Journal of Artificial Intelligence Research 72 (2021) 1029-1082 Submitted 01/2021; published 11/2021

Steady-State Planning in Expected Reward Multichain MDPs

George K. Atia GEORGE.ATIA@UCF.EDU
Department of Electrical and Computer Engineering
Department of Computer Science
University of Central Florida, FL 32816, USA

Andre Beckus ANDRE.BECKUS@US.AF.MIL
Air Force Research Laboratory, NY 13441, USA

Ismail Alkhouri IALKHOURI@KNIGHTS.UCF.EDU
Department of Electrical and Computer Engineering
University of Central Florida, FL 32816, USA

Alvaro Velasquez ALVARO.VELASQUEZ.1@US.AF.MIL
Air Force Research Laboratory, NY 13441, USA

Abstract

The planning domain has experienced increased interest in the formal synthesis of decision-making policies. This formal synthesis typically entails finding a policy which satisfies formal specifications in the form of some well-defined logic. While many such logics have been proposed with varying degrees of expressiveness and complexity in their capacity to capture desirable agent behavior, their value is limited when deriving decision-making policies which satisfy certain types of asymptotic behavior in general system models. In particular, we are interested in specifying constraints on the steady-state behavior of an agent, which captures the proportion of time an agent spends in each state as it interacts for an indefinite period of time with its environment. This is sometimes called the average or expected behavior of the agent and the associated planning problem is faced with significant challenges unless strong restrictions are imposed on the underlying model in terms of the connectivity of its graph structure. In this paper, we explore this steady-state planning problem that consists of deriving a decision-making policy for an agent such that constraints on its steady-state behavior are satisfied. A linear programming solution for the general case of multichain Markov Decision Processes (MDPs) is proposed and we prove that optimal solutions to the proposed programs yield stationary policies with rigorous guarantees of behavior.

1. Introduction

The proliferation and mass adoption of automated solutions in recent years has led to an increased concern in the verification, validation, and trust of the prescribed agent behavior (Schwarting et al., 2018). This motivates the need for traditional techniques which can yield guarantees of behavior. The study of such techniques has largely been the focus of areas such as formal synthesis and planning, where it is common to derive decision-making policies for agents acting in an environment such that some given formal specification is satisfied. The majority of prior art in this area entails the coupling of some formal logic with the model of agent-environment dynamics in order to find optimal policies which satisfy specifications expressed in said logic. Examples include planning with Linear Temporal Logic (LTL) (Guo & Zavlanos, 2018), probabilistic LTL (PLTL) (Kwiatkowska & Parker, 2013), LTL over finite traces (LTL$_f$) (Camacho & McIlraith, 2019), Linear Dynamic Logic (LDL$_f$) (Brafman & De Giacomo, 2019), Computation Tree Logic (CTL) (Pistore et al., 2014), probabilistic CTL (PCTL) (Song et al., 2015), Signal Temporal Logic (STL) (Lindemann & Dimarogonas, 2017), Chance-Constrained Temporal Logic (C2TL) (Jha et al., 2018), Continuous Stochastic Logic (CSL) (Ayala et al., 2014), μ-Calculus (De Giacomo et al., 2010), Metric Temporal Logic (MTL) (Zhou et al., 2016), and logic fragments, such as the Rank-1 Generalized Reactivity (GR[1]) formulas of LTL (Wongpiromsarn et al., 2011). Formal multi-agent planning has also been explored using Dynamic Epistemic Logic (Engesser et al., 2017) and Alternating-time Temporal Logic (ATL) (Jamroga, 2004).

The use of the foregoing logics has facilitated the growth of solutions to the aforementioned planning problems and are a good conduit for verifying, explaining, and yielding provably correct agent behavior and, consequently, establishing a measure of trust. However, these logics are either insufficient to reason about the asymptotic behavior that is captured by the steady-state distribution of the agent as it follows some decision-making policy, or existing solutions to the corresponding planning problems make strong assumptions on the underlying model of the system. Solutions to these challenges have gained traction in recent years. Indeed, there has been increased interest in what we refer to as the steady-state planning problem of computing decision-making policies that satisfy constraints on the resulting steady-state behavior. In particular, progress has been made in easing the restrictions required on the agent-environment dynamics model, usually expressed in the form of a Markov Decision Process (MDP), in order to derive a solution policy. In this paper, we advance the state-of-the-art in steady-state planning by establishing the first solution to steady-state planning in multichain MDPs such that the resulting stationary policy satisfies constraints imposed on the steady-state distribution of the agent. Our approach also dissolves assumptions of ergodicity or recurrence of the underlying MDP which are often made in the literature when reasoning about steady-state distributions.

Steady-state planning has applications in several areas, such as deriving maintenance plans for various systems, including aircraft maintenance, where the asymptotic failure rate of components must be kept below some small threshold (Boussemart & Limnios, 2004; Boussemart, Limnios, & Fillion, 2002). Optimal routing problems for communication networks have also been proposed in which data throughput must be maximized subject to constraints on average delay and packet drop metrics (Lazar, 1983). This includes constraints on steady-state network behavior, which include steady-state network frequency and steady-state phase or timing errors (Skwirzynski, 1981). There is also the potential of leveraging solutions in the steady-state planning problem space to the design of intelligent space satellites. Indeed, this is an area where the steady-state distribution of debris following some orbit can be computed to reason about the probability of a satellite colliding with said debris. Such information has been used to determine human-driven control policies for tasks such as debris mitigation or debris removal (Tian, 2019), sometimes via remote-controlled robots (Baiocchi, 2010) that are amenable to automated approaches.

The steady-state planning problem has been studied under various names, including steady-state control (Akshay et al., 2013), average- or expected-reward constrained MDPs (Altman, 1999), and steady-state policy synthesis (Velasquez, 2019). As pointed out by

Altman, Boularouk, and Josselin (2019), solutions to this problem often require strong assumptions on the ergodicity of the underlying MDP. These assumptions facilitate the search for efficient algorithms by leveraging the one-to-one correspondence between the optimality of solutions to various mathematical programs and the optimality of policies derived thereof. This has been studied at length in the works of Derman (1970), Kallenberg (1983), Puterman (1994), and Altman (1999), who have derived mathematical programs for discounted, total, and expected reward formulations of constrained MDPs. In particular, the work of Kallenberg laid the foundation for Markovian control within the context of multichain constrained MDPs. However, it was noted that deriving optimal policies for the expected-reward formulation was intractable by their approach and there was no guarantee of agent behavior in terms of satisfying steady-state constraints.

Summary of contributions. We make four main contributions. First, we introduce the Steady-State Policy Synthesis (SSPS) problem of finding a policy from a predefined subset of stationary policies in a multichain MDP that maximizes an expected reward signal while enforcing asymptotic behavior that is correct-by-construction (Nilsson et al., 2015) – in the sense that our policies yield provably correct behavior that satisfies the imposed specifications on the steady-state distribution of the Markov chain induced by said policies. Our framework generalizes the steady-state planning problems studied by Akshay et al. (2013) and Velasquez (2019), as we do not impose any restrictions on the underlying MDP. In particular, we dispense with the strong assumption made by Akshay et al. (2013) about the ergodicity of the MDP, according to which every deterministic policy necessarily induces an ergodic Markov chain (i.e., one that is recurrent and aperiodic). In sharp contrast to the work of Velasquez (2019), we do not restrict our search to stochastic policies that induce an irreducible Markov chain (i.e., one in which all states form one communicating class). In general, such a chain may not even exist – normally, many states in a given MDP are inevitably transient. Our search space consists of subsets of the stationary policies that we term edge- or class-preserving, which, apart from a transient phase, restrict the long-term play in the terminal components of the given MDP. We introduce two distinct notions for class preservation that yield policies with different characteristics. These notions will be made precise in Section 4.

As our second contribution, we develop a scalable approach to synthesize policies that provably meet said asymptotic specifications through novel linear programming formulations. While a tractable solution to the SSPS problem has heretofore remained elusive and existing solutions require an enormous amount of calculations with no provable guarantees (Kallenberg, 1983), two key ideas underlie our ability to tackle the associated combinatorial difficulties. The first idea is the aforementioned restriction of the domain to edge- or class-preserving policies, which can be provably obtained from solutions to simple linear programs (LPs). The second idea is to encode constraints on the limiting distributions of the corresponding Markov chains in formulated LPs, whose solutions yield optimal policies maximizing the expected average reward while meeting desired asymptotic specifications on the limit points of the expected state-action frequencies. These LPs are crafted to capture designated state classifications, absorption probabilities in closed communicating components, and recurrence constraints within such components, along with the steady-state specifications.

Our third contribution lies in deriving key theoretical results establishing provable performance and behavior guarantees for the derived policies. Contracting or transient MDP models that use the expected total reward as the optimality criterion are commonplace in constrained MDPs since optimal stationary policies with regard to this criterion can always be found via mathematical programming in view of a well-established one-to-one correspondence between stationary policies and feasible solutions to such programs (Altman, 1998; Feinberg, 2000; Wu & Durfee, 2010; Petrik & Zilberstein, 2009). The notoriously more difficult and equally important expected average reward criterion is much less understood considering that such correspondence ceases to exist for general multichain MDPs. In this paper, we tap into this long-standing dilemma and establish such one-to-one correspondence for classes of stationary policies that are edge- or class-preserving. Theorems 1, 2, 3 and 4 establish the correctness of linear programs yielding optimal policies from said classes. The proof of these theorems rest on few intermediate results. In particular, Lemma 3 characterizes the Markov chains induced by the policies of interest, while Lemma 7 establishes the feasibility of the steady-state distributions induced by these policies. Lemma 6 gives a sufficient condition for the existence of a one-to-one correspondence between feasible solutions to the linear programs and the stationary policies derived from these solutions. Theorem 5 establishes an existence condition of policies found on a more relaxed notion of class preservation, which inspires a constructive approach in Algorithm 1 to compute such policies. Theorem 8 gives a generic sufficient condition for the existence of an optimal stationary policy meeting the desired specifications beyond class-preserving ones.

As our fourth contribution, we introduce an alternative type of specifications applicable in transient states. By augmenting our LPs with appropriate constraints, the synthesized policies provably meet specifications on the expected number of visitations to transient states simultaneously with the foregoing steady-state specifications on the asymptotic frequency with which recurrent states are visited (Proposition 1).

We verify the theoretical findings of our work using a comprehensive set of numerical experiments performed in various environments. The results demonstrate the correctness of the proposed LPs in yielding policies with provably correct behavior and the scalability of the proposed solutions to large problem sizes.

This article brings in and substantially extends the scope of our recent work (Atia et al., 2020), which considered policy synthesis over edge-preserving policies. Such policies constitute only a small subset of the policies considered herein. A particularly appealing characteristic of the newly introduced policies is their greater ability to avert MDP transitions of low return without violating the asymptotic constraints. In turn, they yield larger expected rewards relative to their edge-preserving counterparts – in some cases, we show that this gain can be substantial. Further, we derive general characterizations of optimality over a larger class of policies obtained in terms of the MDP reward signal. In addition, this article advances the aforementioned form of transient specifications that a policy can provably meet together with the steady-state ones. We provide a complete presentation of the steady-state planning problem through linear programming formulations over different families of policies, mathematical analyses establishing correctness of such formulations with optimality guarantees, and a comprehensive set of numerical experiments in diverse environments to support the theoretical findings.

To the best of our knowledge, this work is the first to allow synthesis of stationary policies with provably correct steady-state behavior in general multichain MDPs.

Organization: The paper is organized as follows. Notation and preliminaries are covered in Section 2. Related work in steady-state planning is summarized in Section 3. The SSPS problem is formalized in Section 4. We describe our linear programming approach and present the results of our theoretical analysis in Section 5. Transient specifications and extensions to a larger class of policies are presented in Section 6. Numerical experiments are presented in Section 7 to validate our approach and demonstrate its scalability to large problems. Concluding remarks are presented in Section 8. In Appendix A, we present statements and proof of technical lemmas. The proof of the main results are deferred to Appendix B.

2. Preliminaries and Notation

We introduce some notation and preliminary definitions used throughout the paper. For a matrix A, a_{ij} and $A(i,j)$ are used interchangeably to denote the element in its i^{th} row and j^{th} column. The vectors e and e_s denote the vectors (of appropriate dimension) of all ones, and all zeros except for the s^{th} entry, respectively. Given a vector x and index set V, the vector x_V is the vector with entries $x_v, v \in V$, where x_v is the entry corresponding to index v. By $|S|$, we denote the cardinality of a set S. For an integer $n > 0$, the set $[n] := \{1, \ldots, n\}$, and $A \setminus B$ denotes the set difference of sets A and B. The symbols $\exists$ and $\exists!$ mean "there exists" and "there exists a unique", respectively, and $^\top$ is the transpose operator.

Definition 1 (Markov chain)**.** *A Markov chain is a stochastic model given by a tuple $\mathcal{M} = (S, T, \beta)$, where S is the state space, T the transition function $T : S \times S \to [0,1]$ with $T(s'|s)$ denoting the probability of transitioning from state s to state s', and $\beta : S \to [0,1]$ the initial state distribution. With slight abuse of notation, the transition function can also be thought of as a matrix $T \in [0,1]^{|S|\times|S|}$, where $T(s,s') = T(s'|s)$. The use of T will be clear from the context.*

Classification of states (Norris, 1997; Privault, 2018): Given a finite Markov chain $\mathcal{M} = (S, T, \beta)$, we say state s' is accessible from state s if $(T^t)(s,s') > 0$, for some $t > 0$, where T^t is the $t-$step transition matrix, i.e., if there is a positive probability of transitioning to state s' starting from state s in some number of steps. Two states are said to communicate if they are both accessible from each other. Communication is an equivalence relation which partitions the Markov chain $\mathcal{M}$ into communicating classes such that only members of the same class communicate with each other. A class is closed if the probability of escaping the class is zero. A state $s \in S$ is said to be *transient* if, starting from s, there is a non-zero probability of never returning to s. A set of transient states is termed a transient set. Non-transient states are called recurrent, that is, state s is recurrent if, starting from s, the probability of returning to state s after some number of steps is one. A Markov chain for which there is only one communicating class consisting of the entire state space is called irreducible, whereas a Markov chain that has a single closed communicating class and (possibly) some transient states is termed unichain. A state is periodic with period k if any return to state s must occur in multiples of k time steps, where k is some integer greater than 1. An example illustrating the classification of states in a Markov chain is shown in Figure 1.

Transience and recurrence describe the likelihood of returning to a state *conditioned* on starting from that state, regardless of the initial state distribution β. Given β, we also define an *isolated* component I as a maximal set of states in $\mathcal{M}$ that can never be visited, that is, $\beta_I = \sum_{s \in I} \beta_s = 0$, where β_s is the initial probability of being in state s, and I cannot be reached from any state in $S \setminus I$, i.e., $\sum_{s' \in I} T(s'|s) = 0, \forall s \in S \setminus I$. In Figure 1, the set of states $\{s_3, s_4\}$ is isolated. The term 'reachable' refers to states that are not isolated.

Definition 2 (Markov decision process (MDP)). *An MDP is a tuple $\mathcal{M} = (S, A, T, R, \beta)$, in which S denotes the state space, A the set of actions, $T : S \times A \times S \to [0, 1]$ the transition function with $T(s'|s, a)$ denoting the probability of transitioning from state s to state s' under action a, $R : S \times A \times S \to \mathbb{R}$ a reward obtained when action a is taken in state s and we end up in state s', and $\beta : S \to [0, 1]$ the initial distribution. By $A(s) \subseteq A$, we denote the set of actions available in state s.*

Definition 3 (Transition graph). *We define the transition graph of an MDP $\mathcal{M} = (S, A, T, R, \beta)$ as the directed graph whose vertex set is the state space S, and in which there is a directed edge from vertex s to vertex s' if there exists an action $a \in A(s)$ such that $T(s'|s, a) > 0$. The transition graph of a Markov chain $\mathcal{M} = (S, T, \beta)$ is the directed graph with vertex set S, and which has a directed edge from vertex s to vertex s' if $T(s'|s) > 0$.*

Definition 4 (Terminal strongly connected component (TSCC)). *Consider the transition graph of a Markov chain or MDP $\mathcal{M}$ with state space S and initial distribution β. A strongly connected component (SCC) of the digraph is a maximal subset of vertices C, where for every pair of vertices $s, s' \in C$, there is a directed path*[1] *from s to s' and a directed path from s' to s (Tarjan, 1972). A Terminal Strongly Connected Component (TSCC) $S' \subseteq S$ is an SCC reachable from some initial state $s, \beta_s > 0$ and with no outgoing transitions to any state in $S \setminus S'$. A TSCC is also called a bottom SSC (Courcoubetis & Yannakakis, 1995). We denote by $r_k(\mathcal{M}) \subseteq S$ the set of states in the k^{th} TSCC of $\mathcal{M}$, and by $r(\mathcal{M}) = \bigcup_{k \in [m]} r_k(\mathcal{M})$ the union of all such sets. The complement set is denoted $\bar{r}(\mathcal{M}) := S \setminus r(\mathcal{M})$, which in the case of Markov chains is the set of transient or isolated states.*

Figure 1 illustrates a Markov chain with two TSCCs (highlighted with two separate colors).

Definition 5 (Stationary policy). *Given MDP $\mathcal{M} = (S, A, T, R, \beta)$, a stationary policy $\pi : S \to \Delta^A$ is a mapping of states to probability distributions over the space of actions A, where Δ^A is the probability simplex over A. The policy π specifies the conditional probability $\pi(a|s)$ that action a is taken in state s. The set of all stationary policies is denoted Π_S.*

[1] There is a directed path from node v to node w if it is possible to reach w from v by traversing the directed edges in the directions in which they point.

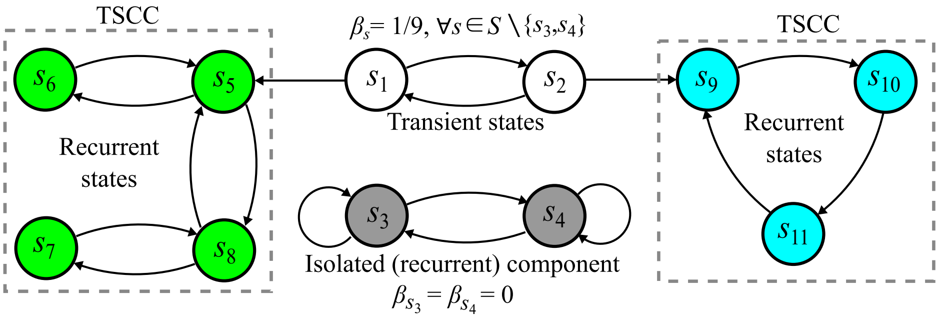

Figure 1: State classification in a Markov chain with four communicating classes. The set $\{s_1, s_2\}$ is transient, and the sets $\{s_3, s_4\}$, $\{s_5, s_6, s_7, s_8\}$ and $\{s_9, s_{10}, s_{11}\}$ are recurrent. This Markov chain is not irreducible since the states do not belong to one communicating class. States s_9, s_{10}, and s_{11} are periodic. The set $\{s_3, s_4\}$ is isolated since it is not reachable from states in $S \setminus \{s_3, s_4\}$ and it has zero initial distribution. The components colored in green and blue are the two TSCCs of the Markov chain, i.e., $r_1(\mathcal{M}) = \{s_5, s_6, s_7, s_8\}$ and $r_2(\mathcal{M}) = \{s_9, s_{10}, s_{11}\}$.

Definition 6 (Markov chain induced by policy)**.** *The tuple $\mathcal{M}_\pi = (S, T_\pi, \beta)$ is the Markov chain induced by a policy π in an underlying MDP $\mathcal{M} = (S, A, T, R, \beta)$, where*

$$T_\pi(s'|s) = \sum_{s \in A(s)} T(s'|s, a)\pi(a|s) \tag{1}$$

Definition 7 (Unichain and multichain MDP)**.** *An MDP is called unichain (Puterman, 1994; Altman, 1999) if every stationary deterministic policy induces a Markov chain that is unichain, that is, consists of exactly one recurrent set and possibly some transient states*[2]*. An MDP is said to be multichain if it is not unichain. See Figure 2 and its caption for an example.*

Definition 8 (Stationary distribution)**.** *Given a Markov chain $\mathcal{M} = (S, T, \beta)$, a stationary distribution* $\Pr^\infty : S \to [0, 1]$ *over the state space is any solution to the set of equations (Norris, 1997)*

$$\Pr^\infty(s) = \sum_{s' \in S} \Pr^\infty(s') T(s|s'),\ \Pr^\infty(s) \geq 0,\ \forall s \in S \tag{2}$$

$$\sum_{s \in S} \Pr^\infty(s) = 1\ . \tag{3}$$

According to the ergodic theorem of Markov chains, the solution to (2) and (3) is unique if and only if T is the transition matrix of a unichain (Gallager, 2013, Chapter 4). If there are

[2] This definition does not require the recurrent class to be ergodic (hence aperiodic). Our analysis dispenses with the aperiodicity precondition as will be clear in the sequel.

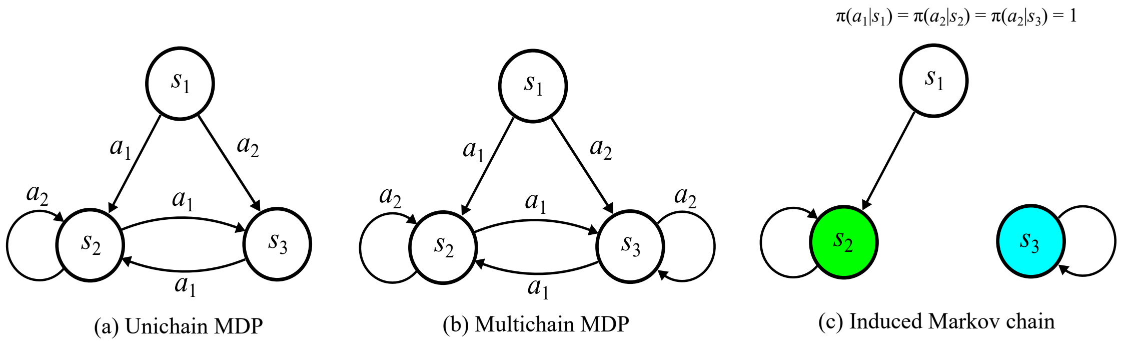

Figure 2: (a) Unichain MDP: every deterministic policy induces a Markov chain that has exactly one recurrent component. (b) Multichain MDP: adding the self-loop to state s_3 yields a multichain MDP. For example, the deterministic policy defined by $\pi(a_1|s_1) = \pi(a_2|s_2) = \pi(a_2|s_3) = 1$ induces the Markov chain in (c), which is multichain with two recurrent components $\{s_2\}$ and $\{s_3\}$.

multiple recurrent classes, then in general there will be many stationary distributions. For example, for the Markov chain of Figure 2(c), one can verify that the distribution $\Pr^\infty(s_1) = \Pr^\infty(s_2) = 0, \Pr^\infty(s_3) = 1$ and the distribution $\Pr^\infty(s_1) = \Pr^\infty(s_3) = 0, \Pr^\infty(s_2) = 1$ both satisfy (2) and (3), thus they are both stationary distributions of the Markov chain. Note that a stationary distribution may not be representative of the true steady-state behavior of the system (c.f. Definition 10 and the following example).

Definition 9 (Stationary matrix of a Markov chain)**.** *Given a Markov chain $\mathcal{M} = (S, T, \beta)$, the stationary matrix T^∞ is given by the Cesàro limit*[3] *(Puterman, 1994)*

$$T^\infty = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} T^t \,. \tag{4}$$

Given a finite multichain Markov chain $\mathcal{M} = (S, T, \beta)$ with transient set F and recurrent (i.e., non-transient) components $E_k, k \in [m]$, the transition matrix T can be expressed in the canonical form (Puterman, 1994, Appendix A)

$$T = \begin{bmatrix} T_1 & 0 & \dots & 0 & 0 \\ 0 & T_2 & & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & T_m & 0 \\ L_1 & L_2 & \dots & L_m & Z \end{bmatrix} \tag{5}$$

where the matrices T_k correspond to transitions between states in E_k, L_k to transitions from states in F to states in $E_k, k \in [m]$, and Z to transitions between states in F. Similarly,

[3] The Cesàro limit always exists and accounts for the non-convergence of powers of transition matrices of periodic chains. Hence, we do not need a precondition about aperiodicity as our analysis does not require that $T^\infty = \lim_{n\to\infty} T^n$.

we use $T_{\pi,k}, L_{\pi,k}$, $k = 1, \ldots, m$, and Z_π to denote the corresponding submatrices of the transition matrix T_π of the Markov chain $\mathcal{M}_\pi$ induced by policy π. Also (Puterman, 1994; Kallenberg, 1983),

$$T^\infty(s', s) = \begin{cases} \eta_s, & s', s \in E_k \text{ for some } k \in [m] \\ p_{s'k}\eta_s & s' \in F, s \in E_k \\ 0 & \textit{otherwise} \end{cases} \tag{6}$$

where, $\eta_s = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} T^t(s', s)$, is the long term proportion of time the chain spends in s from initial states $s' \in E_k$, $\sum_{s\in E_k} \eta_s = 1$, $p_{s'k}$ is the absorption probability from the transient state $s' \in F$ into the recurrent class $E_k, k \in [m]$, and $\sum_{k=1}^{m} p_{s'k} = 1, \forall s' \in F$.

In this paper, we are interested in the asymptotic behavior of an agent's policy in an MDP, as captured by the steady-state distribution of the induced Markov chain defined next.

Definition 10 (Steady-state distribution)**.** *Given an MDP $\mathcal{M}$ and policy π, the steady-state distribution $\Pr_\pi^\infty : S \times A \to [0, 1]$ over the state-action pairs, also known as the occupation measure (Altman, 1999, Chapter 4), is the long-term proportion of time spent in state-action pair (s, a) as the number of transitions approaches ∞, i.e.,*

$$\Pr_\pi^\infty(s, a) = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} \Pr(S_t = s, A_t = a|\beta, \pi), \quad s \in S, a \in A(s) \tag{7}$$

if the limit exists, where S_t and A_t are the state and action at time t. Also, $\Pr_\pi^\infty(s) := \sum_{a\in A(s)} \Pr_\pi^\infty(s, a)$ is the steady-state probability of being in state $s \in S$. The steady-state distribution is a stationary distribution of the Markov chain induced by the policy π.

As an example, consider the MDP in Figure 2(b) with $\beta_{s_1} = 1, \beta_{s_2} = \beta_{s_3} = 0$. The steady-state distribution of the policy $\pi(a_1|s_1) = \pi(a_2|s_2) = \pi(a_2|s_3) = 1$, which induces the Markov chain in Figure 2(c), has $\Pr_\pi^\infty(s_2, a_2) = 1$ and 0 otherwise.

Definition 11. *Given an MDP $\mathcal{M} = (S, A, T, R, \beta)$ and a set of policies $\Pi \subseteq \Pi_S$, we define*

$$\mathcal{P}^\infty(\Pi) := \{\Pr_\pi^\infty | \pi \in \Pi\}$$

as the set of occupation measures induced by policies in Π, where $\Pr_\pi^\infty$ is defined in (7).

Definition 12 (Steady-state specifications and constraints (Velasquez, 2019))**.** *Given an MDP $\mathcal{M} = (S, A, T, R, \beta)$ and a set of labels $L = \{L_1, \ldots, L_{n_L}\}$, where $L_i \subseteq S$, a set of steady-state specifications is given by $\Phi_L^\infty = \{(L_i, [l_i, u_i])\}_{i=1}^{n_L}$. Given a policy π, the specification $(L_i, [l_i, u_i]) \in \Phi_L^\infty$ is satisfied if and only if the steady-state constraint*

$$l_i \leq \sum_{s\in L_i} \Pr_\pi^\infty(s) \leq u_i \tag{8}$$

is satisfied; that is, if the steady-state probability of being in a state $s \in L_i$ in the Markov chain $\mathcal{M}_\pi$ falls within the interval $[l_i, u_i]$.

Definition 13 (Labeled MDP (Velasquez, 2019))**.** *An MDP $\mathcal{M} = (S, A, T, R, \beta, L, \Phi_L^\infty)$ augmented with the label set L and specifications Φ_L^∞ is termed a labeled MDP (LMDP).*

Lemma 1. *(Kallenberg, 1983, Theorem 4.3.2)(Krass & Vrieze, 2002) Given an MDP $\mathcal{M} = (S, A, T, R, \beta)$ and policy $\pi \in \Pi_S$, the steady-state distribution $\Pr_\pi^\infty := \{\Pr_\pi^\infty(s,a)\}_{s,a}$ of the Markov chain $\mathcal{M}_\pi$ is*

$$\Pr_\pi^\infty(s,a) = (\beta^\top T_\pi^\infty)_s \pi(a|s), \;\; s \in S, a \in A(s) \tag{9}$$

where T_π^∞ is the Cesàro limit in (4), i.e., $T_\pi^\infty = \lim_{n\to\infty} \frac{1}{n}\sum_{t=1}^n T_\pi^t$.

Definition 14 (Expected average reward)**.** *Given an MDP $\mathcal{M} = (S, A, T, R, \beta)$, the expected average reward $R_\pi^\infty(\beta)$ of a policy π is defined as*

$$R_\pi^\infty(\beta) = \liminf_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} \mathbb{E}_{\substack{A_t\sim\pi \\ S_0\sim\beta}}[R(S_t, A_t)] \tag{10}$$

where $R(s,a) := \sum_{s'\in S} T(s'|s,a) R(s,a,s')$, and the expectation is w.r.t. the probability measure induced by the initial distribution β and the policy π over the state-action trajectories.

It follows from the definition of the expected average reward in (10) and the steady-state distribution (7) that for a stationary policy π (Krass & Vrieze, 2002; Altman, 1999)

$$R_\pi^\infty(\beta) = \sum_{s\in S} \sum_{a\in A(s)} \Pr_\pi^\infty(s,a) R(s,a) \tag{11}$$

where $\Pr_\pi^\infty(s,a)$ is given in (9).

The primary focus of this paper in the context of steady-state planning is to find stationary policies that maximize the expected average reward (10) while satisfying specifications Φ_L^∞ on the steady-state distribution (see Definition 12). We restrict the search to certain classes of stationary policies which will be introduced and defined precisely in Section 4. Our solution approach to this constrained MDP problem is based on linear programming formulations, which optimize a linear objective function capturing the expected reward, subject to linear equality and inequality constraints. Such constraints encode restrictions on the steady-state distributions induced by policies of interest, as well as desired steady-state specifications on the long-term frequencies for state-actions pairs. The decision variables of the LPs correspond to the occupation measures, and policies are obtained from their optimal solutions. We establish a one-to-one correspondence between optimal solutions of said LPs and optimal policies of the constrained MDP problem.

3. Related Work

Research related to steady-state planning often comes from the field of average- or expected-reward constrained MDPs and has its roots in mathematical programming (Bertsekas, 2005). Many solutions proposed in this area utilize linear programming formulations to derive policies (Altman, 1999). We illustrate these formulations in order of increasing complexity and elucidate the key differences between the formulations in the literature and our own. First, let us consider the simple problem of deriving a policy for an agent which seeks to maximize expected reward without any constraints on its steady-state distribution.

In the unichain MDP case, one may synthesize a policy by solving a linear program (LP) of the form (Manne, 1960; De Ghellinck, 1960)

$$\begin{aligned} \max \quad & \sum_{s\in S}\sum_{a\in A(s)} x_{sa} \sum_{s'\in S} T(s'|s,a)R(s,a,s') \text{ subject to} && \\ & \sum_{s\in S}\sum_{a\in A(s)} x_{sa} T(s' \mid s,a) = \sum_{a\in A(s')} x_{s'a} && \forall s' \in S \\ & x_{sa} \in [0,1] && \forall s \in S, a \in A(s) \\ & \sum_{s\in S}\sum_{a\in A(s)} x_{sa} = 1\ . && \end{aligned} \tag{12}$$

The policy can be derived from the occupation measures given by x_{sa} through a simple calculation. It is worth noting that this combination of occupation measures and linear programming has enabled significant progress in the area of planning within stochastic shortest paths MDPs, where several occupation measure heuristics have been defined to find decision-making policies that maximize the probability of reaching a set of goal states while satisfying multiple cost constraints (Trevizan, Thiébaux, Santana, & Williams, 2016, 2017; Trevizan, Thiébaux, & Haslum, 2017; Baumgartner, Thiébaux, & Trevizan, 2018). While the LP in (12) always produces valid solutions for unichain MDPs, this is not necessarily the case for multichain MDPs due to the fact that there may be more than one ergodic set (Puterman, 1994). This issue is rectified by modifying LP (12) to obtain (Denardo & Fox, 1968; Kallenberg, 1983)

$$\begin{aligned} \max \quad & \sum_{s\in S}\sum_{a\in A(s)} x_{sa} \sum_{s'\in S} T(s'|s,a)R(s,a,s') \text{ subject to} && \\ & \sum_{s\in S}\sum_{a\in A(s)} x_{sa} T(s' \mid s,a) = \sum_{a\in A(s')} x_{s'a} && \forall s' \in S \\ & \sum_{s\in S}\sum_{a\in A(s)} y_{sa} T(s' \mid s,a) = \sum_{a\in A(s')} (x_{s'a} + y_{s'a}) - \beta_{s'} && \forall s' \in S \\ & x_{sa} \in [0,1],\ y_{sa} \geq 0 && \forall s \in S, a \in A(s). \end{aligned} \tag{13}$$

The new y_{sa} variables guide policy formation on the transient states. Both LP (12) and LP (13) yield stationary stochastic policies. Furthermore, there always exists at least one optimal deterministic policy, which can easily be derived from the stochastic policy solution obtained from the LPs (Puterman, 1994).

For producing control policies *with* steady-state specifications, LPs (12) and (13) are extended to include linear steady-state constraints on the occupation measures. When applied to *unichain* MDPs, the constrained version of LP (12) encounters minor difficulties, in that there may not be an optimal deterministic policy (Altman, 1999). Nonetheless, the LP always produces an optimal stochastic stationary policy. In fact, there exists an optimal policy having at most n_L "randomizations", i.e. having at most $|S| + n_L$ state-action pairs with non-zero probability of being selected (Ross, 1989).

On the other hand, serious issues arise when LP (13) is augmented with steady-state constraints and solved for multichain MDPs, as described in the pioneering work of Kallenberg (1983). In particular, it was shown that there is no one-to-one correspondence between the feasible solutions of the augmented LP and the stationary policies. Instead, the space of feasible solutions is partitioned into equivalence classes of various feasible solutions mapping to the same policy. The key deficiency is that the steady-state distribution of the Markov chain induced by the synthesized policy does not match the optimal solution to the LP in general, and so the derived policy does not always meet the steady-state specifications (see Example 1 in Section 4.1). This issue is not easily remedied, since the optimal solution may not be achievable by any stationary policy, or identifying such a policy would generally require combinatorial search. We refer the reader to the paper by Krass and Vrieze (2002) for an overview.

In order to mitigate the preceding problem of integrating steady-state constraints, various assumptions have been made in the literature on the structure of the underlying MDP. Multichain MDPs are also frequently excluded from the conversation altogether. The assumption that the MDP is ergodic, and therefore every policy induces an ergodic Markov chain, has been used by Akshay et al. (2013) to ensure that steady-state equations and constraints on the same are satisfied. This assumption is relaxed to some extent by Ross (1989), Altman (1999), Feinberg (2009), where unichain MDPs are allowed. The assumption of either an ergodic or a unichain MDP requires that no stationary deterministic policy induces more than a single recurrent class, thus severely limiting the applicability of these methods. These assumptions are removed in the recent work of Velasquez (2019), where neither ergodic nor recurrence assumptions are made on the underlying MDP. However, the solution proposed therein finds an irreducible Markov chain in the underlying MDP, if one exists, and is therefore suitable for communicating MDPs where, for any two states s and s', there exists a deterministic stationary policy such that s can reach s' in a finite number of steps (Puterman, 1994). This solution, however, is too restrictive, thus not suitable for reasoning over general multichain MDPs.

Another approach taken to address these challenges is to simply allow solutions to take the form of non-stationary policies. In the work of Kallenberg (1983), this is accomplished by a computationally expensive approach producing a potentially different policy in each time step. Another approach, proposed by Krass and Vrieze (2002), starts by using one policy, and then switches to a second "tail" stationary policy. The time at which the switch occurs is determined by a lottery performed at each time step, and once the switch occurs the tail policy continues to be used indefinitely (thus the policy is "ultimately" stationary *once* the switch occurs). However, this approach has three key limitations. First, the constraints must take the form of a target frequency vector, which imposes an equality constraint on the steady-state distribution over all states. Second, the lottery system does not guarantee that the switch will occur in a finite number of steps, thus the policy is not guaranteed to be ultimately stationary. Third, the policy depends on a marker to track whether or not the switch has occurred. This marker is not part of the MDP, and therefore the MDP machinery must be modified to include a so-called marker-augmented history. As an alternative, the authors also propose a way to extend the given MDP with additional states, such that the problem can be solved using a stationary policy applied to the extended

MDP. However, this approach still cannot produce a stationary policy to solve the original problem.

While most methods for solving constrained MDPs revolve around the use of mathematical programs, some reinforcement learning approaches have also been proposed for optimizing the average-reward objective and, to a lesser extent, for solving constrained instances of average-reward MDPs. Some noteworthy examples include the constrained actor-critic method proposed by Bhatnagar and Lakshmanan (2012), wherein a Lagrangian relaxation of the problem is used to incorporate steady-state costs into the objective function being optimized by the constrained actor-critic algorithm. A similar Lagrangian Q-learning approach is proposed by Lakshmanan and Bhatnagar (2012). Both of these reinforcement learning methods assume that every Markov chain induced by a policy is irreducible, which allows only a single recurrent class as with ergodic and unichain assumptions described earlier. The Lagrangian approach has also been applied to specific stochastic policy linear programming formulations relevant to aircraft maintenance problems where the asymptotic failure is to be kept below some small threshold (Boussemart & Limnios, 2004; Boussemart et al., 2002).

In contrast to the foregoing efforts, our approach is computationally tractable, works with the most general multichain MDPs, and always produces a stationary policy that satisfies the given steady-state specifications, if one exists. Additionally, none of the aforementioned methods consider constraints on the expected visits to transient states, as we are considering in our work.

4. Steady-State Policy Synthesis: Problem Formulation

In this section, we introduce the Steady-State Policy Synthesis (SSPS) problem of finding a stationary policy from predefined classes of policies (edge- and class-preserving) that maximizes the expected average reward subject to steady-state specifications. In contrast to prior work, we do not impose restrictions on the underlying MDP. Before we present our formulation, we briefly discuss the challenges underlying policy synthesis under the average reward optimality criterion and demonstrate the limitations of existing formulations in this context. Subsequently, we specify our search domain of policies and define the SSPS problem of synthesizing optimal policies from this domain.

4.1 Challenges and Limitations

We motivate this section with a simple example. Suppose an autonomous agent is marooned on a set of three connected frozen islands as shown in Figure 3. The agent's goal is to maximize the amount of time it spends fishing for sustenance while at the same time building a canoe to escape the islands. The agent has an equal chance of starting in any state belonging to the larger island of size $n \times n/2$, i.e., we have $\beta_s = 2/n^2$ for each state s in the island. Once the agent moves to one of the two smaller islands, it is unable to return to the larger island. One quarter of the land in the small islands contains logs which can be used to build a canoe, and each of these islands contains one fishing site as well. For the first small island we have steady-state specifications $(L_{\text{log1}}, [0.25, 1])$, $(L_{\text{log2}}, [0.25, 1])$, $(L_{\text{canoe1}}, [0.05, 1.0])$ and reward $R(\cdot, \cdot, L_{\text{fish1}}) = R(\cdot, \cdot, L_{\text{fish2}}) = 1$. Likewise, the second small island has steady-state specifications $(L_{\text{canoe2}}, [0.05, 1.0])$, $(L_{\text{fish1}}, [0.1, 1.0])$, $(L_{\text{fish2}}, [0.1, 1.0])$

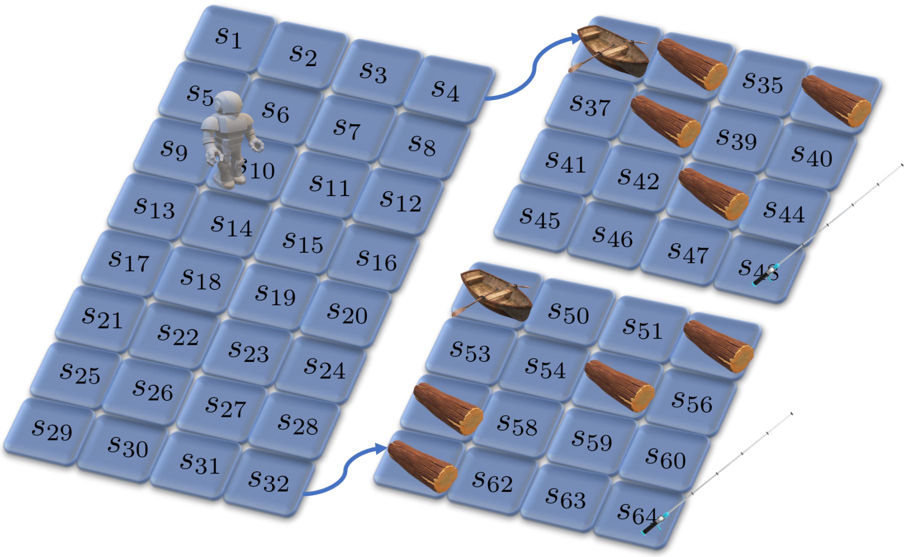

Figure 3: LMDP $\mathcal{M} = (S, A, T, R, \beta, L, \Phi_L^\infty)$ with labels $L_{\text{log1}} = \{s_{34}, s_{36}, s_{38}, s_{43}\}$, $L_{\text{log2}} = \{s_{52}, s_{55}, s_{57}, s_{61}\}$, $L_{\text{canoe1}} = \{s_{33}\}$, $L_{\text{canoe2}} = \{s_{49}\}$, $L_{\text{fish1}} = \{s_{48}\}$, $L_{\text{fish2}} = \{s_{64}\}$, steady-state specifications $(L_{\text{log1}}, [0.25, 1])$, $(L_{\text{log2}}, [0.25, 1])$, $(L_{\text{canoe1}}, [0.05, 1.0])$, $(L_{\text{canoe2}}, [0.05, 1.0])$, $(L_{\text{fish1}}, [0.1, 1.0])$, $(L_{\text{fish2}}, [0.1, 1.0]) \in \Phi_L^\infty$, and rewards $R(\cdot,\cdot,L_{\text{fish1}}) = R(\cdot,\cdot,L_{\text{fish2}}) = 1$, $R(\cdot,\cdot,S \setminus (L_{\text{fish1}} \cup L_{\text{fish2}})) = 0$.

and reward $R(\cdot,\cdot,S \setminus (L_{\text{fish1}} \cup L_{\text{fish2}})) = 0$. Because the islands are covered in ice, the agent has a chance of slipping in three possible directions whenever it moves. Specifically, if the agent attempts to go right (left), it has a 90% chance of transitioning to the right (left), and there is a 5% chance of transitioning instead to either of the states above or below it. Similarly, if the agent tries to go up (down), it moves to the states above (below) it with 90% chance, and to the states to the right and left of it with chance 5% each. This Frozen Island scenario is motivated by that found in OpenAI Gym's FrozenLake environment (Brockman et al., 2016).

For this example, the LP by Velasquez (2019) is infeasible since there exists no policy that induces an irreducible Markov chain, that is, one where all states in S belong to one recurrent class. The LP by Kallenberg (1983) in (13) will return a solution (x, y), from which the stationary policy $\pi := \pi(x, y)$ is computed as follows

$$\pi(a|s) = \begin{cases} \frac{x_{sa}}{x_s} & s \in E_x, a \in A(s) \\ \frac{y_{sa}}{y_s} & s \in E_y \setminus E_x, a \in A(s) \\ \text{arbitrary} & \text{otherwise} \end{cases} \tag{14}$$

where $x_s := \sum_{a \in A(s)} x_{sa}$, $y_s = \sum_{a \in A(s)} y_{sa}$, $E_x := \{s \in S : x_s > 0\}$ and $E_y := \{s \in S : y_s > 0\}$. However, in general the steady-state distribution induced by the policy (14) will not satisfy the specified constraints. This deficiency is best demonstrated via a simple example. The reader is also referred to Example 1 by Krass and Vrieze (2002).

Example 1. *Consider the MDP in Figure 2(b) with initial probability* $\beta_{s_1} = \beta_{s_3} = 0, \beta_{s_2} = 1$. *One feasible solution* x *of the LP in* (13) *(Kallenberg, 1983, Program 4.7.6) has* $x_{s_2 a_2} =$

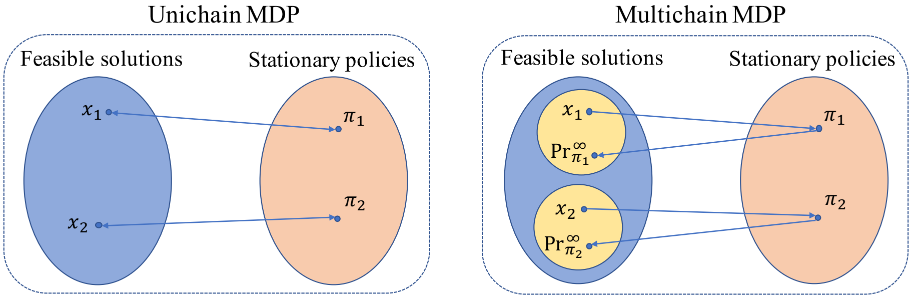

Figure 4: (Left) One-to-one correspondence between the feasible LP solutions and the stationary policies in unichain MDPs. (Right) Equivalence classes of feasible solutions map to stationary policies. The steady-state distribution of the Markov chain induced by a policy need not agree with the LP solution, and could hence fail to meet the LP constraints.

$x_{s_3a_2} = 0.5$. *The policy* π *in* (14) *corresponding to* x *has* $\pi(a_2|s_2) = \pi(a_2|s_3) = 1$*, hence* $\Pr^{\infty}_{\pi}(s_2, a_2) = 1$*. Therefore,* $\Pr^{\infty}_{\pi} \neq x$*.*

The previous example underscores the main challenge underlying steady-state planning in constrained Markov decision models with the average reward criterion: solutions to formulated programs and stationary policies are not in one-to-one correspondence. In other words, given a feasible LP solution (x, y), the steady-state distribution $\Pr^{\infty}_{\pi}$ induced by the policy $\pi(x, y)$ derived from that solution is not equal to x in general. As a result, unlike unichain MDPs (Altman, 1999), steady-state specifications encoded as constraints on the state-action variables are generally not met by π. Figure 5 (Left) illustrates the one-to-one correspondence between LP solutions and stationary policies found in unichain MDPs. Figure 5 (Right) illustrates the lack of such correspondence in multichain MDPs, where instead, equivalence classes of feasible LP solutions (yellow circles) map to the same policy (Kallenberg, 1983; Puterman, 1994).

4.2 Problem Setup

The previous example motivates the work of this paper in which we develop an approach to synthesizing policies with provably correct asymptotic behavior based on the notions of edge preservation and class equivalence. First, we will define sets of policies under which certain class structures are preserved and give an example of such policies, then define the SSPS problem of finding an optimal policy from such classes.

Definition 15 (Edge-preserving policies)**.** *Given an MDP* $\mathcal{M}$*, we define the set of Edge-Preserving (EP) policies* Π_{EP} *as the set of stationary policies that play every action available at states in the TSCCs* $r(\mathcal{M})$ *of* $\mathcal{M}$ *and for which* $r(\mathcal{M}_\pi) = r(\mathcal{M})$*, i.e.,*

$$\Pi_{EP} = \left\{\pi \in \Pi_S : r(\mathcal{M}_\pi) = r(\mathcal{M}) \ \wedge \ \pi(a|s) > 0, \forall s \in r(\mathcal{M}), a \in A(s)\right\} . \qquad (15)$$

Hence, for every state $s \in r(\mathcal{M})$ (see Definition 4), an EP policy assigns a non-zero probability to every action in $A(s)$, and every state in $\bar{r}(\mathcal{M})$ is either transient or isolated in the Markov chain induced by the policy. For example, the uniform policy which has

$\pi(a|s) = 1/|A(s)|, \forall s \in S$ is in Π_{EP}. Note that other policies in Π_{EP} could assign a very small probability (as long as it is non-zero) to non-rewarding transitions in $r(\mathcal{M})$. Using an open set definition in (15) simplifies the exposition and the subsequent theoretical analysis, however, it does not guarantee that an optimal policy from the set always exists. We discuss and analyze variations of the problem formulation to address this issue at length in Section 5.4.4.

Next, we introduce two sets of policies whose definitions rest on two distinct notions of class preservation.

Definition 16 (Class-preserving policies)**.** *Given an MDP $\mathcal{M}$ with TSCCs $r_k(\mathcal{M}), k = 1, \ldots, m$, we define the set of Class-Preserving (CP) policies Π_{CP} as the set of stationary policies that induce Markov chains with the same TSCCs as those of $\mathcal{M}$, i.e.,*

$$\Pi_{CP} = \left\{\pi \in \Pi_S : r(\mathcal{M}_\pi) = r(\mathcal{M}) \ \wedge \ \forall k \in [m], r_k(\mathcal{M}_\pi) = r_k(\mathcal{M})\right\} \ . \tag{16}$$

Note that the condition $r(\mathcal{M}_\pi) = r(\mathcal{M})$ in Definitions 15 and 16 implies that $\bar{r}(\mathcal{M})$ consists of transient or isolated states in $\mathcal{M}_\pi$ for any π in Π_{EP} or Π_{CP}. Per (16), a CP policy preserves the recurrence of all states in the TSCCs of the MDP but, unlike EP policies, its support need not be the entire set of actions available at said states. Therefore, CP policies can conceivably achieve larger rewards than EP policies by averting non-rewarding transitions.

Definition 17 (Class-preserving up to unichain)**.** *Given an MDP $\mathcal{M}$ with TSCCs $r_k(\mathcal{M}), k = 1, \ldots, m$, we define the set of Class-Preserving-up-to-Unichain (CPU) policies Π_{CPU} as*

$$\Pi_{CPU} = \left\{\pi \in \Pi_S : r(\mathcal{M}_\pi) \subseteq r(\mathcal{M}) \ \wedge \ \forall k \in [m], \exists!\, r_k(\mathcal{M}_\pi) \subseteq r_k(\mathcal{M})\right\} \ , \tag{17}$$

that is, the set of stationary policies that induce Markov chains $\mathcal{M}_\pi$ in which the TSCCs of the MDP $\mathcal{M}$ are reachable and unichain (i.e., each contains exactly one non-isolated, recurrent component) and the recurrent states are a subset of the recurrent states of $\mathcal{M}$ (Recalling that the notation $\exists!$ in (17) *refers to the existence of a unique set).*

This definition captures a more relaxed notion of class preservation than (16) for CP policies in that it relaxes the requirement that all states in the TSCCs of $\mathcal{M}$ be recurrent and reachable in the Markov chain $\mathcal{M}_\pi$ induced by the policy π, to the milder requirement that in $\mathcal{M}_\pi$ there exists a unique reachable recurrent class in each of the TSCCs of $\mathcal{M}$.

The aforementioned definitions are best illustrated by an example. Figure 5(b) illustrates a Markov chain induced by an EP policy, i.e., one that plays every action available in the TSCCs of the MDP of Figure 5(a) with non-zero probability. As shown, s_1 is isolated and s_2 is transient – these would both be transient under the uniform policy. The TSCCs of the induced chain are highlighted with two separate colors. Examples of Markov chains induced by a CP and a CPU policy are shown in Figure 5(c) and (d), respectively. The Markov chain of Figure 5(c) has the exact same TSCCs of the MDP and of the Markov chain of Figure 5(b) induced by the EP policy, with the fundamental difference that the CP policy is not supported on every action available in the TSCCs (e.g., see the recurrent component highlighted in blue). By contrast, states s_3, s_4, s_6 and s_7 are transient in the Markov chain of Figure 5(d). The set consisting of states s_3, s_4, s_5 is unichain, having exactly one recurrent

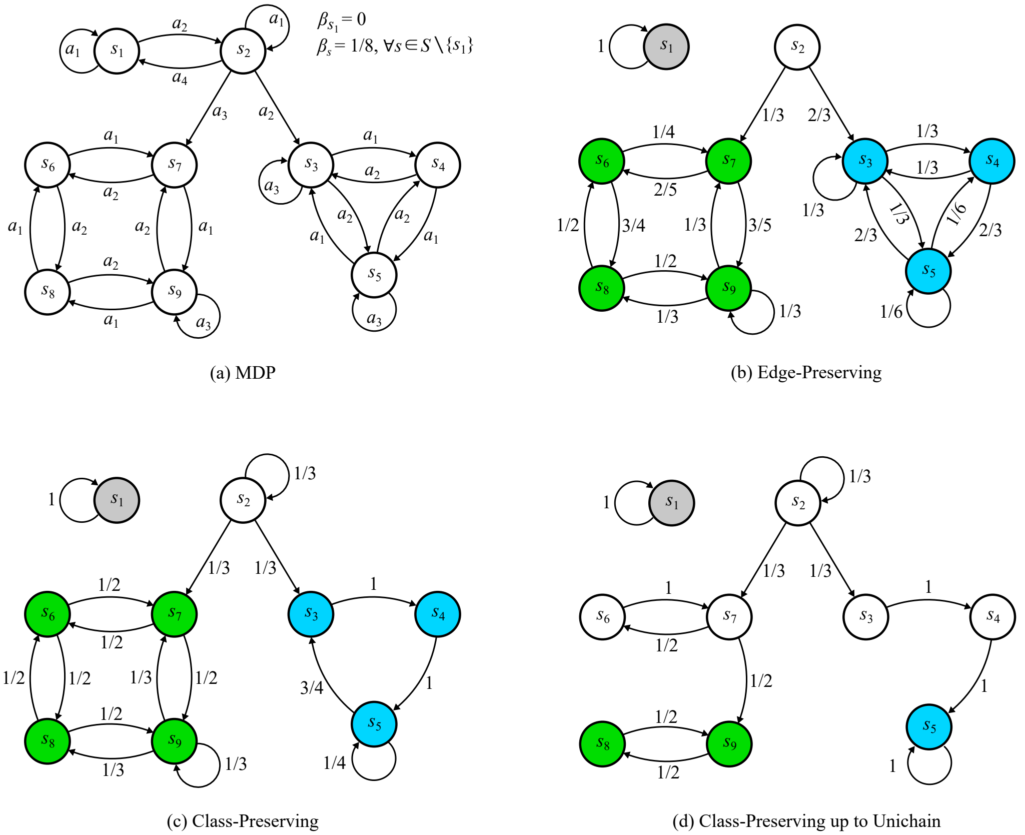

Figure 5: (a) MDP and Markov chains induced by (b) EP, (c) CP, and (d) CPU policies. For the MDP, all transitions are deterministic, i.e., $T(s'|s,a) \in \{0,1\}$ indicating if there is an outgoing edge from s to s' under action a, the rewards are defined such that $R(s_5, a_3) = R(s_8, a_1) = 1$ and 0 otherwise, and the initial probabilities are $\beta_{s_1} = 0$ and $\beta_s = 1/8, \forall s \in S \setminus \{s_1\}$. The numbers next to the edges of the Markov chains are the conditional probabilities $\pi(a|s)$ of the different actions given the states specifying the policies.

component (state s_5) and two transient states (s_3 and s_4). Similarly, the set composed of states s_6, s_7, s_8, s_9 is unichain with one recurrent component (s_8 and s_9) and two transient states (s_6 and s_7).

The classes of policies defined in (15), (16) and (17) satisfy the following relations.

Lemma 2. $\Pi_{EP} \subseteq \Pi_{CP} \subseteq \Pi_{CPU}$.

We remark that the inclusions in Lemma 2 are generally strict, except for some special MDPs. Specifically, given a general MDP $\mathcal{M}$, there may exist a policy $\pi \in \Pi_{CP}$ for which $r(\mathcal{M}_\pi) = r(\mathcal{M})$ and $\pi(a|s) = 0$ for some $a \in A(s), s \in r(\mathcal{M})$, in which case $\pi \notin \Pi_{EP}$. Similarly, since a unichain may contain some transient states, Π_{CP} is generally a proper subset of Π_{CPU}.

Problem Definition. We can readily define the class of problems SSPS(Π), parametrized by a predefined set of stationary policies Π, of finding a policy in the set Π that maximizes the expected average reward while satisfying a given set of steady-state specifications.

Definition 18 (Steady-state policy synthesis (SSPS)). *Given an LMDP $\mathcal{M} = (S, A, T, R, \beta, L, \Phi_L^\infty)$ and a domain of policies $\Pi \subseteq \Pi_S$, the SSPS(Π) problem is to find an optimal stochastic policy $\pi \in \Pi$ that maximizes the expected average reward defined in* (10) *and satisfies the steady-state specifications Φ_L^∞ (Definition 12), i.e.,*

$$\begin{aligned} &\max_{\pi \in \Pi} \sum_{s \in S} \sum_{a \in A(s)} \Pr_\pi^\infty(s,a) R(s,a) \text{ subject to} \\ &\sum_{s \in L_i} \sum_{a \in A(s)} \Pr_\pi^\infty(s,a) \in [l,u], \quad \forall (L_i, [l,u]) \in \Phi_L^\infty \end{aligned} \tag{18}$$

If the maximum in (18) *cannot be attained over the domain* Π, *we define SSPS*(Π) *as the problem of finding a policy $\pi \in \Pi$ that satisfies the specifications Φ_L^∞ and whose expected average reward $R_\pi^\infty(\beta) \geq \sup_{\pi' \in \Pi} R_{\pi'}^\infty(\beta) - \epsilon$, for some arbitrarily small $\epsilon > 0$ (See Section 5.4.4).*

In this paper, we present solutions to SSPS(Π_{EP}), SSPS(Π_{CP}) and SSPS(Π_{CPU}), where Π in (18) is set to Π_{EP}, Π_{CP}, and Π_{CPU}, respectively[4]. To this end, we first determine the TSCCs $r(\mathcal{M})$ of $\mathcal{M}$ and the complement set $\bar{r}(\mathcal{M})$ using standard techniques from graph theory (Tarjan, 1972). These are then used to define an LP from which the solution policy is derived.

5. Linear Programming Based Solutions

In this section, we present our linear-programming-based solution to the SSPS problem (18) over edge- and class-preserving policies. We formulate linear programs that encode constraints on the limiting distributions of said policies to solve SSPS(Π_{EP}) and SSPS(Π_{CP}). The optimal solutions to the formulated programs provably yield optimal edge- and class-preserving policies that meet the desired specifications. The encoded constraints are also at the center of an iterative algorithm described in Section 5.3 to generate CPU policies

[4] Our work (Atia et al., 2020) has presented preliminary results for the SSPS(Π_{EP}) problem.

for SSPS(Π_{CPU}). Our main results on SSPS in edge- and class-preserving policies are presented in Sections 5.1, 5.2 and 5.3. To simplify the exposition, all proofs are deferred to the appendix.

We present three main programs. The first, used for the synthesis of optimal EP policies, is the most constrained as it encodes the requirement that every action in the terminal components must be played with non-zero probability. While this condition is not necessary in order to ensure one-to-one correspondence between the feasible solutions and the induced steady-state distributions, it results in a simple program whose solution provably yields an optimal EP policy that meets the specifications. The second program relaxes this condition to the milder requirement that every state in the terminal components is visited infinitely often (which may not require playing every action available), but at the expense of additional complexity. Specifically, its solution yields an optimal policy that meets the specifications from the class of CP policies (a superset of EP policies) but uses more complex flow constraints to encode said requirement. The third program is the least constrained and is used to synthesize a policy from the larger class of CPU policies. While its solution is not guaranteed to yield a CPU policy, we derive a characterization of its optimal solution, which inspires a greedy algorithm to construct such policy. We augment the program by iteratively adding constraints until convergence. The algorithm is guaranteed to converge in a finite number of steps to a (possibly) suboptimal CPU policy that meets the specifications.

By encoding constraints on the limiting distribution of the Markov chain induced by a stationary policy derived from an LP solution, the policy is ultimately absorbed in the TSCCs of the MDP. This restricts the long-term play to the TSCCs, which once reached, cannot be escaped. By imposing strict positivity on state-action pairs or flow constraints in the TSCCs, we further ensure that these components are unichain, and in turn, the long-term frequencies induced by the policy match the solution from which the policy is generated.

5.1 SSPS(Π_{EP}) – Synthesis over Edge-Preserving Policies

In this section, we formulate a linear program to solve SSPS(Π_{EP}) defined in (18), which seeks to maximize the expected average reward subject to specification constraints over the class of policies Π_{EP} in (15).

Given MDP $\mathcal{M}$, define Q_0 to be the set of vectors x, y satisfying

$$\begin{cases} (i) \quad \sum_{s\in S}\sum_{a\in A(s)} x_{sa}T(s' \mid s,a) = \sum_{a\in A(s')} x_{s'a}, \quad \forall s' \in S \\ (ii) \quad \sum_{s\in S}\sum_{a\in A(s)} y_{sa}T(s' \mid s,a) = \sum_{a\in A(s')}(x_{s'a} + y_{s'a}) - \beta_{s'}, \quad \forall s' \in S \\ (iii) \quad \sum_{f\in \bar{r}(\mathcal{M})}\sum_{a\in A(f)} x_{fa} = 0, \\ x_{sa} \in [0,1], y_{sa} \geq 0, \forall s \in S, a \in A(s), f \in \bar{r}(\mathcal{M}), k \in [m] \end{cases} \tag{19}$$

We can readily formulate LP$_1$ (20) to synthesize optimal EP policies, which incorporates two additional constraints beside the constraints in (19).

$$\begin{array}{lll} & \max & \displaystyle\sum_{s\in S}\sum_{a\in A(s)} x_{sa}R(s,a) \text{ subject to } (x,y)\in Q_0 \\ (\mathrm{LP}_1) & (iv) & l_i \le \displaystyle\sum_{s\in L_i}\sum_{a\in A(s)} x_{sa} \le u_i,\ \forall (L_i,[l_i,u_i])\in \Phi_L^\infty \\ & (v) & x_{sa} > 0,\ \forall s\in r_k(\mathcal{M}), k\in[m], a\in A(s) \end{array} \tag{20}$$

Constraints (i) – (iii) constrain the limiting distributions of Markov chains induced by the policies of interest, and are thus part of the constraint set of all programs we formulate in this work. In particular, they capture the structure of the stationary matrix T^∞ corresponding to the classifications $r(\mathcal{M})$ and $\bar{r}(\mathcal{M})$ (See Definition 4). Constraint (i) ensures that x is a stationary distribution (Altman, 1999; Puterman, 1994); constraint (ii), which is described in (Kallenberg, 1983, Chapter 4) and (Puterman, 1994, Sec 9.3), enforces consistency in the expected average number of visits y_{fa} for any transient state-action pair $f\in\bar{r}(\mathcal{M}), a\in A(f)$; constraint (iii) preserves the non-recurrence of the states $f\in\bar{r}(\mathcal{M})$ by forcing zero steady-state probability. Constraint (iv) encodes the steady-state specifications. The strict positivity constraint (v) preserves the transitions in the TSCCs to yield EP policies. In practice, we transform the strict inequalities to bounded ones by introducing an arbitrarily small constant on the right-hand side, thereby ensuring an optimal solution always exists (See Section 5.4.4). Enforcing constraints on the occupation measures ensures that, from any state $f\in\bar{r}(\mathcal{M})$, the process will be ultimately absorbed into the TSCCs $r_k(\mathcal{M}), k\in[m]$.

The next theorem guarantees that every feasible solution to LP$_1$ yields an EP policy.

Theorem 1. *Given an LMDP $\mathcal{M}$, let $(x,y)\in Q_1$, where Q_1 is the feasible set of solutions to* LP$_1$ (20)*, and let $\pi := \pi(x,y)$ be defined as in* (14)*. Then, $\pi\in\Pi_{EP}$.*

We can readily state the following theorem establishing the correctness of LP$_1$. It guarantees that the policy synthesized from an optimal solution to (20) (if one exists) is not only in Π_{EP}, but also is optimal among all such policies and meets the steady-state specifications, i.e., solves SSPS(Π_{EP}).

Theorem 2. *Given an LMDP $\mathcal{M}=(S,A,T,R,\beta,L,\Phi_L^\infty)$,* LP$_1$ *in* (20) *is feasible iff there exists a policy $\pi\in\Pi_{EP}$ such that the Markov chain $\mathcal{M}_\pi=(S,T_\pi,\beta)$ satisfies the specifications Φ_L^∞. Further, given an optimal solution x^*, y^* of* (20)*, the policy $\pi^* := \pi(x^*,y^*)$ as defined in* (14) *is optimal in the class of policies Π_{EP} and meets the specifications Φ_L^∞.*

5.2 SSPS(Π_{CP}) – Synthesis over Class-Preserving Policies

The strict positivity constraint (v) of LP$_1$ forces the policy to play every action in the TSCCs of the MDP $\mathcal{M}$ (by assigning non-zero probability to every action available), which may be restrictive and often unnecessary. Indeed, as we show, in order to ensure one-to-one correspondence between the optimal solutions of a formulated LP and the optimal policies of the constrained MDP derived from these solutions, it suffices to preserve the recurrence or the unichain property of these components.

To address this restriction, we introduce flow constraints in LP$_2$ given in (21) (replacing constraint (v)) to ensure the recurrence of the components $r_k(\mathcal{M}), k \in [m]$, in the induced chain. It helps to introduce some notation in order to express such constraints. We define the transition relation of an MDP by $T^{\text{rel}} = \{(s, s') \in S \times S | s \neq s' \wedge \exists a \in A(s), T(s'|s, a) > 0\}$ (Velasquez, 2019). This corresponds to the graph structure of the MDP. For each TSCC $r_k(\mathcal{M})$, we further define its graph structure as $T_k^{\text{rel}} = T^{\text{rel}} \cap r_k(\mathcal{M}) \times r_k(\mathcal{M})$. We can now add flow constraints in order to ensure that, for the Markov chain induced by the solution policy, each set $r_k(\mathcal{M})$ remains a recurrent class without necessarily having to take every action available in that set.

$$
\begin{array}{lll}
& \max \quad \sum_{s \in S} \sum_{a \in A(s)} x_{sa} \sum_{s' \in S} T(s'|s, a) R(s, a, s') \text{ subject to } (x, y) \in Q_0, & \\
& (iv) \quad l_i \leq \sum_{s \in L_i} \sum_{a \in A(s)} x_{sa} \leq u_i, & \forall (L_i, [l_i, u_i]) \in \Phi_L^{\infty} \\
& (vi) \quad f_{s_i s'} = \sum_{a \in A(s_i)} T(s'|s_i, a) x_{s_i a} & \forall (s_i, s') \in T_k^{\text{rel}}, k \in [m] \\
& (vii) \quad f_{s_i s'}^{\text{rev}} = \sum_{a \in A(s')} T(s_i|s', a) x_{s' a} & \forall (s', s_i) \in T_k^{\text{rel}}, k \in [m] \\
& (viii) \quad f_{ss'} \leq \sum_{a \in A(s)} T(s'|s, a) x_{sa} & \forall (s, s') \in T_k^{\text{rel}}, k \in [m] \\
(\text{LP}_2) & (ix) \quad f_{ss'}^{\text{rev}} \leq \sum_{a \in A(s')} T(s|s', a) x_{s' a} & \forall (s', s) \in T_k^{\text{rel}}, k \in [m] \\
& (x) \quad \sum_{(s', s) \in T^{\text{rel}}} f_{s' s} > \sum_{(s, s') \in T^{\text{rel}}} f_{ss'} & \forall s \in r(\mathcal{M}) \setminus \{s_i\} \\
& (xi) \quad \sum_{(s, s') \in T^{\text{rel}}} f_{s' s}^{\text{rev}} > \sum_{(s', s) \in T^{\text{rel}}} f_{ss'}^{\text{rev}} & \forall s \in r(\mathcal{M}) \setminus \{s_i\} \\
& (xii) \quad \sum_{(s', s) \in T^{\text{rel}}} f_{s' s} > 0 & \forall s \in r(\mathcal{M}) \\
& (xiii) \quad \sum_{(s, s') \in T^{\text{rel}}} f_{s' s}^{\text{rev}} > 0 & \forall s \in r(\mathcal{M}) \\
& (xiv) \quad f_{ss'}, f_{ss'}^{\text{rev}} \in [0, 1] & \forall (s, s') \in T^{\text{rel}}
\end{array}
\tag{21}
$$

The program LP$_2$ in (21) is such that every state in $r_k(\mathcal{M})$ can reach and is reachable from every other state in $r_k(\mathcal{M})$. For each $k \in [m]$, constraint (vi) induces an initial flow out of a randomly chosen state $s_i \in r_k(\mathcal{M})$ and into its neighbors s', that is proportional to the transition probability $T_\pi(s'|s_i)$ in the Markov chain induced by the solution policy; constraint $(viii)$ establishes the flow capacity between states in a similar manner; (x) ensures that the incoming flow into every state in $r_k(\mathcal{M})$ is greater than the outgoing flow; finally, constraint (xii) ensures that there is incoming flow into every state in $r_k(\mathcal{M})$. These constraints ensure that every state in $r_k(\mathcal{M})$ is reachable from s_i, whereas constraints (vii), (ix), (xi), $(xiii)$ address the foregoing in the reverse graph structure of the MDP, thereby ensuring that s_i is reachable from all states in $r_k(\mathcal{M})$. We remark that the feasible set in (21) is a superset of that in (20) since the flow constraints (vi)–(xiv) are implied by (v), hence LP$_2$ is less-constrained than LP$_1$.

We can readily state the following two theorems establishing the correctness of LP$_2$, which are the counterparts of Theorem 1 and 2. In particular, Theorem 3 guarantees that the solution to LP$_2$ is a CP policy, while Theorem 4 establishes that the policy (14) derived from the optimal solution to LP$_2$ in (21) solves SSPS(Π_{CP}), i.e., is optimal among the class of CP policies and meets the steady-state specifications.

Theorem 3. *Given an LMDP $\mathcal{M}$, let $(x, y) \in Q_2$ and π be defined as in* (14)*, where Q_2 is the feasible set of solutions to* LP$_2$ (21)*. Then, $\pi \in \Pi_{CP}$.*

Theorem 4. *Given an LMDP $\mathcal{M} = (S, A, T, R, \beta, L, \Phi_L^\infty)$, the LP in* (21) *is feasible iff there exists a policy $\pi \in \Pi_{CP}$, where Π_{CP} is defined in* (16)*, such that the Markov chain $\mathcal{M}_\pi = (S, T_\pi, \beta)$ satisfies the specifications Φ_L^∞. Further, given an optimal x^*, y^* of* (21)*, the policy $\pi(x^*, y^*)$ defined in* (14) *is optimal in the class of policies Π_{CP} and meets the specifications Φ_L^∞.*

5.3 SSPS(Π_{CPU}) – Synthesis over Class-Preserving up to Unichain Policies

In this section, we discuss policy synthesis over the larger set of policies Π_{CPU}. We provide a sufficient condition under which we can identify an optimal policy $\pi \in \Pi_{CPU}$ that meets the specifications. Based on this result, we develop an iterative algorithm to construct a policy in Π_{CPU} that provably meets the desired specifications.

Next, we give a sufficient condition for SSPS(Π_{CPU}), characterized in terms of the set of optimal solutions to LP$_3$ in (22).

$$\text{LP}_3 : \max \quad \sum_{s \in S} \sum_{a \in A(s)} x_{sa} \sum_{s' \in S} T(s'|s,a) R(s,a,s') \text{ subject to } (x,y) \in Q_0 \text{ and } (iv) \qquad (22)$$

Note that the feasible set of LP$_3$ is the intersection of the set Q_0 in (19) and the set of variables satisfying the steady-state specifications, that is, without the positivity or flow constraints in (20) and (21), respectively.

Recall that a strongly connected digraph is one in which it is possible to reach any node starting from any other node by traversing the directed edges in the directions in which they point. Theorem 5 states that the policy π in (14), derived from an optimal solution to LP$_3$ in (22), solves SSPS(Π_{CPU}) if the directed subgraphs corresponding to the support of the optimal solution in the TSCCs of $\mathcal{M}$ are strongly connected. In Theorem 5, we define the digraph associated with the support of a given solution x as the graph whose vertices are all states s with $x_s > 0$ and whose edges correspond to actions a for which $x_{sa} > 0$.

Theorem 5. *Given LMDP $\mathcal{M}$, let Q^* be the set of optimal solutions of* LP$_3$ (22) *and $X^* := \{x : (x, y) \in Q^* \text{ for some } y\}$. Given $x \in X^*$, let $V_k^+(x) := \{s \in r_k(\mathcal{M}) : x_s > 0\}$ and $E_k^+(x) := \{(s, a) \in r_k(\mathcal{M}) \times A(s) : x_{sa} > 0\}$. If the directed subgraph $(V_k^+(x), E_k^+(x))$ is strongly connected $\forall k \in [m]$, then the policy π in* (14) *is optimal in the class of policies Π_{CPU} and meets the specifications in Φ_L^∞.*

Corollary 1. *If the condition in the statement of Theorem 5 holds for all $x \in X^*$, then* LP$_3$ (22) *solves SSPS*(Π_{CPU})*.*

5.3.1 Generation of policies in Π_{CPU}

Inspired by Theorem 5, we devise a row-generation-based algorithm to search for a policy $\pi \in \Pi_{CPU}$ as shown in Algorithm 1. First, LP$_3$ (22) is solved. If the digraph corresponding to the support of the obtained solution is strongly connected for each of the TSCCs of LMDP $\mathcal{M}$, the policy in (14) is computed and the search stops. However, if the solution does not correspond to a strongly connected digraph in some TSCC, then there must exist a non-empty set of states with no outgoing edges to the rest of the states in that TSCC. Hence, for some $k \in [m]$ we find a cut, that is, a set of states C that has no outgoing edges to the complement set $r_k(\mathcal{M}) \setminus C$. The constraint in (23) corresponding to this cut is added to include the edges in the support, where $A' = \{a \in A(s) : T(s'|s,a) > 0, s' \in r_k(\mathcal{M}) \setminus C\}$. The constraint forces the addition of missing edges across this cut in a greedy manner (by forcing the sum of the state-action variables corresponding to these edges to be non-zero) to eventually produce a strongly connected digraph. The process is repeated until a strongly connected solution is found.

$$\sum_{s \in C} \sum_{a \in A'} x_{sa} > 0 \tag{23}$$

Algorithm 1 is guaranteed to converge to a (possibly suboptimal) policy in Π_{CPU} in a finite number of steps, since in the worst case (when all edges are included) it will yield a policy in $\Pi_{EP} \subseteq \Pi_{CPU}$ under which all edges in the TSCCs of $\mathcal{M}$ are retained. The finiteness of the number of steps is because the number of cuts in the finite MDP is bounded above by $O(\max_{k \in [m]} 2^{|r_k(\mathcal{M})|})$. Our experiments have shown that Algorithm 1 converges to a policy in Π_{CPU} after a small number of iterations.

Algorithm 1 Generation of a policy $\pi \in \Pi_{CPU}$

```
Input: LMDP $\mathcal{M}$ with specifications $\Phi_L^\infty$.
Output: Stationary policy $\pi \in \Pi_{CPU}$ which satisfies $\Phi_L^\infty$.
  Determine the TSCCs $r_k(\mathcal{M}), k \in m$ of $\mathcal{M}$
  $isSConnected = False$, $C = \{.\}$, $A' = \{.\}$
  while $isSConnected = False$ do
    Solve LP (22) with constraint (23) to get optimal values $x^*_{sa}, y^*_{sa}, \forall (s,a) \in S \times A(s)$.
    Compute the support $E_k^+(x^*)$ of each TSCC corresponding to $x^*$ (See Theorem 5).
    if digraph $(V_k^+(x^*), E_k^+(x^*))$ forms a SCC for every $k \in [m]$ then
      compute $\pi$ using (14)
      $isSConnected = True$
    else
      find a cut and update $C$ and $A'$
    end if
  end while
```

5.4 Additional Insights

This section provides additional remarks and examples to shed more light on the linear programming formulations. The section may be skipped without loss of continuity.

5.4.1 Non-Surjective Mapping

All occupation measures induced by the policies of interest are elements of Q_0 (19), that is, $\Pr_\pi^\infty \in X_0 := \{x : (x, y) \in Q_0 \text{ for some } y\}$ if $\pi \in \Pi_{CPU}$, which is affirmed by Lemma 7 stated in Appendix A. However, in general, $\mathcal{P}^\infty(\Pi_{CPU}) \subset X_0$, i.e., the set $\mathcal{P}^\infty(\Pi_{CPU})$ is a *proper* subset of X_0. In mathematical terms, the mapping (9) between the set of policies Π_{CPU} and the set X_0 is injective but non-surjective. In turn, there may exist elements of X_0 that are unpaired with policies in Π_{CPU}. This is illustrated by the following example.

Example 2. *Consider the MDP in Figure 2(b). It is easy to see that x for which $x_{s_2a_2} = x_{s_3a_2} = 1/2$, $x_{s_1a_1} = x_{s_2a_1} = x_{s_3a_1} = 0$ is in X_0, i.e., $x \in X_0$. However, the only policies in Π_{CPU} that satisfy $\Pr_\pi^\infty(s_2, a_2) + \Pr_\pi^\infty(s_3, a_2) = 1$ are the deterministic policies π_1, π_2, which have $\pi_1(a_2|s_2) = 1, \pi_1(a_2|s_3) = 0$ (for which state s_2 is recurrent and s_3 is transient) and $\pi_2(a_2|s_2) = 0, \pi_2(a_2|s_3) = 1$ (for which state s_3 is recurrent and s_2 is transient). However, $\Pr_{\pi_1}^\infty(s_2, a_2) = \Pr_{\pi_2}^\infty(s_3, a_2) = 1$. Thus, $x \notin \mathcal{P}^\infty(\Pi_{CPU})$.*

5.4.2 Insufficient Constraint Set

The set Q_0 correctly encodes constraints on the limiting distributions of Markov chains induced by policies in Π_{CPU} (with the state classification corresponding to $r(\mathcal{M})$ and $\bar{r}(\mathcal{M})$). However, the lack of one-to-one correspondence between feasible solutions and policies (see Section 4.1) is not fully resolved by the constraint set (19) without the additional constraints in (20) or (21). In particular, consider the linear program LP_0 with feasible set Q_0

$$(\text{LP}_0) : \max \quad \sum_{s\in S} \sum_{a\in A(s)} x_{sa} \sum_{s'\in S} T(s'|s, a)R(s, a, s') \text{ subject to } (x, y) \in Q_0 \qquad (24)$$

The steady-state distribution of the policy (14) derived from an optimal solution (x^*, y^*) to LP_0 is generally not equal to x^*. In turn, specifications encoded as constraints on the state-action variables as in (22) will not necessarily be met by the policy. This is best illustrated via a simple example.

Example 3. *Revisit the three-state example of Figure 2(b) and define the rewards $R(s_1, a_1) = R(s_1, a_2) = R(s_2, a_1) = R(s_3, a_1) = 0$, $R(s_2, a_2) = R(s_3, a_2) = 1$ and initial distribution $\beta_{s_1} = 0, \beta_{s_2} = \beta_{s_3} = 1/2$. The MDP has one TSCC such that, $r(\mathcal{M}) = r_1(\mathcal{M}) = \{s_2, s_3\}, \bar{r}(\mathcal{M}) = \{s_1\}$. The solution to LP_0 in (24) which has $x^*_{s_1a_1} = x^*_{s_1a_2} = x^*_{s_2a_1} = x^*_{s_3a_1} = 0, x^*_{s_2a_2} = 1/3, x^*_{s_3a_2} = 2/3, y^*_{s_1a_1} = y^*_{s_1a_2} = y^*_{s_3a_1} = 0, y^*_{s_2a_1} = 1/6$, is optimal (albeit not unique). However, the policy $\pi := \pi(x^*, y^*)$ has $\Pr_\pi^\infty(s_2, a_2) = \Pr_\pi^\infty(s_3, a_2) = 1/2$, hence in general $\Pr_\pi^\infty \neq x^*$.*

5.4.3 Remarks on SSPS(Π_{CPU})

1) Note that, in the previous example, the derived policy $\pi \notin \Pi_{CPU}$. However, if $\pi := \pi(x^*, y^*) \in \Pi_{CPU}$, where (x^*, y^*) is an optimal solution to (22), then π will be optimal over $\Pi_{CPU} \supseteq \Pi_{CP}$ i.e., solves SSPS(Π_{CPU}). This follows from the optimality of (x^*, y^*) and Lemma 6 in the appendix, which gives a sufficient condition for the existence of a one-to-one correspondence between the elements of Q_0 and the steady-state distribution of policy (14).
2) In general, if we dispense with the flow constraints in (21), we have no guarantee that the TSCCs $r_k(\mathcal{M})$ will be unichain in $\mathcal{M}_\pi$ under such π. For example, $\mathcal{M}_\pi$ induced by the

policy π given in Example 3 has $r_1(\mathcal{M}_\pi) = \{2\}, r_2(\mathcal{M}_\pi) = \{3\}$, i.e., $\pi \notin \Pi_{CPU}$. However, if the rewards in this example are modified such that $R(s_1, a_1) \neq R(s_2, a_1)$ while keeping all other rewards unchanged, then $\pi \in \Pi_{CPU}$. Therefore, under certain sufficient conditions on the reward vector, LP_3 in (22) solves $\text{SSPS}(\Pi_{CPU})$.

5.4.4 Existence of Optimal Policies

Modified LP. The feasible set Q_1 for LP_1 is not compact given the strict inequalities of constraint (v) in (20). Therefore, the maximum in (20) may not always be attained on the set. This can be easily remedied by replacing the strict inequalities with bounded ones via introducing an arbitrarily small constant $\epsilon > 0$ on the right-hand side. Even when such requirement is not made explicit, a constant ϵ is dictated by the numerical precision of the LP solvers. We define $\text{LP}_1(\epsilon)$ similar to (20), with constraints (v) replaced with the bounded inequalities in $(v)'$ for some $\epsilon > 0$,

$$\text{LP}_1(\epsilon) \qquad \begin{array}{ll} \max & \displaystyle\sum_{s\in S}\sum_{a\in A(s)} x_{sa}R(s,a) \text{ subject to } (x,y)\in Q_0,\ (iv), \\ (v)' & x_{sa} \geq \epsilon,\ \forall s \in r_k(\mathcal{M}), k\in[m], a\in A(s)\ . \end{array} \tag{25}$$

Theorem 6 stated next is analogous to Theorem 2 with the modified program $\text{LP}_1(\epsilon)$; it establishes that every feasible solution of $\text{LP}_1(\epsilon)$ yields a policy that is in Π_{EP}, and conversely, for every EP policy that meets the steady-state specifications, there exists an $\epsilon > 0$ such that its steady-state distribution is $\text{LP}_1(\epsilon)$-feasible. Moreover, the policy obtained from the optimal solution to $\text{LP}_1(\epsilon)$ solves $\text{SSPS}(\Pi_{EP})$, that is, its expected average reward can be made arbitrarily close to the supremum over the set Π_{EP} as $\epsilon \to 0$.

Theorem 6. *Given an LMDP $\mathcal{M} = (S, A, T, R, \beta, L, \Phi_L^\infty)$ and $\text{LP}_1(\epsilon)$ as in* (25)*, then*

(1) The policy π in (14) *corresponding to a feasible solution of $\text{LP}_1(\epsilon)$ is in Π_{EP}.*

(2) If $\exists\pi \in \Pi_{EP}$ and π meets the specifications Φ_L^∞, then $\exists\epsilon > 0$ such that Pr_π^∞ is a feasible solution of $\text{LP}_1(\epsilon)$.

(3) Let x^*, y^* be an optimal solution to $\text{LP}_1(\epsilon)$ and $\pi^* := \pi(x^*, y^*)$ the corresponding policy in (14)*. Then,*

$$\lim_{\epsilon\to 0}\left(\sup_{\pi\in\Pi_{EP}} R_\pi^\infty(\beta) - R_{\pi^*}^\infty(\beta)\right) = 0 \tag{26}$$

A similar result holds for CP policies if the maximum in (21) cannot be attained over the feasible set by transforming constraints (x)–$(xiii)$ to bounded ones. The generalization is straightforward, thus omitted for brevity.

Compact policy set – policies with bounded support. The foregoing existence issue stems from the open set definition of Π_{EP} in (15), a result of which is that an optimal policy from the set (i.e., one that maximizes the average reward) may not always exist. Therefore, we introduce a slightly modified definition next, in which we force a lower bound on the values a policy assumes on its support, i.e., require that $\pi(a|s) \geq \delta$, for some arbitrarily small constant $\delta > 0$. We formally introduce the definition of the compact set of policies, then state a result analogous to Theorem 2 based on this definition for completeness.

Definition 19. *Given an MDP $\mathcal{M}$ and some small δ, where $0 < \delta < 1/\max_{s \in r(\mathcal{M})} |A(s)|$, we define the set $\Pi_{EP}(\delta) \subset \Pi_{EP}$ of EP policies of bounded support as,*

$$\Pi_{EP}(\delta) = \left\{ \pi \in \Pi_S : r(\mathcal{M}_\pi) = r(\mathcal{M}) \ \wedge \ \pi(a|s) \geq \delta, \forall s \in r(\mathcal{M}), a \in A(s) \right\} . \tag{27}$$

Theorem 7. *Given an LMDP $\mathcal{M} = (S, A, T, R, \beta, L, \Phi_L^\infty)$ and the set $\Pi_{EP}(\delta)$ in (27),*

(1) The policy π in (14) corresponding to a feasible solution of $\mathrm{LP}_1(\delta)$ is in $\Pi_{EP}(\delta)$.

(2) Let x^*, y^* be an optimal solution to $\mathrm{LP}_1(\delta)$ and $R^*(\delta)$ the average reward of the corresponding policy in (14). Then,

$$\lim_{\delta \to 0} \max_{\pi \in \Pi_{EP}(\delta)} R_\pi^\infty(\beta) - R^*(\delta) = 0. \tag{28}$$

According to Theorem 7, every feasible solution to $\mathrm{LP}_1(\delta)$ yields a policy in $\Pi_{EP}(\delta)$. Also, the gap between the optimal expected average reward over the set $\Pi_{EP}(\delta)$ and the optimal reward of $\mathrm{LP}_1(\delta)$ approaches zero as $\delta \to 0$.

6. Extensions

In this section, we explore extensions beyond class-preserving policies, as well as an alternative type of specifications applicable to transient states.

6.1 Beyond Class-Preserving Policies

In this section, we derive an alternative condition given in Theorem 8 under which LP_3 in (22) is guaranteed to yield a stationary policy whose steady-state distribution meets the desired specifications. The policy generated need not be in Π_{CPU}. The proof of Theorem 8 follows from the sufficient and necessary optimality conditions of program (22) (Bertsimas & Tsitsiklis, 1997). The condition is characterized in terms of the rewards vector $R = [R(s,a)], s \in S, a \in A(s)$. First, we introduce the following definition.

Definition 20 (Cone of feasible directions)**.** *The cone $V(x,y)$, where (x,y) is any feasible solution to LP (22), is defined as*

$$V(x,y) := \left\{ \begin{array}{ll} v = (h,z) \in \mathbb{R}^{2|S||A|} : & \\ \sum_{a \in A(s)} h_{sa} = \sum_{s' \in S} \sum_{a \in A(s')} h_{s'a} T(s|s',a), & \forall s \in S, \\ \sum_{a \in A(s)} (h_{sa} + z_{sa}) = \sum_{s' \in S} z_{s'a} T(s|s',a), & \forall s \in S, \\ \sum_{s \in L_i} \sum_{a \in A(s)} h_{sa} \leq 0, & i \in u(x), \\ \sum_{s \in L_j} \sum_{a \in A(s)} h_{sa} \geq 0, & j \in l(x), \\ h_{fa} = 0, & \forall f \in \bar{r}(\mathcal{M}), a \in A(f), \\ h_{sa} \geq 0, & \forall (s,a) \in n(x), \\ z_{sa} \geq 0, & \forall (s,a) \in m(y) \end{array} \right\} \tag{29}$$

where $u(x) := \{i : \sum_{L_i} \sum_{a \in A(s)} x_{sa} = u_i\}, l(x) := \{j : \sum_{L_j} \sum_{a \in A(s)} x_{sa} = l_j\}, n(x) := \{(s,a) \in r(\mathcal{M}) \times A(s) : x_{sa} = 0\}, m(y) := \{(s,a) \in S \times A(s) : y_{sa} = 0\}$.

Theorem 8. *Given LMDP $\mathcal{M}$, let (x, y) be a feasible solution of* (22)*. If $R = [R(s,a)], s \in S, a \in A$ is an interior point of the dual cone*

$$V^*(x,y) := \{u \in \mathbb{R}^{2|S||A|} : \ \langle u, v \rangle \leq 0, \ \textit{for every } v \in V(x,y)\} ,$$

where $\langle , \rangle$ denotes the inner product, then the policy π in (14) *meets the specifications Φ_L^∞. Further, π is the unique optimal policy in the class of policies for which $\bar{r}(\mathcal{M}) \subseteq \bar{r}(\mathcal{M}_\pi)$.*

We remark that the policy could be outside of Π_{CPU}, but preserves the transience (or isolation) of the states in $\bar{r}(\mathcal{M})$. While the statement of Theorem 8 imposes a conservative assumption on the rewards vector which may be generally hard to verify, it opens up possibilities for further research on steady-state planning over larger sets of policies (beyond Π_{EP}, Π_{CP} and Π_{CPU} considered in this paper) – in this case, sets of policies that preserve the transience of $\bar{r}(\mathcal{M})$. Ultimately, one would hope to tackle SSPS(Π) for arbitrary sets of stationary policies Π. These are directions for future investigation.

6.2 Transient Specifications

In Definition 12, we introduced specifications on the steady-state distribution. However, such specifications are only useful in the recurrent sets where states are visited infinitely often. A transient state $f \in \bar{r}(\mathcal{M}_\pi)$ on the other hand will only be visited a finite number of times, i.e., $\Pr_\pi^\infty(f) = 0$ for any stationary policy $\pi \in \Pi_S$. In this section, we present an alternative specification type which can be applied to transient states.

We first describe a suitable property of transient states against which specifications can be applied. We then define transient specifications based on this property.

Definition 21 (Expected number of visits (Kemeny & Snell, 1963))**.** *Given an MDP $\mathcal{M}$ and policy $\pi \in \Pi_S$, the expected total number of times that state $f \in \bar{r}(\mathcal{M}_\pi)$ is visited under policy π is*

$$\zeta_\pi(f) = \beta_{\bar{r}(\mathcal{M}_\pi)}^T (I - Z_\pi)^{-1} e_f \ . \tag{30}$$

Definition 22 (Transient specification)**.** *Given an MDP and a set of labels $L = \{L_1, \ldots, L_{n_L}\}$, where $L_i \subseteq \bar{r}(\mathcal{M})$, a set of transient specifications is given by $\Phi_L^{tr} = \{(L_i, [l_i, u_i])\}_{i=1}^{n_L}$. Given a policy π, the specification $(L_i, [l_i, u_i]) \in \Phi_L^{tr}$ is satisfied if and only if $\sum_{f \in L_i} \zeta_\pi(f) \in [l_i, u_i]$; that is, if the expected number of visits to transient states $f \in L_i$ in the Markov chain $\mathcal{M}_\pi$ falls within the interval $[l_i, u_i]$.*

Suppose that we have a set of labels L^{tr} over transient states, and a set of transient specifications $\Phi_{L^{tr}}^{tr}$. We can augment the LMDP found in Definition 12 to incorporate these transient specifications as follows. Let L^∞ be the set of steady-state labels, and let $\Phi_{L^\infty}^\infty$ be corresponding steady-state specifications. We define a complete set of labels $L = (L^\infty, L^{tr})$ and specifications $\Phi_L = (\Phi_{L^\infty}^\infty, \Phi_{L^{tr}}^{tr})$, and define an LMDP as $\mathcal{M} = (S, A, T, R, \beta, L, \Phi_L)$.

Our next result regarding y and ζ_π for the transient states gives a sufficient condition for the policy to meet the transient specification.

Proposition 1. *Given an MDP $\mathcal{M}$, let $(x, y) \in Q_0$ and π as in* (14)*, where Q_0 is defined in* (19)*. If $\pi \in \Pi_{CPU}$, then $y_f = \zeta_\pi(f)$ for any state $f \in \bar{r}(\mathcal{M})$.*

We remark that this is analogous to Lemma 6 stated in the appendix, which establishes that if we have a point $(x, y) \in Q_0$ for which (14) yields a CPU policy, then $\Pr^\infty_\pi = x$.

Given this characterization, we can augment LP$_1$ (20) and LP$_2$ (21) with constraint (xv) to synthesize policies subject to transient specifications.

$$(xv) \quad l_i \leq \sum_{s \in L_i} \sum_{a \in A(s)} y_{sa} \leq u_i, \ \forall (L_i, [l_i, u_i]) \in \Phi^{tr}_{L^{tr}} \tag{31}$$

7. Numerical Results

In this section, we present a set of numerical results to corroborate the findings of the theoretical analysis. In Section 7.1, we verify the steady-state behavior of the policies derived from the proposed LPs using the Frozen Islands example of Figure 3, followed by a study of their behavior in the presence of additional transient specifications in Section 7.2. In Section 7.3, we present the results of a study which shows that the empirical steady-state distributions and average number of state visitations induced by the derived policies converge to values that meet the desired specifications. In Section 7.4, we evaluate the average reward achieved by said policies and examine the impact of various restrictions in their respective LPs on the optimal values of the objective using the Toll Collector example of Figure 13. A case study is also presented featuring the progress of the iterative Algorithm 1 for generating a CPU policy. A natural generalization of the specifications to the product space of state-action pairs is presented in Section 7.5. We present two numerical experiments to support the theoretical findings of Section 5.4.4 in Section 7.6. Finally, we examine the scalability of the proposed formulations in Section 7.7, where we present the runtime results for problems with increasing size conducted in various environments.

7.1 Steady-State Specifications

In this section, we demonstrate the correct-by-construction behavior of the policies proposed. As an illustrative example, we first examine the behavior of a policy in Π_{EP} using the Frozen Island example shown in Figure 3. We run our proposed LP$_1$ (20) to calculate the steady-state distribution $\Pr^\infty_\pi(s)$, and show the values for the two TSCCs (the two small islands) in Figure 6.

The heat map gives insight into the means by which the agent satisfies the specifications. After the agent enters an island, it spends a large amount of time in states s_{33}, s_{36}, s_{48}, s_{49}, s_{61}, and s_{64}, in the sense of asymptotic frequency of visits as given by $\Pr^\infty_\pi(s)$. The agent also frequently visits states s_{36} and s_{61} to satisfy the steady-state specifications $(L_{\text{log1}}, [0.25, 1])$ and $(L_{\text{log2}}, [0.25, 1])$, respectively. Likewise, to meet specifications $(L_{\text{canoe1}}, [0.05, 1.0])$, $(L_{\text{canoe2}}, [0.05, 1.0])$ $(L_{\text{fish1}}, [0.1, 1.0])$, $(L_{\text{fish2}}, [0.1, 1.0])$ the agent often visits states s_{33}, s_{49}, s_{48}, and s_{64}, respectively. In addition to visiting the aforementioned states to satisfy the constraints, the agent also visits state s_{48} over 25% of the time to maximize its expected reward (recall that $R(\cdot, \cdot, s_{48}) = R(\cdot, \cdot, s_{64}) = 1$).

The right three plots of Figure 7 show the values of $\Pr^\infty_\pi(s)$ along with the optimal values x^*_s obtained from LP$_1$ (20) for SSPS(Π_{EP}), LP$_2$ for SSPS(Π_{CP}), and Algorithm 1 for

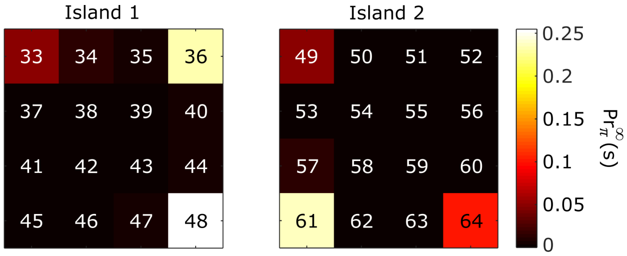

Figure 6: Heat maps showing the steady-state probabilities $\Pr_\pi^\infty(s)$ for states $s \in r(\mathcal{M})$ belonging to the two TSCCs of the Frozen Lakes example in Figure 3.

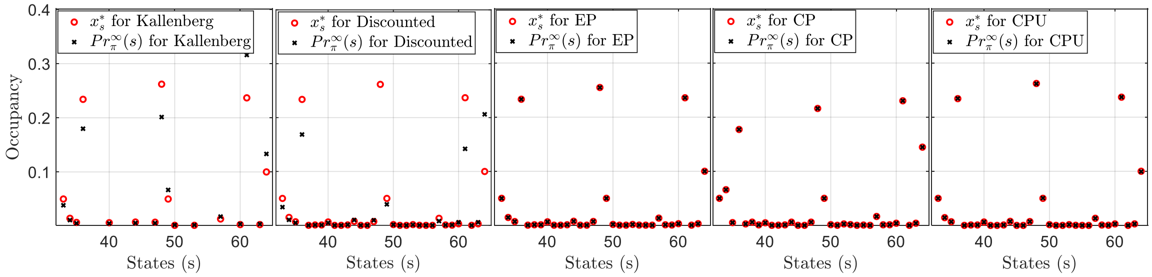

Figure 7: Example showing that $\Pr_\pi^\infty(s) = x_s^*, s \in r(\mathcal{M})$ for policies in Π_{EP}, Π_{CP} and Π_{CPU} derived from the proposed LP_1, LP_2 and Algorithm 1, but not for Kallenberg's and the discounted case (discount factor $\gamma = 0.9999$).

SSPS(Π_{CPU}). In each of these, the steady-state distribution matches the one estimated by the LP for every state. This in fact holds for all state-action pairs as well, i.e., $\Pr^\infty = x^*$. This condition is essential to the proof of Theorems 2, 4 and 5 and ensures that the policy is both optimal and satisfies the steady-state specifications. We calculate the policy corresponding to the optimal solution of LP (4.7.6) by Kallenberg (1983) given in (13) with the additional specification constraints for comparison. As shown in Figure 7 (*left*), the derived policy fails to give a steady-state distribution equal to x^*. In addition, we obtain a policy from the solution to LP (3.5) by Altman (1999) of a discounted reward MDP (with the additional specification constraints) using a discount factor $\gamma = 0.9999$. As observed in the second from the left plot of Figure 7, the steady-state distribution of the derived policy does not match x^*. In Table 1, we show the ramifications when $\Pr_\pi^\infty \neq x^*$. For each specification $(L_i, [l_i, u_i]) \in \Phi_L^\infty$, Table 1 shows the values of $e^\top x_{L_i}^* := \sum_{s \in L_i} x_s^*$ and $\Pr_\pi^\infty(L_i) := \sum_{s \in L_i} \Pr_\pi^\infty(s)$, demonstrating that all of the specifications are met for the proposed methods. For Kallenberg's and the discounted formulations, however, although $x_{L_{\text{log1}}}^*$ and $x_{L_{\text{canoe1}}}^*$ satisfy the specification, the policy yields steady-state distributions $\Pr_\pi^\infty(L_{\text{log1}})$ and $\Pr_\pi^\infty(L_{\text{canoe1}})$ which violate the specifications (these violations are highlighted with bold red text). In other words, $e^\top x_{L_{\text{canoe1}}}^* \neq \Pr_\pi^\infty(L_{\text{canoe1}})$ and $e^\top x_{L_{\text{log1}}}^* \neq \Pr_\pi^\infty(L_{\text{log1}})$ for the Kallenberg and discounted formulations. The table also shows the optimal reward R^* given by our proposed methods, as well as the expected average reward yielded by the policy, i.e., $R_\pi^\infty := \sum_{s \in S} \sum_{a \in A(s)} \Pr_\pi^\infty(s, a) R(s, a)$. While R^* obtained by Kallenberg's formulation is larger than that of the EP and CP methods, the proposed LPs produce policies

Method	**SS Specifications**												**Rewards**	
	Logs (≥ 0.25)				**Canoes (≥ 0.05)**				**Fish Rods (≥ 0.1)**					
	Island 1		**Island 2**		**Island 1**		**Island 2**		**Island 1**		**Island 2**		**$\mathbf{R}^*$**	**$\mathbf{R}_\pi^\infty$**
	$\boldsymbol{x}^*$	$\mathbf{Pr}^\infty$	$\boldsymbol{x}^*$	$\mathbf{Pr}^\infty$	$\boldsymbol{x}^*$	$\mathbf{Pr}^\infty$	$\boldsymbol{x}^*$	$\mathbf{Pr}^\infty$	$\boldsymbol{x}^*$	$\mathbf{Pr}^\infty$	$\boldsymbol{x}^*$	$\mathbf{Pr}^\infty$		
CPU	0.25	0.25	0.25	0.25	0.05	0.05	0.05	0.05	0.26	0.26	0.10	0.10	0.3621	0.3621
CP	0.26	0.26	0.25	0.25	0.05	0.05	0.05	0.05	0.21	0.21	0.14	0.14	0.3605	0.3605
EP	0.25	0.25	0.25	0.25	0.05	0.05	0.05	0.05	0.25	0.25	0.10	0.10	0.3547	0.3547
Kallenberg	0.25	**0.17**	0.25	0.36	0.05	**0.04**	0.05	0.07	0.26	0.19	0.10	0.14	0.3621	**0.3278**
Discounted ($\gamma = 0.999$)	0.25	**0.04**	0.25	**0**	0.05	**0.0037**	0.05	**0.0013**	0.26	0.52	0.10	0.39	0.3576	**0.9061**
Discounted ($\gamma = 0.9999$)	0.25	**0.18**	0.25	**0.15**	0.05	**0.03**	0.05	**0.04**	0.26	0.35	0.10	0.21	0.3617	**0.5530**

Table 1: Steady-state specification comparison. Bold red text indicates violated steady-state specifications. Constraints are specified in the header for each label type.

which yield larger values of R^∞. Additionally, as the discount factor γ approaches 1, the discounted reward does not converge to the expected reward. The policies obtained from the discounted reward formulation achieve larger rewards R^∞ by violating the steady-state constraints and spending larger proportions of time in the rewarding fishing sites.

7.2 Synthesis for Transient Specifications

In this section, we demonstrate the behavior of the policies derived subject to transient specifications following the framework described in Section 6.2. We again compare the policies derived from our proposed formulations to that of Kallenberg with regard to meeting such specifications for the Frozen Islands example of Figure 3. The labels over states in $\bar{r}(\mathcal{M})$ are set to $L_{\text{tools}} = \{s_7, s_{13}, s_{23}\}$, $L_{\text{gas}} = \{s_{10}, s_{16}\}$, and $L_{\text{supplies}} = \{s_2, s_{15}, s_{29}\}$ as shown in Figure 9 (*left*). The agent sets out to collect some tools, fill up enough gas, and pick up the required fishing supplies before transitioning to one of the smaller islands which correspond to TSCCs. This is reflected in the transient specifications $(L_{\text{tools}}, [10, N_{\text{tr}}])$, $(L_{\text{gas}}, [12, N_{\text{tr}}])$, $(L_{\text{supplies}}, [15, N_{\text{tr}}]) \in \Phi_L^{tr}$. These specifications bound the expected total number of visitations to certain states in $\bar{r}(\mathcal{M})$, where $N_{\text{tr}} = 200$. Figure 8 presents the values of the expected total number of times a state $s \in \bar{r}(\mathcal{M})$ is visited under policy π, denoted by $\zeta_\pi(s)$, along with the optimal values y_s^*, obtained from Kallenberg's LP, LP$_1$ (20), LP$_2$ (21), and Algorithm 1. As shown, the results match the expected number of visitations for the proposed methods for every state.

For each transient specification $(L_i, [l_i, u_i]) \in \Phi_L^{tr}$, Table 2 shows $e^\top y_{L_i}^* := \sum_{f \in L_i} y_f^*$ and the expected total number of visitations achieved by the policy in corresponding states $\zeta_\pi(L_i) := \sum_{f \in L_i} \zeta_\pi(f)$. As shown, LP$_1$, LP$_2$ and Algorithm 1 yield policies that satisfy the given specifications, while the policy derived from the Kallenberg LP does not. The last column shows the expected total number of visitations achieved by the policy on the larger (transient) island, where $\zeta_\pi(\bar{r}(\mathcal{M})) = \sum_{f \in \bar{r}(\mathcal{M})} \zeta_\pi(f)$.

Figure 9 (*right*) shows a heat map for the expected number of visits to the transient states, i.e. the large island. The policy is calculated using LP$_1$ (20). In addition to the constraints $(L_{\text{tools}}, [10, N_{\text{tr}}])$, $(L_{\text{gas}}, [12, N_{\text{tr}}])$, and $(L_{\text{supplies}}, [15, N_{\text{tr}}])$, we also add the constraint $(\bar{r}(\mathcal{M}) \setminus (L_{\text{tools}} \cup L_{\text{gas}} \cup L_{\text{supplies}}), [0, 10])$ to reduce the amount of time spent in transient states with no resources. As shown, the agent meets the specifications largely by visiting states s_{13}, s_{16}, and s_{29} to collect tools, gas, and supplies, respectively.

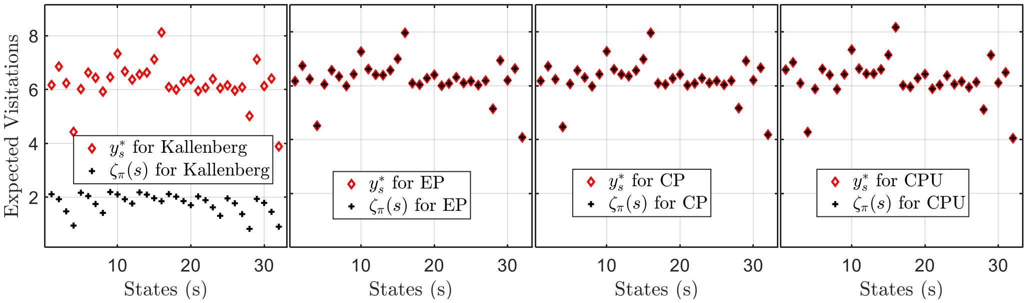

Figure 8: Example showing that $\zeta_\pi(s) = y_s^*, s \in \bar{r}(\mathcal{M})$ for the proposed methods, but not for Kallenberg's formulation.

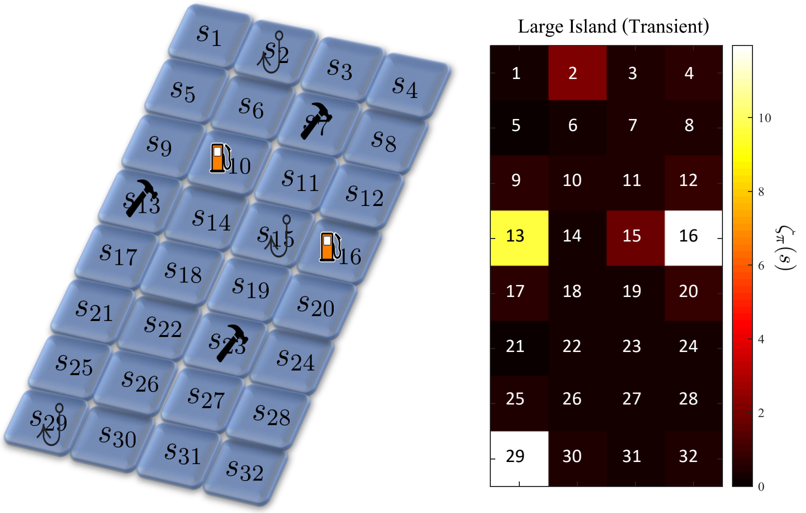

Figure 9: Distribution of labels on the large island, $L_{\text{tools}} = \{s_7, s_{13}, s_{23}\}$, $L_{\text{gas}} = \{s_{10}, s_{16}\}$, and $L_{\text{supplies}} = \{s_2, s_{15}, s_{29}\}$ (*left*). Heat map showing the expected number of visits $\zeta_\pi(s)$ for states $s \in \bar{r}(\mathcal{M})$, i.e., the states belonging to the large island in Figure 3 (*left*).

	Transient Specifications ($N_{\text{tr}} = 200$)						**Results**	
Method	**Tools ($\geq$ 10)**		**Gas ($\geq$ 12)**		**Supplies ($\geq$ 15)**			
	$\boldsymbol{y}^*$	ζ_π	$\boldsymbol{y}^*$	ζ_π	$\boldsymbol{y}^*$	ζ_π	$\mathbf{R}^*$	$\zeta_\pi(\bar{r}(\mathcal{M}))$
CPU	19.22	19.22	15.51	15.51	21.15	21.15	0.3621	200
CP	18.97	18.97	15.26	15.26	20.67	20.67	0.3607	200
EP	19.32	19.32	15.51	15.51	21.15	21.15	0.3547	200
Kallenberg	19.34	**5.55**	15.53	**3.95**	21.17	**5.82**	0.3621	**56.5**

Table 2: Bold red text indicates violated transient specifications. Constraints are specified in the header for each label type.

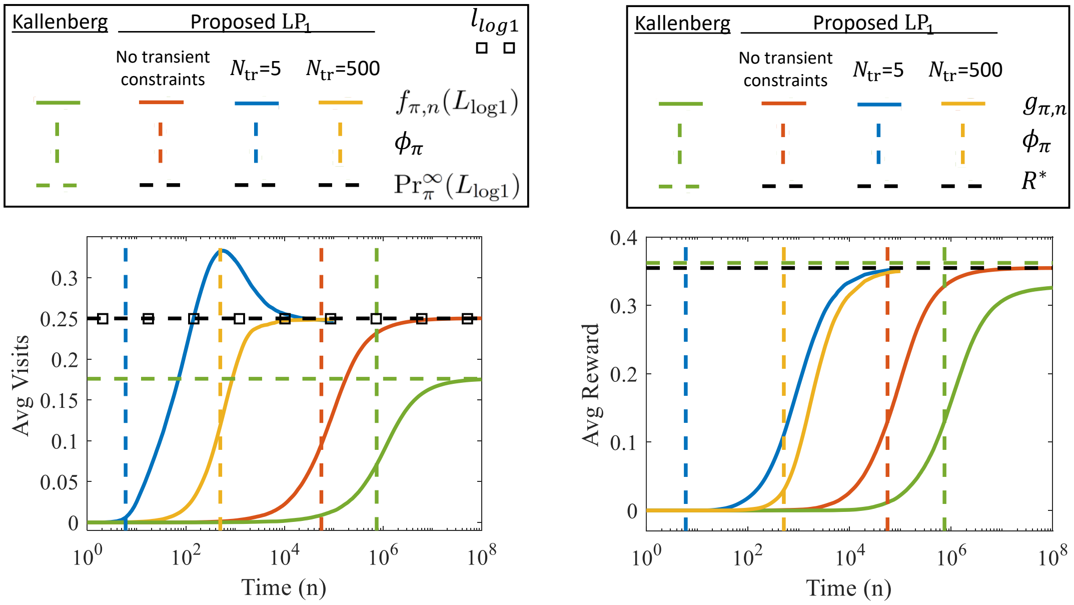

Figure 10: Execution of policy, showing (*left*) average visits and (*right*) average reward up to time n.

7.3 Empirical Study

In this section, we simulate the policies derived from our LPs to show the validity of our formulations and to further demonstrate the failure of the Kallenberg formulation to yield optimal rewards and meet specifications.

Let S_t and A_t denote the state and action, respectively, of the Frozen Island example at time t assuming policy π and initial distribution β. The average number of visits $f_{\pi,n}$ and average reward $g_{\pi,n}$ up to time n are defined as

$$f_{\pi,n}(L) = \frac{1}{n}\sum_{t=1}^{n} \mathbb{1}_L(S_t), \quad \mathbb{1}_L(s) = \begin{cases} 1 & s \in L \\ 0 & s \notin L \end{cases} \tag{32}$$

$$g_{\pi,n} = \frac{1}{n}\sum_{t=1}^{n} R(S_t, A_t, S_{t+1}). \tag{33}$$

We take an ensemble average over 5000 paths.

First, we solve LP (4.7.6) of Kallenberg (1983). In Figure 10 (*left*), the solid green line shows the average number of visits to the states in $L_{\log 1} = \{s_{34}, s_{36}, s_{38}, s_{43}\}$, and the horizontal dashed green line indicates the steady-state distribution. The square markers show the lower bound of the specification on the logs. While the value of $f_{\pi,n}(L_{\log 1})$ converges to the steady-state distribution, the policy fails to meet the steady-state specification. This follows from the fact that $\Pr_\pi^\infty \neq x^*$.

Next, we produce an EP policy by executing LP$_1$ (20), using no transient specifications. The average number of visits to the states in $L_{\log 1}$ is shown as a solid red line in Figure 10 (*left*). Not only does the average number of visits converge to the corresponding steady-state distribution (dashed black line), but the specification is met. We observe similar results for CP and CPU policies as well.

In Figure 10 (*right*), the solid green and red lines show the average reward for the Kallenberg LP and our proposed LP_1, respectively. The dashed green line indicates R^* for Kallenberg's formulation. As can be seen, for similar reasons as before, the average reward converges to a reward other than that output by the LP. On the other hand, LP_1 converges to the corresponding LP reward R^* (black dashed line).

The vertical green and red dashed lines in Figure 10 indicate the average time of entry into $r(\mathcal{M})$ for LP_1 and Kallenberg's LP, respectively, where the time of entry ϕ_π is given by

$$\phi_\pi = \min\{n \mid S_n \in r(\mathcal{M})\}. \tag{34}$$

In both cases, the agent spends an unduly amount of time in the transient states before transitioning to a recurrent set, which may be undesirable. To reduce the amount of time spent in the transient states, we next introduce a transient specification $(\bar{r}(\mathcal{M}), [0, N_{\text{tr}}])$ and rerun LP_1. The constant N_{tr} is used to control the time of entry into the recurrent sets. The results are shown for $N_{\text{tr}} = 5$ (blue lines) and $N_{\text{tr}} = 500$ (yellow lines). In both cases, convergence of the average visits to $\text{Pr}^\infty_\pi(L_{\text{log1}})$ occurs at a much faster rate, leading to a much faster accumulation of reward.

We now comment further on the simulation for $N_{\text{tr}} = 5$. The policy produced by LP_1 separates the first small island into two main subsets. The agent tends to visit state s_{33} repeatedly after entering the first small island, leading to an above average number of visits to log1 states. This results in an initial "overshoot" of $\text{Pr}^\infty_\pi(L_{\text{log1}})$. This effect is not seen for $N_{\text{tr}} = 500$ due to the averaging effect of $f_{\pi,n}(L)$. Likewise, the policy tends to delay the entry of the agent into state s_{48} where rewards are accumulated. This delay is especially noticeable in $g_{\pi,n}$ for $N_{\text{tr}} = 5$ due to the logarithmic time scale.

In the same vein, we explore the simulated behavior of our policies in terms of the number of visits to transient states, where the number of visits $h_{\pi,n}(L)$ to states in $L \subseteq \bar{r}(\mathcal{M})$ up to time n is defined as

$$h_{\pi,n}(L) = \sum_{t=1}^{n} \mathbb{1}_L(S_t). \tag{35}$$

In Figure 11, we show the number of visits to the transient states for the same policies as shown in Figure 10. For the policies produced by LP_1, $h_{\pi,n}(\bar{r}(\mathcal{M}))$ converges to the optimal $y^*_{\bar{r}(\mathcal{M})}$. On the other hand, as described in Section 7.2, for the Kallenberg formulation we have $\zeta_\pi \neq y^*$ and so the derived policy fails to converge to $y^*_{\bar{r}(\mathcal{M})}$.

7.4 Comparison of Policies

Recall that policies in Π_{EP} exercise all transitions in the TSCCs of an MDP. By contrast, policies in Π_{CP} and Π_{CPU} are less restrictive in that they only preserve the state classification and the unichain property of these components, respectively. In turn, they often yield larger expected rewards while simultaneously satisfying desired specifications. In this section, we verify the correctness of such policies and compare their optimal rewards.

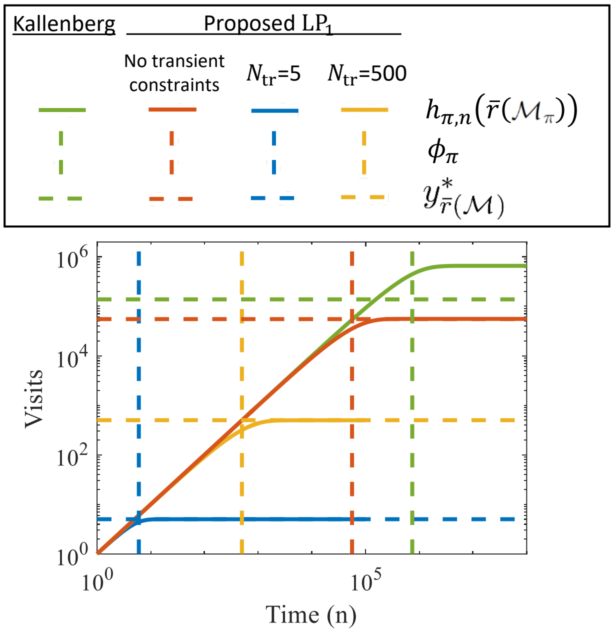

Figure 11: Execution of policy with transient specifications showing the number of visits to transient states up to time n.

Figure 12 illustrates the Markov chains induced by policies in Π_{EP}, Π_{CP}, and Π_{CPU}, respectively, for the MDP shown in its first column. The policies are obtained from the optimal solutions of the corresponding LPs. As observed, the Markov chain induced by the EP policy (Column 2) contains all transitions in the TSCCs of the underlying MDP. The self loops of states s_3,s_5, and s_9 are missing in the Markov chain induced by the CP policy without affecting the recurrence of each of the TSCCs. In the case of the Markov chain induced by the CPU policy, the state s_3 is transient in the TSCC $\{s_3, s_4, s_5\}$, but all TSCCs of $\mathcal{M}$ remain unichain in the induced Markov chain. That is, each TSCC contains exactly one recurrent component.

In order to compare the performance of our EP, CP, and CPU policies, we define the Toll Collector example given by the LMDP $\mathcal{M}$ of Figure 13. In this problem, an agent must choose one of m cities to visit, each of which corresponds to a TSCC of $\mathcal{M}$. The k-th city consists of $n_k, k \in [m]$ counties represented as vertices and roads connecting these counties represented by edges. The roads with toll booths yield a positive reward for collecting a toll. However, the agent needs to spend some time on roads without toll booths in order to build them. We consider an instance of the Toll Collector problem for which $m = 3$ and the number of states per TSCC $n_k = n, \forall k$. To highlight the gap between the optimal rewards of the different policies, we define the labels $L_k = \{s \in r_k(\mathcal{M}) : R(s, a, s') = 0, \forall a \in A(s), s' \in r_k(\mathcal{M})\}$ and $\Phi_L^\infty = (L_k, [l, 1])$, respectively, $\forall k \in [m]$. As such, per Φ_L^∞, the steady-state probability of states with no rewardful transitions is forced to be bounded below by l. We will use this steady-state specification with various values of l to show that, for lower values of l, there is a significant gap in expected rewards observed by the various policies. As this l value is increased, the gap can be shown to diminish.

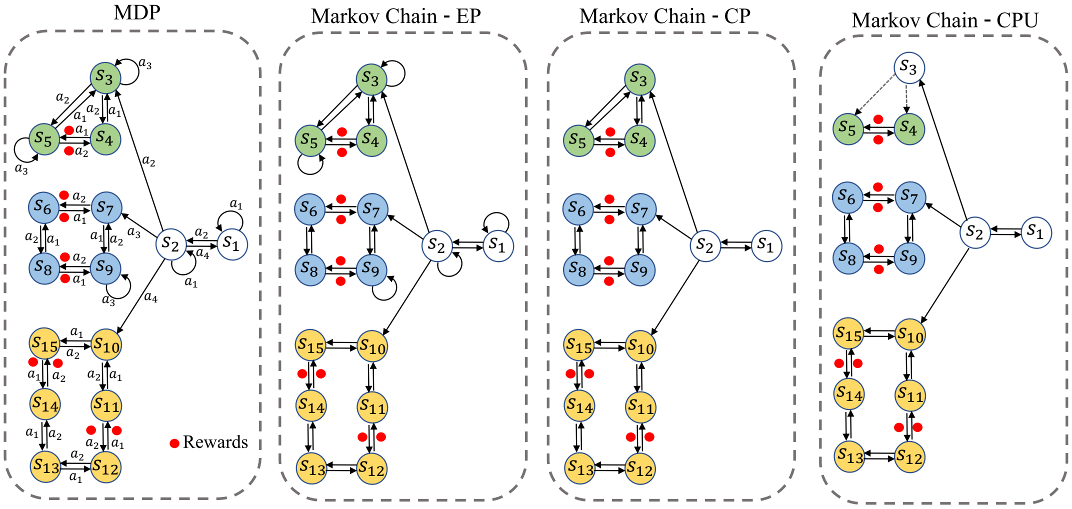

Figure 12: Markov chains induced by the EP, CP, and CPU policies. The first column shows the underlying MDP $\mathcal{M} = (S, A, T, R, \beta)$. Transitions designated with circles have unit reward, otherwise the reward is 0. The initial distribution β is uniform over S.

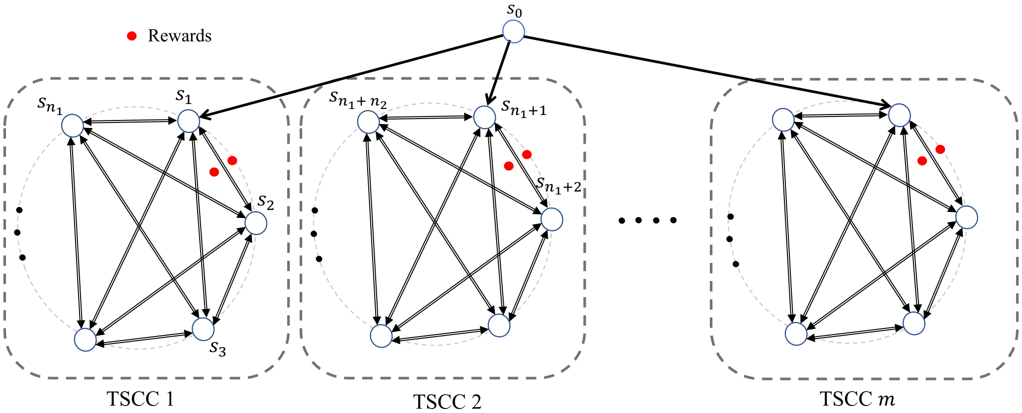

Figure 13: Toll Collector problem given by LMDP $\mathcal{M} = (S, A, T, R, \beta, L, \Phi_L^\infty)$ consisting of m fully-connected TSCCs $r_k(\mathcal{M}), k \in [m]$ and $\bar{r}(\mathcal{M}) = \{s_0\}$. The k-th TSCC consists of n_k states. State s_0 has m actions, each of which leads to one of the m TSCCs with probability 1. For each state s_i in $r_k(\mathcal{M})$, there are $n_k - 1$ actions, each of which causes a transition to another state in $r_k(\mathcal{M})$ with probability 1. The reward function is defined such that, in each TSCC, there is a positive reward by taking the action that leads from some state s_i to its neighbor s_{i+1} and vice-versa. That is, $R(s_i, \cdot, s_{i+1}) = R(s_{i+1}, \cdot, s_i) = 1$ for some i. These rewards are designated with red solid circles in each TSCC. All other rewards are 0. The initial distribution β is uniform over S. The labels and steady-state specifications are given by $L_k = \{s \in r_k(\mathcal{M}) : R(s, a, s') = 0, \forall a \in A(s), s' \in r_k(\mathcal{M})\}$ and $\Phi_L^\infty = (L_k, [l, 1])$, respectively, for all $k \in [m]$.

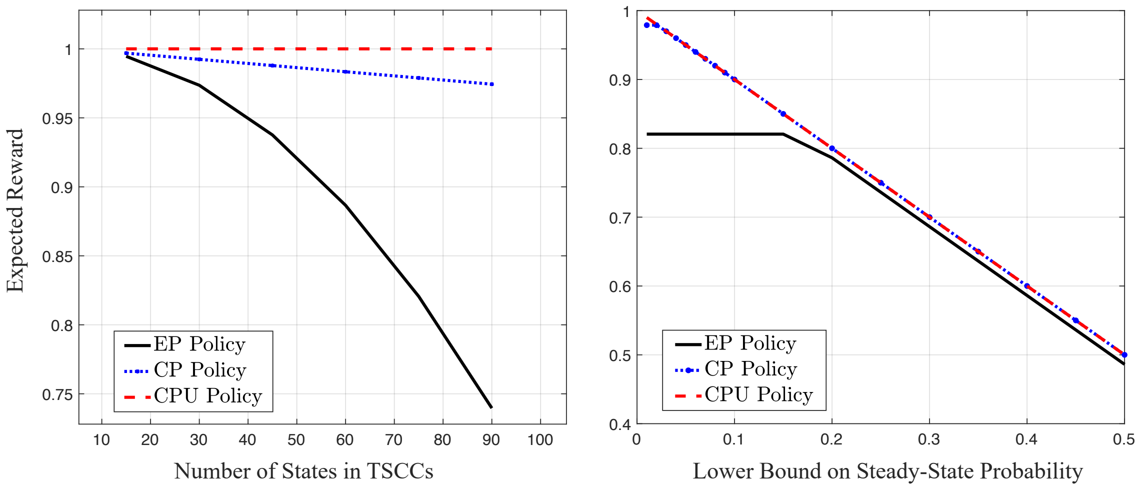

Figure 14: Comparison of EP, CP, and CPU policies for the Toll Collector example. (Left) Expected reward as function of the total number of states in each TSCC when $l = 0$. (Right) Expected reward as function of the lower bound l on the steady-state probability when $n = 25$.

Figure 14 (left) compares the optimal rewards achieved by the different policies as a function of the total number of states in the TSCCs (i.e., $3n$) when $l = 0$. As the number of states increases, the gap between the average reward of the EP policies and their CP and CPU counterparts increases. In this scenario, the EP policies incur a quadratic loss relative to CPU policies since they are forced to exercise all existing transitions equally and there are $O(n_k^2)$ such transitions in the k-th TSCC. On the other hand, an optimal CPU policy preserves the unichain property while exercising exclusively the two transitions with positive reward in each TSCC. A smaller loss is incurred by CP policies since they are only required to preserve the recurrence of the TSCCs and can thus restrict themselves to visiting the outer perimeter of each TSCC. In doing so, the CP policies incur a linear loss when compared to the CPU policies because they must visit every state in a TSCC infinitely often in order to preserve the recurrent classification of these states.

Figure 14 (right) illustrates the average rewards as a function of the lower bound l for the three types of policies when the number of states in each TSCC is $n = 25$. When l increases, the average reward gap between the different policies diminishes since the agent has to spend more time in states with no rewards to meet the desired specifications.

7.4.1 Operation of Algorithm 1

In this subsection, we present in detail the operation of the proposed Algorithm 1 to generate a CPU policy. Consider the LMDP given in Figure 15. For each iteration, LP (22) is solved and the digraph of the support of the solution in each TSCC is shown (colored nodes). In the first iteration, for the first TSCC, states s_4 and s_5 form a SCC, while s_3 does not belong to the support of the solution. For both the second and third TSCCs, all respective states belong to the support but they do not form a SCC, thus we can find cut(s) (as can be seen at the bottom of the figure of the second iteration). In the second iteration, the dotted edges are added which results in one SCC for the second TSCC (no additional constraints are needed) but not for the third TSCC. Thus, we consider additional cuts as shown at the bottom of the figure of the third iteration. In the last iteration, the stopping criteria is met (the digraph in each of the three TSCCs is strongly connected). The final Markov chain induced by the CPU policy derived from the solution to the LP of the third iteration is shown on the right side of Figure 15. States $\{s_3, s_4, s_5\}$ form a unichain component, states $\{s_6, s_7, s_8, s_9\}$ form a recurrent component, and states $\{s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}\}$ belong to a TSCC where all edges are preserved.

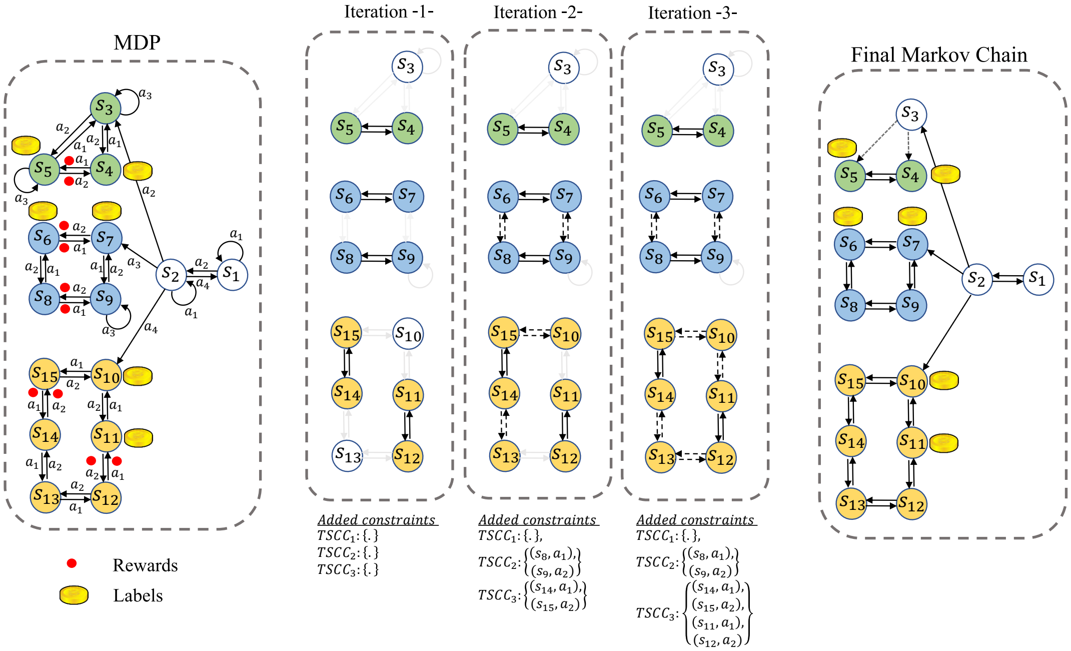
MDP
Iteration -1-
Iteration -2-
Iteration -3-
Final Markov Chain
s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_8 s_9 s_{10} s_{11} s_{12} s_{13} s_{14} s_{15}
a_1 a_2 a_3 a_4
Added constraints
$TSCC_1$: {.}
$TSCC_2$: {.}
$TSCC_3$: {.}
Added constraints
$TSCC_1$: {.},
$TSCC_2$: $\{(s_8, a_1), (s_9, a_2)\}$
$TSCC_3$: $\{(s_{14}, a_1), (s_{15}, a_2)\}$
Added constraints
$TSCC_1$: {.},
$TSCC_2$: $\{(s_8, a_1), (s_9, a_2)\}$
$TSCC_3$: $\{(s_{14}, a_1), (s_{15}, a_2), (s_{11}, a_1), (s_{12}, a_2)\}$
● Rewards
Labels

Figure 15: Illustration of the progress of Algorithm 1 for generating a policy in Π_{CPU}. The LMDP $\mathcal{M} = (S, A, T, R, \beta, L, \Phi_L^\infty)$, where S, A, and R are given in the MDP (first column) and β is uniform. labels are $L_{\text{gold1}} = \{s_4, s_5\}$, $L_{\text{gold2}} = \{s_6, s_7\}$, $L_{\text{gold3}} = \{s_{10}, s_{11}\}$, and the steady-state specifications are $(L_{\text{gold1}}, [0.20, 1])$,$(L_{\text{gold2}}, [0.10, 1])$, $(L_{\text{gold3}}, [0.15, 1])$ (first column). In each iteration, we illustrate the support of the optimal solution for the TSCCs of $\mathcal{M}$ (middle columns) along with the edges (state-action pairs) for a given cut. After three iterations, the support of the optimal solution corresponds to SCCs in each of the TSCCs. Every TSCC of $\mathcal{M}$ is a unichain component in the Markov chain $\mathcal{M}_\pi$ induced by the resulting policy (last column).

7.5 Specifications on State-Action Pairs

Up to this point, we have defined steady-state and transient specifications over states. However, the framework proposed can be used to synthesize policies with provably correct behavior on the level of state-action pairs as well. As an example, consider the LMDP

Policy	**Specifications**				
	Steady-State			**Transient**	
	$\Pr^{\infty}_{\pi}(s_4, a_1)$	$\Pr^{\infty}_{\pi}(s_6, a_2)$	$\Pr^{\infty}_{\pi}(s_{10}, a_1)$	$\zeta_\pi(s_2, a_1)$	$\zeta_\pi(\bar{r}(\mathcal{M}))$
EP	0.11	0.12	0.20	26.2	50
CP	0.10	0.12	0.20	31.03	50
CPU	0.11	0.12	0.20	28.82	50

Table 3: The policies meet transient and steady-state specifications on state-action pairs for the MDP defined in Figure 15.

$\mathcal{M} = (S, A, T, R, \beta, L, \Phi_L)$ defined in Figure 15 (*left*). We define labels $L = (L^\infty, L^{tr})$ over state-action pairs, i.e., $L^{tr}_{\text{tool}} = \{(s_2, a_1)\}$, and $L^\infty_{\text{gold1}} = \{(s_4, a_1)\}$, $L^\infty_{\text{gold2}} = \{(s_6, a_2)\}$, and $L^\infty_{\text{gold3}} = \{(s_{10}, a_1)\}$. The specifications are given as $\Phi_L = (\Phi^\infty_{L^\infty}, \Phi^{tr}_{L^{tr}})$, where the steady-state specifications are $(L^\infty_{\text{gold1}}, [0.10, 1])$, $(L^\infty_{\text{gold2}}, [0.12, 1])$, and $(L^\infty_{\text{gold3}}, [0.20, 1])$, and the transient specifications are given as $(L^{tr}_{\text{tool}}, [20, 50])$. We also set the average total number of visitations $N_{\text{tr}} = 50$. Table 3 shows the steady-state distributions $\Pr^\infty_\pi(s, a)$ and the expected number of visitations $\zeta_\pi(s, a)$ for the labeled sets, as well as the total number of visitations $\zeta_\pi(\bar{r}(\mathcal{M}))$ to the set $\bar{r}(\mathcal{M})$ for EP, CP and CPU policies. As shown, the policies meet both steady-state and transient specifications defined over the product space $S \times A$.

7.6 Modified LP and Policy Set

Impact of ϵ in $\text{LP}_1(\epsilon)$**.** In this section, we use the MDP example of Figure 15 to investigate the impact of the parameter $\epsilon > 0$ on the total reward induced by an optimal EP policy. In particular, we solve $\text{LP}_1(\epsilon)$ with descending values of ϵ and compute the optimal reward $R^*(\epsilon)$. As shown in Figure 16, $R^*(\epsilon)$ increases monotonically as we decrease ϵ with diminishing return, and converges to nearly 0.36 as $\epsilon \to 0$. For values of ϵ below 10^{-4}, the change in average reward if we further decrease ϵ is insignificant. Therefore, in our experiments we have set $\epsilon = 10^{-4}$.

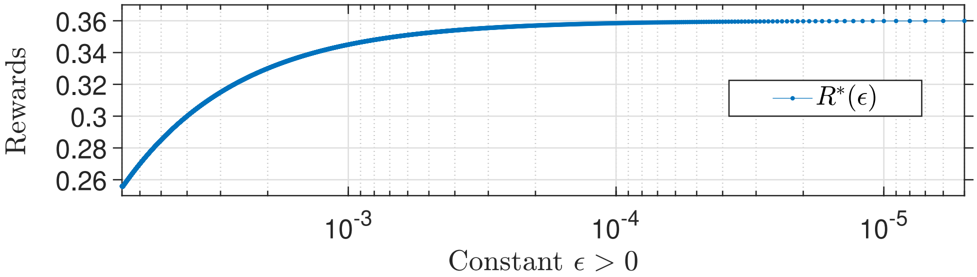

Figure 16: Convergence of the average expected reward as we vary the parameter ϵ in (25) for the MDP example of Figure 15.

Policies with bounded support. Here, we verify the result of Theorem 7. Consider the example of Fig. 2 where $R(s_2, a_1) = R(s_3, a_1) = R(s_3, a_2) = 0.1$ and $R(s_2, a_2) = 0.5$.

Recalling that δ is a lower bound on the support of the policies in $\Pi_{EP}(\delta)$ in (27), we can show that the optimal average reward over $\Pi_{EP}(\delta)$ is $\max_{\pi\in\Pi_{EP}(\delta)} R_\pi^\infty = 0.5(1-\delta)^2 + 0.2\delta(1-\delta) + 0.1\delta^2$, achieved by the policy π^* which has $\pi^*(s_2|a_2) = \pi^*(s_3|a_1) = 1-\delta$, and $\pi^*(a_1|s_2) = \pi^*(a_2|s_3) = \delta$. The reward $R^*(\delta)$ of the policy π in (14) obtained from the optimal solution to $\mathrm{LP}_1(\delta)$ in (25) is $R^*(\delta) = 0.5 - 1.2\delta$, where $\pi(a_1|s_2) = \delta/(1-2\delta)$, $\pi(a_2|s_2) = (1-\delta)/(1-2\delta)$ and $\pi(a_1|s_3) = \pi(a_2|s_3) = 1/2$. Fig. 17 shows that the difference $R_{\pi^*}^\infty - R^*(\delta) \to 0$ as $\delta \to 0$ as per Theorem 7.

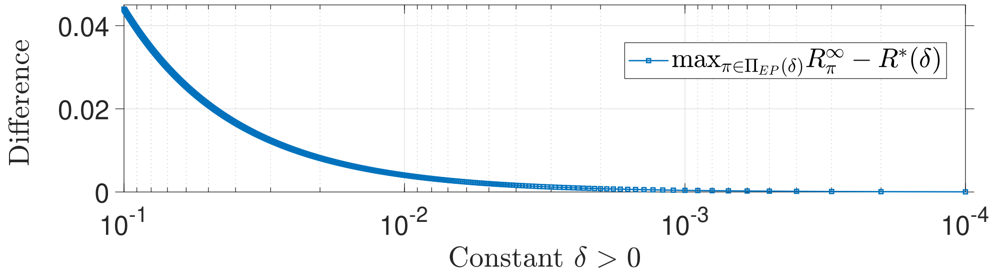

Figure 17: The difference $R_{\pi^*}^\infty - R^*(\delta) \to 0$ as $\delta \to 0$, where $\pi^* = \arg\max_{\pi\in\Pi_{EP}(\delta)} R_\pi^\infty$ and $R^*(\delta)$ is the reward of the policy obtained from an optimal solution to $\mathrm{LP}_1(\delta)$.

7.7 Scalability

We demonstrate the scalability of the proposed formulations using two sets of experiments. The first set of experiments are performed on a standard desktop with 16GB of RAM using the Matlab CVX package for convex optimization (Grant & Boyd, 2014, 2008). We also perform a second set of experiments on a standard desktop of 128GB of RAM using the commercial CPLEX Optimizer, which provides a higher-precision mathematical solver for large-scale linear programming.

For the first set of experiments, we experiment with instances of increasing size of the Toll Collector problem (Figure 13), the Frozen Islands environment (Figure 3) and random partition graphs from the NetworkX library (Hagberg, Swart, & S Chult, 2008). The Toll Collector problem uses an MDP with three TSCCs, each of size n, while the Frozen Islands environment consists of an $n \times n$ grid. We also experiment with random MDPs constructed from n-node directed Gaussian partition graphs generated using the NetworkX toolbox (Hagberg et al., 2008). For such graphs, the cluster sizes are drawn from a normal distribution with mean and variance $n/5$, and two nodes within the same cluster are connected with probability p_{in}, while two nodes in different clusters are connected with probability p_{out} (Brandes, Gaertler, & Wagner, 2003). For these partition graphs, the state space corresponds to the vertex set, the number of actions is equal to the maximum node outdegree and the transitions are deterministic. The initial distribution is uniform over the set $\bar{r}(\mathcal{M})$ and the rewards are selected such that only the first action from every state yields a positive reward, i.e., $R(s \in r(\mathcal{M}), a_1, \cdot) = 1$ and 0 otherwise. An instance of an MDP constructed from a 40-node Gaussian partition graph is illustrated in Figure 18. The specifications for the three environments are given in the caption of Table 4.

For each example, we generate EP, CP and CPU policies using $\mathrm{LP}_1, \mathrm{LP}_2$ and Algorithm 1, respectively, and report on the runtime as we increase n. All instances were verified to meet the given specifications. The results are summarized in Table 4 demonstrating the scalability of the proposed formulations. As shown, LP_2 incurs the largest runtime as it incorporates additional variables in the flow constraints to enforce the recurrence of the TSCCs.

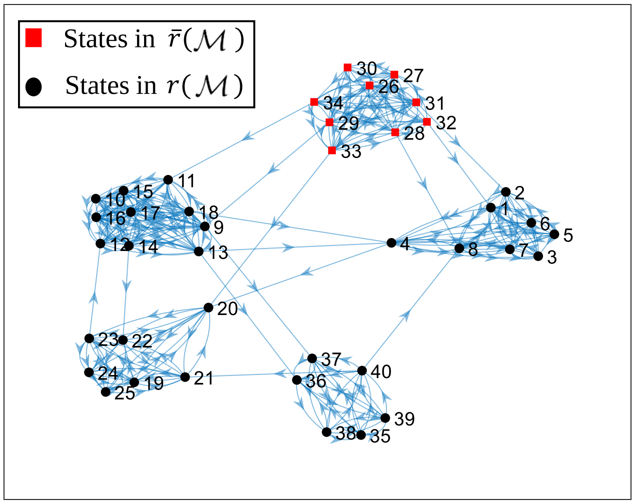

Figure 18: A NetworkX random 40-node digraph used to generate a LMDP for the third example of Table 4.

To further examine the scalability of the LPs underlying the different policies to much larger problem sizes, additional experiments are conducted using the CPLEX 12.8 solver. We run simulations of the LP in (22), LP_1 (with the positivity constraints) and LP_2 (with the flow constraints) for random instances of the Frozen Islands problem. The runtime results are reported in Table 5. For a 64×64 and a 128×128 grid, LP_2 (the most complex) is solved in about 20 seconds and 15 minutes, respectively, demonstrating the effectiveness of the developed formulations even for MDPs with over ten thousand states.

8. Conclusion

A framework for steady-state policy synthesis in general MDPs was developed to derive policies that satisfy constraints on the steady-state behavior of an agent. Linear programming solutions were proposed and their correctness proved for classes of edge-preserving and class-preserving policies. The framework also enables policies that meet specified constraints on the expected number of times the agent visits transient states. Numerical simulations of the resulting policies demonstrate that our approach overcomes limitations in the literature.

The article provides the first solution to the highly understudied problem of steady-state planning over stationary policies in constrained expected average reward multichain MDPs. The policies derived come with rigorous guarantees on the asymptotic long-term

Policy	**Example**												
	Toll Collector, $3n$					**Frozen Lake,** $n \times n$				**Random Gaussian,** n			
	15	30	45	60	75	8×8	12×12	16×16	20×20	20	40	60	80
EP	0.5	1.18	2.4	4.7	6.99	1.96	2.87	5.17	7.97	0.61	1.58	2.34	4.16
CP	1.16	3.85	8.37	15.1	24.48	5.08	25.81	108.49	431.38	4.56	5.84	19.06	35.74
CPU	0.45	0.87	1.54	2.48	3.72	1.97	4.77	7.6	12.62	4.8	6.74	12.04	17.35

Table 4: Average runtime results (in seconds) for 20 instances of the Toll Collector, Frozen Islands, and Gaussian partition graphs of increasing problem size n. The Toll Collector MDP consists of three TSCCs, each of size n. The detailed LMDP parameters are given in the caption of Figure 13 with a steady-state specification lower bound $l = 0.05$. The three-island problem described in Figure 3 forms an $n \times n$ grid. In each of the smaller islands, logs are randomly distributed over 1/4 of the states and a canoe (fishing rod) is placed in the top-left (bottom-right) tile. For these experiments, we have the constraints $(L_{\text{log1}} \cup L_{\text{log2}}, [0.3, 1]), (L_{\text{canoe1}} \cup L_{\text{canoe2}}, [0.05, 1])$ and reward function $R(\cdot, \cdot, L_{\text{fish1}} \cup L_{\text{fish2}}) = 1, R(\cdot, \cdot, S \setminus L_{\text{fish1}} \cup L_{\text{fish2}}) = 0$. For the Gaussian partition graphs, we define a steady-state specification $(L, [0.05, 1])$, where $L = \{s_i\}$, for some $s_i \in r(\mathcal{M})$. The probability of intra-cluster connection $p_{in} = 0.9$ and the probability of inter-cluster connection p_{out} is $0.05, 0.01, 0.01, 0.005$ for the $20, 40, 60, 80$ nodes, respectively.

LP	**Frozen Islands Example**				
	Size, $n \times n$				
	8×8	16×16	32×32	64×64	128×128
LP_1 (20)	0.0001	0.0017	0.0170	0.1187	20.306
LP_2 (21)	0.003	0.038	0.595	20.251	933.821
LP_3 (22)	0.0001	0.0018	0.0168	0.1553	5.425

Table 5: Average runtime (in seconds) of 20 instances per LP for the three-island problem described in Figure 3. These islands combined form an $n \times n$ grid. In each of the smaller islands, logs are randomly distributed over 1/4 of the states and a canoe (fishing rod) is placed in the top-left (bottom-right) tile. For these experiments, we have the constraints $(L_{\text{log1}} \cup L_{\text{log2}}, [0.3, 1]), (L_{\text{canoe1}} \cup L_{\text{canoe2}}, [0.05, 1])$ and reward function $R(\cdot, \cdot, L_{\text{fish1}} \cup L_{\text{fish2}}) = 1, R(\cdot, \cdot, S \setminus L_{\text{fish1}} \cup L_{\text{fish2}}) = 0$.

behavior of agents. The research findings have bearing on the fields of explainable, safe and trustworthy AI, where there is increased concern about explaining AI decisions, ensuring safety constraints are met, and building trust in the behavior of autonomous agents.

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.

Appendix A. Technical Lemmas

Proof of Lemma 2

Let $\pi \in \Pi_{EP}$. Hence, $r(\mathcal{M}_\pi) = r(\mathcal{M})$ according to (15). Consider the set of states $r_k(\mathcal{M})$ in a TSCC of $\mathcal{M}$, for some $k \in [m]$. We will show that this set is also a TSCC of $\mathcal{M}_\pi$. To this end, we first show that they form a SCC in the transition graph of $\mathcal{M}_\pi$. Since $\pi \in \Pi_{EP}$, then $\pi(a|s) > 0, \forall s \in r(\mathcal{M}), a \in A(s)$. Hence, every action $a \in A(s)$ available in state $s \in r_k(\mathcal{M})$ is played with non-zero probability. From (1), for a pair of states $s, s' \in r_k(\mathcal{M})$, $T_\pi(s'|s) > 0$ if $\exists a \in A(s)$ such that $T(s'|s,a) > 0$. Thus, for every directed path between a pair of nodes in $r_k(\mathcal{M})$ in the transition graph of $\mathcal{M}$, there is a similar path between the same nodes in the transition graph of $\mathcal{M}_\pi$. Therefore, the states $r_k(\mathcal{M})$ also form a SCC in the transition graph of $\mathcal{M}_\pi$. Also, the set $r_k(\mathcal{M})$ is reachable in $\mathcal{M}_\pi$ since $r_k(\mathcal{M}) \subseteq r(\mathcal{M}_\pi)$. Finally, there are no outgoing edges to states in $S \setminus r_k(\mathcal{M}_\pi)$ since the edge set of the transition graph of $\mathcal{M}_\pi$ is a subset of the edge set of the transition graph of $\mathcal{M}$. We conclude that $r_k(\mathcal{M}_\pi) = r_k(\mathcal{M}), \forall k \in [m]$. From the definition of the set of CP policies in (16), it follows that $\pi \in \Pi_{CP}$, proving that $\Pi_{EP} \subseteq \Pi_{CP}$. The two conditions in (16) are special cases of the more general requirements in (17), hence $\Pi_{CP} \subseteq \Pi_{CPU}$. □

The following lemma gives a characterization of the Markov chain state classification induced by a policy (14) derived from a feasible point of the constrained set Q_0 in (19).

Lemma 3. *Given an MDP $\mathcal{M}$, let $(x,y) \in Q_0$ defined in* (19) *and $\pi := \pi(x,y)$ as in* (14). *The following holds for the Markov chain $\mathcal{M}_\pi$.*

(a) If $s \in \bar{r}(\mathcal{M})$, then $s \in \bar{r}(\mathcal{M}_\pi)$, i.e., $\bar{r}(\mathcal{M}) \subseteq \bar{r}(\mathcal{M}_\pi)$.

(b) If $s \in r_k(\mathcal{M}) \cap E_x$ for some $k \in [m]$, then $s \in r(\mathcal{M}_\pi)$. As a consequence, if $s \in r_k(\mathcal{M}) \cap \bar{r}(\mathcal{M}_\pi)$, then $s \in \overline{E}_x$, i.e., $x_s = 0$.

Proof of Lemma 3

First, we show part (a), according to which every state in $\bar{r}(\mathcal{M})$ is either transient or isolated in the Markov chain $\mathcal{M}_\pi$ induced by a policy of the form (14) derived from a point in Q_0. Consider $f \in \bar{r}(\mathcal{M})$. From constraint (*iii*), we have $x_f = 0$. Thus, from constraint (*ii*), (14) and the fact that f is only reachable from states in $\bar{r}(\mathcal{M})$,

$$\begin{aligned} y_f &= \beta_f + \sum_{f' \in \bar{r}(\mathcal{M})} \sum_{a \in A(f')} y_{f'a} T(f|f',a) \\ &= \beta_f + \sum_{f' \in \bar{r}(\mathcal{M})} y_{f'} \sum_{a \in A(f')} T(f|f',a)\pi(a|f') \\ &= \beta_f + \sum_{f' \in \bar{r}(\mathcal{M})} y_{f'} T_\pi(f|f') \end{aligned} \tag{36}$$

Note that the second equality above follows from the definition of π in (14) for a state $f \notin E_x$. Two cases arise. If $f \notin E_y$, then $y_f = 0$. Hence, $\beta_f = 0$ and $T_\pi(f|f') = 0, \forall f' \in E_y$. Thus, we have shown that every state f in $\overline{E}_y \cap \bar{r}(\mathcal{M})$ can only be reached from states in $\overline{E}_y \cap \bar{r}(\mathcal{M})$, and that all such states have zero initial probability. Thus, every such state is either isolated or resides in an isolated component. Therefore, $f \in \bar{r}(\mathcal{M}_\pi)$, where $\bar{r}(\mathcal{M}_\pi)$ consists of transient or isolated states. Now consider the other case where $f \in E_y$, i.e., $y_f > 0$. Assume, for the sake of contradiction, that $f \in r(\mathcal{M}_\pi)$. Hence, $f \in F$, for some TSCC F (this subsumes the case where f is absorbing with $|F| = 1$). Then, it must be that $F \subseteq \bar{r}(\mathcal{M})$ since f is not reachable from states $S \setminus \bar{r}(\mathcal{M})$ even under an EP policy. Summing (36) over the set F, we have

$$\begin{aligned}\sum_{f' \in F} y_{f'} &= \sum_{f' \in F} \beta_{f'} + \sum_{j \in F} \sum_{f' \in \bar{r}(\mathcal{M})} y_{f'} T_\pi(j|f') \\ &= \sum_{f' \in F} \beta_{f'} + \sum_{f' \in \bar{r}(\mathcal{M}) \setminus F} y_{f'} \sum_{j \in F} T_\pi(j|f') + \sum_{f' \in F} y_{f'} \,, \end{aligned} \tag{37}$$

where the second equality is due to the closure of the set F, implying that $\sum_{j \in F} T_\pi(j|f') = 1$ for $f' \in F$. It follows that $\beta_{f'} = 0, \forall f' \in F$, and $T_\pi(f|f') = 0, \forall f' \in (\bar{r}(\mathcal{M}) \setminus F) \cap E_y$. Therefore, $F \subseteq \bar{r}(\mathcal{M}_\pi)$, yielding a contradiction. Hence, $f \in \bar{r}(\mathcal{M}_\pi)$. We conclude that $\bar{r}(\mathcal{M}) \subseteq \bar{r}(\mathcal{M}_\pi)$.

Next, we prove part (b) which states that every state s in a TSCC of $\mathcal{M}$ for which $x_s > 0$ is both recurrent and non-isolated in $\mathcal{M}_\pi$. Consider a state $s \in r_k(\mathcal{M}) \cap E_x$ for some $k \in [m]$, so $x_s > 0$. Assume, for the sake of contradiction, that $s \in \bar{r}(\mathcal{M}_\pi)$, i.e., the state s is either transient or isolated. If s is transient, then the column of the matrix T_π^∞ corresponding to state s is zero. Therefore, from constraint (i) in (19), we have $x_s = 0$, i.e., $s \notin E_x$, yielding a contradiction. If $s \in F$ for some isolated component F, then

$$\sum_{s' \in F} (x_{s'} + y_{s'}) = \sum_{s' \in F} \sum_{f \in F} \sum_{a \in A(f)} y_{fa} T(s'|f, a)$$

by summing constraint (ii) over states $s' \in F$, and using the fact that $\beta_f = 0, \forall f \in F$ and that $s' \in F$ is only reachable from states in the isolated set F. Since $\sum_{s' \in F} T(s'|f, a) = 1, \forall f \in F, a \in A(f)$, by the closure of F, we get that $\sum_{s' \in F} x_{s'} = 0$ by interchanging the order of the sums, i.e., $s \in \overline{E}_x$, also yielding a contradiction. Hence, $s \in r(\mathcal{M}_\pi)$.

The second clause of Lemma 3 (b) remains to be proved, i.e., $s \in r_k(\mathcal{M}) \cap \bar{r}(\mathcal{M}_\pi) \implies x_s = 0$. Consider $s \in r_k(\mathcal{M}) \cap \bar{r}(\mathcal{M}_\pi)$. Thus, $s \notin r(\mathcal{M}_\pi)$, so it follows from the result we have just shown that $s \in \bar{r}(\mathcal{M}) \cup \overline{E}_x$. However, since $s \in r_k(\mathcal{M})$ for some k, then $s \notin \bar{r}(\mathcal{M})$. Hence, $s \in \overline{E}_x$. □

Next, we state and prove two lemmas that will be useful in the proof of Lemma 6, which establishes a sufficient condition for the existence of a one-to-one correspondence between a feasible point in Q_0 and the steady-state distribution of the Markov chain induced by the policy in (14) derived from this solution.

Lemma 4. *Given MDP $\mathcal{M}$, if $(x, y) \in Q_0$, where Q_0 is the set of points in (19), then x is a stationary distribution of the Markov chain $\mathcal{M}_\pi$ induced by the policy π in (14).*

Proof of Lemma 4

First, consider $s' \in \bar{r}(\mathcal{M})$, and define $X_0 := \{x : (x, y) \in Q_0 \text{ for some } y\}$. Since $x \in X_0$, we have $x_{s'} = 0$ by constraint (iii). Also,

$$\sum_{s \in S} x_s T_\pi(s'|s) = \sum_{s \in \bar{r}(\mathcal{M})} x_s T_\pi(s'|s) = 0 \,, \tag{38}$$

where the first equality holds since $s' \in \bar{r}(\mathcal{M})$ is only reachable from states $s \in \bar{r}(\mathcal{M})$ even when all edges defining possible transitions in the MDP are preserved. Next, consider $s' \in S \setminus \bar{r}(\mathcal{M})$. We have

$$x_{s'} := \sum_{a \in A(s')} x_{s'a} = \sum_{s \in S} \sum_{a \in A(s)} x_{sa} T(s'|s, a) = \sum_{s \in E_x} \sum_{a \in A(s)} x_s \pi(a|s) T(s'|s, a) = \sum_{s \in S} x_s T_\pi(s'|s) \tag{39}$$

The first equality follows from the fact that $x \in X_0$, the second from the definition of π in (14), and the last from the definition of T_π in (1) and that $x_s = 0, \forall s \in S \setminus E_x$. Finally, $x^\top e = 1$, by summing constraints (ii) over all $s' \in S$. □

Lemma 5. *Given an MDP $\mathcal{M}$, let $(x, y) \in Q_0$ and $\pi := \pi(x, y)$ as in* (14). *If $\pi \in \Pi_{CPU}$, then the subvector $x_{r_k(\mathcal{M}_\pi)}$ of x must satisfy the following identity for all $k \in [m]$*

$$x^\top_{r_k(\mathcal{M}_\pi)} e = \beta^\top_{r_k(\mathcal{M}_\pi)} e + \beta^\top_{\bar{r}(\mathcal{M}_\pi)} P_{\pi,k} \,, \tag{40}$$

where, $P_{\pi,k} = [p_{fk}], f \in \bar{r}(\mathcal{M}_\pi)$, is the vector of absorption probabilities from $\bar{r}(\mathcal{M}_\pi)$ into $r_k(\mathcal{M}_\pi)$ under policy π.

Proof of Lemma 5

To show (40), note that, since $\pi \in \Pi_{CPU}$, we have that $r_k(\mathcal{M}_\pi) \subseteq r_k(\mathcal{M}), k \in [m]$, where $r_k(\mathcal{M}_\pi) \subset r(\mathcal{M}_\pi)$ denotes the k-th TSCC of $\mathcal{M}_\pi$. Since $(x, y) \in Q_0$, by summing constraints (ii) in (19) over the set $r_k(\mathcal{M}_\pi)$, we get

$$\sum_{s \in r_k(\mathcal{M}_\pi)} \beta_s = \sum_{s \in r_k(\mathcal{M}_\pi)} x_s + \sum_{s \in r_k(\mathcal{M}_\pi)} y_s - \sum_{s \in r_k(\mathcal{M}_\pi)} \sum_{s' \in r_k(\mathcal{M}_\pi) \cup \bar{r}(\mathcal{M}_\pi)} \sum_{a \in A(s')} T(s|s', a) y_{s'a} \tag{41}$$

where we used the fact that $r_k(\mathcal{M}_\pi)$ is only reachable from states in $r_k(\mathcal{M}_\pi) \cup \bar{r}(\mathcal{M}_\pi)$. By breaking the summation in the last term on the RHS of (41) over states s' in the union of the disjoint sets $r_k(\mathcal{M}_\pi)$ and $\bar{r}(\mathcal{M}_\pi)$ and interchanging the order of the summations over s and s', the last term in (41) simplifies to

$$\begin{aligned} &\sum_{s' \in r_k(\mathcal{M}_\pi)} \sum_{a \in A(s')} y_{s'a} \sum_{s \in r_k(\mathcal{M}_\pi)} T(s|s', a) + \sum_{s' \in \bar{r}(\mathcal{M}_\pi)} \sum_{s \in r_k(\mathcal{M}_\pi)} \sum_{a \in A(s')} T(s|s', a) y_{s'a} \\ &= \sum_{s' \in r_k(\mathcal{M}_\pi)} y_{s'} + \sum_{s' \in \bar{r}(\mathcal{M}_\pi)} y_{s'} \sum_{s \in r_k(\mathcal{M}_\pi)} T_\pi(s|s') \,, \end{aligned} \tag{42}$$

where the first term on the RHS of the equality (42) follows from the closure of $r_k(\mathcal{M}_\pi)$ (which implies that $\sum_{s\in r_k(\mathcal{M}_\pi)} T(s|s',a) = 1$ for $s' \in r_k(\mathcal{M}_\pi)$), and the second term from the definition of the policy in (14) for states in $\overline{E}_x$ (noting that $s' \in \bar{r}(\mathcal{M}_\pi)$ implies $x_{s'} = 0$). Replacing (42) in (41), we get that

$$\sum_{s\in r_k(\mathcal{M}_\pi)} \beta_s = \sum_{s\in r_k(\mathcal{M}_\pi)} x_s - \sum_{s'\in\bar{r}(\mathcal{M}_\pi)} y_{s'} \sum_{s\in r_k(\mathcal{M}_\pi)} T_\pi(s|s') \tag{43}$$

We proceed to further simplify the second term on the RHS of (43). Since $x_s = 0, \forall s \in \bar{r}(\mathcal{M}_\pi)$, it follows from constraint (*ii*) of (19) and (14) that

$$y_s = \beta_s + \sum_{s'\in\bar{r}(\mathcal{M}_\pi)} y_{s'} \sum_{a\in A(s')} \pi(a|s')T(s|s',a) = \beta_s + \sum_{s'\in\bar{r}(\mathcal{M}_\pi)} y_{s'} T_\pi(s|s'),\ \forall s \in \bar{r}(\mathcal{M}_\pi)\,. \tag{44}$$

In matrix form, this can be rewritten as

$$y_{\bar{r}(\mathcal{M}_\pi)} = (I - Z_\pi^\top)^{-1}\beta_{\bar{r}(\mathcal{M}_\pi)}\,, \tag{45}$$

where $Z_\pi = [z_{s's}] \in [0,1]^{|\bar{r}(\mathcal{M}_\pi)|\times|\bar{r}(\mathcal{M}_\pi)|}$, with $z_{s's} := T_\pi(s|s')$. Hence, the second summation in (43) can be written as

$$\sum_{s'\in\bar{r}(\mathcal{M}_\pi)} y_{s'} \sum_{s\in r_k(\mathcal{M}_\pi)} T_\pi(s|s') = y^\top L_{\pi,k}e = \beta^\top_{\bar{r}(\mathcal{M}_\pi)}(I - Z_\pi)^{-1}L_{\pi,k}e\,, \tag{46}$$

where $L_{\pi,k}$ is the submatrix of T_π of transitions from $\bar{r}(\mathcal{M}_\pi)$ to $r_k(\mathcal{M})$ under policy π as in (5). From (43) and (46),

$$\sum_{s\in r_k(\mathcal{M}_\pi)} x_s = \beta^\top_{r_k(\mathcal{M}_\pi)}e + \beta^\top_{\bar{r}(\mathcal{M}_\pi)}(I - Z_\pi)^{-1}L_{\pi,k}e\,. \tag{47}$$

Since the vector $P_{\pi,k}$ is the scaled (by the inverse of η_s) s-th column of the submatrix of the matrix T_π^∞ defining transitions from $\bar{r}(\mathcal{M}_\pi)$ to $r_k(\mathcal{M}_\pi)$, we have (Feller, 1968; Puterman, 1994),

$$P_{\pi,k} = (I - Z_\pi)^{-1}L_{\pi,k}e\,, \tag{48}$$

which proves the identity (40) of Lemma 5. □

We can readily state the next Lemma which establishes the aforementioned sufficiency condition.

Lemma 6. *Given an MDP $\mathcal{M}$, let $(x,y) \in Q_0$ and $\pi := \pi(x,y)$ as in* (14)*. If $\pi \in \Pi_{CPU}$, then* $\Pr_\pi^\infty = x$.

Before we prove Lemma 6, we remark that this result also holds for policies in Π_{EP} and Π_{CP} since these are subsets of Π_{CPU} per Lemma 2.

Proof of Lemma 6

We seek to show that the steady-state distribution of the Markov chain $\mathcal{M}_\pi$ induced by the policy π (14) derived from a feasible point $(x, y) \in Q_0$ matches x, provided that π is a CPU policy, where Q_0 is as defined in (19).

First, we consider the states in $\bar{r}(\mathcal{M}_\pi)$. We have that $\Pr_\pi^\infty(s) = 0, \forall s \in \bar{r}(\mathcal{M}_\pi)$ since such states are either transient or isolated in the Markov chain $\mathcal{M}_\pi$ induced by policy π. Next, we argue that $x_s = 0$ for all such states. From Lemma 3 (a), we have that $\bar{r}(\mathcal{M}) \subseteq \bar{r}(\mathcal{M}_\pi)$. For states $s \in \bar{r}(\mathcal{M})$, $x_s = 0$ by constraint (iii) in (19). Thus, we have shown that $\Pr_\pi^\infty(s) = x_s$ for every $s \in \bar{r}(\mathcal{M})$. Now, consider a state $s \in \bar{r}(\mathcal{M}_\pi) \setminus \bar{r}(\mathcal{M})$. The state s must belong to $r_k(\mathcal{M}) \cap \bar{r}(\mathcal{M}_\pi)$ for some $k \in [m]$, where m is the number of TSCCs in $\mathcal{M}$. Hence, $x_s = 0$ by Lemma 3 (b). Therefore, we have argued that $x_s = \Pr_\pi^\infty(s) = 0, \forall s \in \bar{r}(\mathcal{M}_\pi)$.

Second, we consider states in $r(\mathcal{M}_\pi)$. According to Lemma 4, x satisfies

$$x_{r_k(\mathcal{M}_\pi)}^\top = x_{r_k(\mathcal{M}_\pi)}^\top T_{\pi,k}, \forall k \in [m] \, , \tag{49}$$

where $T_{\pi,k}$ is the submatrix of T_π of transitions between states in $r_k(\mathcal{M}_\pi)$. We have also shown that $x_{r_k(\mathcal{M}_\pi)}$ satisfies the identity (40) stated in Lemma 5.

Given the definition of $\Pr_\pi^\infty(s)$ in Lemma 1 and (6), $\Pr_\pi^\infty(s) = \eta_s \left(\beta_{r_k(\mathcal{M})}^\top e + \beta_{\bar{r}(\mathcal{M})}^\top P_{\pi,k}\right)$. Hence,

$$\sum_{s \in r_k(\mathcal{M}_\pi)} \Pr_\pi^\infty(s) = \beta_{r_k(\mathcal{M}_\pi)}^\top e + \beta_{\bar{r}(\mathcal{M}_\pi)}^\top P_{\pi,k} \, . \tag{50}$$

From (40) and (50), we conclude that

$$\sum_{s \in r_k(\mathcal{M}_\pi)} \Pr_\pi^\infty(s) = \sum_{s \in r_k(\mathcal{M}_\pi)} x_s. \tag{51}$$

The ergodic theorem of Markov chains asserts that the solution to $x^\top T = x^\top$, where $x^\top e = 1, x \geq 0$, is unique iff T is the transition matrix of a unichain (Gallager, 2013; Altman, 1999). From (49), (50) and (51), we have shown that

$$x_k^\top T_{\pi,k} = x_k^\top, \text{ where } x_k^\top e = c_k, \; x_k \geq 0$$

for TSCCs $k \in [m]$, where $x_k := x_{r_k(\mathcal{M}_\pi)}$, c_k is the RHS of (50), and $\sum_{k=1}^m c_k = 1$. Further, since $\pi \in \Pi_{CPU}$, every TSCC is a unichain. Hence, by the ergodic theorem, the solution $x_{r_k(\mathcal{M}_\pi)}$ to (49) and (40) is unique for each component $r_k(\mathcal{M}_\pi), k \in [m]$, thus x is equal to the unique steady-state distribution, i.e., $x = \Pr_\pi^\infty$. □

We also make use of the following lemma in the proof of the converse part of Theorem 2. The lemma establishes that all occupation measures induced by the policies of interest are Q_0-feasible.

Lemma 7. *Given MDP $\mathcal{M}$, let $X_0 := \{x : (x, y) \in Q_0$ for some $y\}$, where Q_0 is as defined in* (19)*. Then, $\mathcal{P}^\infty(\Pi_{CPU}) \subseteq X_0$.*

Proof of Lemma 7

We show that the steady-state distribution induced by every CPU policy is in X_0. To this end, let $x \in \mathcal{P}^\infty(\Pi_{CPU})$, i.e., $\exists \pi \in \Pi_{CPU} : \Pr_\pi^\infty = x$, where $\Pr_\pi^\infty$ is as defined in Lemma 1. Therefore, x is a stationary distribution of the Markov chain $\mathcal{M}_\pi$, in which $r_k(\mathcal{M}_\pi), k \in [m]$ are TSCCs and states $\bar{r}(\mathcal{M}_\pi)$ are either transient or isolated. Hence, $x^\top = x^\top T_\pi$. Therefore,

$$\begin{aligned} x_{s'} := \sum_{a \in A(s')} x_{s'a} &= \sum_{s \in S} x_s T_\pi(s'|s) = \sum_{s \in S} \sum_{a \in A(s)} x_s \pi(a|s) T(s'|s,a) \\ &= \sum_{s \in S} \sum_{a \in A(s)} x_{sa} T(s'|s,a) \,, \end{aligned} \tag{52}$$

where the last equality follows since $\Pr_\pi^\infty(s,a) = \Pr_\pi^\infty(s)\pi(a|s)$. Thus, the steady-state distribution x satisfies constraint (i) in (19). From the definition of Π_{CPU} in (17), every $f \in \bar{r}(\mathcal{M})$ is either transient or isolated under π. Thus, $x_{\bar{r}(\mathcal{M})} := \{\Pr_\pi^\infty(f,a)\}_{f \in \bar{r}(\mathcal{M}),a} = 0$, satisfying constraint (iii).

The variables $y_{fa}, f \in \bar{r}(\mathcal{M}_\pi), a \in A(f)$, can be set as in (45), i.e., choose $y_{fa} = \beta_{\bar{r}(\mathcal{M}_\pi)}^\top (I - Z_\pi)^{-1} e_f \pi(a|f), f \in \bar{r}(\mathcal{M}_\pi), a \in A(f)$, where Z_π is the submatrix of T_π defined in (5), which satisfies the constraints (ii) as we have already shown in (44). The remaining variables $y_{sa}, s \in r_k(\mathcal{M}_\pi), a \in A(s)$, can now be chosen in terms of $x_{sa}, y_{fa}, T(s'|s,a)$ and β such that the corresponding constraints (ii) are satisfied. Thus, for the given x, we have shown the existence of a feasible y such that $(x,y) \in Q_0$. Therefore, $x \in X_0$. □

Appendix B. Proof of Main Theorems

Proof of Theorem 1

Since (x,y) is a feasible point of LP$_1$, we have that $(x,y) \in Q_0$ per (20). From Lemma 3 (a), $\bar{r}(\mathcal{M}) \subseteq \bar{r}(\mathcal{M}_\pi)$, thus $r(\mathcal{M}) \supseteq r(\mathcal{M}_\pi)$. Consider a state $s \in r(\mathcal{M})$. Then, $s \in r_k(\mathcal{M})$ for some $k \in [m]$. From the positivity constraint (v) of LP$_1$, we also have that $x_s > 0$, i.e., $s \in E_x$. Since $s \in r_k(\mathcal{M}) \cap E_x$, it follows that $s \in r(\mathcal{M}_\pi)$ by Lemma 3 (b). Therefore, $r(\mathcal{M}) \subseteq r(\mathcal{M}_\pi)$. We conclude that $r(\mathcal{M}_\pi) = r(\mathcal{M})$. From constraint (v), $x_{sa} > 0, \forall s \in r(\mathcal{M}), a \in A(s)$. It follows from the definition of π in (14) for states $s \in E_x$ that $\pi(a|s) > 0, \forall s \in r(\mathcal{M}), a \in A(s)$. We have shown that π satisfies both requirements in (15), hence $\pi \in \Pi_{EP}$. □

Proof of Theorem 2

($\Longrightarrow$) First, we show that if (20) is feasible, then there exists an EP policy that meets the specifications Φ_L^∞. Let $(x,y) \in Q_1$ denote a feasible solution to (20) and let π be defined as in (14). By Theorem 1, $\pi \in \Pi_{EP}$. By Lemma 2, we also have that $\pi \in \Pi_{CPU}$. Invoking Lemma 6, we conclude that $\Pr_\pi^\infty(s,a) = x_{sa}, s \in S, a \in A(s)$, i.e., x is equal to the steady-state distribution of the Markov chain $\mathcal{M}_\pi$ induced by policy π. Since x satisfies constraint (iv), this implies that $\mathcal{M}_\pi$ meets the specifications Φ_L^∞.

($\Longleftarrow$) Now, we show the converse, that is, the existence of an EP policy that meets the specifications implies that LP$_1$ in (20) is feasible. Define $V := \{x : (iv) \text{ and } (v) \text{ satisfied}\}$. Thus, we have that $X_{LP_1} = X_0 \cap V$, where $X_{LP_1} = \{x : (x,y) \in Q_1 \text{ for some } y\}$. Suppose

$\exists \pi \in \Pi_{EP}$ that satisfies the specifications Φ_L^∞ as in the statement of Theorem 2. Then $\Pr_\pi^\infty \in \mathcal{P}^\infty(\Pi_{EP})$ is well-defined as in Lemma 1. We have $\Pr_\pi^\infty(s) := (\beta^\top T_\pi^\infty)_s > 0, \forall s \in r(\mathcal{M}_\pi)$, since all such states are recurrent in the Markov chain $\mathcal{M}_\pi$. Since $\pi \in \Pi_{EP}$, $\pi(a|s) > 0, \forall s \in r(\mathcal{M}), a \in A(s)$, from (15). Hence, by Lemma 1, $\Pr_\pi^\infty(s,a) > 0, \forall s \in r(\mathcal{M}), a \in A(s)$. Therefore, $\Pr_\pi^\infty \in V$. Hence, $\mathcal{P}^\infty(\Pi_{EP}) \cap V$ is non-empty. Set $x_{sa} = \Pr_\pi^\infty(s,a), s \in S, a \in A(s)$. Recall that $\bar{r}(\mathcal{M}_\pi) = \bar{r}(\mathcal{M})$ since $\pi \in \Pi_{EP}$, so we have $x_{sa} = \Pr_\pi^\infty(s,a) = 0, \forall s \in \bar{r}(\mathcal{M})$. From Lemma 7, $\mathcal{P}^\infty(\Pi_{EP}) \subseteq X_0$, where we also use the fact that $\mathcal{P}^\infty(\Pi_{EP}) \subseteq \mathcal{P}^\infty(\Pi_{CPU})$ as a consequence of Lemma 2. The variables y_{sa} can be defined in terms of $x_{sa}, T(s'|s,a)$ and β such that the constraints (ii) are satisfied. Hence, X_{LP_1}, and in turn Q_1, is non-empty. The optimality of π^* follows from the optimality of (x^*, y^*), Theorem 1 and the established equality $\Pr_{\pi^*}^\infty = x^*$. □

Proof of Theorem 3

Let $f \in \bar{r}(\mathcal{M})$. From Lemma 3 (a), $f \in \bar{r}(\mathcal{M}_\pi)$. Now consider $s \in r_k(\mathcal{M})$ for some $k \in [m]$. As argued earlier, every state in $r_k(\mathcal{M})$ is reachable from s given constraints $(viii)$, (x), (xii) of (21). In addition, s is reachable from all states in $r_k(\mathcal{M})$, which follows from constraints (vii), (ix), (xi), $(xiii)$. Hence, $s \in r(\mathcal{M}_\pi)$. Therefore, $r(\mathcal{M}) \subseteq r(\mathcal{M}_\pi)$. Since we have already shown that $\bar{r}(\mathcal{M}) \subseteq \bar{r}(\mathcal{M}_\pi)$, we conclude that $r(\mathcal{M}_\pi) = r(\mathcal{M})$. Therefore, $\pi \in \Pi_{CP}$ defined in (16). □

Proof of Theorem 4

The proof follows the same reasoning as that of Theorem 2.
($\Longrightarrow$) Let $(x, y, f, f^{\text{rev}}) \in Q_2$ denote a feasible solution to (21) and let π be defined as in (14). By Theorem 3, $\pi \in \Pi_{CP}$. Invoking Lemma 6 and Lemma 2, we have that $\Pr_\pi^\infty(s,a) = x_{sa}, s \in S, a \in A(s)$, which implies that $\mathcal{M}_\pi$ meets the specifications Φ_L^∞ per constraint (iv).
($\Longleftarrow$) Define $V := \{x : (iv) \text{ and } (vi) - (xiv) \text{ satisfied}\}$. Thus, we have that $X_{LP_2} = X_0 \cap V$, where $X_{LP_2} = \{x : (x, y, f, f^{\text{rev}}) \in Q_2\}$. Suppose $\exists \pi \in \Pi_{CP}$ that satisfies the specifications Φ_L^∞ as in the statement of Theorem 4. Hence, $r_k(\mathcal{M}_\pi), k \in [m]$ are the recurrent components of $\mathcal{M}_\pi$. Then, $\Pr_\pi^\infty \in \mathcal{P}^\infty(\Pi_{CP})$ is well-defined as in Lemma 1. We can set $x_{sa} = \Pr_\pi^\infty(s,a) = \pi(a|s)\Pr_\pi^\infty(s)$ for every $s \in S, a \in A(s)$. The flow variables in (vi) - (vii) can be defined in terms of x_{sa} and $T(s'|s,a)$ such that the constraints (x) - $(xiii)$ are satisfied. Hence, $x \in V$, i.e., $\mathcal{P}^\infty(\Pi_{CP}) \cap V$ is non-empty. By Lemma 7, X_{LP_2} and Q_2 are non-empty. The optimality of π^* follows from the optimality of (x^*, y^*), Theorem 3 and the established equality $\Pr_{\pi^*}^\infty = x^*$. □

Proof of Theorem 5

Assume $(x, y) \in Q^*$. We have that $V_k^+(x) \subseteq r_k(\mathcal{M}_\pi)$ for π in (14) by Lemma 3 (b). Further, consider $s \in r_k(\mathcal{M}) \cap \overline{E}_x$. By constraint (i) and the definition of π in (14), $T_\pi(s|s') = 0, \forall s' \in V_k^+(x)$. For the sake of contradiction, assume $s \in r(\mathcal{M}_\pi)$. Hence, $s \in F \subseteq r_k(\mathcal{M})$ for some TSCC F of $\mathcal{M}_\pi$. Summing constraints (ii) over the set F, we get that $\beta_s = 0, \forall s \in F$ and $T_\pi(s|s') = 0, s' \in \bar{r}(\mathcal{M}), s \in F$. Hence, $s \in \bar{r}(\mathcal{M}_\pi)$, yielding a contradiction. We conclude that $V_k^+(x) = r(\mathcal{M}_\pi) \cap r_k(\mathcal{M})$. Therefore, if for every $k \in [m]$

we have that the subgraph $(V_k^+(x), E_k^+(x))$ is strongly connected, then $V_k^+(x)$ is a SCC in $\mathcal{M}_\pi$, for π in (14). Hence, $V_k^+(x)$ is the unique TSCC $r_k(\mathcal{M}_\pi)$ in the set $r_k(\mathcal{M})$, i.e., $\pi \in \Pi_{CPU}$. The result now follows from Lemma 6. □

Proof of Theorem 6

(1) Every feasible solution of $\text{LP}_1(\epsilon)$ is also LP_1-feasible. Hence, the result follows as an immediate consequence of Theorem 1.

(2) The proof of part (2) follows the same reasoning as in the proof of the converse of Theorem 2. Specifically, we have shown that, if $\pi \in \Pi_{EP}$, then $\Pr_\pi^\infty(s,a) > 0, \forall s \in r(\mathcal{M}_\pi), a \in A(s)$. Hence, $\exists \epsilon > 0$ such that $\Pr_\pi^\infty \in V'$, where $V' := \{x : (iv) \text{ and } (v)' \text{ satisfied})\}$. Therefore, $X_{LP_1(\epsilon)} := X_0 \cap V'$ is non-empty, and in turn $\text{LP}_1(\epsilon)$ is feasible.

(3) Let $\epsilon_n \to 0, n \in \mathbb{N}$, be a monotonically decreasing sequence, π_n^* the EP policy in (14) corresponding to an optimal solution to $\text{LP}_1(\epsilon_n)$, and $R_n := R_{\pi_n^*}^\infty(\beta)$. The sequence $(R_n)_{n\in\mathbb{N}}$ is monotonically non-decreasing since $R_n \geq R_m$ whenever $\epsilon_n < \epsilon_m$. Further, from (11), we have that the sequence is bounded above since $\sup_{\pi\in\Pi_{EP}} R_\pi^\infty(\beta) \leq r_{\max}$, where $r_{\max} := \max_{s\in S, a\in A(s)} R(s,a)$. Since the sequence $(R_n)_{n\in\mathbb{N}}$ is both increasing and bounded, it converges to the limit $\sup_n R_n$ by the monotone convergence theorem (Royden & Fitzpatrick, 2010). We are only left to show that $\sup_n R_n = \sup_{\pi\in\Pi_{EP}} R_\pi^\infty$. To this end, assume for the sake of contradiction that $\sup_n R_n < \sup_{\pi\in\Pi_{EP}} R_\pi^\infty$. Since the RHS of the inequality is the least upper bound on the average reward of EP policies, then for any $\delta > 0, \exists \pi' \in \Pi_{EP} : R_{\pi'}^\infty > \sup_{\pi\in\Pi_{EP}} R_\pi^\infty - \delta$. We can choose δ small enough such that $R_{\pi'}^\infty > \sup_n R_n$. From part (2) above, $\exists \epsilon > 0$, such that $\Pr_{\pi'}^\infty$ is $\text{LP}_1(\epsilon')$-feasible for all $\epsilon' \leq \epsilon$. Hence, from the definition of π_n^*, we get that $\sup_n R_n \geq R_{\pi'}^\infty$, yielding a contradiction. □

Proof of Theorem 7

(1) Let x be $\text{LP}_1(\delta)$-feasible. Since the feasible set for $\text{LP}_1(\delta)$ is a subset of the feasible set of LP_1, then $\pi \in \Pi_{EP}$ by Theorem 1. Therefore, we only need to verify the bounded support requirement in (27). For $s \in r(\mathcal{M})$, we have that $x_{sa} \geq \delta > 0, a \in A(s)$, from constraint $(v)'$ in $\text{LP}_1(\delta)$. Hence, $\pi(a|s) = x_{sa}/x_s \geq \delta$. Therefore, $\pi \in \Pi_{EP}(\delta)$.

(2) Assume $\pi \in \Pi_{EP}(\delta)$ and meets the specifications Φ_L^∞. Noting that $\Pi_{EP}(\delta) \subset \Pi_{EP}$, then there exists an $0 < \epsilon \leq \delta$ such that $\Pr_\pi^\infty$ is a feasible solution of $\text{LP}_1(\epsilon)$, which follows from part (2) of Theorem 6. Hence, $\max_{\pi\in\Pi_{EP}(\delta)} R_\pi^\infty(\beta) \leq R^*(\epsilon)$ since $R^*(\epsilon)$ is the optimal value of $\text{LP}_1(\epsilon)$, where $\epsilon \leq \delta$ is a function of δ. As $\delta \to 0$, the sequence of rewards $R^*(\delta)$ is monotonically non-decreasing and bounded above. Hence, as $\delta \to 0$, the sequence $R^*(\delta)$ converges to a limit. Every convergent sequence is a Cauchy sequence (Royden & Fitzpatrick, 2010), i.e., the elements of the sequence become arbitrarily close to each other as $\delta \to 0$. Hence, $R^*(\epsilon) - R^*(\delta) \to 0$, as $\delta \to 0$. □

Proof of Theorem 8

The cone $V(x,y)$ in (29) is the cone of feasible directions from a feasible point (x,y), i.e., directions $v = (h,z)$ along which $\exists\lambda > 0$ such that $(x,y) + \lambda(h,z)$ is feasible. The sets $u(x)$, $l(x)$, $n(x)$ and $m(y)$ denote the sets of active (upper and lower) specification and non-negativity (of state-action variables x and y) constraints, respectively. Since the rewards vector R is an interior point of the dual cone $V^*(x, y)$ designated in the statement of Theorem 8, moving away from (x, y) along any feasible direction can only reduce the value of the objective, i.e., $\sum_{s\in S}\sum_{a\in A(s)} R(s, a)h_{sa} < 0$. Hence, (x, y) is the unique optimal solution to (22). We have already shown that the set of occupation measures induced by policies for which $\bar{r}(\mathcal{M}) \subseteq \bar{r}(\mathcal{M}_\pi)$ is contained in the feasible set of (22). Since π in (14) is one such policy by Lemma 3 (a), we have $\Pr_\pi^\infty = x$ and π meets the specifications Φ_L^∞. The uniqueness of π in this class of policies follows from the established uniqueness of the optimal solution x. □

Proof of Proposition 1

By Lemma 3, we have that $f \in \bar{r}(\mathcal{M}_\pi)$. If $\pi \in \Pi_{CPU}$, then the condition of Lemma 6 is met, and it follows from (45) that $y_f = \beta_{\bar{r}(\mathcal{M}_\pi)}^\top (I - Z_\pi)^{-1} e_f = \zeta_\pi(f)$. □

References

Akshay, S., Bertrand, N., Haddad, S., & Hélouët, L. (2013). The steady-state control problem for Markov decision processes. In *International Conference on Quantitative Evaluation of Systems*, pp. 290–304, Berlin Heidelberg. Springer.

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

Altman, E. (1999). *Constrained Markov decision processes.* CRC Press, Boca Raton.

Altman, E., Boularouk, S., & Josselin, D. (2019). Constrained Markov decision processes with total expected cost criteria. In *Proceedings of the 12th EAI International Conference on Performance Evaluation Methodologies and Tools*, pp. 191–192. ACM.

Atia, G., Beckus, A., Alkhouri, I., & Velasquez, A. (2020). Steady-state policy synthesis in multichain Markov decision processes. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4069–4075. International Joint Conferences on Artificial Intelligence Organization.

Ayala, A. M., Andersson, S. B., & Belta, C. (2014). Formal synthesis of control policies for continuous time markov processes from time-bounded temporal logic specifications. *IEEE Transactions on Automatic Control*, *59*(9), 2568–2573.

Baiocchi, D. (2010). *Confronting Space Debris: Strategies and Warnings from Comparable Examples Including Deepwater Horizon.* Rand Corporation.

Baumgartner, P., Thiébaux, S., & Trevizan, F. (2018). Heuristic search planning with multi-objective probabilistic LTL constraints. In *Sixteenth International Conference on Principles of Knowledge Representation and Reasoning*, pp. 415–424.

Bertsekas, D. (2005). *Dynamic programming and optimal control*, Vol. 2. Athena Scientific, Belmont, Mass.

Bertsimas, D., & Tsitsiklis, J. (1997). *Introduction to Linear Optimization* (1st edition). Athena Scientific.

Bhatnagar, S., & Lakshmanan, K. (2012). An online actor–critic algorithm with function approximation for constrained Markov decision processes. *Journal of Optimization Theory and Applications*, *153*(3), 688–708.

Boussemart, M., & Limnios, N. (2004). Markov decision processes with asymptotic average failure rate constraint. *Communications in Statistics-Theory and Methods*, *33*(7), 1689–1714.

Boussemart, M., Limnios, N., & Fillion, J. (2002). Non-ergodic Markov decision processes with a constraint on the asymptotic failure rate: general class of policies. *Stochastic models*, *18*(1), 173–191.

Brafman, R. I., & De Giacomo, G. (2019). Planning for LTLf /LDLf goals in non-Markovian fully observable nondeterministic domains. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1602–1608. International Joint Conferences on Artificial Intelligence Organization.

Brandes, U., Gaertler, M., & Wagner, D. (2003). Experiments on graph clustering algorithms. In *European Symposium on Algorithms*, pp. 568–579. Springer.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). OpenAI Gym. *arXiv preprint arXiv:1606.01540*.

Camacho, A., & McIlraith, S. A. (2019). Strong fully observable non-deterministic planning with LTL and LTLf goals. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 5523–5531. International Joint Conferences on Artificial Intelligence Organization.

Courcoubetis, C., & Yannakakis, M. (1995). The complexity of probabilistic verification. *Journal of the ACM*, *42*(4), 857–907.

De Ghellinck, G. (1960). Les problèmes de décisions séquentielles. *Cahiers du Centre d'Etudes de Recherche Opérationnelle*, *2*(2), 161–179.

De Giacomo, G., Felli, P., Patrizi, F., & Sardina, S. (2010). Two-player game structures for generalized planning and agent composition. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Denardo, E. V., & Fox, B. L. (1968). Multichain Markov renewal programs. *SIAM Journal on Applied Mathematics*, *16*(3), 468–487.

Derman, C. (1970). *Finite State Markovian Decision Processes*. Academic Press, Inc., Orlando, FL, USA.

Engesser, T., Bolander, T., & Nebel, B. (2017). Cooperative epistemic multi-agent planning with implicit coordination. In *Proceedings of the 3rd Workshop on Distributed and Multi-Agent Planning (DMAP)*, p. 68.

Feinberg, E. A. (2009). Adaptive computation of optimal nonrandomized policies in constrained average-reward MDPs. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pp. 96–100.

Feinberg, E. A. (2000). Constrained discounted Markov decision processes and Hamiltonian cycles. *Mathematics of Operations Research*, *25*(1), 130–140.

Feller, W. (1968). *An Introduction to Probability Theory and its Applications* (3rd edition)., Vol. 1. Wiley.

Gallager, R. G. (2013). *Stochastic Processes: Theory for Applications*. Cambridge University Press, New York.

Grant, M., & Boyd, S. (2008). Graph implementations for nonsmooth convex programs. In Blondel, V., Boyd, S., & Kimura, H. (Eds.), *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pp. 95–110. Springer-Verlag Limited.

Grant, M., & Boyd, S. (2014). CVX: Matlab software for disciplined convex programming, version 2.1..

Guo, M., & Zavlanos, M. M. (2018). Probabilistic motion planning under temporal tasks and soft constraints. *IEEE Transactions on Automatic Control*, *63*(12), 4051–4066.

Hagberg, A., Swart, P., & S Chult, D. (2008). Exploring network structure, dynamics, and function using networkx. Tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Jamroga, W. (2004). Strategic planning through model checking of atl formulae. In *International Conference on Artificial Intelligence and Soft Computing*, pp. 879–884. Springer.

Jha, S., Raman, V., Sadigh, D., & Seshia, S. A. (2018). Safe autonomy under perception uncertainty using chance-constrained temporal logic. *Journal of Automated Reasoning*, *60*(1), 43–62.

Kallenberg, L. C. M. (1983). *Linear programming and finite Markovian control problems*. Mathematisch Centrum, Amsterdam.

Kemeny, J., & Snell, J. L. (1963). *Finite Markov chains*. Springer-Verlag, New York.

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

Kwiatkowska, M., & Parker, D. (2013). Automated verification and strategy synthesis for probabilistic systems. In *Automated Technology for Verification and Analysis*, pp. 5–22. Springer.

Lakshmanan, K., & Bhatnagar, S. (2012). A novel Q-learning algorithm with function approximation for constrained Markov decision processes. In *50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 400–405. IEEE.

Lazar, A. (1983). Optimal flow control of a class of queueing networks in equilibrium. *IEEE Transactions on Automatic Control*, *28*(11), 1001–1007.

Lindemann, L., & Dimarogonas, D. V. (2017). Robust motion planning employing signal temporal logic. In *2017 American Control Conference (ACC)*, pp. 2950–2955. IEEE.

Manne, A. S. (1960). Linear programming and sequential decisions. *Management Science*, *6*(3), 259–267.

Nilsson, P., Hussien, O., Balkan, A., Chen, Y., Ames, A. D., Grizzle, J. W., Ozay, N., Peng, H., & Tabuada, P. (2015). Correct-by-construction adaptive cruise control: Two approaches. *IEEE Transactions on Control Systems Technology*, *24*(4), 1294–1307.

Norris, J. R. (1997). *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Petrik, M., & Zilberstein, S. (2009). A bilinear programming approach for multiagent planning. *Journal of Artificial Intelligence Research*, *35*, 235–274.

Pistore, M., Bettin, R., & Traverso, P. (2014). Symbolic techniques for planning with extended goals in non-deterministic domains. In *Sixth European Conference on Planning*.

Privault, N. (2018). *Understanding Markov Chains: Examples and Applications*. Springer Singapore, Singapore.

Puterman, M. (1994). *Markov decision processes : discrete stochastic dynamic programming*. Wiley, New York.

Ross, K. W. (1989). Randomized and past-dependent policies for Markov decision processes with multiple constraints. *Operations Research*, *37*(3), 474–477.

Royden, H., & Fitzpatrick (2010). *Real Analysis* (4th edition). Pearson.

Schwarting, W., Alonso-Mora, J., & Rus, D. (2018). Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, *1*(1), 187–210.

Skwirzynski, J. K. (1981). *New concepts in multi-user communication*, Vol. 43. Springer Science & Business Media.

Song, L., Feng, Y., & Zhang, L. (2015). Planning for stochastic games with co-safe objectives. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Tarjan, R. E. (1972). Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, *1*, 146–160.

Tian, Z. (2019). United states law and policy on space debris. In *Space Security and Legal Aspects of Active Debris Removal*, pp. 155–167. Springer.

Trevizan, F., Thiébaux, S., & Haslum, P. (2017). Occupation measure heuristics for probabilistic planning. In *Twenty-Seventh International Conference on Automated Planning and Scheduling*.

Trevizan, F., Thiébaux, S., Santana, P., & Williams, B. (2016). Heuristic search in dual space for constrained stochastic shortest path problems. In *Twenty-Sixth International Conference on Automated Planning and Scheduling*.

Trevizan, F., Thiébaux, S., Santana, P., & Williams, B. (2017). I-dual: solving constrained ssps via heuristic search in the dual space. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 4954–4958.

Velasquez, A. (2019). Steady-state policy synthesis for verifiable control. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 5653–5661. AAAI Press.

Wongpiromsarn, T., Topcu, U., Ozay, N., Xu, H., & Murray, R. M. (2011). TuLiP: a software toolbox for receding horizon temporal logic planning. In *Proceedings of the 14th International Conference on Hybrid Systems: Computation and Control*, pp. 313–314. ACM.

Wu, J., & Durfee, E. H. (2010). Resource-driven mission-phasing techniques for constrained agents in stochastic environments. *Journal of Artificial Intelligence Research (JAIR)*, *38*, 415–473.

Zhou, Y., Maity, D., & Baras, J. S. (2016). Timed automata approach for motion planning using metric interval temporal logic. In *2016 European Control Conference (ECC)*, pp. 690–695. IEEE.