Steady-State Policy Synthesis for Verifiable Control

Alvaro Velasquez
Information Directorate, Air Force Research Laboratory
alvaro.velasquez.1@us.af.mil

Abstract

In this paper, we introduce the Steady-State Policy Synthesis (SSPS) problem which consists of finding a stochastic decision-making policy that maximizes expected rewards while satisfying a set of asymptotic behavioral specifications. These specifications are determined by the steady-state probability distribution resulting from the Markov chain induced by a given policy. Since such distributions necessitate recurrence, we propose a solution which finds policies that induce recurrent Markov chains within possibly non-recurrent Markov Decision Processes (MDPs). The SSPS problem functions as a generalization of steady-state control, which has been shown to be in **PSPACE**. We improve upon this result by showing that SSPS is in **P** via linear programming. Our results are validated using CPLEX simulations on MDPs with over 10000 states. We also prove that the deterministic variant of SSPS is **NP-hard**.

1 Introduction

The problem of designing decision-making agents which satisfy formal behavioral properties is an important challenge to overcome in an increasingly automated and autonomous world [Russell *et al.*, 2015]. We explore this problem through the lens of verifiable probabilistic planning and present a model-based solution to the problem of finding an optimal stochastic policy in a Markov Decision Process (MDP) subject to constraints on the steady-state behavior of the agent. As such, we must reason about the frequency with which states are visited by computing the steady-state distribution of the Markov chain induced by a given decision-making policy. However, this distribution is only well-defined if the Markov chain is recurrent. That is, there must exist a path between every pair of nodes [Cinlar, 2013]. Akshay *et al.* solve this problem for ergodic MDPs where it is assumed that all policies lead to recurrent Markov chains and the desired steady-state distribution over all states is provided by the user [Akshay *et al.*, 2013]. This is called the Steady-State Control (SSC) problem. In the labeled SSC (L-SSC) variant, the steady-state probability is defined over a partition of the state space. In this case, the problem can be posed as a bilinearly-constrained program and it is therefore determined that L-SSC is in **PSPACE** [Akshay *et al.*, 2013].

In this paper, we make three main contributions. First, we introduce the Steady-State Policy Synthesis (SSPS) problem as a generalization of the SSC/L-SSC problems and as a means of enforcing verifiable behavior in terms of steady-state properties within probabilistic state-transition systems whose goal is to maximize an expected reward signal. Our approach does not assume that the underlying MDP is recurrent or ergodic. An ergodic MDP, also known as a unichain MDP, is called ergodic if every policy induces an ergodic Markov chain [Kearns and Singh, 2002]. This is a particularly strong assumption considering that the problem of checking whether an MDP is ergodic is **NP-complete** [Tsitsiklis, 2007]. Interestingly, checking whether a deterministic MDP is ergodic is in **P** [McCuaig, 1993]. Our approach finds a stochastic policy which induces a recurrent Markov chain within a possibly non-recurrent MDP, if one exists. As our second contribution, we improve upon the complexity results in [Akshay *et al.*, 2013] by demonstrating that SSPS is in **P** via a linear programming formulation and, therefore, the L-SSC problem is also in **P**. Finally, we prove that the deterministic policy variant of the proposed SSPS problem is **NP-hard**. To the best of the authors' knowledge, this is the first attempt at verifiable control within non-ergodic MDPs which allows for steady-state specifications as well as an optimization objective.

The remainder of this paper is organized as follows. Section 2 covers preliminary material on probabilistic planning and properties of Markov chains. Section 3 provides a brief exposition of related work in verifiable control. The Steady-State Policy Synthesis (SSPS) problem is formally defined in Section 4 and its connection to existing steady-state control problems is explored. Section 5 follows with a solution to SSPS and complexity results for the deterministic variant of SSPS are established in Section 6. CPLEX simulations are used in Section 7 to validate our approach as well as demonstrate its efficiency in solving problems with over ten thousand states. A brief discussion on the challenges of integrating temporal logic and steady-state constraints is presented in Section 8. Concluding remarks follow in Section 9.

Approved for public release: distribution unlimited, case 88ABW-2018-4584. Cleared September 13, 2018.

2 Preliminaries

We briefly define the structures and types of specifications used in our approach.

Definition 1 (Markov Decision Process (MDP))**.** *An MDP is a non-deterministic probabilistic automaton represented by the tuple $\mathcal{M} = (S, A, T, R)$, where S is the set of states, A is the set of actions, $T : S \times A \times S \mapsto [0, 1]$ is the transition function with $T(s'|s, a)$ denoting the probability of transitioning from state s to state s' when action a is taken, and $R : S \times A \mapsto \mathbb{R}$ is a reward signal observed when action a is taken in state s. We denote by $A(s) \subseteq A$ the set of actions available in state s.*

It is worth noting that MDPs and Markov chains are usually defined with an initial state $s_{\text{init}} \in S$ or initial probability distribution over states. However, since we are reasoning about steady-state distributions, our approach is independent of the initial state. As such, we will only refer to such states for illustrative purposes and will omit them from technical discussion.

Solutions to an MDP take the form of a policy $\pi : S \times A \mapsto [0, 1]$ specifying the probability of taking an action in a given state. Thus, we can think of $\pi(a|s), \sum_{a \in A(s)} \pi(a|s) = 1$ as a conditional probability distribution. Any such policy resolves the non-determinism in the underlying MDP and gives rise to a Markov chain (See Definition 2). In particular, we are interested in the asymptotic behavior of the agent's policy as captured by the steady-state distribution of the resulting Markov chain (See Definition 3).

Definition 2 (Markov Chain)**.** *A Markov chain is a pair $\mathcal{M} = (S, T)$ with state set S and transition probability function $T : S \times S \mapsto [0, 1]$, where $T(s'|s)$ denotes the probability of transitioning from s to s' and $\sum_{s' \in S} T(s'|s) = 1$ for every state $s \in S$. For convenience, the transition function can also be thought of as a matrix $T \in [0, 1]^{|S| \times |S|}$, where $T \ni T_{ss'} = T(s'|s)$. We do not belabor this difference when the use of T is clear from the context.*

Definition 3 (Steady-State Distribution)**.** *Given a Markov chain $\mathcal{M} = (S, T)$, the steady-state distribution $Pr^{\infty} : S \mapsto [0, 1]$, $\sum_{s \in S} Pr^{\infty}(s) = 1$, also known as the stationary or invariant distributions, over the state space denotes the proportion of time spent in each state as the number of transitions within $\mathcal{M}$ approaches ∞. This distribution is given by the solution to the system of equations in (1) [Konstantopoulos, 2009].*

Given an MDP $\mathcal{M} = (S, A, T, R)$, we will often reason about the binary transition relation $T^{\text{rel}} = \{(s, s') \in S \times S | s \neq s' \wedge \exists a \in A(s), T(s'|s, a) > 0\}$ consisting of the edges in $\mathcal{M}$, not including self-loops. Here, the term edge is used to refer to a non-zero probability of transitioning between two different states. This binary transition relation will be used in program (7) to simplify the equations used to determine whether the Markov chain induced by a solution policy is recurrent.

Whenever we refer to a Markov chain induced by a policy $\pi : S \times A \mapsto [0, 1]$ in an underlyng MDP $\mathcal{M} = (S, A, T, R)$, we will utilize the notation $\mathcal{M}_\pi = (S, T_\pi)$, where T_π can be computed from T via equation (2). We will similarly use this subscript to refer to the steady-state distribution of such a Markov chain by $\Pr^{\infty}_{\pi}(\cdot)$.

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

$$T_\pi(s'|s) = \sum_{a \in A(s)} \Pr_\pi(s', a|s) = \sum_{a \in A(s)} T(s'|s, a)\pi(a|s) \tag{2}$$

It is worth noting that there is a valid solution to the steady-state set of equations (1) if the underlying Markov chain is recurrent [Ross, 2014]. That is, if all states can be reached from each other [Cinlar, 2013]. When this condition does not hold, the steady-state equations (1) can admit solutions which are not indicative of the true behavior of the system. See Figure 1 for an example.

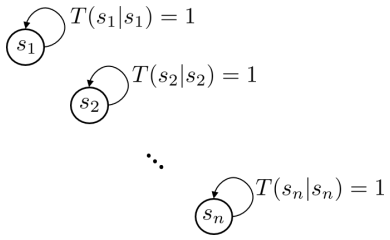

Figure 1: Consider the Markov chain $\mathcal{M} = (S, T)$, where each state has a single transition consisting of a self-loop. In effect, $T = I$ and a solution to (1) would be $\Pr^{\infty}(s_1) = \cdots = \Pr^{\infty}(s_n) = 1/n$, which is not representative of the true steady-state behavior of the system, i.e. $\Pr^{\infty}(s_{\text{init}}) = 1$, where s_{init} is the initial state of the system.

To elaborate on the definition of recurrence and how it is different from other Markov chain properties, we briefly discuss some of the theory behind Markov chains. In particular, the notion of recurrence can be defined in terms of communicating classes and irreducibility. Given a Markov chain $\mathcal{M} = (S, T)$, two states s_i and s_j are said to be in a communicating class if there exists a sequence of transitions such that there is a non-zero probability of transitioning from s_i to s_j and vice-versa. If all states in a Markov chain belong to the same communicating class (i.e. if the chain is strongly connected), then the chain is said to be irreducible. Furthermore, an irreducible Markov chain with finitely many states is also known as a recurrent Markov chain ([Konstantopoulos, 2009], Theorem 22). Alternatively, a state in a Markov chain is said to be recurrent if the chain returns to it infinitely many times. For Markov chains with a finite state space (i.e. the ones considered in this paper), there is always at least one recurrent state and all states in a communicating class are either recurrent or transient ([Konstantopoulos, 2009], Corollary 8). A Markov chain whose states are all recurrent is also called a recurrent Markov chain. We thus have two equivalent definitions for recurrence that are derived from the strong connectivity of the underlying Markov chain. This motivates our approach for finding recurrent Markov chains within possibly non-recurrent MDPs by searching for strongly connected chains. This is in contrast to ergodic assumptions on the MDP that are sometimes

made in the literature [Akshay *et al.*, 2013]. This assumption states that all policies in an MDP induce a Markov chain that is both recurrent and aperiodic. The addition of aperiodicity makes it so that $\lim_{n\to\infty}(T^n)_{s_i,s_j} = \Pr^{\infty}(s_j), \forall s_i \in S$ ([Konstantopoulos, 2009], Theorem 17). That is, as the number of transitions within $\mathcal{M}$ approaches ∞, the probability of transitioning from an arbitrary state s_i to state s_j is equal to the steady-state probability of s_j. This facilitates analysis in certain problem spaces. A chain is said to be aperiodic if all of its states have a period of 1, where the period of a state s is defined as the greatest common divisor over $n \in \mathbb{N}$ such that there is a non-zero probability of going from s to itself in n transitions.

It is the view of the authors that the assumption of ergodicity or recurrence in the underlying MDP is quite a strong one. Thus, the method proposed in this paper finds a recurrent Markov chain in a possibly non-recurrent MDP, if one exists. As previously mentioned, such recurrence implies a valid steady-state distribution of the Markov chain. We are interested in solving MDPs subject to verifiable asymptotic behavior by imposing constraints on said steady-state distribution of the state space. The addition of these constraints on an MDP leads to a labeled MDP (LMDP) (See Definition 5). Labels are often used in verification to partition the state space into subsets of states that share similar properties. A solution to an LMDP $\mathcal{M}$ is a stochastic policy $\pi : S \times A \mapsto [0, 1]$ whose induced Markov chain $\mathcal{M}_\pi$ maximizes the expected reward signal while satisfying the given steady-state specifications defined below, which allow us to bound the frequency with which sets of states are visited.

Definition 4 (Steady-State Specification)**.** *Given an MDP $\mathcal{M} = (S, A, T, R)$ 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 $\Phi^{\infty_{\mathbb{L}}} = \{(L_i, [l_i, u_i])\}_{i=1}^{n_L}$. Given a policy π, the specification $(L_i, [l, u]) \in \Phi^{\infty_{\mathbb{L}}}$ is satisfied if and only if $\sum_{s\in L_i} Pr_\pi^\infty(s) \in [l, u]$. That is, if the steady-state probability of being in some state $s \in L_i$ in the Markov chain induced by π falls within the interval $[l, u]$. In the case of equality constraints, we use $\{u\}$ to denote the interval $[l, u]$ when $l = u$.*

Definition 5 (Labeled MDP (LMDP))**.** *An LMDP is an augmented MDP $\mathcal{M} = (S, A, T, R, L, \Phi^{\infty_{\mathbb{L}}})$, where $L = \{L_1, \ldots, L_{n_L}\}, (L_i \subseteq S)$ is a set of labels and specifications $\Phi^{\infty_{\mathbb{L}}} = \{(L_i, [l_i, u_i])\}_{i=1}^{n_L}$ are of the form described in Definition 4.*

Note that we can also bound the frequency of individual states by defining labels that consist of a single element. Specifications of the form $(L_i, \{0\})$ capture the safety notion that the probability of visiting certain states should be zero. For every such unsafe state(s), we would like to ensure that $\Pr_\pi^\infty(s) = 0$ for any feasible policy π. For simplicity, we preprocess the LMDP in order to remove any such states as well as any states for which every policy leads to the undesirable state(s). See Figure 2 an example.

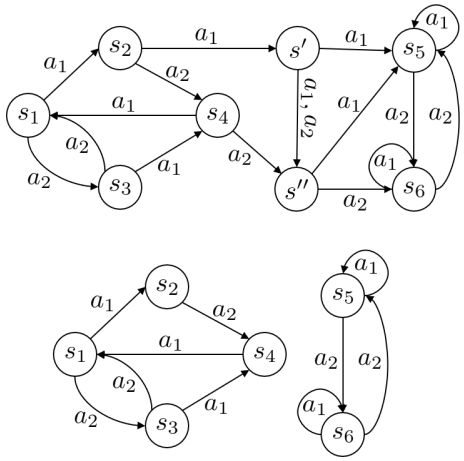

Figure 2: Given an LMDP $\mathcal{M}' = (S', A, T', R, L, \Phi^{\infty_{\mathbb{L}}})$ with steady-state specification $(\{s''\}, \{0\}) \in \Phi^{\infty_{\mathbb{L}}}$ (*top*), note that, for any policy $\pi : S \times A \mapsto [0, 1]$, the induced Markov chain $\mathcal{M}_\pi = (S', T_\pi)$ will be such that $T_\pi(s''|s') > 0$. That is, the probability of transitioning from s' to s'' is non-zero regardless of the policy. Thus, any policy which satisfies $\Phi^{\infty_{\mathbb{L}}}$ must not visit s', s'' and we can remove these states from the LMDP. This leads to the modified LMDP $\mathcal{M} = (S, A, T, R, L, \Phi^{\infty_{\mathbb{L}}} \setminus (\{s''\}, \{0\}))$ (*bottom*).

3 Related Work

Mathematical programming approaches have been proposed to solve optimal control problems subject to formal behavioral specifications encoded in some logic [Feyzabadi, 2017] [Wolff and Murray, 2016]. An interesting approach based on linear programming heuristics is proposed in [Trevizan *et al.*, 2016] and [Baumgartner *et al.*, 2018] to solve a constrained stochastic shortest path problem. This problem entails the minimization of cost subject to reachability and probabilistic Linear Temporal Logic (pLTL) constraints. When a polynomial-time solution is not known for the non-convex primal linear program, its convex dual formulation is used in a sub-procedure to solve the original problem. In the primal linear program, variables are naturally used to represent reachability costs. However, the dual linear program is defined over occupancy measures which denote the frequency with which a state-action pair is observed. This functions as the state-action pair analogue of steady-state probabilities, which capture the frequency of visiting individual states. Occupancy measures have also been used to find policies that satisfy certain probabilistic Computation Tree Logic (pCTL) operations, such as the probabilistic *strong until* operator [Teichteil-Königsbuch, 2012] via the use of iterative linear programming, and to minimize various cost metrics associated with reaching a set of states in constrained MDPs [Altman, 1999].

There has been some work done in reasoning about steady-state specifications as well. Indeed, the method proposed in [Akshay *et al.*, 2013] finds stochastic policies that satisfy a given steady-state distribution via the use of linear programs. The case where the state space S is partitioned into pairwise disjoint labeled subsets $\{L_1, \ldots, L_{n_L}\}, (L_i \subset S, L_i \cap L_j = \emptyset)$ is also considered. This is called the labeled SSC (L-SSC) problem. When the partition defined by these labels must satisfy a given distribution, the problem is shown to be in **PSPACE** by formulating it as a bilinearly-constrained program. This follows from the fact that bilinearly-constrained

programming in general is in **PSPACE** [Canny, 1988]. We improve upon this complexity result by showing that L-SSC is in **P**.

It is worth noting that the expressiveness of specifications allowed by the SSPS problem is not equivalent to that of the logics mentioned. There exist specifications that can be formulated within SSPS which cannot be formulated by pCTL/pLTL, and vice-versa. For example, steady-state properties have no pCTL/pLTL equivalent and specifications of the form $\phi \mathbf{U} \psi, (\phi, \psi \in \{0, 1\})$ cannot be specified within SSPS using steady-state specifications, where $\phi \mathbf{U} \psi$ holds if and only if ψ holds in some state in the future and ϕ holds in all states until ψ holds.

In the literature, it is common to make strong assumptions about the underlying MDP when reasoning about occupancy measures and steady-state distributions. For example, the assumption that the MDP is transient [Altman, 1999] or ergodic [Akshay *et al.*, 2013]. The former implies that every policy induces a Markov chain wherein a subset of states have finite expected time spent while the latter assumes that all policies yield recurrent, aperiodic Markov chains. In particular, we have previously mentioned how the ergodicity assumption is convenient as recurrence implies a well-defined steady-state distribution that is independent of the initial state of the system. In this work, we make no such assumptions on the underlying MDP. We propose a solution which finds a recurrent Markov chain within a possibly non-recurrent MDP, if one exists, subject to constraints on the steady-state distribution of the state space while maximizing the average reward signal of the Markov chain induced by the solution policy.

4 Steady-State Policy Synthesis

We begin our exposition of steady-state policy synthesis (SSPS) with a simple example. Suppose there is a robotic agent in an environment as shown in Figure 3, where the underlying dynamics are given by the LMDP $\mathcal{M} = (S, A, T, R, L, \Phi^{\infty_{\mathbb{L}}})$. We have $S = \{s_1, \ldots, s_{16}\}$, $A = \{\leftarrow, \uparrow, \rightarrow, \downarrow\}$, and $T(s'|s, a) \in \{0, 1\}$ can be visualized in the figure. The robot starts in the top-left corner of the map and observes a positive reward $R(s_9, \downarrow) = R(s_{14}, \leftarrow) \in \mathbb{R}^+$ associated with recharging its energy source in state s_{13}. The mission of the agent is to attempt to establish communication with a satellite via the telecommunication links in states s_3, s_4, s_7, s_8. Thus, we want to make sure the agent spends at least 70% of its time in these states. Successfully establishing communication will cause a spaceship to arrive at the rendezvous point s_{16} and take the agent home, thereby accomplishing the mission. The agent does not know if or when the spaceship will arrive, so it must check back periodically to state s_{16}, but not so often that it interrupts the task of establishing communication or recharging batteries. We therefore want to visit s_{16} at least 1% of the time, but no more than 10% of the time. During this mission of establishing communication, recharging batteries, and visiting the rendezvous point, the agent must avoid states s_9, s_{10}, s_{12} which contain radioactive material. It follows that our specifications can be given by $\Phi^{\infty_{\mathbb{L}}} = \{(L_{\text{comm}}, [0.7, 1]), (L_{\text{unsafe}}, \{0\}), (\{s_{16}\}, [0.01, 0.1])\}$, where

Figure 3: An agent within an environment whose dynamics can be modeled by an LMDP $\mathcal{M} = (S, A, T, R, \Phi^{\infty_{\mathbb{L}}})$. In this example, the state-transition probability is given by $T(s_j|s_i, a) = 1$ for the appropriate action, e.g. $T(s_2|s_1, \rightarrow) = 1$ and $T(\cdot|s_1, \leftarrow) = 0$.

labels $L_{\text{comm}} = \{s_3, s_4, s_7, s_8\}$ and $L_{\text{unsafe}} = \{s_9, s_{10}, s_{12}\}$ denote the sets of communication links and unsafe states, respectively.

The goal of SSPS is to find a stochastic policy $\pi : S \times A \mapsto [0, 1]$ which induces a Markov chain $\mathcal{M}_\pi$ that satisfies all the specifications in a given LMDP $\mathcal{M} = (S, A, T, R, \Phi^{\infty_{\mathbb{L}}})$ while maximizing an objective function based on R. In the next section, we propose a novel linear programming formulation to solve SSPS. For the preceding example, this linear program yields the policy π (3). In the Markov chain $\mathcal{M}_\pi = (S, T_\pi)$ induced by this policy, where T_π is computed from T and π as shown in (2), we have: $\Pr_\pi^\infty(s_{16}) \approxeq 0.0179$, $\sum_{s \in L_{\text{comm}}} \Pr_\pi^\infty(s) \approxeq 0.70967$, and $\sum_{s \in L_{\text{unsafe}}} \Pr_\pi^\infty(s) = 0$. Thus, π satisfies all of the desired specifications.

$$\pi(a|s) = \begin{array}{c} \begin{matrix} \leftarrow & \uparrow & \rightarrow & \downarrow \end{matrix} \\ \begin{pmatrix} 0.0 & 0.0 & 0.53543 & 0.46457 \\ 0.25545 & 0.0 & 0.54298 & 0.20157 \\ 0.12793 & 0.0 & 0.58195 & 0.29012 \\ 0.66285 & 0.0 & 0.0 & 0.33715 \\ 0.0 & 0.53893 & 0.46107 & 0.0 \\ 0.25369 & 0.25807 & 0.48824 & 0.0 \\ 0.09746 & 0.176 & 0.63112 & 0.09542 \\ 0.7107 & 0.2893 & 0.0 & 0.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.0 & 0.56166 & 0.0 & 0.43834 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.2939 & 0.0 & 0.7061 & 0.0 \\ 0.45054 & 0.26172 & 0.28774 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 \end{pmatrix} \end{array} \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \\ s_9 \\ s_{10} \\ s_{11} \\ s_{12} \\ s_{13} \\ s_{14} \\ s_{15} \\ s_{16} \end{matrix} \quad (3)$$

For the remainder of this section, we formally define the SSPS problem and the SSC and L-SSC problems that it subsumes. Namely, the SSC and L-SSC problems are degenerate cases of SSPS where it is assumed that the underlying MDP is ergodic, there is no optimization objective, no reward signal, labels are pairwise disjoint, and steady-state specifica-

tions only permit equalities. We also formulate our objective function in terms of the steady-state distribution and propose a linear program to maximize said objective while computing the steady-state distribution of the Markov chain induced by the solution policy. In the next section, we build upon this foundation in order to integrate steady-state specifications and recurrence to solve the SSPS problem.

Definition 6 (Steady-State Control (SSC) Problem [Akshay *et al.*, 2013])**.** *Given an ergodic LMDP $\mathcal{M} = (S, A, T, R = \emptyset, L = \emptyset, \Phi^{\infty_{\mathbb{L}}})$ and a desired distribution over the state space $\delta : S \mapsto [0, 1], \sum_{s \in S} \delta(s) = 1$, find a stochastic policy $\pi : S \times A \mapsto [0, 1]$ such that the Markov Chain $\mathcal{M}_\pi$ induced by π satisfies $(s, \{\delta(s)\}) \in \Phi^{\infty_{\mathbb{L}}}$ for every $s \in S$.*

Definition 7 (Labeled Steady-State Control (L-SSC) Problem [Akshay *et al.*, 2013])**.** *Given an ergodic LMDP $\mathcal{M} = (S, A, T, R = \emptyset, L, \Phi^{\infty_{\mathbb{L}}})$ with pairwise disjoint labels $L = \{L_1, \ldots, L_{n_L}\}, L_i \cap L_j = \emptyset$ and a desired distribution over the label space $\delta : L \mapsto [0, 1], \sum_{L_i \in L} \delta(L_i) = 1$, find a stochastic policy $\pi : S \times A \mapsto [0, 1]$ such that the Markov Chain $\mathcal{M}_\pi$ induced by π satisfies $(L_i, \{\delta(L_i)\}) \in \Phi^{\infty_{\mathbb{L}}}$ for every $L_i \in L$.*

It is worth noting that, in [Akshay *et al.*, 2013], the steady-state control problems are originally defined as history-dependent. That is, they require a history of state-action pairs from the agent in order to determine the policy $\pi : (S \times A)^* \mapsto [0, 1]$ output. However, the authors then go on to show that finding a Markovian policy which is a function of a single state-action pair suffices and the linear programs proposed therein find such Markovian policies $\pi : S \times A \mapsto [0, 1]$, as we do in our approach, as opposed to history-dependent ones. Note that the steady-state control (SSC) problem is a special case of L-SSC where each label is defined by a single state. It follows from Definitions 6 and 7 that SSC and L-SSC only reason about restricted cases of LMDPs, whereas the SSPS problem defined below pertains to general LMDPs.

Definition 8 (Steady-State Policy Synthesis (SSPS) Problem)**.** *Given an LMDP $\mathcal{M} = (S, A, T, R, L, \Phi^{\infty_{\mathbb{L}}})$, find a stochastic policy $\pi : S \times A \mapsto [0, 1]$ such that the Markov chain $\mathcal{M}_\pi = (S, T_\pi)$ induced by π is recurrent, maximizes the objective function (4), and satisfies the steady-state specifications in $\Phi^{\infty_{\mathbb{L}}}$.*

$$\max \sum_{s \in S} \Pr_\pi^\infty(s) \sum_{a \in A(s)} \pi(a|s) R(s, a) \tag{4}$$

We have defined our objective function (4) in a way which leverages the steady-state distribution values that must be computed. This formulation is referred to as the average-reward objective [Sutton and Barto, 2018] that is used in reinforcement learning with continuing tasks as opposed to episodic ones. This definition allows us to do away with user-defined discount factors by considering the asymptotic reward obtained by an agent as opposed to its discounted future reward. For simplicity, we first present mathematical programs that only attempt to solve this optimization objective and associated steady-state equations without taking into account any steady-state specifications. These simple programs (5), (6) also do not search for a policy that induces a recurrent Markov chain. In the next section, we demonstrate how steady-state specifications and the search for recurrent Markov chains can then be added in the form of constraints to the linear program (6). Recall that the policy π specifies the probability of taking action a in state s and can be modeled as a conditional probability $\pi(a|s)$.

$$\begin{aligned}
&\max \sum_{s \in S} \Pr_\pi^\infty(s) \sum_{a \in A(s)} \pi(a|s) R(s, a) \text{ subject to} \\
&(i) \sum_{s \in S} \Pr_\pi^\infty(s) \sum_{a \in A(s)} \pi(a|s) T(s'|s, a) = \Pr_\pi^\infty(s') \;\; \forall s' \in S \\
&(ii) \sum_{s \in S} \Pr_\pi^\infty(s) = 1 \\
&(iii) \sum_{a \in A(s)} \pi(a|s) = 1 \qquad \forall s \in S \\
&\Pr_\pi^\infty(s), \pi(a|s) \in [0, 1] \qquad \forall s \in S, a \in A
\end{aligned} \tag{5}$$

The first constraint in program (5) is bilinear and entails the system of equations used to determine the steady-state probabilities of $\mathcal{M}_\pi = (S, T_\pi)$. Recall from (2) that the inner sum is equivalent to $T_\pi(s'|s)$. Clearly, all such probabilities must add up to unity, as given by the second constraint. The third constraint encodes the restriction that an action must be taken in each state and the action probabilities must add up to unity. The preceding program may not have an efficient solution since bilinearly-constrained programs are **NP-hard** problems in general [Floudas and Visweswaran, 1995]. Since these programs are also in **PSPACE** [Canny, 1988], it is determined in [Akshay *et al.*, 2013] that L-SSC must be in **PSPACE** due to a program formulation similar to (5). However, we can reformulate (5) as the following linear program (6).

$$\begin{aligned}
&\max \sum_{s \in S} \sum_{a \in A(s)} x_{sa} R(s, a) \text{ subject to} \\
&(i) \sum_{s \in S} \sum_{a \in A(s)} x_{sa} T(s'|s, a) = \Pr_\pi^\infty(s') \quad \forall s' \in S \\
&(ii) \sum_{s \in S} \Pr_\pi^\infty(s) = 1 \\
&(iii) \sum_{s \in S} \sum_{a \in A(s)} x_{sa} = 1 \\
&(iv) \sum_{a \in A(s)} x_{sa} = \Pr_\pi^\infty(s) \qquad \forall s \in S \\
&\Pr_\pi^\infty(s), x_{sa} \in [0, 1] \qquad \forall s \in S, a \in A
\end{aligned} \tag{6}$$

We have introduced a new variable x_{sa} to replace the bilinear term $\Pr_\pi^\infty(s)\pi(a|s)$. However, we have not added a constraint to specify that $x_{sa} = \Pr_\pi^\infty(s)\pi(a|s)$ since this would again yield a bilinear program. Therefore, we must prove that this equation holds for any feasible solution to (6).

Theorem 1. *If $(x_{sa})_{|S| \times |A|}, (Pr_\pi^\infty(s))_{|S|}$ is a feasible solution to (6), then $\pi(a|s) = x_{sa}/Pr_\pi^\infty(s)$ is a feasible policy.*

Proof. It follows from constraint (iii) that x_{sa} is a probability distribution $\Pr(s,a)$ over the space of state-action pairs. Constraint (iv) ensures that the marginal distribution of x_{sa} over the state space is equal to the steady-state distribution $\Pr^{\infty}_{\pi}(s)$. Thus, we have $x_{sa} = \Pr(s,a) = \pi(a|s)\Pr^{\infty}_{\pi}(s)$. Note that, for any state $s \in S$, the following holds due to (iv):

$$\sum_{a \in A(s)} \pi(a|s) = \frac{\sum_{a \in A(s)} x_{sa}}{\Pr^{\infty}_{\pi}(s)} = 1$$

Thus, $\pi(a|s) = x_{sa}/\Pr^{\infty}_{\pi}(s)$ is a distribution and the resulting policy.

□

For states with $\Pr^{\infty}_{\pi}(s) = 0$, an arbitrary choice of action can be made. In the next section, we propose a general linear program to solve any LMDP $\mathcal{M} = (S, A, T, R, L, \Phi^{\infty_{\mathbb{L}}})$. Specifications in $\Phi^{\infty_{\mathbb{L}}}$ can be easily accounted for using linear constraints. However, ensuring that the resulting Markov chain is recurrent requires a more complex use of network flow theory.

5 Recurrence and Steady-State Specifications

In this section, we take the linear program (6) and add constraints to ensure that the Markov chain $\mathcal{M}_\pi = (S, T_\pi)$ induced on an arbitrary LMDP $\mathcal{M} = (S, A, T, R, L, \Phi^{\infty_{\mathbb{L}}})$ by the solution policy $\pi : S \times A \mapsto [0,1]$ is recurrent and satisfies the given steady-state specifications $\Phi^{\infty_{\mathbb{L}}}$. Constraints $(i), (iv)$ and $(ii), (iii)$ in (6) are combined into two constraints in order to eliminate the variables $\Pr^{\infty}_{\pi}(s) = \sum_{a \in A(s)} x_{sa}$ corresponding to the steady-state probabilities. In the extended program (7), new variables $f_{s,s'}, f^{\text{rev}}_{s,s'} \in [0,1]$ are introduced for every $(s,s') \in T^{\text{rel}}$. Recall that $T^{\text{rel}} = \{(s,s') \in S \times S | s \neq s' \wedge \exists a \in A(s), T(s'|s,a) > 0\}$ is the graph structure of $\mathcal{M}$ consisting of the transitions that have non-zero probability measure, not including self-loops. Similarly, let $T^{\text{rel}}_{\pi} = \{(s,s') \in S \times S | s \neq s' \wedge T_\pi(s'|s) > 0\} \subseteq T^{\text{rel}}$ denote the graph structure of $\mathcal{M}_\pi$. These new variables capture the notion of flow between two adjacent nodes along the direction of the edge $(s,s') \in T^{\text{rel}}$ and its reverse (s',s), respectively. In effect, we seek to use flow arguments to ensure that the graph formed by T^{rel}_{π} is strongly connected in order to have a well-defined steady-state distribution. Constraints (iii) and (iv) of program (7) originate flow from an arbitrary random initial state s_{init} to adjacent outgoing and incoming states, respectively. In this context, flow is the consequence of non-zero probability of transitioning between two states in $\mathcal{M}_\pi$. We can thus think of $f_{s,s'}$ for any $s, s' \in S$ as being similar to the transition probability $T_\pi(s'|s)$ of $\mathcal{M}_\pi$. However, we cannot pose constraints explicitly as a function of $T_\pi(s'|s)$ since this would require access to the policy variables $\pi(a|s)$ as can be seen in equation (2). For this reason, $f_{s_{\text{init}},s'}, f^{\text{rev}}_{s_{\text{init}},s'}$ are initialized to be implicitly proportional to $T_\pi(s'|s)$ as can be seen in equation (8). Similarly, in constraints (v) and (vi), the flow capacity between two states s, s' is implicitly set proportionally to $T_\pi(s'|s)$ as can be seen in equation (9). The flow transfer constraints (vii) and $(viii)$ ensure that the incoming flow into a node is strictly greater the outgoing flow for all states, except the randomly chosen initial state s_{init}. This serves a critical purpose. Namely, it forces the initial state to act as the sole producer of flow while all other states act as consumers. It follows that there is incoming flow $f_{s',s}$ ($f^{\text{rev}}_{s',s}$) into an arbitrary state $s \in S$ if and only if there exists a path from s_{init} to s (s to s_{init}) in $\mathcal{M}_\pi$. Constraints (ix) and (x) ensure that there is incoming flow into all states. In practice, such strict inequality constraints can be transformed to bounded inequalities by adding an arbitrarily small constant to the right-hand side of the constraint. For a visualization of these flow constraints, see Figure 4. Finally, (xi) is satisfied if and only if every steady-state specification in $\Phi^{\infty_{\mathbb{L}}}$ is satisfied. We prove the correctness of program (7) in Theorem 2.

$$
\begin{aligned}
&\max \sum_{s \in S} \sum_{a \in A(s)} x_{sa} R(s,a) \text{ subject to} \\
&(i) \sum_{s \in S} \sum_{a \in A(s)} x_{sa} T(s'|s,a) = \sum_{a \in A(s')} x_{s',a} && \forall s' \in S \\
&(ii) \sum_{s \in S} \sum_{a \in A(s)} x_{sa} = 1 \\
&(iii)\ f_{s_{\text{init}},s'} = \sum_{a \in A(s_{\text{init}})} T(s'|s_{\text{init}},a) x_{s_{\text{init}},a} && \forall (s_{\text{init}}, s') \in T^{\text{rel}} \\
&(iv)\ f^{\text{rev}}_{s_{\text{init}},s'} = \sum_{a \in A(s')} T(s_{\text{init}}|s',a) x_{s',a} && \forall (s', s_{\text{init}}) \in T^{\text{rel}} \\
&(v)\ f_{s,s'} \leq \sum_{a \in A(s)} T(s'|s,a) x_{s,a} && \forall (s,s') \in T^{\text{rel}} \\
&(vi)\ f^{\text{rev}}_{s,s'} \leq \sum_{a \in A(s')} T(s|s',a) x_{s',a} && \forall (s',s) \in T^{\text{rel}} \\
&(vii) \sum_{(s',s) \in T^{\text{rel}}} f_{s',s} > \sum_{(s,s') \in T^{\text{rel}}} f_{s,s'} && \forall s \in S \setminus \{s_{\text{init}}\} \\
&(viii) \sum_{(s,s') \in T^{\text{rel}}} f^{\text{rev}}_{s',s} > \sum_{(s',s) \in T^{\text{rel}}} f^{\text{rev}}_{s,s'} && \forall s \in S \setminus \{s_{\text{init}}\} \\
&(ix) \sum_{(s',s) \in T^{\text{rel}}} f_{s',s} > 0 && \forall s \in S \\
&(x) \sum_{(s,s') \in T^{\text{rel}}} f^{\text{rev}}_{s',s} > 0 && \forall s \in S \\
&(xi)\ l \leq \sum_{s \in L_i} \sum_{a \in A(s)} x_{sa} \leq u && \forall (L_i, [l,u]) \in \Phi^{\infty_{\mathbb{L}}} \\
&\qquad x_{sa}, f_{s,s'}, f^{\text{rev}}_{s,s'} \in [0,1] \quad \forall s \in S, a \in A, (s,s') \in T^{\text{rel}}
\end{aligned}
\tag{7}
$$

Theorem 2. *Given an LMDP $\mathcal{M} = (S, A, T, R, L, \Phi^{\infty_{\mathbb{L}}})$, the linear program (7) is feasible if and only if there exists a stochastic policy $\pi : S \times A \mapsto [0,1]$ such that the induced Markov chain $\mathcal{M}_\pi = (S, T_\pi)$ is recurrent and satisfies the specifications in $\Phi^{\infty_{\mathbb{L}}}$.*

Proof. ($\implies$) Suppose (7) is feasible and let $(x_{sa})_{|S|\times|A|}, (f_{s,s'})_{|T^{\text{rel}}|}, (f^{\text{rev}}_{s,s'})_{|T^{\text{rel}}|}$ denote an arbitrary fea-

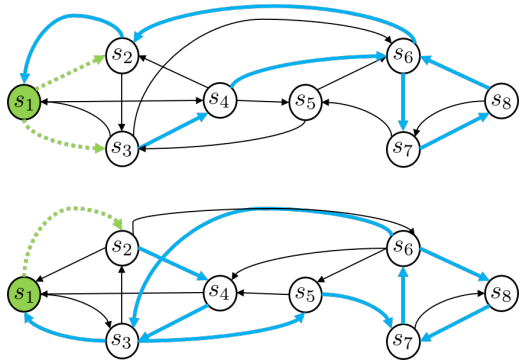

Figure 4: Visualization of the flow constraints in linear program (7). The top (bottom) graph represents the original (reversed) MDP with arbitrary initial state s_1. Dashed green lines indicate initial flow generated by constraints $(iii), (iv)$. Thick blue lines denote the edges enabled by constraints $(v), (vi)$. It is along these edges that flow can be transferred via constraints $(vii), (viii)$. Recall that the outgoing flow from a node is strictly less than the incoming flow. Thus, in order for the resulting Markov chain to be recurrent, all nodes must be reachable from the initial state in both graphs, which is the sole producer of flow. This is enforced by constraints $(ix), (x)$, which ensure that the incoming flow to all nodes is non-zero.

sible solution. Recall that we can compute the policy $\pi(a|s) = x_{sa}/\Pr_\pi^\infty(s) = x_{sa}/\sum_{a' \in A(s)} x_{s,a'}$ as stated in Theorem 1 and let $\mathcal{M}_\pi = (S, T_\pi)$ denote the Markov chain induced by π. We can reformulate constraints (iii) and (v) in terms of T_π as follows.

$$
\begin{aligned}
f_{s_{\text{init}},s'} &= \sum_{a \in A(s_{\text{init}})} T(s'|s_{\text{init}}, a) x_{s_{\text{init}},a} \\
&= \Pr_\pi^\infty(s_{\text{init}}) \sum_{a \in A(s_{\text{init}})} T(s'|s_{\text{init}}, a)\pi(a|s_{\text{init}}) \\
&= \Pr_\pi^\infty(s_{\text{init}}) \sum_{a \in A(s_{\text{init}})} \Pr_\pi(s', a|s_{\text{init}}) \\
&= T_\pi(s'|s_{\text{init}})\Pr_\pi^\infty(s_{\text{init}}) \qquad (8) \\
f_{s,s'} &\leq \sum_{a \in A(s)} T(s'|s, a) x_{s,a} = T_\pi(s'|s)\Pr_\pi^\infty(s) \qquad (9)
\end{aligned}
$$

From (8), we have an initial flow on the outgoing edges of s_{init} in $\mathcal{M}_\pi$. Equation (9) sets the flow capacities on edges (s, s') in $\mathcal{M}_\pi$ to be proportional to $T_\pi(s'|s)$ so that flow will be transferred from incoming to outgoing edges as stipulated by constraint (vii). Note that only those edges which correspond to transitions $T_\pi(s'|s) > 0$ will have flow. Indeed, consider an arbitrary state $s \in S$. From constraint (ix), the incoming flow into s must be non-zero. It follows that $f_{s',s} > 0$ for some $(s', s) \in T^{\text{rel}}$ in any feasible solution policy. Thus, from (9), we have $T_\pi(s|s') > 0$. It follows that there is an incoming transition into s in $\mathcal{M}_\pi$. Since there is only initial flow outgoing from s_{init} and for all other states the outgoing flow is less than the incoming flow, the preceding flow arguments inductively imply that all nodes in S are reachable from s_{init} in $\mathcal{M}_\pi$. By inverting the edges of $\mathcal{M}_\pi$, the same line of reasoning can be used to show that reverse reachability also holds via constraints $(iv), (vi), (viii), (x)$. Thus, $\mathcal{M}_\pi$ is recurrent. By observation, the specifications in $\Phi^{\infty_{\mathbb{L}}}$ are satisfied via constraint (xi).

($\Longleftarrow$) Suppose there exists a stochastic policy $\pi : S \times A \mapsto [0, 1]$ such that the induced Markov chain $\mathcal{M}_\pi = (S, T_\pi)$ is recurrent and satisfies the specifications in $\Phi^{\infty_{\mathbb{L}}}$. Then, by definition of recurrence, the steady-state distribution $\Pr_\pi^\infty(s)$ is well-defined. Thus, we can set $x_{sa} = \pi(a|s)\Pr_\pi^\infty(s)$ for every $s \in S$. Note that the flow variables in $(iii) - (vi)$ can now be defined in terms of x_{sa} and $T(s'|s, a)$ such that constraints $(vii) - (x)$ are satisfied.

□

6 Complexity

It follows from Theorem 2 that SSPS and, therefore, SSC and L-SSC, are solvable in polynomial time due to the time complexity of linear programming. However, if a deterministic policy $\pi : S \mapsto A$ is required as opposed to a stochastic one, then SSPS becomes **NP-hard**. We call this variant the deterministic SSPS (D-SSPS) problem, which is formally defined below.

Corollary 1. *SSPS, SSC, and L-SSC are in **P**.*

Definition 9 (Deterministic SSPS Problem (D-SSPS))**.** *Given an LMDP $\mathcal{M} = (S, A, T, R, L, \Phi^{\infty_{\mathbb{L}}})$, find a deterministic policy $\pi : S \mapsto A$ such that the Markov Chain $\mathcal{M}_\pi$ induced by π is recurrent, maximizes (4), and satisfies steady-state specifications in $\Phi^{\infty_{\mathbb{L}}}$.*

We prove that D-SSPS belongs to the **NP-hard** complexity class via a reduction from the classic HAMILTONIAN-CYCLE problem [Karp, 1972].

Definition 10 (HAMILTONIAN-CYCLE)**.** *Given a graph $G = (V, E)$, determine whether there is a simple cycle containing every node in V.*

Theorem 3. *D-SSPS is **NP-hard**.*

Proof. We prove that there is a polynomial-time reduction from HAMILTONIAN-CYCLE to D-SSPS, thereby establishing that D-SSPS is at least as difficult as HAMILTONIAN-CYCLE. Given a HAMILTONIAN-CYCLE instance $G = (V, E), |V| = n$, we create the D-SSPS instance $\mathcal{M} = (S, A, T, R = \emptyset, L, \Phi^{\infty_{\mathbb{L}}})$ as follows. For every vertex $v \in V$, we have a state $s \in S$. Let $A(s_i) = \{a_j | (v_i, v_j) \in E\}$. That is, s_i has as many actions as the out-degree of v_i. Let $T(s_j|s_i, a_j) = 1$ if $a_j \in A(s_i)$ and $T(s_j|s_i, a_j) = 0$ otherwise. The steady-state specifications are given by $\Phi^{\infty_{\mathbb{L}}} = \{(\{s_1\}, \{1/n\}), \ldots, (\{s_n\}, \{1/n\})\}$.

We will show that G has a Hamiltonian cycle if and only if there is a feasible solution to $\mathcal{M}$. That is, if there exists a policy $\pi : S \mapsto A$ such that the Markov Chain $\mathcal{M}_\pi$ is recurrent and satisfies the specifications in $\Phi^{\infty_{\mathbb{L}}}$.

($\Longrightarrow$) Without loss of generality, suppose G has a Hamiltonian cycle $\tau = (v_1, \ldots, v_n, v_1)$. Note that τ is a path from v_1 to itself with $|\tau| = n + 1$. We can construct a feasible policy π as follows. For every $(v_i, v_{i+1}) \subset \tau$, let $\pi(s_i) = a_{i+1}$. Since $T(s_{i+1}|a_{i+1}, s_i) = 1$, π will induce a Markov Chain $\mathcal{M}_\pi$ consisting of a cycle. Its transition probability matrix is

defined by $T_\pi(s_{i+1}|s_i) = 1$ and $T_\pi(s_1|s_n) = 1$. The steady-state probabilities are then given by the solution to the system of equations (1). Due to the cyclic structure of T_π, this leads to equations (10).

$$\begin{aligned}(\Pr_\pi^\infty(s_n), \Pr_\pi^\infty(s_1), \ldots, \Pr_\pi^\infty(s_{n-1})) = \\ (\Pr_\pi^\infty(s_1), \Pr_\pi^\infty(s_2), \ldots, \Pr_\pi^\infty(s_n)) \\ \sum_{s\in S} \Pr_\pi^\infty(s) = 1\end{aligned} \tag{10}$$

Note that these equations can only be satisfied by setting $\Pr_\pi^\infty(s_i) = 1/n$ for all $i \in [n]$. Thus, π satisfies the steady-state specifications in $\Phi^{\infty_\mathbb{L}}$. Furthermore, since $\mathcal{M}_\pi$ is a cycle containing every state in S, we have reachability between every pair of nodes. This establishes the recurrence of $\mathcal{M}_\pi$.

($\Longleftarrow$) Suppose we have a policy π whose induced Markov Chain $\mathcal{M}_\pi$ is recurrent and satisfies the specifications in $\Phi^{\infty_\mathbb{L}}$. From our reduction, we can show that every state in $\mathcal{M}_\pi$ will have exactly one outgoing edge. Indeed, recall that $T(s_j|s_i, a) \in \{0, 1\}$ and note that, for every state $s_i \in S$, the sum of all outgoing transition probabilities is one. This can be seen in equation (11), where $\mathcal{I}(*) = 1$ if its argument is true and 0 otherwise.

$$\begin{aligned}\sum_{s_j\in S} T_\pi(s_j|s_i) &= \sum_{s_j\in S}\sum_{a\in A(s_i)} \Pr_\pi(s_j, a|s_i) \\ &= \sum_{s_j\in S}\sum_{a\in A(s_i)} T(s_j|s_i, a)\mathcal{I}(\pi(s_i) = a) = 1\end{aligned} \tag{11}$$

Since there must exist a path between every pair of states, we have, without loss of generality, that $T_\pi(s_2|s_1) = T_\pi(s_3|s_2) = \cdots = T_\pi(s_n|s_{n-1}) = 1$. There is some j such that $T_\pi(s_j|s_n) = 1$. Note that in order to satisty $\Pr_\pi^\infty(s_1) = \cdots = \Pr_\pi^\infty(s_n) = 1/n$, it must be that $T_\pi(s_1|s_n) = 1$. Thus, the Hamiltonian cycle in G is derived directly from $\mathcal{M}_\pi$. □

Theorem 3 establishes the complexity of solving D-SSPS. In particular, note that finding a feasible (not necessarily optimal) solution is **NP-hard**. Indeed, no reference to the objective function is made in the reduction from HAMILTONIAN-CYCLE. Thus, the complexity stems exclusively from finding a satisfying assignment to the constraint set defined by $\Phi^{\infty_\mathbb{L}}$.

7 Experimental Results

Simulations of program (7) were performed using CPLEX version 12.8 [CPL, 2017] on a machine with a 3.6 GHz Intel Core i7-6850K processor and 128 GB of RAM. The traditional simplex algorithm was used as well as the barrier method for comparison. Each LMDP $\mathcal{M} = (S, A, T, R, L, \Phi^{\infty_\mathbb{L}})$ instance was defined as follows for various state-space sizes $|S|$. There are four actions associated with each state and taking an action causes a transition to one of two possible random states. Each state-action pair observes a random reward in $\{1, 2, 3, 4\}$. Two labels $L_1, L_2 \subset S, |L_1| = |L_2| = \lfloor \log(|S|) \rfloor$ were randomly defined for each instance and used for the steady-state constraints $\Phi^{\infty_\mathbb{L}} = (L_1, [10/|S|, 1000/|S|]), (L_2, \{0\})$. See Table 1 for runtime results.

Number of States	Barrier Solver (Seconds)	Simplex Solver (Seconds)
1000	3.16	0.76
2000	15.17	7.12
3000	42.10	18.99
4000	95.05	50.32
5000	158.79	86.38
6000	277.05	133.27
7000	461.56	223.53
8000	601.15	293.48
9000	979.55	435.25
10000	1514.29	549.77

Table 1: Simulation runtimes using CPLEX version 12.8 for LMDPs with a varying number of states subject to steady-state constraints. Each entry denotes the average runtime in seconds to solve ten random SSPS problem instances for each pair of state-space size and solver type.

8 Discussion

While many verifiable control methods have been proposed for systems under temporal logic specifications such as pCTL and pLTL, steady-state specifications have not received as much attention. To the best of the authors' knowledge, the integration of these different specifications within the same system has been virtually unexplored. We argue that the foregoing solution to the SSPS problem facilitates said integration by posing the problem as a linear program, which is the method of choice for solving multi-objective verifiable control problems subject to pCTL and pLTL specifications. Based on the observation that $x_{sa} = \pi(a|s)\Pr_\pi^\infty(s)$, there is an obvious integration using this formula as a quadratic constraint. However, this leads to a non-convex quadratic program, which is hard to solve. Typically, there is a variable for $\pi(a|s) \in [0, 1]$ in linear programs for pCTL/pLTL. However, we do not have access to this variable in our case. Instead, we have x_{sa} which is a function of $\pi(a|s)$. A linear program could be formulated with both $\pi(a|s)$ for pCTL/pLTL and x_{sa} for steady-state specifications as variables, but we cannot ensure that the policy derived from x_{sa} and the one given by $\pi(a|s)$ would be the same as this seems to require the aforementioned quadratic constraint.

9 Conclusion

We have demonstrated how the problem of computing optimal stochastic policies for verifiable control under steady-state constraints can be solved efficiently. Our approach is extensible by allowing for additional constraints to be added to the linear programming formulation. We have improved upon the complexity results in prior art and presented new ones for the deterministic policy variant of SSPS. We plan to investigate the integration of temporal logic specifications into the proposed framework.

Acknowledgments

The authors would like to thank anonymous reviewers for valuable inputs that helped shape this manuscript.

References

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

[Altman, 1999] Eitan Altman. *Constrained Markov decision processes*, volume 7. CRC Press, 1999.

[Baumgartner *et al.*, 2018] Peter Baumgartner, Sylvie Thiébaux, and Felipe Trevizan. Heuristic search planning with multi-objective probabilistic ltl constraints. 2018.

[Canny, 1988] John Canny. Some algebraic and geometric computations in pspace. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 460–467. ACM, 1988.

[Cinlar, 2013] Erhan Cinlar. *Introduction to stochastic processes*. Courier Corporation, 2013.

[CPL, 2017] Ibm ilog cplex optimization studio v12.8.0 documentation. *IBM Knowledge Center*, 2017.

[Feyzabadi, 2017] Seyedshams Feyzabadi. *Robot Planning with Constrained Markov Decision Processes*. PhD thesis, UNIVERSITY OF CALIFORNIA, MERCED, 2017.

[Floudas and Visweswaran, 1995] Christodoulos A Floudas and V Visweswaran. Quadratic optimization. In *Handbook of global optimization*, pages 217–269. Springer, 1995.

[Karp, 1972] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[Kearns and Singh, 2002] Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2-3):209–232, 2002.

[Konstantopoulos, 2009] Takis Konstantopoulos. Markov chains and random walks. *Lecture notes*, 2009.

[McCuaig, 1993] William McCuaig. Intercyclic digraphs. *Contemporary Mathematics*, 147:203–203, 1993.

[Ross, 2014] Sheldon M Ross. *Introduction to probability models*. Academic press, 2014.

[Russell *et al.*, 2015] Stuart Russell, Daniel Dewey, and Max Tegmark. Research priorities for robust and beneficial artificial intelligence. *Ai Magazine*, 36(4):105–114, 2015.

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

[Teichteil-Königsbuch, 2012] Florent Teichteil-Königsbuch. Path-constrained markov decision processes: bridging the gap between probabilistic model-checking and decision-theoretic planning. In *20th European Conference on Artificial Intelligence (ECAI 2012)*, 2012.

[Trevizan *et al.*, 2016] Felipe W Trevizan, Sylvie Thiébaux, Pedro Henrique Santana, and Brian Charles Williams. Heuristic search in dual space for constrained stochastic shortest path problems. In *ICAPS*, pages 326–334, 2016.

[Tsitsiklis, 2007] John N Tsitsiklis. Np-hardness of checking the unichain condition in average cost mdps. *Operations research letters*, 35(3):319–323, 2007.

[Wolff and Murray, 2016] Eric M Wolff and Richard M Murray. Optimal control of nonlinear systems with temporal logic specifications. In *Robotics Research*, pages 21–37. Springer, 2016.