in S$ multiple times as $S^\times \ni (s,q), (s,q'), \ldots$ We will show that the partition $[s] = \{(s,q)\}_q$ naturally defined over the states of the product LMC to yield the states of the original LMC is such that $\Pr^\infty(s) = \sum_{(s,q) \in [s]} \Pr^\infty(s,q)$. That is, we can compute the steady-state probabilities over the product LMC and use these to derive

those in the original LMC over which the SS+LTL specification is defined. This is enabled by the lumpability of the product LMC, defined below.

Definition 5.1 (Lumpability [31], Def. 1). Given an irreducible Markov chain $\mathcal{M} = (S, T)$ and a partition $\bigcup_{k=1}^{K} S_k$ ($S_k \subset S, S_i \cap S_j = \emptyset$) of S, then $\bigcup_k S_k$ is called ordinarily lumpable if and only if $(\mathbf{e}_\alpha - \mathbf{e}_\beta)TV = \mathbf{0}$ for all $s_\alpha, s_\beta \in S_k, k \leq K$, where $\mathbf{e}$ is the standard basis vector and V is defined so that $v_{ik} = 1$ if $s_i \in S_k$ and $v_{ik} = 0$ otherwise. The vector $\mathbf{e}_k$ is the all-zeroes vector with a value of 1 only for the k^{th} entry.

A partition over an LMC naturally defines another LMC, known as the aggregated LMC, where each state of the latter corresponds to one of the partition sets of the former. As we will show in Corollary 1, the original LMC is the aggregated LMC resulting from a lumpable partition of the product LMC.

Definition 5.2 (Aggregated Markov Chain). Given a product LMC $\mathcal{M}^\times = (S^\times = S \times Q, \beta^\times, T^\times, R^\times, AP, L^\times)$ and a partition $\bigcup_{s \in S}[s]$ such that $[s] = \{(s, q) | (s, q) \in S^\times\}$, the aggregated LMC is given by $\mathcal{M}^* = (S^*, \beta^*, T^*, R^*, AP, L^*)$, where $S^* = \{s | [s] \in \bigcup_{s \in S}[s]\}$, $\beta^*(s) = \sum_{(s,q) \in [s]} \beta^\times(s, q)$, $T^*(s)(s') = T^\times(s, \cdot)(s', \cdot)$, $R^*(s, s') = R^\times((s, \cdot), (s', \cdot))$, and $L^*(s) = L^\times((s, \cdot))$.

Lemma 5.3. *Given an arbitrary BSCC $(\hat{S}, \hat{T})$ of a product LMC $\mathcal{M}^\times = (S^\times = S \times Q, T^\times)$, the partition $\bigcup_{s \in S}[s]$ given by equivalence classes $[s] = \{(s, q) | (s, q) \in \hat{S}\}$ is ordinarily lumpable. The proof is in Appendix A of the extended version [32].*

We adapt a theorem from [31] and modify it for our product LMC setting below.

Theorem 5.4 ([33], [31], Theorem 4). *Given an irreducible product LMC $\mathcal{M}^\times = (S^\times = S \times Q, T^\times)$ and an ordinarily lumpable partition $\bigcup_{s \in S}[s]$, where $[s] = \{(s, q) | (s, q) \in S^\times\}$, the steady-state distribution of the aggregated LMC $\mathcal{M} = (S, T)$ satisfies $Pr^\infty(s) = \sum_{(s,q) \in [s]} Pr^\infty(s, q)$ for every $s \in S$. Furthermore, the transition function of the aggregated LMC is given by $T(s)(s') = \mathbf{e}_i T([s])([s'])\mathbf{e}^T$, where i is an arbitrary index in the set $\{i | (s_i, \cdot) \in [s]\}$. The proof is in Appendix B of the extended version [32].*

Corollary 5.5. *Given an irreducible product LMC $\mathcal{M}^\times = (S^\times, T^\times)$, the original LMC $\mathcal{M} = (S, T)$ is the aggregated LMC resulting from the ordinarily lumpable partition $\bigcup_{s \in S}[s]$, where $[s] = \{(s, q) | (s, q) \in S^\times\}$. The proof is in Appendix B of the extended version [32].*

Lemma 5.3, Theorem 5.4, and Corollary 5.5 establish the one-to-one correspondence between the steady-state probability derived for an irreducible product LMC and the steady-state distribution for the original LMC. Note that this result also holds for unichains since the steady-state probability measure of transient states therein would be zero. Now, let us consider the case where the product LMC is a multichain. We establish sufficient conditions for yielding the same one-to-one correspondence of steady-state distributions. Furthermore, we establish necessary and sufficient conditions for the multichain product LMC $(\mathcal{M} \times \mathcal{A})_\pi$ to yield a unichain original LMC $\mathcal{M}_\pi$.

Lemma 5.6. *Let $\mathcal{M}^\times = (S^\times, T^\times)$ denote a multichain product LMC and let $(S^k)_k$, $S^k \subset S^\times$ denote its BSCCs. Then its corresponding original LMC $\mathcal{M} = (S, T)$ is a unichain iff some state $(s, \cdot) \in S^\times$ shows up in every BSCC S^k of $\mathcal{M}^\times$. That is, for some $s \in S$ and all k, there exists $q \in Q$ such that $(s, q) \in S^k$.*

Proof. This follows from the one-to-one correspondence between paths in $\mathcal{M}$ and paths in $\mathcal{M}^\times$. Furthermore, the single BSCC $S' \subseteq S$ of $\mathcal{M}$ is given by $S' = \{s | (s, q) \in \bigcup_k S^k\}$. □

Lemma 5.7. *Given a multichain product LMC $\mathcal{M}^\times = (S^\times, T^\times)$ with m identical BSCCs given by transition probability matrices $T_1 = T_2 = \cdots = T_m$, and an irreducible LMC $\mathcal{M}' = (S', T')$, where S' contains exactly the states in the first BSCC and $T' = T_1$ (w.l.o.g.), the steady-state probability of an arbitrary state $(s, q) \in S'$ is equivalent to the sum of steady-state probabilities of all states isomorphic to it in the BSCCs of $\mathcal{M}^\times$. The proof is in Appendix C of the extended version [32].*

To illustrate Theorem 5.4 and Lemmas 5.6 and 5.7, consider the multichain product LMC in Figure 3, where T_1 and T_2 denote the transition probability matrices for the two BSCCs. We will show that, because these two BSCCs are identical in terms of their transition matrices (rows may need to be reordered to reflect this), we have $\Pr^\infty(s) = \sum_{(s,q) \in [s]} \Pr^\infty(s, q)$ for states s in the original LMC shown in Figure 4. The solution to the steady-state equations for this product LMC yields $\Pr^\infty(s_0, q_0) = 0, \Pr^\infty(s_1, \cdot) = 1/6, \Pr^\infty(s_2, \cdot) = 1/12$. The order of states in T_2 differs from that of T_1 in order to reflect that BSCCs can be identical up to row ordering. Note that the equivalence classes can be defined in terms of the isomorphic sets as $[s_1] = \langle s_1 \rangle \bigcup \langle s_1' \rangle$ and $[s_2] = \langle s_2 \rangle \bigcup \langle s_2' \rangle$.

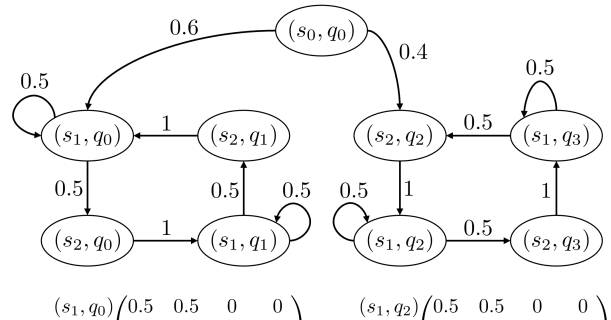

Figure 3: Product LMC with isomorphic sets given by $\langle s_1 \rangle = \{(s_1, q_0), (s_1, q_2)\}$, $\langle s_1' \rangle = \{(s_1, q_1), (s_1, q_3)\}$, $\langle s_2 \rangle = \{(s_2, q_0), (s_2, q_3)\}$, $\langle s_2' \rangle = \{(s_2, q_1), (s_2, q_2)\}$.

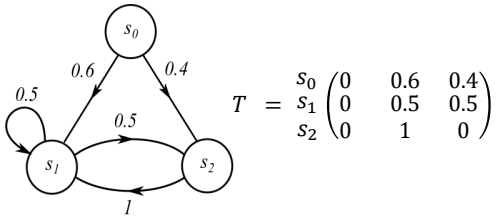

Figure 4: The unichain original LMC.

Now, consider the original LMC shown in Figure 4 corresponding to this product LMC. Solving the steady-state equations (1) for the original LMC yields $\Pr^{\infty}(s_0) = 0, \Pr^{\infty}(s_1) = 2/3, \Pr^{\infty}(s_2) = 1/3$. Note that $\Pr^{\infty}(s_1) = \sum_{(s,q)\in[s_1]} \Pr^{\infty}(s,q)$ and $\Pr^{\infty}(s_2) = \sum_{(s,q)\in[s_2]} \Pr^{\infty}(s,q)$ in accordance with Theorem 5.4 by leveraging the fact that $\Pr^{\infty}(s,q) = \sum_{(s',q')\in\langle s\rangle} \Pr^{\infty}(s',q')$ per Lemma 5.7.

Theorem 5.4 and Lemma 5.7 establish necessary and sufficient conditions for a multichain product LMC to yield a unichain original LMC in the original LMDP such that there is a one-to-one correspondence between the sum of steady-state probabilities $\sum_q \Pr^{\infty}(s,q)$ in the former and the steady-state distribution $\Pr^{\infty}(s)$ in the latter. Indeed, note that

$$\Pr^{\infty}(s) = \sum_{(s,q)\in[s]} \Pr^{\infty}(s,q) = \sum_{s\in S} \sum_{(s',q')\in\langle s\rangle} \Pr^{\infty}(s',q')$$

holds when the product LMC is unichain or multichain given that the original LMC is unichain. This is the case when all BSCCs in the product LMC are identical as mentioned in Lemma 5.7, which is the case when all BSCCs in the product LMC share some state in S per Lemma 5.6.

We can now add constraints to program (2) to ensure that the solution policy π is deterministic and yields a unichain original LMC $\mathcal{M}_\pi$ in the original LMDP $\mathcal{M}$ even though the product LMC $(\mathcal{M}\times\mathcal{A})_\pi$ induced in the product LMDP $\mathcal{M}\times\mathcal{A}$ may be multichain. We begin with constraints to enforce a deterministic policy. Constraint (iii) ensures that a positive occupation measure implies that the action corresponding to it is selected as part of the solution policy and (iv) enforces a valid probability distribution, where $\pi(a|s,q) \in \{0,1\}$.

$$(iii)\ x_{sqa} \le \pi(a|s,q) \qquad \forall (s,q)\in S^{\times}, a\in A$$

$$(iv)\ \sum_{a\in A(s)} \pi(a|s,q) = 1 \qquad \forall (s,q)\in S^{\times}$$

Lemma 5.8. *Let (x,π) denote a feasible solution to constraints (i) through (iv) and assume that the product LMC $(\mathcal{M}\times\mathcal{A})_\pi$ induced by π is such that all BSCCs share some state $s\in S$. Then $x_{sqa} = \pi(a|s,q)Pr^{\infty}_{\pi}(s,q) = \pi(a|s,q)\sum_a x_{sqa}$ for all recurrent states $(s,q)\in S^{\times}$. The proof is in Appendix D of the extended version [32].*

We can now add additional constraints that utilize the policy π in constraints (iii) and (iv) in order to establish that some accepting state within an AMEC is reached by this policy and visited infinitely often. This would, in turn, satisfy the LTL specification ϕ_{LTL} of the given SS+LTL specification θ. For simplicity, we assume an initial state s_0 in the underlying LMDP $\mathcal{M}$ (i.e. $\beta(s_0) = 1$). In order to ensure that there is a path from the initial state $(s_0, \delta(q_0, L(s_0))) \in S^{\times}$ in the product LMDP $\mathcal{M}^{\times}$ to some recurrent component in the union of AMECs which contains nodes in $\bigcup_i G_i$ (i.e. nodes that are part of the DRA acceptance pairs), we will use flow transfer constraints. This notion of flow reflects the probability of transitioning between states given a policy. Constraint (v) sets the flow capacities, where $f_{sqs'q'}$ denotes flow from $(s,q)\in S^{\times}$ to $(s',q')\in S^{\times}$. Constraint (vi) ensures that, for every state (except the starting state), if there is incoming flow, then it is strictly greater than the outgoing flow. This is handled by the product of some small constant ϵ and an indicator variable $\mathcal{I}_{sq}\in\{0,1\}$ denoting whether flow is being transferred from state (s,q) to some other state. If there is no incoming flow, then there is no outgoing flow and $\mathcal{I}_{sq}$ must necessarily be zero. Constraint (vii) ensures that, if there is incoming flow into a state $(s,q)\in S^{\times}$, then $\mathcal{I}_{sq} = 1$. Constraint $(viii)$ ensures that, whenever there is incoming flow, there must also be some arbitrary amount of outgoing flow. The choice of denominator 2 here is arbitrary.

$$(v)\ f_{sqs'q'} \le \sum_{a\in A(s)} T((s,q),a)(s',q')\pi(a|s,q) \qquad \forall((s,q),(s',q'))\in T^G$$

$$(vi)\ \sum_{((s',q'),(s,q))\in T^G} f_{s'q'sq} \ge \sum_{((s,q),(s',q'))\in T^G} f_{sqs'q'} + \epsilon\mathcal{I}_{sq} \qquad \forall (s,q)\in S^{\times}\setminus\{(s_0,\delta(q_0,L(s_0)))\}$$

$$(vii)\ \sum_{((s',q'),(s,q))\in T^G} f_{s'q'sq} \le \mathcal{I}_{sq} \qquad \forall (s,q)\in S^{\times}$$

$$(viii)\ \sum_{((s,q),(s',q'))\in T^G} f_{sqs'q'} \ge \sum_{((s',q'),(s,q))\in T^G} f_{s'q'sq}/2 \qquad \forall (s,q)\in S^{\times}$$

Constraint (ix) ensures that the steady-state probability of states with no incoming flow (as determined by $\mathcal{I}_{sq}$ in constraint (vi)) is 0. This makes it so that unreachable BSCCs in the product LMC $(\mathcal{M}\times\mathcal{A})_\pi$ do not contribute to the steady-state distribution. Constraint (x) encodes the steady-state specifications given by operators **SS** in θ and constraint (xi) ensures that some state in the acceptance sets $\bigcup_i G_i$ of $\mathcal{A}$ is visited infinitely often to satisfy the LTL specification.

$$(ix)\ \sum_{a\in A(s)} x_{sqa} \le \mathcal{I}_{sq} \qquad \forall (s,q)\in S^{\times}$$

$$(x)\ l \le \sum_{s\in L^{-1}(\psi)} \sum_{q\in Q} \sum_{a\in A(s)} x_{sqa} \le u \qquad \forall \mathbf{SS}_{[l,u]}\psi \in \theta$$

$$(xi)\ \sum_{s\in S} \sum_{q\in\bigcup_i G_i} \sum_{a\in A(s)} x_{sqa} > 0$$

Recall that constraints (i) and (ii) yield the correct steady-state distribution if there is a single BSCC (per the unichain condition) or if all BSCCs in the product LMC $(\mathcal{M}\times\mathcal{A})_\pi$ are identical and the induced original LMC $\mathcal{M}_\pi$ is a unichain (per Lemma 5.7). This is the case when all BSCCs of $(\mathcal{M}\times\mathcal{A})_\pi$ share some state in $s\in S$ per Lemma 5.6. We must therefore ensure that, in the product LMC, some state in S is shared (This is trivially true if there is only one BSCC). The one-to-one correspondence of paths between the original LMC and the product LMC will then guarantee that the former is unichain even if the latter is multichain. To accomplish this, we define three indicator variables $\mathcal{I}_s, \mathcal{I}^k, \mathcal{I}^k_s \in \{0,1\}$ whose value is 1 iff $(s,\cdot)$ shows up in some BSCC of the product LMC, the k^{th} AMEC of the product LMDP has some state with positive steady-state probability (meaning that the AMEC, or a subset of it, will show up as a BSCC in the product LMC), and $(s,\cdot)$ has positive steady-state probability in the k^{th} AMEC, respectively.

Let *AMEC* denote the set of all AMECs in the product LMDP $\mathcal{M}\times\mathcal{A}$ and let $AMEC_k$ denote the k^{th} AMEC. Constraint (xii) ensures that $\mathcal{I}^k$ is 1 if some state in the k^{th} AMEC has positive steady-state probability. Constraints $(xiii)$ and (xiv) ensure that, for a given state $s\in S$ and the k^{th} AMEC, some indicator variable $\mathcal{I}_{sq}$ is 1 for (s,q) in the k^{th} AMEC if and only if $\mathcal{I}^k_s = 1$. Constraint (xv) ensures that, if $\mathcal{I}_s$ is 1, then $(s,\cdot)$ shows up in every BSCC

of the product LMC $(\mathcal{M} \times \mathcal{A})_\pi$, thereby enforcing that all BSCCs in $(\mathcal{M} \times \mathcal{A})_\pi$ are identical. Per Lemma 5.6, this ensures that the original LMC $\mathcal{M}_\pi$ is a unichain. Note that the sum $\sum_k (\mathcal{I}_s^k - \mathcal{I}^k)$ is always non-positive and dividing by the number of AMECs bounds this result to be within $[-1, 0]$. Finally, constraint (xvi) ensures that some such shared state exists across all BSCCs of the product LMC.

$$(xii) \quad \sum_{(s,q)\in AMEC_k} \sum_a x_{sqa} \leq \mathcal{I}^k \qquad \forall 1 \leq k \leq |AMEC|$$

$$(xiii) \quad \mathcal{I}_s^k \leq \sum_{(s,q)\in AMEC_k} \mathcal{I}_{sq} \qquad \forall s \in S, 1 \leq k \leq |AMEC|$$

$$(xiv) \quad \sum_{(s,q)\in AMEC_k} \frac{\mathcal{I}_{sq}}{|Q|} \leq \mathcal{I}_s^k \qquad \forall s \in S, 1 \leq k \leq |AMEC|$$

$$(xv) \quad \mathcal{I}_s - 1 \leq \frac{\sum_k \left(\mathcal{I}_s^k - \mathcal{I}^k\right)}{|AMEC|} \qquad \forall s \in S$$

$$(xvi) \quad \sum_s \mathcal{I}_s \geq 1$$

The program is summarized below.

$$\begin{aligned} &\max \sum_{(s,q)\in S^\times} \sum_{a\in A(s)} x_{sqa} \sum_{s'\in S} T(s,a)(s')R(s,a,s') \text{ s.t. } (i)-(xvi) \\ &x_{sqa}, f_{sqs'q'} \in [0,1], \qquad \forall ((s,q),a,(s',q')) \in S^\times \times A \times S^\times \\ &\pi(a|s,q), \mathcal{I}_{sq}, \mathcal{I}_s, \mathcal{I}^k, \mathcal{I}_s^k \in \{0,1\}, \\ &\qquad \forall ((s,q),a) \in S^\times \times A, 1 \leq k \leq |AMEC| \end{aligned} \tag{3}$$

Theorem 5.9. *Given an LMDP $\mathcal{M} = (S, \beta, A, T, R, AP, L)$ and an SS+LTL objective $\theta = (\phi_{LTL}, (\mathbf{SS}_{[l_i,u_i]}\psi_i)_i)$, let $(x, f, \pi, \mathcal{I})$ denote an assignment to the variables in program (3). Then $(x, f, \pi, \mathcal{I})$ is a feasible solution if and only if $\mathcal{M}_\pi = (S_\pi, \beta, T_\pi, AP, L)$ satisfies θ and is a unichain. The proof is in Appendix E of the extended version [32].*

6 EXPERIMENTAL RESULTS

Simulations of program (3) were performed using CPLEX version 12.8 [34] on a machine with a 3.6 GHz Intel Core i7-6850K processor and 128 GB of RAM. We generated random 4×4, 8×8, and 16×16 gridworld environments given by the LMDP $\mathcal{M} = (S, \beta, T, R, AP, L)$ subject to various SS+LTL specifications θ and with the top-left corner of the grid as the inital state. There are four actions corresponding to the four cardinal directions and a deterministic transition function $T(s,a)(s') \in \{0,1\}$ defined in the obvious manner. Each state-action pair observes a uniformly distributed random reward in $\{0,1\}$. See the figure in Appendix F for an example. It is worth noting that the solutions illustrated in Figure 1 and Figure 2 were also generated using program (3).

In the following experiments, the set of atomic propositions is given by $AP = \{a, b, c, d\}$, with each atomic proposition allocated to one-fourth of the states chosen at random. See Table 1 for results. These results demonstrate that the proposed program (3) can scale to state spaces of moderate size on the order of a few minutes.

$$\theta_1 = ((\mathbf{G}\neg b) \wedge (\mathbf{GF}a), \mathbf{SS}_{[0.01,0.5]}d)$$
$$\theta_2 = ((\mathbf{GF}a) \vee (\mathbf{FG}b), \mathbf{SS}_{[0.01,0.5]}d)$$
$$\theta_3 = ((\mathbf{FG}a)\mathbf{U}(b \vee \mathbf{X}(b \vee \mathbf{X}(b \vee \mathbf{X}b))), \mathbf{SS}_{[0.01,0.5]}d)$$
$$\theta_4 = ((\mathbf{F}a)\mathbf{U}b, \mathbf{SS}_{[0.01,0.5]}d)$$
$$\theta_5 = ((\mathbf{F}a) \wedge \mathbf{F}(a\mathbf{U}b), \mathbf{SS}_{[0.01,0.5]}d)$$
$$\theta_6 = (\mathbf{F}(a \wedge \mathbf{X}(a \wedge \mathbf{X}a)), \mathbf{SS}_{[0.01,0.5]}d)$$
$$\theta_7 = ((\mathbf{F}a \wedge \mathbf{F}b) \wedge ((\mathbf{F}a \wedge \mathbf{F}b)\mathbf{U}(c \vee \mathbf{X}a)), \mathbf{SS}_{[0.01,0.5]}d)$$
$$\theta_8 = (\mathbf{F}a \wedge \mathbf{F}b \wedge \mathbf{F}c, \mathbf{SS}_{[0.01,0.5]}d)$$

θ	4×4	8×8	16×16
θ_1	0.42 (0.43)	13.09 (83.86)	35.21 (70.09)
θ_2	0.28 (0.72)	0.15 (0.06)	1.42 (0.74)
θ_3	1.13 (3.27)	0.72 (0.40)	52.74 (59.42)
θ_4	0.58 (2.04)	1.19 (0.53)	78.29 (53.94)
θ_5	0.64 (1.93)	1.56 (0.70)	125.42 (93.79)
θ_6	0.25 (0.62)	1.03 (0.43)	155.60 (130.84)
θ_7	1.50 (5.08)	4.95 (2.77)	195.87 (145.07)
θ_8	2.28 (6.41)	9.50 (6.37)	338.88 (205.29)

Table 1: Average runtimes and standard deviations for 100 random instances of program (3) using CPLEX version 12.8 for the listed SS+LTL specifications $\theta_1, \ldots, \theta_8$ and for grids of sizes 4×4, 8×8, and 16×16.

7 CONCLUSION

In this paper, we proposed and solved the deterministic controller synthesis problem for labeled Markov Decision Processes (LMDPs) subject to specifications on both the linear-time and visitation frequency behaviors of an agent. The proposed approach uses a novel integer programming formulation to find a policy that induces a unichain labeled Markov chain (LMC). The program reasons about the product LMDP computed from the original LMDP and the deterministic Rabin automaton (DRA) representation of the linear-time property. Though the product LMC induced by the solution policy may be a multichain, we established necessary and sufficient conditions for the one-to-one correspondence between the visitation frequencies derived from the product LMC and the true steady-state behavior of the agent captured by the unichain original LMC. The foregoing is a step toward infinite-horizon formal synthesis of control policies in general decision processes. For future work, we will explore how similar correct-by-construction policies can be computed such that guarantees of behavior hold for general multichain LMCs induced by said policies in the original LMDP.

ACKNOWLEDGMENTS

This research was supported in part by the Air Force Research Laboratory through the Information Directorate's Information Institute® Contract Number FA8750-20-3-1003 and FA8750-20-3-1004, the Air Force Office of Scientific Research through Award 20RICOR012, and the National Science Foundation through CAREER Award CCF-1552497 and Award CCF-2106339.

REFERENCES

[1] W. Thomas. *Handbook of Theoretical Computer Science*, chapter Automata on Infinite Objects, pages 133–191. The MIT Press/Elsevier, 1990.

[2] D. Perrin and J.-É. Pin. *Infinite Words: Automata, Semigroups, Logic and Games*. Elsevier, 2004.

[3] L. de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford University, 1998.

[4] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT press, 2008.

[5] Sundararaman Akshay, Nathalie Bertrand, Serge Haddad, and Loic Helouet. The steady-state control problem for markov decision processes. In *International Conference on Quantitative Evaluation of Systems*, pages 290–304. Springer, 2013.

[6] Alvaro Velasquez. Steady-state policy synthesis for verifiable control. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5653–5661. International Joint Conferences on Artificial Intelligence Organization, 2019.

[7] George Atia, Andre Beckus, Ismail Alkhouri, and Alvaro Velasquez. Steady-state policy synthesis in multichain markov decision processes. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4069–4075. International Joint Conferences on Artificial Intelligence Organization, 2020.

[8] George K Atia, Andre Beckus, Ismail Alkhouri, and Alvaro Velasquez. Steady-state planning in expected reward multichain mdps. *Journal of Artificial Intelligence Research*, 72:1029–1082, 2021.

[9] Jan Křetínskỳ. Ltl-constrained steady-state policy synthesis. *arXiv preprint arXiv:2105.14894*, 2021.

[10] Vasanth Sarathy, Daniel Kasenberg, Shivam Goel, Jivko Sinapov, and Matthias Scheutz. Spotter: Extending symbolic planning operators through targeted reinforcement learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1118–1126, 2021.

[11] Xu Chu Dennis Ding, Stephen L Smith, Calin Belta, and Daniela Rus. Ltl control in uncertain environments with probabilistic satisfaction guarantees. *IFAC Proceedings Volumes*, 44(1):3515–3520, 2011.

[12] Bruno Lacerda, David Parker, and Nick Hawes. Optimal and dynamic planning for markov decision processes with co-safe LTL specifications. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1511–1516. IEEE, 2014.

[13] James R Norris and James Robert Norris. *Markov chains*. Number 2. Cambridge university press, 1998.

[14] Kousha Etessami, Marta Kwiatkowska, Moshe Y Vardi, and Mihalis Yannakakis. Multi-objective model checking of markov decision processes. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 50–65. Springer, 2007.

[15] Mihalis Yannakakis, Moshe Y Vardi, Marta Kwiatkowska, and Kousha Etessami. Multi-objective model checking of markov decision processes. *Logical Methods in Computer Science*, 4, 2008.

[16] Vojtěch Forejt, Marta Kwiatkowska, Gethin Norman, and David Parker. Automated verification techniques for probabilistic systems. In *International School on Formal Methods for the Design of Computer, Communication and Software Systems*, pages 53–113. Springer, 2011.

[17] K. Chatterjee and M. Henzinger. Faster and dynamic algorithms for maximal end-component decomposition and related graph problems in probabilistic verification. In *Symposium on Discrete Algorithms (SODA)*, pages 1318–1336, January 2011.

[18] L. C. M. Kallenberg. *Linear programming and finite Markovian control problems*. Mathematisch Centrum, Amsterdam, 1983.

[19] Martin L. Puterman. *Markov Decision Processes*. Wiley, 1994.

[20] Eitan Altman. Constrained Markov decision processes with total cost criteria: Lagrangian approach and dual linear program. *Mathematical Methods of Operations Research*, 48(3):387–417, 1998.

[21] E. A. Feinberg. Adaptive computation of optimal nonrandomized policies in constrained average-reward MDPs. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 96–100, March 2009.

[22] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[23] Patricia Bouyer, Nicolas Markey, and Raj Mohan Matteplackel. Averaging in ltl. In *International Conference on Concurrency Theory*, pages 266–280. Springer, 2014.

[24] Shaull Almagor, Udi Boker, and Orna Kupferman. Discounting in ltl. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 424–439. Springer, 2014.

[25] Udi Boker, Krishnendu Chatterjee, Thomas A Henzinger, and Orna Kupferman. Temporal specifications with accumulative values. *ACM Transactions on Computational Logic (TOCL)*, 15(4):1–25, 2014.

[26] Benedikt Bollig, Normann Decker, and Martin Leucker. Frequency linear-time temporal logic. In *2012 Sixth International Symposium on Theoretical Aspects of Software Engineering*, pages 85–92. IEEE, 2012.

[27] Mária Svoreňová, Ivana Černá, and Calin Belta. Optimal control of mdps with temporal logic constraints. In *52nd IEEE Conference on Decision and Control*, pages 3938–3943. IEEE, 2013.

[28] Eitan Altman, Said Boularouk, and Didier Josselin. Constrained Markov decision processes with total expected cost criteria. In *Proceedings of the 12th EAI International Conference on Performance Evaluation Methodologies and Tools*, pages 191–192. ACM, 2019.

[29] Dmitry Krass and O. J. Vrieze. Achieving target state-action frequencies in multichain average-reward Markov decision processes. *Mathematics of Operations Research*, 27(3):545–566, 2002.

[30] Felipe W Trevizan, Sylvie Thiébaux, and Patrik Haslum. Occupation measure heuristics for probabilistic planning. In *ICAPS*, pages 306–315, 2017.

[31] Peter Buchholz. Exact and ordinary lumpability in finite markov chains. *Journal of applied probability*, pages 59–75, 1994.

[32] Alvaro Velasquez, Ashutosh Trivedi, Ismail Alkhouri, Andre Beckus, and George Atia. Controller synthesis for omega-regular and steady-state specifications. *arXiv preprint arXiv:2106.02951*, 2021.

[33] Ushio Sumita and Maria Rieders. Lumpability and time reversibility in the aggregation-disaggregation method for large markov chains. *Stochastic Models*, 5(1):63–81, 1989.

[34] ILOG, Inc. ILOG CPLEX: High-performance software for mathematical programming and optimization, 2006. See http://www.ilog.com/products/cplex/.