Linear Programming Formulation for Non-stationary, Finite-Horizon Markov Decision Process Models

Arnab Bhattacharya[1] and Jeffrey P. Kharoufeh[2]
Department of Industrial Engineering
University of Pittsburgh
1025 Benedum Hall
3700 O'Hara Street
Pittsburgh, PA 15261 USA

Accepted Version: September 5, 2017
To appear in *Operations Research Letters*

Abstract

Linear programming (LP) formulations are often employed to solve stationary, infinite-horizon Markov decision process (MDP) models. We present an LP approach to solving non-stationary, finite-horizon MDP models that can potentially overcome the computational challenges of standard MDP solution procedures. Specifically, we establish the existence of an LP formulation for risk-neutral MDP models whose states and transition probabilities are temporally heterogeneous. This formulation can be recast as an approximate linear programming formulation with significantly fewer decision variables.

Keywords: Non-stationary MDP, linear programming

1 Introduction

It is well known that stationary Markov decision process (MDP) models can be reformulated as linear programs and solved efficiently using linear programming (LP) algorithms [2, 17, 18, 19]. The appeal of the LP formulation stems from the fact that it allows for the inclusion of additional model constraints and facilitates sensitivity analysis in sequential decision making problems. Furthermore, duality theory allows one to characterize the optimal decisions in a MDP model via the optimal solution of the associated dual problem [18, 19]. Owing to recent advances in the computational speed of LP solvers, the LP approach has been successfully employed to solve large-scale, stationary MDP models (see [1, 4, 5, 15, 17, 21]).

While the LP reformulation has been most prevalent in the case of stationary, infinite-horizon MDP models [6, 18], by contrast, this formulation is seldom used as a solution strategy for non-stationary, finite-horizon MDP models. This is due to the ease of implementation of the backward dynamic programming (BDP) procedure to solve finite-horizon problems as a sequence of simpler

[1]Ph: 412-626-1799; Email: cfcarnabiitkgp@gmail.com
[2]Corresponding author. Ph: 412-624-9832; Email: jkharouf@pitt.edu

single-stage problems using the optimality equations. It is well known that, for an N-stage MDP model with K states and L actions in each stage, the BDP procedure requires $(N-1)LK^2$ multiplicative operations to determine an optimal policy [18]. The BDP procedure is computationally viable for models with low-dimensional state and action spaces, as the number of such operations is relatively small. However, for models with multidimensional state and action spaces, BDP becomes computationally intractable due to the curses of dimensionality [2, 17]. The problem is further exacerbated for models with a large number of decision stages. For example, solving a 10-stage model comprised of four state and three action variables, each with five feasible values in each stage, requires more than 2.2×10^9 multiplicative operations, which is prohibitively large. The computational burden may increase further for *non-stationary* MDP models that include temporal heterogeneity in the states and transition probabilities.

In this paper, we present a linear programming formulation for non-stationary, finite-horizon MDP models as a viable approach to overcome these computational challenges. Specifically, we prove the existence of a general LP formulation for such models with countable state and action spaces under a risk-neutral objective. We establish lower and upper bounds of the value functions, which are used to formulate the primal LP model. The solution of this model is the optimal value function of the MDP model, while the solution of its dual problem recovers the optimal policy. Although the LP approach does not (in and of itself) overcome the curses of dimensionality, it lays the groundwork for implementing approximate linear programming (ALP) procedures [4] to solve computationally intractable finite-horizon models. Specifically, we suggest an ALP formulation that utilizes parametric basis functions to approximate the value functions at each stage. In light of recent advances in LP solvers, the ALP approach offers computational advantages over traditional MDP solution procedures (such as the value and policy iteration algorithms) for solving high-dimensional finite-horizon problems.

The remainder of the paper is organized as follows. Section 2 introduces some preliminaries of the non-stationary, finite-horizon MDP model, while Section 3 presents our main results which establish existence of the LP formulation. In Section 4, we discuss the computational advantages of using the LP approach as compared to standard MDP solution procedures. Some concluding remarks are provided in Section 5.

2 Preliminaries

Consider a finite planning horizon $T = \{1, 2, \ldots, N\}$ with N decision stages (or decision epochs) and let $t \in T$ be the index of the tth decision epoch. For convenience, define $T' \equiv T \setminus \{N\}$. In what follows, all random variables are defined on a common, complete probability space $(\Omega, \mathscr{A}, \mathbb{P})$, where Ω is a sample space, $\mathscr{A}$ is a σ-field of subsets of Ω and $\mathbb{P}$ is a probability measure on $(\Omega, \mathscr{A})$. In what follows, all vectors are assumed to be column vectors, unless otherwise noted. The state of the process at the start of stage t is denoted by the random vector $\boldsymbol{S}_t$ whose state space is a countable

set $\mathcal{S}_t \subset \mathbb{R}^n$. A realization of $\boldsymbol{S}_t$ is denoted by $\boldsymbol{s} \in \mathcal{S}_t$. When the state of the process is $\boldsymbol{s}$, the set of feasible decisions (or action space) is denoted by a countable set $\mathcal{X}_t(\boldsymbol{s}) \subset \mathbb{R}^m$. For notational convenience, we suppress the dependence of this set on $\boldsymbol{s}$ and simply write $\mathcal{X}_t$. A decision rule is a vector-valued mapping $\boldsymbol{x}_t : \mathcal{S}_t \to \mathcal{X}_t$ that prescribes feasible actions for each $\boldsymbol{s} \in \mathcal{S}_t$. A set of decision rules, one for each stage $t \in T'$, is called a policy and is denoted by $\pi = \{\boldsymbol{x}_t : t \in T'\} \in \Pi$, where Π is the collection of all feasible Markov deterministic (MD) policies. It is noted that no decisions are made in the terminal stage $t = N$. For a given decision rule $\boldsymbol{x}_{t-1}$, the temporally-heterogeneous transition probabilities are denoted by $\mathbb{P}_t(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{x}_{t-1}(\boldsymbol{s}))$, where $(\boldsymbol{s}', \boldsymbol{s}) \in \mathcal{S}_t \times \mathcal{S}_{t-1}$ and $\boldsymbol{x}_{t-1}(\boldsymbol{s}) \in \mathcal{X}_{t-1}$. Let $p : \mathcal{S}_1 \to [0, 1]$ be the probability mass function of the initial state $\boldsymbol{S}_1$ such that $0 \leq p(\boldsymbol{s}) \leq 1$ for all $\boldsymbol{s} \in \mathcal{S}_1$ and $\sum_{\boldsymbol{s} \in \mathcal{S}_1} p(\boldsymbol{s}) = 1$. For a given policy π, the transition probability matrix at stage t, denoted by $\boldsymbol{Q}_t^\pi$, is defined as

$$\boldsymbol{Q}_t^\pi = \left(\mathbb{P}_t(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{x}_{t-1}(\boldsymbol{s})) : (\boldsymbol{s}', \boldsymbol{s}) \in \mathcal{S}_t \times \mathcal{S}_{t-1}, \boldsymbol{x}_{t-1} \in \pi\right), \quad t = 2, \ldots, N,$$

where $\sum_{\boldsymbol{s}' \in \mathcal{S}_t} \mathbb{P}_t(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{x}_{t-1}(\boldsymbol{s})) = 1$ for all $\boldsymbol{s} \in \mathcal{S}_{t-1}$. The random one-step cost incurred in stage $t \in T'$ is denoted by $c_t(\boldsymbol{S}_t, \boldsymbol{x}_t(\boldsymbol{S}_t))$, while the terminal cost is $c_N(\boldsymbol{S}_N)$. For a given policy π, the vector of one-step costs at stage t is

$$\boldsymbol{c}_t^\pi = \begin{cases} (c_t(\boldsymbol{s}, \boldsymbol{x}_t(\boldsymbol{s})) : \boldsymbol{s} \in \mathcal{S}_t, \boldsymbol{x}_t \in \pi), & t \in T', \\ (c_N(\boldsymbol{s}) : \boldsymbol{s} \in \mathcal{S}_N), & t = N, \end{cases}$$

where we assume that $|c_t(\boldsymbol{s}, \boldsymbol{x}_t(\boldsymbol{s}))| < \infty$ and $|c_N(\boldsymbol{s})| < \infty$. Consider a mapping $V_t^\pi : \mathcal{S}_t \to \mathbb{R}$, where $V_t^\pi(\boldsymbol{s})$ denotes the expected future cost incurred under policy π starting in state $\boldsymbol{s}$ at stage t, and let $\boldsymbol{V}_t^\pi \equiv (V_t^\pi(\boldsymbol{s}) : \boldsymbol{s} \in \mathcal{S}_t)$. By definition,

$$\begin{aligned} \boldsymbol{V}_t^\pi &= \boldsymbol{c}_t^\pi + \delta \boldsymbol{Q}_{t+1}^\pi \boldsymbol{V}_{t+1}^\pi, \quad t \in T', \\ \boldsymbol{V}_N^\pi &= \boldsymbol{c}_N, \end{aligned}$$

where $\delta \in (0, 1]$ is a discount factor. Given an initial state $\boldsymbol{s}$, the risk-neutral objective is to minimize the expected total discounted costs incurred over the planning horizon as follows:

$$\begin{aligned} z^*(\boldsymbol{s}) &= \inf_{\pi \in \Pi} \{\boldsymbol{V}_1^\pi(\boldsymbol{s})\} \\ &= \inf_{\pi \in \Pi} \left\{ \mathbb{E}_\pi \left(\sum_{t \in T'} \delta^{t-1} c_t(\boldsymbol{S}_t, \boldsymbol{x}_t(\boldsymbol{S}_t)) + \delta^{N-1} c_N(\boldsymbol{S}_N) \Big| \boldsymbol{S}_1 = \boldsymbol{s} \right) \right\}. \end{aligned} \tag{1}$$

We denote an optimal policy of (1) by π^* and the corresponding value function at stage t by $V_t^* \equiv V_t^{\pi^*}$. Let $\boldsymbol{V}_t^* = (V_t^*(\boldsymbol{s}_t) : \boldsymbol{s}_t \in \mathcal{S}_t)$ be the vector of optimal values in stage t. Then, the optimality equations (in vector form) are given by

$$\boldsymbol{V}_t^* = \inf_{\pi \in \Pi} \left\{ \boldsymbol{c}_t^\pi + \delta \boldsymbol{Q}_{t+1}^\pi \boldsymbol{V}_{t+1}^* \right\}, \quad t \in T', \tag{2a}$$

$$\boldsymbol{V}_N^* = \boldsymbol{c}_N. \tag{2b}$$

3 Linear Programming Formulation

In this section, we establish the existence of an LP formulation for the model in (1) whose optimal solutions are the value functions defined in (2). Additionally, we present an associated dual LP formulation whose optimal solutions can be used to obtain an optimal policy π^*.

Let $\mathbb{V}$ denote the set of all real-valued, bounded functions on $\mathcal{S}_t$. For each $t \in T$, consider a complete, normed linear space $(\mathbb{V}, \|\cdot\|_\infty)$ of bounded functions on $\mathcal{S}_t$ that is equipped with the supremum norm $\|\cdot\|_\infty$ and component-wise partial order $\preceq$. Let $J_t : \mathcal{S}_t \to \mathbb{R}$ be a function that belongs to $\mathbb{V}$, and let $\boldsymbol{J}_t = (J_t(\boldsymbol{s}) : \boldsymbol{s} \in \mathcal{S}_t)$ be the vector form of J_t so that its supremum norm is $\|\boldsymbol{J}_t\|_\infty = \sup_{\boldsymbol{s}\in\mathcal{S}_t}\{|J_t(\boldsymbol{s})|\}$. Moreover, for any two functions $J_t^1, J_t^2 \in \mathbb{V}$, the relation $J_t^1 \preceq J_t^2$ implies that $J_t^1(\boldsymbol{s}) \leq J_t^2(\boldsymbol{s})$ for all $\boldsymbol{s} \in \mathcal{S}_t$, or simply that $\boldsymbol{J}_t^1 \leq \boldsymbol{J}_t^2$. For each $t \in T'$, define a nonlinear operator $\Lambda_t : \mathbb{V} \to \mathbb{V}$, such that for any $J_{t+1} \in \mathbb{V}$,

$$\Lambda_t \boldsymbol{J}_{t+1} = \inf_{\pi\in\Pi} \left\{\boldsymbol{c}_t^\pi + \delta \boldsymbol{Q}_{t+1}^\pi \boldsymbol{J}_{t+1}\right\}. \tag{3}$$

For stage N, define another operator, $\Psi : \mathbb{V} \to \mathbb{V}$, such that for all $\boldsymbol{J}_N \in \mathbb{V}$,

$$\Psi \boldsymbol{J}_N = \boldsymbol{c}_N. \tag{4}$$

Thus, the operator Ψ maps any bounded function in stage N to the terminal cost function in stage N so that $\Psi \boldsymbol{J}_N(\boldsymbol{s}) = \boldsymbol{c}_N(\boldsymbol{s})$ for each $\boldsymbol{s} \in \mathcal{S}_N$. Next, denote an arbitrary vector of functions, one for each stage $t \in T$, by $\boldsymbol{J} = (\boldsymbol{J}_t : t \in T) \in \mathbb{V}^N$ and define the operator $\Lambda : \mathbb{V}^N \to \mathbb{V}^N$ such that for any $\boldsymbol{J} \in \mathbb{V}^N$,

$$\Lambda \boldsymbol{J} = (\Lambda_1 \boldsymbol{J}_2, \Lambda_2 \boldsymbol{J}_3, \ldots, \Lambda_{N-1} \boldsymbol{J}_N, \Psi \boldsymbol{J}_N). \tag{5}$$

A fixed point of the operator Λ is any vector $\boldsymbol{J}^* \in \mathbb{V}^N$ satisfying the equality

$$\Lambda \boldsymbol{J}^* = \boldsymbol{J}^*. \tag{6}$$

Any $\boldsymbol{J} \in \mathbb{V}^N$ that satisfies the inequality $\boldsymbol{J} \leq \Lambda \boldsymbol{J}$ is called a sub-solution of (6), while a super-solution of (6) satisfies $\boldsymbol{J} \geq \Lambda \boldsymbol{J}$. Proposition 1 shows that sub- and super-solutions of (6) are, respectively, lower and upper bounds of the value function vector $\boldsymbol{V}^* = (\boldsymbol{V}_t^* : t \in T)$.

Proposition 1 *For any $\boldsymbol{J} \in \mathbb{V}^N$, if $\boldsymbol{J} \leq \Lambda \boldsymbol{J}$ ($\boldsymbol{J} \geq \Lambda \boldsymbol{J}$), then $\boldsymbol{J} \leq \boldsymbol{V}^*$ ($\boldsymbol{J} \geq \boldsymbol{V}^*$).*

Proof. Let $\bar{\pi}$ be a feasible policy of (1). First consider the case $\boldsymbol{J} \leq \Lambda \boldsymbol{J}$ for some $\boldsymbol{J} \in \mathbb{V}^N$. In this case, $\boldsymbol{J}_t \leq \Lambda_t \boldsymbol{J}_{t+1}$, for each $t \in T'$ and $\boldsymbol{J}_N \leq \Psi \boldsymbol{J}_N = \boldsymbol{c}_N$. Using equalities (3) and (4), respectively, we obtain the following system of inequalities:

$$\boldsymbol{J}_t \leq \inf_{\pi\in\Pi}\{\boldsymbol{c}_t^\pi + \delta \boldsymbol{Q}_{t+1}^\pi \boldsymbol{J}_{t+1}\} \leq \boldsymbol{c}_t^{\bar{\pi}} + \delta \boldsymbol{Q}_{t+1}^{\bar{\pi}} \boldsymbol{J}_{t+1}, \quad t \in T', \tag{7a}$$

$$\boldsymbol{J}_N \leq \boldsymbol{c}_N = \boldsymbol{V}_N^*. \tag{7b}$$

The right-most inequality in (7a) holds because $\bar{\pi}$ is feasible, but not necessarily optimal, for $\boldsymbol{J}_{t+1}$ in (3). Starting in stage 1 and sequentially applying constraints (7) for stages $t = 2, \ldots, N$, we have

$$\begin{aligned}
\boldsymbol{J}_1 &\leq \boldsymbol{c}_1^{\bar{\pi}} + \delta \boldsymbol{Q}_2^{\bar{\pi}} \boldsymbol{J}_2, \\
&\leq \boldsymbol{c}_1^{\bar{\pi}} + \delta \boldsymbol{Q}_2^{\bar{\pi}} (\boldsymbol{c}_2^{\bar{\pi}} + \delta \boldsymbol{Q}_3^{\bar{\pi}} \boldsymbol{J}_3) = \boldsymbol{c}_1^{\bar{\pi}} + \delta \boldsymbol{Q}_2^{\bar{\pi}} \boldsymbol{c}_2^{\bar{\pi}} + \delta^2 \boldsymbol{Q}_2^{\bar{\pi}} \boldsymbol{Q}_3^{\bar{\pi}} \boldsymbol{J}_3, \\
&\;\;\vdots \\
&\leq \boldsymbol{c}_1^{\bar{\pi}} + \delta \boldsymbol{Q}_2^{\bar{\pi}} \boldsymbol{c}_2^{\bar{\pi}} + \cdots + \delta^{N-1} \left(\prod_{t \in T'} \boldsymbol{Q}_{t+1}^{\bar{\pi}} \right) \boldsymbol{c}_N \\
&= \boldsymbol{J}_1^{\bar{\pi}},
\end{aligned}$$

where $\boldsymbol{J}_1^{\bar{\pi}}$ is the vector of expected total costs from stage 1 forward under policy $\bar{\pi}$. As $\boldsymbol{J}_1 \leq \boldsymbol{J}_1^{\bar{\pi}}$ and $\bar{\pi}$ is any feasible policy, we have $\boldsymbol{J}_1 \leq \inf_{\bar{\pi} \in \Pi} \{\boldsymbol{J}_1^{\bar{\pi}}\} = \boldsymbol{V}_1^* \Rightarrow \boldsymbol{J}_1 \leq \boldsymbol{V}_1^*$. Using similar arguments, we can show that $\boldsymbol{J}_t \leq \boldsymbol{V}_t^*$ for $t = 2, 3, \ldots, N-1$. Thus, we have $\boldsymbol{J} \leq \boldsymbol{V}^*$.

Next, consider the case when $\boldsymbol{J} \geq \Lambda \boldsymbol{J}$. By the definition of infimum, for any non-negative sequence of column vectors $\boldsymbol{\epsilon} = \{\boldsymbol{\epsilon}_t : t \in T'\}$, there exists a policy $\hat{\pi} \in \Pi$, such that

$$\boldsymbol{J}_t \geq \inf_{\pi \in \Pi} \{\boldsymbol{c}_t^{\pi} + \delta \boldsymbol{Q}_{t+1}^{\pi} \boldsymbol{J}_{t+1}\} \geq \boldsymbol{c}_t^{\hat{\pi}} + \delta \boldsymbol{Q}_{t+1}^{\hat{\pi}} \boldsymbol{J}_{t+1} - \boldsymbol{\epsilon}_t, \quad t \in T', \tag{8a}$$

$$\boldsymbol{J}_N \geq \boldsymbol{c}_N = \boldsymbol{V}_N^*. \tag{8b}$$

Next construct a positive, nondecreasing sequence $\boldsymbol{\epsilon}$ by choosing an appropriate auxiliary sequence of vectors $\hat{\boldsymbol{\epsilon}} = \{\hat{\boldsymbol{\epsilon}}_t : t \in T'\}$, such that

$$\boldsymbol{\epsilon}_1 = \hat{\boldsymbol{\epsilon}}_1; \qquad \boldsymbol{\epsilon}_t \geq \sum_{i=1}^{t} \delta^{i-1} \hat{\boldsymbol{\epsilon}}_i \prod_{j=1}^{i-1} \boldsymbol{Q}_{j+1}^{\hat{\pi}}, \quad t \in \{2, \ldots, N-1\}. \tag{9}$$

Note that the entries of $\boldsymbol{\epsilon}_t$ can be made arbitrarily small by choosing an appropriate $\hat{\boldsymbol{\epsilon}}_t$. Then, starting from stage 1 and sequentially applying constraints (8) and (9) in stages $t = 2, \ldots, N$, we obtain

$$\boldsymbol{J}_1 \geq \boldsymbol{c}_1^{\hat{\pi}} + \delta \boldsymbol{Q}_2^{\hat{\pi}} \boldsymbol{c}_2^{\hat{\pi}} + \cdots + \delta^{N-1} \left(\prod_{t \in T'} \boldsymbol{Q}_{t+1}^{\hat{\pi}} \right) \boldsymbol{c}_N - \boldsymbol{\epsilon}_{N-1} = \boldsymbol{J}_1^{\hat{\pi}} - \boldsymbol{\epsilon}_{N-1},$$

where $\boldsymbol{J}_1^{\hat{\pi}}$ is the vector of expected total costs from stage 1 forward under policy $\hat{\pi}$. As $\hat{\pi}$ is any feasible policy, we have $\boldsymbol{J}_1 \geq \inf_{\hat{\pi} \in \Pi} \{\boldsymbol{J}_1^{\hat{\pi}}\} - \boldsymbol{\epsilon}_{N-1} = \boldsymbol{V}_1^* - \boldsymbol{\epsilon}_{N-1}$. As $\boldsymbol{\epsilon}_{N-1} > \boldsymbol{0}$ with arbitrarily small entries, we have that $\boldsymbol{J}_1 \geq \boldsymbol{V}_1^*$. Similarly, it can be shown that $\boldsymbol{J}_t \geq \boldsymbol{V}_t^*$ for $t = 2, \ldots, N-1$; therefore, $\boldsymbol{J} \geq \boldsymbol{V}^*$ and the proof is complete. ∎

By Proposition 1, $\boldsymbol{V}^*$ is both a sub- and super-solution of Λ and is, therefore, a fixed point of Λ. Moreover, $\boldsymbol{V}^*$ is the unique fixed point of Λ in $\mathbb{V}^N$ since the fixed point equation $\Lambda \boldsymbol{J} = \boldsymbol{J}$ corresponds to the system of equalities,

$$\begin{aligned}
\boldsymbol{J}_t &= \inf_{\pi \in \Pi} \left\{ \boldsymbol{c}_t^{\pi} + \delta \boldsymbol{Q}_{t+1}^{\pi} \boldsymbol{J}_{t+1} \right\}, \quad t \in T', \\
\boldsymbol{J}_N &= \boldsymbol{c}_N,
\end{aligned}$$

which are uniquely satisfied by the value functions $\boldsymbol{V}_1^*, \ldots, \boldsymbol{V}_N^*$ (via Bellman's optimality principle). Next, we present our main result (Theorem 1) establishing the existence of an LP formulation for the model in (1). In what follows, for each state $\boldsymbol{s}$ and stage t, define the decision variables $u_t(\boldsymbol{s})$.

Theorem 1 *For any real, positive vector $(\beta_t(\boldsymbol{s}) : t \in T,\, \boldsymbol{s} \in \mathcal{S}_t)$, such that $\beta_t(\boldsymbol{s}) < \infty$ for each t and $\boldsymbol{s}$, the value functions of (2) are obtained by solving the following linear program:*

$$\max \quad \sum_{t \in T} \sum_{\boldsymbol{s} \in \mathcal{S}_t} \beta_t(\boldsymbol{s}) u_t(\boldsymbol{s}) \tag{10a}$$

$$\text{s.t.} \quad u_t(\boldsymbol{s}) \leq c_t(\boldsymbol{s}, \boldsymbol{x}) + \sum_{\boldsymbol{s}' \in \mathcal{S}_{t+1}} \delta \mathbb{P}_{t+1}(\boldsymbol{s}' | \boldsymbol{s}, \boldsymbol{x}) u_{t+1}(\boldsymbol{s}') \qquad \forall t \in T',\, \boldsymbol{s} \in \mathcal{S}_t,\, \boldsymbol{x} \in \mathcal{X}_t, \tag{10b}$$

$$u_N(\boldsymbol{s}) \leq c_N(\boldsymbol{s}) \qquad \forall \boldsymbol{s} \in \mathcal{S}_N, \tag{10c}$$

$$u_t(\boldsymbol{s}) \in \mathbb{R} \qquad \forall t \in T,\, \boldsymbol{s} \in \mathcal{S}_t. \tag{10d}$$

i.e., $u_t^*(\boldsymbol{s}) = V_t^*(\boldsymbol{s})$ for each $t \in T$ and $\boldsymbol{s} \in \mathcal{S}_t$.

Proof. Let $\boldsymbol{\beta} = (\beta_t(\boldsymbol{s}) : t \in T,\, \boldsymbol{s} \in \mathcal{S}_t)$ be any positive, bounded vector of real values. Then, consider the following maximization problem in standard form:

$$\max \ \boldsymbol{\beta}' \boldsymbol{J} \tag{11a}$$

$$\text{s.t.} \ \boldsymbol{J} \leq \Lambda \boldsymbol{J}. \tag{11b}$$

By Proposition 1, any $\boldsymbol{J} \in \mathbb{V}^N$ that satisfies (11b) is a valid lower bound of $\boldsymbol{V}^*$. As $\boldsymbol{\beta} \geq \boldsymbol{0}$, at the optimal solution $\boldsymbol{J}^*$, constraint (11b) holds with equality, i.e. $\boldsymbol{J}^* = \Lambda \boldsymbol{J}^*$. Hence, $\boldsymbol{J}^*$ is a fixed point of Λ and is, therefore, equal to the unique fixed point $\boldsymbol{V}^*$. Objective (11a) is linear, and although the operator Λ is nonlinear, the constraint $\boldsymbol{J} \leq \Lambda \boldsymbol{J}$ is equivalent to the following system of linear inequalities:

$$J_t(\boldsymbol{s}) \leq c_t(\boldsymbol{s}, \boldsymbol{x}) + \sum_{\boldsymbol{s}' \in \mathcal{S}_{t+1}} \delta \mathbb{P}_{t+1}(\boldsymbol{s}' | \boldsymbol{s}, \boldsymbol{x}) J_{t+1}(\boldsymbol{s}'), \qquad \forall t \in T',\, \boldsymbol{s} \in \mathcal{S}_t,\, \boldsymbol{x} \in \mathcal{X}_t,$$

$$J_N(\boldsymbol{s}) \leq c_N(\boldsymbol{s}), \qquad \forall \boldsymbol{s} \in \mathcal{S}_N.$$

Hence, problem (11) is an LP, and replacing the variables $J_t(\boldsymbol{s})$ by $u_t(\boldsymbol{s})$, we obtain the result. ∎

Note that the optimal solutions of the primal LP model in (10) correspond to the value functions defined via (2), i.e., $u_t^*(\boldsymbol{s}) = V_t^*(\boldsymbol{s})$. To recover the optimal policy π^* directly, we exploit the dual of formulation (10). Let $\boldsymbol{v} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N)^T$ be the vector of dual variables, such that

$$\boldsymbol{v}_t = \begin{cases} (v_t(\boldsymbol{s}, \boldsymbol{x}) : \boldsymbol{s} \in \mathcal{S}_t,\, \boldsymbol{x} \in \mathcal{X}_t), & t \in T', \\ (v_N(\boldsymbol{s}) : \boldsymbol{s} \in \mathcal{S}_N), & t = N. \end{cases}$$

Define $T'' = \{2, \ldots, N-1\}$. Then, the dual problem of (10) is the linear program

$$\min \quad \sum_{t \in T'} \sum_{\boldsymbol{s} \in \mathcal{S}_t} \sum_{\boldsymbol{x} \in \mathcal{X}_t} c_t(\boldsymbol{s}, \boldsymbol{x}) v_t(\boldsymbol{s}, \boldsymbol{x}) + \sum_{\boldsymbol{s} \in \mathcal{S}_N} c_N(\boldsymbol{s}) v_N(\boldsymbol{s}) \tag{12a}$$

$$\text{s.t.} \quad \sum_{\boldsymbol{x} \in \mathcal{X}_1} v_1(\boldsymbol{s}, \boldsymbol{x}) = \beta_1(\boldsymbol{s}) \qquad \forall \boldsymbol{s} \in \mathcal{S}_1, \tag{12b}$$

$$\sum_{\boldsymbol{x} \in \mathcal{X}_t} v_t(\boldsymbol{s}, \boldsymbol{x}) - \sum_{\boldsymbol{s}' \in \mathcal{S}_{t-1}} \sum_{\boldsymbol{x}' \in \mathcal{X}_{t-1}} \delta \mathbb{P}_t(\boldsymbol{s} | \boldsymbol{s}', \boldsymbol{x}') v_{t-1}(\boldsymbol{s}', \boldsymbol{x}') = \beta_t(\boldsymbol{s}) \ \forall t \in T'', \boldsymbol{s} \in \mathcal{S}_t, \tag{12c}$$

$$v_t(\boldsymbol{s}) - \sum_{\boldsymbol{s}' \in \mathcal{S}_{t-1}} \sum_{\boldsymbol{x}' \in \mathcal{X}_{t-1}} \delta \mathbb{P}_t(\boldsymbol{s} | \boldsymbol{s}', \boldsymbol{x}') v_{t-1}(\boldsymbol{s}', \boldsymbol{x}') = \beta_t(\boldsymbol{s}) \qquad t = N, \forall \boldsymbol{s} \in \mathcal{S}_N, \tag{12d}$$

$$v_t(\boldsymbol{s}) \geq 0 \qquad \forall t \in T, \ \boldsymbol{s} \in \mathcal{S}_t. \tag{12e}$$

The dual LP model (12) possesses some interesting properties. First, it can be shown that any basic feasible solution of (12) corresponds to a unique Markov deterministic policy $\pi \in \Pi$ (see Proposition 6.9.3 in [18]). Second, the optimal solution $\boldsymbol{v}^*$ of (12) has a one-to-one correspondence with the optimal solution π^* of (1). That is, for each $\boldsymbol{s} \in \mathcal{S}_t$, $v_t^*(\boldsymbol{s}, \boldsymbol{x}) > 0$ when $\boldsymbol{x} = \boldsymbol{x}_t^*$, and $v_t^*(\boldsymbol{s}, \boldsymbol{x}) = 0$ otherwise (the proof, which we omit here, is similar to that of Theorem 6.9.3 in [18]). Thus, π^* can be recovered from the optimal solution $\boldsymbol{v}^*$ without the need to enumerate the value functions via (2).

4 Complexity Issues

Unfortunately, the primal and dual linear programs ((10) and (12), respectively) have countably many variables and constraints, rendering them computationally intractable. A first step towards solving such linear programs is to select a finite number of states and formulate an approximate LP with a finite number of variables (see [20, 22, 23] for details). Additionally, constraint sampling algorithms with provable convergence guarantees, such as those described in [5], can be employed to sequentially sample finitely many constraints in each iteration and solve the resulting finite model. However, to obtain high-quality solutions for problems with high-dimensional state and action spaces, solving even approximate linear programs may be challenging. This dimensionality issue is common to standard MDP solution procedures, such as the value iteration (VI) and policy iteration (PI) algorithms. However, to illustrate the advantages of the LP formulation of Section 3, we consider models with finite state and action spaces (i.e., those with $|\mathcal{S}_t| < \infty$ and $|\mathcal{X}_t| < \infty$).

It is well known that each iteration of VI entails $O(|\mathcal{X}_t||\mathcal{S}_t|^2)$ arithmetic operations, which can be significant for large state and action spaces. Moreover, the number of iterations in VI grows exponentially in the discount factor δ; therefore, it is not pragmatic for solving high-dimensional MDP problems (see [3, 10]). In practice, the PI algorithm typically requires fewer iterations than VI; however, each iteration involves $O(|\mathcal{X}_t||\mathcal{S}_t|^2 + |\mathcal{S}_t|^3)$ operations, which is computationally more expensive than a single VI iteration. Although, a worst-case bound on the number of iterations is not known for the general implementation of PI [10], some naïve implementations are known to

exhibit exponential worst-case running times, similar to the simplex algorithm for LPs [11]. In fact, it has been established that there exists no strongly-polynomial algorithm that can solve model (1) in a number of operations that is strictly polynomial only in $|\mathcal{S}_t|$ and $|\mathcal{X}_t|$ [10]. However, MDPs are inherently related to linear programs in that they both belong to the class of P-complete problems that can potentially be solved in polynomial time using fast parallel algorithms [14]. It is well known that linear programs, such as (10) and (12), can be solved using interior-point methods (cf. [8]) that are polynomial in $|\mathcal{S}_t|$, $|\mathcal{X}_t|$ and B, where B denotes the maximum number of bits required to store any component in $\boldsymbol{Q}_t^\pi$ or $\boldsymbol{c}_t^\pi$. Thus, from a theoretical perspective, a polynomial-time LP algorithm offers obvious computational benefits over the VI algorithm, and may have potential advantages over the PI algorithm as well; however, the latter assertion remains an important open research question [10]. From a computational perspective, recent improvements in interior-point and simplex algorithms offer great promise for efficiently solving large-scale instances of (10) and (12) (see [9, 12, 13]).

The LP approach is even more appealing when concepts from approximate linear programming (ALP) can be used to solve our LP models in a computationally tractable way [4, 5]. In the ALP approach, the value function V_t^* is replaced by a surrogate function $\hat{V}_t$ that closely approximates V_t^*. This surrogate function is usually formed by selecting a set of parametric functions (basis functions) that project a high-dimensional state space to a lower-dimensional feature space. Typically, the number of basis functions (M) at each stage is significantly smaller than the cardinality of the state space, i.e., $M \ll |\mathcal{S}_t|$ for all $t \in T$. One common approximation scheme is a class of linearly parameterized functions of the form

$$V_t^*(\boldsymbol{s}) \approx \hat{V}_t(\boldsymbol{s}) = \sum_{i=1}^{M} r_t^i \phi_t^i(\boldsymbol{s}), \quad \boldsymbol{s} \in \mathcal{S}_t, \tag{13}$$

where $\phi_t^i : \mathcal{S}_t \to \mathbb{R}$ is the ith basis function, and $\boldsymbol{r}_t \equiv (r_t^1, \ldots, r_t^M)^T$ is a vector of basis function weights. Define the $|\mathcal{S}_t| \times M$-dimensional matrix $\boldsymbol{\Phi}_t$, where for each state $\boldsymbol{s} \in \mathcal{S}_t$ the corresponding row of $\boldsymbol{\Phi}_t$ is $(\phi_t^1(\boldsymbol{s}), \ldots, \phi_t^M(\boldsymbol{s}))$. The objective is to obtain a weight vector $\hat{\boldsymbol{r}}_t \in \mathbb{R}^M$ for each $t \in T$, such that $\hat{\boldsymbol{V}}_t \equiv \boldsymbol{\Phi}_t \hat{\boldsymbol{r}}_t$ closely approximates $\boldsymbol{V}_t^*$. The ALP formulation of model (1) is given by

$$\max \quad \sum_{t \in T} \boldsymbol{\beta}_t' \boldsymbol{\Phi}_t \boldsymbol{r}_t \tag{14a}$$

$$\text{s.t.} \quad (\boldsymbol{\Phi}_t \boldsymbol{r}_t)(\boldsymbol{s}) \le c_t(\boldsymbol{s}, \boldsymbol{x}) + \sum_{\boldsymbol{s}' \in \mathcal{S}_{t+1}} \delta \mathbb{P}_{t+1}(\boldsymbol{s}' | \boldsymbol{s}, \boldsymbol{x}) \boldsymbol{\Phi}_{t+1} \boldsymbol{r}_{t+1}(\boldsymbol{s}') \quad \forall t \in T', \, \boldsymbol{s} \in \mathcal{S}_t, \, \boldsymbol{x} \in \mathcal{X}_t, \tag{14b}$$

$$(\boldsymbol{\Phi}_N \boldsymbol{r}_N)(\boldsymbol{s}) \le c_N(\boldsymbol{s}) \qquad \forall \boldsymbol{s} \in \mathcal{S}_N, \tag{14c}$$

$$\boldsymbol{r}_t \in \mathbb{R}^M \qquad \forall t \in T. \tag{14d}$$

Note that the original primal model (10) has $\sum_{t \in T} |\mathcal{S}_t|$ variables and

$$|\mathcal{S}_N| + \sum_{t \in T'} \sum_{\boldsymbol{s} \in \mathcal{S}_t} |\mathcal{X}_t(\boldsymbol{s})|$$

constraints. By comparison, the approximate linear program (14) has only $N \cdot M$ variables, though the number of constraints remains as large as in (10). To reduce the number of constraints in (14), constraint sampling algorithms and fast ϵ-approximation algorithms can be used to solve (14) within reasonable time bounds for a given level of accuracy ϵ (see [16]). Solving the approximate LP is significantly easier than solving the original primal LP, owing to the reduced number of variables. However, the quality of solutions obtained by using (14) is highly dependent on the choice of the basis functions. One way to improve solution quality is to choose basis functions that possess properties of the value functions (e.g., convexity and monotonicity). Fortunately, for many finite-horizon problems arising in energy, healthcare, finance, transportation, and supply chain management applications, the value functions exhibit such structural properties (cf. [7] and the references therein).

5 Conclusions

We have established the existence of a linear programming formulation for general non-stationary, finite-horizon MDP models under a risk-neutral objective. The formulation is also extendable to MDP models with risk-averse objectives. We have shown that the value functions of the finite-horizon MDP model are the optimal solutions of the associated primal LP model, while the optimal policy can be fully recovered via the optimal solutions of the dual LP model.

From a theoretical perspective, solving the LP formulations can provide computational advantages over the standard value and policy iteration algorithms. Perhaps most importantly, these formulations naturally facilitate the use of approximate LP procedures for finite-horizon MDP models. We proposed parameterized basis functions that can be used to approximate the value functions, and which lead to a computationally tractable formulation (see (14)). This approximate formulation drastically reduces the number of decision variables, and constraint sampling algorithms can be used to deal with the large number of constraints. Although solution quality is highly dependent on basis function selection, this approach provides sufficient flexibility to model value functions possessing useful properties, such as convexity or monotonicity. In the future, it will be instructive to examine performance guarantees related to formulation (14) and how they compare to those of the infinite-horizon case.

Acknowledgements

We gratefully acknowledge the constructive and helpful comments of an anonymous referee, as well as those of Professor Daniel Jiang of the University of Pittsburgh.

References

[1] D. Adelman. Dynamic bid prices in revenue management. *Operations Research*, 55(4):647–661, 2007.

[2] D.P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. II, 4th Edition*. Athena Scientific, 2012.

[3] D.P. Bertsekas and J.N Tsitsiklis. *Neurodynamic Programming*. Athena Scientific, Belmont, 1996.

[4] D.P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865, 2003.

[5] D.P. de Farias and B. Van Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, 29(3):462–478, 2004.

[6] A. Hordijk and L.C.M. Kallenberg. Linear programming and Markov decision chains. *Management Science*, 25(4):353–362, 1979.

[7] D. Jiang and W.B. Powell. An approximate dynamic programming algorithm for monotone value functions. *Operations Research*, 63(6):1489–1511, 2015.

[8] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

[9] Y.T. Lee and A. Sidford. Path finding methods for linear programming: Solving linear programs in $\tilde{O}(\sqrt{\text{rank}})$ iterations and faster algorithms for maximum flow. In *Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 424–433, 2014.

[10] M.L. Littman, T.D. Dean, and L.P. Kaelbling. On the complexity of solving Markov decision problems. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, pages 394–402, 1995.

[11] M. Melekopoglou and A. Condon. On the complexity of the policy iteration algorithm for stochastic games. Technical report, CS-TR-90-941, Department of Computer Science, University of Wisconsin, Madison, WI, 1990.

[12] Y. Nesterov and A. Nemirovski. *Interior-point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics (SIAM), 1994.

[13] P. Pan. A fast simplex algorithm for linear programming. *Journal of Computational Mathematics*, 28(6):837–847, 2010.

[14] C. Papadimitriou and J.N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[15] J. Patrick, M.L. Puterman, and M. Queyranne. Dynamic multipriority patient scheduling for a diagnostic resource. *Operations Research*, 56(6):1507–1525, 2008.

[16] S.A. Plotkin, D.B. Shmoys, and E. Tardos. Fast approximation algorithms for fractional packing and covering problems. In *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science*, pages 495–504, 1991.

[17] W.B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality, 2nd Edition*. John Wiley & Sons, 2011.

[18] M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2009.

[19] S.M. Ross. *Introduction to Stochastic Dynamic Programming, 1st Edition*. Academic Press, 1983.

[20] R.S. Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Proceedings of Advances in Neural Information Processing Systems 8*, pages 1038–1044, 1995.

[21] H. Topaloglu and W.B. Powell. A distributed decision-making structure for dynamic resource allocation using nonlinear functional approximations. *Operations Research*, 53(2):281–297, 2005.

[22] C.J.C.H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3):279–292, 1992.

[23] M. Weiring and M. van Otterlo. *Reinforcement Learning: State-of-the-Art, 1st Edition*. Springer, 2012.