A Shadow Simplex Method for Infinite Linear Programs

Archis Ghate
The University of Washington
Seattle, WA 98195

Dushyant Sharma
The University of Michigan
Ann Arbor, MI 48109

Robert L. Smith
The University of Michigan
Ann Arbor, MI 48109

May 25, 2009

Abstract

We present a Simplex-type algorithm, that is, an algorithm that moves from one extreme point of the infinite-dimensional feasible region to another not necessarily adjacent extreme point, for solving a class of linear programs with countably infinite variables and constraints. Each iteration of this method can be implemented in finite time, while the solution values converge to the optimal value as the number of iterations increases. This Simplex-type algorithm moves to an adjacent extreme point and hence reduces to a true infinite-dimensional Simplex method for the important special cases of non-stationary infinite-horizon deterministic and stochastic dynamic programs.

1 Introduction

In this paper, we present a Simplex-type algorithm to solve a class of *countably infinite linear programs* (henceforth CILPs), i.e., linear programming problems with countably infinite variables and countably infinite constraints. CILPs often arise from infinite-horizon dynamic planning problems [27, 28] in a variety of models in Operations Research, most notably, a class of deterministic or stochastic dynamic programs with countable states [17, 32, 41, 45] whose special cases include infinite-horizon problems with time-indexed states considered in Sections 4 and 5. Other interesting special cases of CILPs include infinite network flow problems [42, 47], infinite extensions of Leontief systems [50, 51], and semi-infinite linear programs [5, 25, 26], i.e., problems in which *either* the number of variables *or* the number of constraints is allowed to be countably infinite. CILPs also arise in the analysis of games with partial information [13], linear search problems with applications to robotics [15] and infinite-horizon stochastic programs [30, 34].

Unfortunately, positive results on CILPs are scarce due to a disturbing variety of mathematical pathologies in subspaces of R^{∞}. For example, weak duality and complementary slackness may not hold [44], the primal and the dual may have a duality gap [5], the primal may not have any extreme points even when each variable is bounded [5], extreme points may not be characterized as basic feasible solutions, and finally, basis matrices, reduced costs, and optimality conditions are not straightforward [5].

It is perhaps due to the complications outlined above that almost all published work on concrete solution algorithms for infinite linear programs has focused on the semi-infinite and/or the uncountable, i.e., "continuous" case. A partially successful attempt to extend the Simplex method to semi-infinite linear programs with finitely many variables and uncountably many constraints was made in [4]. A simplex-type method for semi-infinite linear programs in Euclidean spaces was developed in [3] using the idea of locally polyhedral linear inequality systems from [2]. A value convergent approximation scheme for linear programs in function spaces was proposed in [31]. Weiss worked on separated continuous linear programs [52] and developed an implementable algorithm for their solution in MATLAB. Earlier, Pulan also worked on similar problems [39, 40]. Continuous network flow problems were studied in [6, 7, 38]. There has been a recent surge of interest in applying the theory developed in [5] to uncountable state-space stationary Markov and semi-Markov decision problems [17, 18, 32, 35, 36].

Recent work [47] on countably infinite network flow problems used the characterization of extreme points through positive variables for countably infinite network flow problems from [42] to devise a Simplex method. It was noted that each pivot operation in that Simplex method may require infinite computation in general. On the other hand, each pivot could be performed in finite time for a restricted subclass of inequality constrained network flow problems. Unfortunately, this class excludes countable state dynamic programming problems that are often the motivation for studying CILPs.

The above observations have recently motivated our theoretical work on CILPs where we developed duality theory [23] and sufficient conditions for a basic feasible characterization of extreme points [22]. In this paper, we focus on algorithmic aspects of CILPs. From this perspective, since CILPs are characterized by *infinite information*, our first question is whether it is possible to design a procedure that uses *finite information* to perform *finite computations* in each iteration and yet approximates the optimal value of the original infinite problem. Any such implementable procedure could proceed by solving a sequence of finite-dimensional truncations of the original infinite problem [27, 28, 29]. However, constructing an appropriate finite truncation of countably infinite equality constraints in countably infinite variables is not straightforward. When such a truncation is naturally available owing to amenable structure of the constraint matrix as in [44] or in this paper, it would indeed be sufficient to solve a large enough truncation by any solution method to approximate the optimal value of the infinite problem assuming it is embedded in appropriate infinite-dimensional vector spaces. Then the second, far more demanding question is whether it is possible to approximate an optimal policy and in particular an optimal extreme point policy. Note that our special interest in extreme point solutions is motivated by their useful properties, for example, correspondence to deterministic policies in Markov decision problems [41]. An additional complication in this context is that countably infinite linear programs may have an *uncountable* number of extreme points, and unlike finite-dimensional linear programs, values of a sequence of extreme point solutions with strictly decreasing values may not converge to the optimal value as illustrated in the binary tree example in Section 2.

The approach presented in this paper surmounts difficulties listed above for a class of CILPs with a finite number of variables appearing in every constraint. These CILPs subsume the important class of non-stationary infinite-horizon dynamic programs in Sections 4 and 5 and more generally are common in dynamic planning problems where problem data are allowed to vary over time owing to technological and economic change hence providing a versatile modeling and optimization tool. We devise an *implementable* Simplex-type procedure, i.e., an algorithm that implicitly constructs a sequence of extreme points in the infinite-dimensional feasible region and asymptotically converges to the optimal value *while performing finite computations on finite information in each step*. This is achieved by employing results in [14], which assert that each extreme point of any finite-dimensional projection, i.e., *shadow* of the infinite-dimensional feasible region can be appended to form some extreme point of the infinite-dimensional feasible region. Specifically, we develop a polyhedral characterization of these shadows and employ the standard Simplex method for solving the resulting finite-dimensional linear programs of increasing dimensions. For the important special case of non-stationary infinite-horizon dynamic programs in Sections 4 and 5, our Simplex-type method moves through adjacent extreme points of the infinite-dimensional feasible region and hence reduces to a true Simplex method in the conventional sense. This partly answers a question from [5] as to whether it is possible to design a finitely implementable Simplex method for any non-trivial class of CILPs in the affirmative. Owing to the critical role that finite-dimensional shadows play in our approach, we call it the *Shadow Simplex* method.

2 Problem Formulation, Preliminary Results and Examples

We focus on problem (P) formulated as follows:

$$(P) \quad \min \sum_{j=1}^{\infty} c_j x_j$$

$$\text{subject to} \quad \sum_{j=1}^{\infty} a_{ij}x_j = b_i, \quad i = 1, 2, \ldots$$
$$x_j \geq 0, \quad j = 1, 2, \ldots$$

where c, b and x are sequences in R^∞ and for $i = 1, 2, \ldots$, $a_i \equiv \{a_{ij}\}_{j=1}^{\infty} \in R^\infty$ is the ith row vector of a doubly infinite matrix A. Note that CILPs with inequality constraints and free variables can be converted to the standard form (P) above as in the finite-dimensional case. Our assumptions below ensure that all infinite sums in the above formulation are well-defined and finite. We also show later (Proposition 2.7) that (P) indeed has an optimal solution justifying the use of "min" instead of "inf". We employ the product topology, i.e., the topology of componentwise convergence on R^∞ throughout this paper.

We now discuss our assumptions in detail. The first assumption is natural.

Assumption 2.1. Feasibility: *The feasible region F of problem (P) is non-empty.*

Assumption 2.2. Finitely Supported Rows: *Every row vector a_i of matrix A has a finite number of non-zero components, i.e., each equality constraint has a finite number of variables.*

Note that this assumption *does not* require the number of variables appearing in each constraint to be uniformly bounded. Note as a simple example that this assumption holds in infinite network flow problems if node degrees are bounded. Similarly in infinite-horizon deterministic production planning problems where there is one inventory balance constraint in each period and every such constraint has three variables with non-zero coefficients. In addition, this assumption is also satisfied in CILP formulations of deterministic and stochastic dynamic programming problems discussed in detail in Sections 4 and 5. Assumption 2.2 helps in the proof of closedness of the feasible region F in Lemma A.1 in Appendix A. In addition, it is used in Section 3 to design finite-dimensional truncations of F that ensure a finite implementation of iterations of Shadow Simplex.

Assumption 2.3. Variable Bounds: *There exists a sequence of non-negative numbers $\{u_j\}_{j=1}^{\infty}$ such that for every $x \in F$ and for every j, $x_j \leq u_j$.*

Note that this assumption *does not* require a uniform upper bound on variable values. It holds by construction in CILP formulations of dynamic programming problems in Sections 4 and 5, and in capacitated network flow problems. Assumption 2.3 implies that F is contained in a compact subset of R^∞ and hence ensures, along with closedness of F proved in Lemma A.1, that F is compact as in Corollary A.2 in Appendix A.

Assumption 2.4. Uniform Convergence: *There exists a sequence of non-negative numbers $\{u_j\}_{j=1}^{\infty}$ as in Assumption 2.3 for which $\sum_{j=1}^{\infty} |c_j|u_j < \infty$.*

Remark 2.5. *Let $\{u_j\}$ be as in Assumption 2.4. Then the series $\sum_{i=1}^{\infty} c_i x_i$ converges uniformly over $X = \{x \in R^\infty : 0 \leq x_j \leq u_j\}$ by Weierstrass M-test [8] hence the name uniform convergence. The objective function may be written as $C(x) \equiv \sum_{i=1}^{\infty} c_i x_i$, $C : X \to R$. Since the functions $f_i : X \to R$ defined as $f_i(x) = c_i x_i$ that form the above series are each continuous over X, the function $C(x)$ is also continuous over X [8] and hence over $F \subseteq X$. Nevertheless we prove this continuity from first principles in Appendix A Lemma A.3.*

Assumption 2.4 is motivated by a similar assumption in the general infinite-horizon optimization framework of [46] and other more specific work in this area [44]. It is conceptually similar to the ubiquitous assumption (see Chapter 3 of [5]) in infinite-dimensional linear programming that the costs and the variables are embedded in a (continuous) dual pair of vector spaces. Such assumptions are also common in mathematical economics where commodity consumptions and prices are embedded in dual pairs of vector spaces [1]. It allows us to treat problems embedded in a variety of sequence spaces in R^∞ within one common framework as for example shown in the following Lemma proven in Appendix A.

Lemma 2.6. *Assumption 2.4 holds in each of the following situations:*

1. *When c is in l_1, the space of absolutely summable sequences, and u in Assumption 2.3 is in l_∞, the space of bounded sequences. This situation is common in planning problems where activity levels are uniformly bounded by finite resources and costs are discounted over time.*
2. *When c is in l_∞, and u in Assumption 2.3 is in l_1. This situation arises in CILP equivalents of Bellman's equations for discounted dynamic programming as in Sections 4 and 5 where immediate costs are uniformly bounded and variables correspond to state-action frequencies that sum to a finite number often normalized to one.*

(The reader may recall here that $\langle l_1, l_\infty\rangle$ is a dual pair of sequence spaces [1]).

Proposition 2.7. *Under Assumptions 2.1, 2.2, 2.3, and 2.4, problem (P) has an extreme point optimal solution.*

The proof of this proposition, provided in Appendix A, follows the standard approach of confirming that the objective function and the feasible region of (P) satisfy the hypotheses of the well-known Bauer Maximum Principle (Theorem 7.69 page 298 of [1]), which implies that (Corollary 7.70 page 299 of [1]) a continuous linear functional has an extreme point minimizer over a nonempty convex compact subset of a locally convex Hausdorff space (such as R^∞ with its product topology). In the sequel, we denote the set of optimal solutions to problem (P) by F^*.

We first show (Value Convergence Theorem 2.9) that optimal values of mathematical programs with feasible regions formed by finite-dimensional projections, i.e., *shadows* of the infinite-dimensional feasible region F converge to the optimal value of (P). The necessary mathematical background and notation from [14] is briefly reviewed here. Specifically, we recall from [14] the concept of a projection of a non-empty, compact, convex set in R^∞ such as F. The projection function $p_N : R^\infty \to R^N$ is defined as $p_N(x) = (x_1, \ldots, x_N)$ and the projection of F onto R^N as

$$F_N = \{p_N(x) : x \in F\} \subset R^N \tag{1}$$

for each $N = 1, 2, 3, \ldots$. The set F_N can also be viewed as a subset of R^∞ by appending it with zeros as follows

$$F_N = \{(p_N(x); 0, 0, \ldots) : x \in F\} \subset R^\infty. \tag{2}$$

Using F_N to denote both these sets should not cause any confusion since the meaning will be clear from context. Our value convergence result uses the following lemma from [14].

Lemma 2.8. *([14]) The sequence of projections F_N converges in the Kuratowski sense to F as $N \to \infty$, i.e.,*

$$\liminf F_N = \limsup F_N = \lim F_N = F.$$

Now consider the following sequence of optimization problems for $N = 1, 2 \ldots$:

$$P(N) \quad \min \sum_{i=1}^{N} c_i x_i, \quad x \in F_N.$$

Set F_N is non-empty, convex, compact (inheriting these properties from F), finite-dimensional and the objective function is linear implying that $P(N)$ has an extreme point optimal solution. Let F_N^* be the set of optimal solutions to $P(N)$. We have the following convergence result.

Theorem 2.9. Value Convergence: *The optimal value $V(P(N))$ in problem $P(N)$ converges to the optimum value $V(P)$ in problem (P) as $N \to \infty$. Moreover, if $N_k \to \infty$ as $k \to \infty$ and $x^k \in F^*_{N_k}$ for each k, then the sequence $\{x^k\}$ has a limit point in F^*.*

Our proof of this theorem in Appendix A employs Berge's Maximum Theorem (Theorem 17.31 page 570 of [1]), where the key intuitive idea is to use convergence of feasible regions of problems $P(N)$ to the feasible region of problem (P) from Lemma 2.8 and continuity of their objective functions to establish value convergence. Theorem 2.9 implies that the optimal values in $P(N)$ arbitrarily well-approximate the optimum value in (P). In addition, when (P) has a unique optimal solution x^*, a sequence of optimal solutions to finite-dimensional shadow problems converges to x^*. Similar value convergence for finite-dimensional approximations of a different class of CILPs was earlier established in [28]. Finally, we remark that since $V(P(N))$ is a sequence of real numbers that converges to $V(P)$ as $N \to \infty$, $V(P(N_k))$ also converges to $V(P)$ as $k \to \infty$ for any subsequence N_k of positive integers. This fact will be useful in Section 3.

2.1 Examples

Infinite horizon non-stationary dynamic programs, one of our most important and largest class of applications, is discussed in Sections 4 and 5. Here we present two concrete prototypical examples where Assumptions 2.1-2.4 are easy to check.

Production planning: Consider the problem of minimizing infinite-horizon discounted production and inventory costs while meeting an infinite stream of integer demand for a single product [43]. Demand during time-period $n = 1, 2 \ldots$ is $D_n \le D$, unit cost of production is $0 \le k_n \le K$ during period n and unit inventory holding cost is $0 \le h_n \le H$ at the end of period n. The discount factor is $0 < \alpha < 1$. Production capacity (integer) in period $n = 1, 2, \ldots$ is $P_n \le P$ and inventory warehouse capacity (integer) is $I_n \le I$ ending period $n = 0, 1, 2, \ldots$. Then letting the decision variable x_n for $n = 1, 2, \ldots$ denote production level in period n, and y_n denote the inventory ending period n for $n = 0, 1, \ldots$, where y_0 is fixed, we obtain the following CILP:

$$
\begin{array}{rcl}
(PROD) \min & \sum\limits_{n=1}^{\infty} & \alpha^{n-1}(k_n x_n + h_n y_n) \\
x_n & \le & P_n, \ \ n = 1, 2, \ldots \\
y_n & \le & I_n, \ \ n = 1, 2, \ldots \\
y_{n-1} + x_n - y_n & = & D_n, \ \ n = 1, 2, \ldots \\
x_n, y_n & \ge & 0, \ \ n = 1, 2, \ldots
\end{array}
$$

Note that problem $(PROD)$ can be converted into form (P) after adding non-negative slack variables in the production and inventory capacity constraints respectively. A sufficient condition for Assumption 2.1 to hold is that production capacity dominates demand meaning $P_n \ge D_n$ for $n = 1, 2, \ldots$. Assumption 2.2 is satisfied as the inventory balance constraints have three variables each and the capacity constraints (after adding slack variables) have two variables each. Assumption 2.3 holds because $|x_n| \le P$, $|y_n| \le I$. Finally, for Assumption 2.4, note that $|k_n| \le K$, $|h_n| \le H$, and

$$
\sum_{n=1}^{\infty} \alpha^{n-1} KP + \sum_{n=1}^{\infty} \alpha^{n-1} HI = \frac{KP + HI}{1 - \alpha} < \infty.
$$

Dynamic resource procurement and allocation: We present a dynamic extension of a prototypical planning problem in linear programming [37]. Consider a resource allocation problem with n activities and m resources with opportunities to purchase resources from an external source with limited availability. In particular, during time-period $t = 1, 2, \ldots$, amount $0 \le b_i(t) \le B_i$ of resource $i = 1, \ldots, m$ is available for consumption. An additional amount up to $0 \le D_i(t) \le D_i$ of resource i may be purchased at unit cost

$0 \leq c_i(t) \leq c_i$ in period t. Each unit of activity j consumes amount $0 < a_{ij}(t)$ of resource i in period t for $j = 1, \ldots, n$, and $i = 1, \ldots, m$. Let $a_{ij} \equiv \inf_t a_{ij}(t)$. We assume that $a_{ij} > 0$, that is, a unit of activity j consumes a strictly positive amount of resource i in all time periods. Each unit of activity j yields revenue $0 \leq r_j(t) \leq r_j$ in period t. Resources left over from one period can be consumed in future periods however the carrying capacity for resource i is $0 \leq E_i(t) \leq E_i$ from period t to $t+1$. The cost of carrying a unit of resource i from period t to period $t+1$ is $0 \leq h_i(t) \leq h_i$. The discount factor is $0 < \alpha < 1$. Our goal is to determine an infinite-horizon resource procurement and allocation plan to maximize net revenue. Let $x_j(t)$ denote the level of activity j in period t, $y_i(t)$ denote the amount of resource i carried from period t to $t+1$, and $z_i(t)$ denote the amount of resource i purchased in period t. Let $y_i(0) = 0$ for all i. The optimization problem at hand in these decision variables can be formulated as the following CILP:

$$(RES-PROC-ALL)\ \max \sum_{t=1}^{\infty} \alpha^{t-1} \left(\sum_{j=1}^{n} r_j(t)x_j(t) - \sum_{i=1}^{m} c_i(t)z_i(t) - \sum_{i=1}^{m} h_i(t)y_i(t) \right)$$

$$\begin{aligned}
z_i(t) &\leq D_i(t), \quad i = 1, \ldots, m; \quad t = 1, 2, \ldots \\
y_i(t) &\leq E_i(t), \quad i = 1, \ldots, m; \quad t = 1, 2, \ldots \\
\sum_{j=1}^{m} a_{ij}(t)x_j(t) + y_i(t) - y_i(t-1) - z_i(t) &= b_i(t), \quad i = 1, \ldots, m; \quad t = 1, 2, \ldots \\
z_i(t) &\geq 0, \quad i = 1, \ldots, m; \quad t = 1, 2, \ldots \\
y_i(t) &\geq 0, \quad i = 1, \ldots, m; \quad t = 1, 2, \ldots \\
x_j(t) &\geq 0, \quad j = 1, \ldots, n; \quad t = 1, 2, \ldots
\end{aligned}$$

Problem $(RES-PROC-ALL)$ can be converted into standard form (P) by transforming into a net cost minimization problem and adding non-negative slack variables in the capacity constraints. This problem is feasible. For example, select and fix any activity j and set $x_j(t) = b_i(t)/a_{ij}(t)$, $x_k(t) = 0$ for all $k \neq j$, $y_i(t) = z_i(t) = 0$ for all resources i and all t yielding a feasible solution. Thus Assumption 2.1 holds. Each material balance constraint includes $m+3$ variables whereas each capacity constraint includes two variables (after adding slacks) hence satisfying Assumption 2.2. Note that $x_j(t) \leq \frac{\max_i (B_i + D_i + E_i)}{\max_i a_{ij}} \equiv F_j$ for all time periods t. Then Assumption 2.3 holds with vector u whose components all equal $B \equiv \max\{\max_i D_i, \max_i E_i, \max_j F_j\}$. This u is in l_∞. Let $M_1 = \sum_{j=1}^{n} r_j$, $M_2 = \sum_{i=1}^{m} c_i$, $M_3 = \sum_{i=1}^{m} h_i$, and $M = \max\{M_1, M_2, M_3\}$. Then Assumption 2.4 is satisfied because the cost vector is in l_1 as $\sum_{t=1}^{\infty} \alpha^{t-1} M = \frac{M}{1-\alpha} < \infty$.

Recall that our goal is to design an implementable Simplex-type procedure for solving problem (P). Since this involves implicitly constructing a value convergent sequence of extreme points of the infinite-dimensional feasible region F using finite computations, we must devise a "finite representation" of these extreme points. This is achieved in Section 3, however we first illustrate with a binary tree example some of the challenges involved in designing a value convergent sequence of extreme points even for CILPs that appear simple.

A binary tree example: Consider a network flow problem on the infinite directed binary tree shown in Figure 1. The nodes are numbered $1, 2, \ldots$ starting at the root node. Tuple (i, j) denotes a directed arc from node i to node j. There is a source of $(1/4)^i$ at nodes at depth i in the tree, where the root is assumed to be at depth 0. The cost of sending a unit of flow through "up" arcs at depth i in the tree is $(1/4)^i$, where the arcs emerging from the root are assumed to be at depth 0. The cost of pumping unit flow through "down" arcs is always 0. The objective is to push the flow out of each node to infinity at minimum cost while satisfying the flow balance constraints. The unique optimal solution of this problem is to push the flow through the "down" arc at each node at zero total cost. This flow problem over an

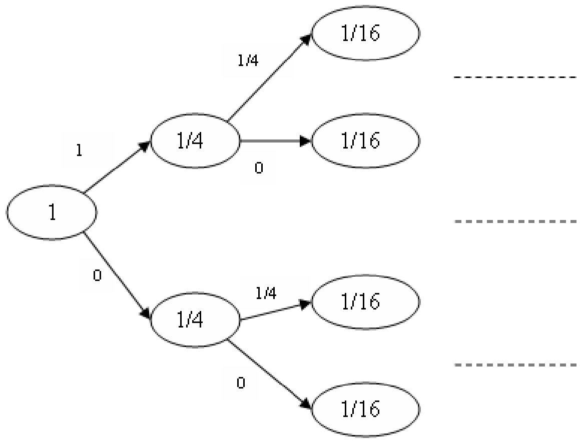

Figure 1: An infinite network flow problem.

infinite binary tree can be formulated as a CILP that fits the framework of (P) above. This CILP has flow balance (equality) constraints in non-negative flow variables. It is clearly feasible satisfying Assumption 2.1 and satisfies Assumption 2.2 as each flow balance constraint has three entries two of which are -1 the third being $+1$. As for Assumption 2.3, note that the flow in any arc must be less than the total supply at all nodes, which equals $\sum_{i=0}^{\infty} 2^i \frac{1}{4^i} = \sum_{i=0}^{\infty} \frac{1}{2^i} = 2$ implying that the choice $u_j = 2$ for all j suffices hence this u is in l_∞. Moreover, the cost vector is in l_1 since the sum of costs associated with all arcs is again $\sum_{i=0}^{\infty} 2^i \frac{1}{4^i} = 2$. Thus Assumption 2.4 holds.

Extreme points of feasible regions of infinite network flow linear programs were defined in [21, 42]. For the network flow problem in Figure 1, a feasible flow is an extreme point if it has exactly one path to infinity out of every node, where a "path to infinity" is defined as a sequence of directed arcs $(i_1, i_2), (i_2, i_3), \ldots$ with positive flows. In other words, a feasible flow is an extreme point if every node pushes the total incoming flow out through exactly one of the two emerging arcs. Thus, this feasible region has an *uncountable* number of extreme points. We will say that two arcs are "complementary" if they emerge from the same node. A pivot operation involves increasing the flow through one arc from zero to an appropriate positive value and decreasing the flow through its complementary arc to zero. It is then possible to construct an infinite sequence of adjacent extreme point solutions whose values (strictly) monotonically decrease but do not converge to the optimal value.

More specifically, suppose we start at the extreme point solution illustrated in Figure 2 (a), where the arcs with positive flows are shown with solid lines and the ones with zero flows with dotted lines. In this extreme point solution, a flow of $(1/4)^i$ is pushed through 2^i paths each with total cost $\sum_{j=i}^{\infty}(1/4)^j$ for $i = 0, 1, 2, \ldots$. Thus the cost of this extreme point solution equals

$$\sum_{i=0}^{\infty} \left(\frac{1}{4}\right)^i (2^i) \sum_{j=i}^{\infty} \left(\frac{1}{4}\right)^j = \sum_{i=0}^{\infty} \left(\frac{1}{4}\right)^i (2^i) \left(\frac{1}{4}\right)^i \sum_{j=0}^{\infty} \left(\frac{1}{4}\right)^j = \sum_{i=0}^{\infty} \left(\frac{1}{8}\right)^i \left(\frac{4}{3}\right) = \frac{32}{21}.$$

The adjacent extreme point formed by increasing the flow in arc a_1 and decreasing the flow in arc b_1 to zero is shown in Figure 2 (b). This extreme point has a strictly lower cost than the initial extreme point. Similarly, the extreme point obtained by another pivot operation that increases the flow in arc a_2 and decreases the flow in arc b_2 to zero is shown in Figure 2 (c). Again, this extreme point has strictly lower cost than the one in Figure 2 (b). Repeating this infinitely often, we obtain a sequence of adjacent extreme points with strictly decreasing values that remain above one — the cost of the "up" arc out of the root node. Note that such a situation cannot occur in feasible, finite-dimensional problems where every extreme point is non-degenerate and the optimal cost is finite, as the Simplex method that chooses a non-basic variable with negative reduced cost to enter the basis in every iteration reaches an optimal extreme point in a finite number of iterations.

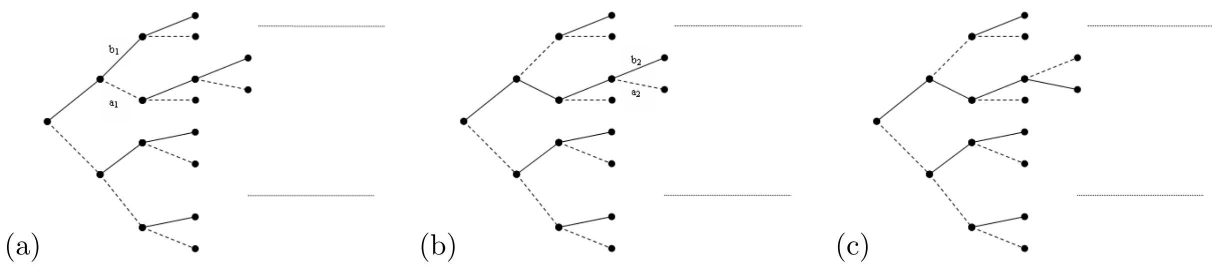

Figure 2: Pivots in the binary tree example described in the text.

3 The Shadow Simplex method

The Shadow Simplex method builds upon results in [14], which provide that for every n, and for every extreme point $x(n)$ of an n-dimensional shadow, there exist extreme points of all higher dimensional shadows, including the infinite-dimensional one, such that their first n coordinates exactly coincide with those of $x(n)$. These results from [14] are stated here.

Lemma 3.1. *([14]) For every extreme point x of F_N there exists an extreme point of F_{N+1} which is identical to x in its first N components.*

Lemma 3.2. *([14]) For every extreme point x of F_N there exists an extreme point of F_M, for every $M > N$ which is identical to x in its first N components.*

Lemma 3.3. *([14]) For every extreme point x of F_N there exists an extreme point of F which is identical to x in its first N components.*

Remark 3.4. *In view of Lemmas 3.1, 3.2 and 3.3 we informally say that every extreme point of F_N is* liftable *for each $N = 1, 2, \ldots$. In other words every extreme point of a finite-dimensional shadow F_N is a projection of some extreme point of F. As a result, a sequence of extreme points of finite-dimensional shadows of F is in fact a projection of a sequence of extreme points of F. Thus an algorithm that moves from one extreme point of a finite-dimensional shadow to another implicitly constructs a sequence of extreme points of the infinite-dimensional feasible region. This observation is central to the Shadow Simplex method.*

Unfortunately, it is not in general easy to characterize shadows of F. We therefore focus attention on "nice" CILPs where shadows of F equal feasible regions of finite-dimensional linear programs derived from (P). This requires some more notation and an assumption that are discussed here. Under Assumption 2.2, without loss of generality (with possibly reordering the variables) we assume that (P) is formulated so that for every N there is an $L_N < \infty$ such that variables $x_{L_N+1}, x_{L_N+2}, \ldots$ do not appear in the first N constraints. See [44] for a detailed mathematical discussion on this issue. As a simple example, let v_n and w_n be the non-negative slack variables added in the production and inventory capacity constraints of the production planning problem described in Section 2. We order the variables as $(x_1, v_1, y_1, w_1, x_2, v_2, \ldots)$ and the equality constraints by time-period:

$$x_1 + v_1 = P, \;\; y_1 + w_1 = I, \;\; x_1 - y_1 = D_1 - y_0,$$

$$x_2 + v_2 = P,\ \ y_2 + w_2 = I,\ \ y_1 + x_2 - y_2 = D_2\ \ \ldots$$

Then it is easy to see that $L_1 = 2,\ L_2 = 4,\ L_3 = 4,\ L_4 = 6,\ L_5 = 8,\ L_6 = 8$ .... Since CILPs most commonly arise from infinite-horizon planning problems, such an ordering of variables and constraints is often the most natural as there is a set of one or more constraints grouped together corresponding to every time period (see [27, 44]).

Now consider *truncations* T_N of F defined as

$$T_N = \{x \in R^{L_N} : \sum_{j=1}^{L_N} a_{ij}x_j = b_i,\ i = 1, \ldots, N;\ x_j \geq 0,\ j = 1, 2, \ldots, L_N\}.$$

That is, T_N is the feasible region formed by ignoring all variables beyond the L_Nth and all equality constraints beyond the Nth. Observe that $F_{L_N} \subseteq T_N$. To see this, let $y \in F_{L_N}$. Then by definition of F_{L_N}, $y_j \geq 0$ for $j = 1, 2, \ldots, L_N$, $y \in R^{L_N}$ and there is some $x \in F$ whose first L_N components match with y. Thus $y \in T_N$ because the variables beyond the L_Nth do not appear in the first N equality constraints in problem (P).

Definition 3.5. *The truncation T_N is said to be extendable if for any $x \in T_N$ there exist real numbers $y_{L_N+1}, y_{L_N+2}, \ldots$ such that*

$$(x_1, x_2, \ldots, x_{L_N}, y_{L_N+1}, y_{L_N+2}, \ldots) \in F,$$

i.e., if any solution feasible to truncation T_N can be appended with an infinite sequence of variables to form a solution feasible to (P).

Lemma 3.6. *Truncation T_N is extendable if and only if $T_N = F_{L_N}$.*

In non-stationary infinite-horizon deterministic dynamic programs we consider in Section 4, we assume that in a finite-horizon truncation, a finite sequence of decisions that reaches a "terminal" state can be appended with an infinite sequence of decisions to construct a decision sequence feasible to the original infinite-horizon problem. *This assumption is without loss of generality by following a "big-M" approach mentioned in Section 4 and discussed in more detail in Appendix A.* Consequently, extendability of finite-horizon truncations can be forced in CILP formulations of *all* non-stationary infinite-horizon dynamic programs — our largest class of deterministic sequential decision problems. Thus we work with the following assumption in the rest of this paper.

Assumption 3.7. *There exists an increasing sequence of integers $\{N_n\}_{n=1}^{\infty}$ for which truncations T_{N_n} are extendable.*

In infinite-horizon planning problems the above sequence of integers is often indexed by lengths of finite horizons $n = 1, 2, \ldots$ and N_n corresponds to the number of equality constraints that appear in the n-horizon problem. We discuss three concrete examples.

In the production planning problem $(PROD)$ in Section 2, we consider the sequence $N_n = 3n$ for $n = 1, 2, \ldots$ since there are 3 equality constraints in every period after adding slack variables. If the production capacity dominates demand in every period, i.e., $P_n \geq D_n$ for all n, then truncations T_{3n} are extendable for all n. Notice that this dominance is not necessary for extendability, which in fact can be forced without loss of generality whenever $(PROD)$ has a feasible solution by adding "sufficient inventory" inequality constraints in $(PROD)$. In particular, let $\Delta_m^n = \sum_{i=n+1}^{m} (D_i - P_i)$ for $n = 0, 1, \ldots$, and $m = n+1, n+2, \ldots$. Also let $\Delta^n = \max\left(\{0, \Delta_{n+1}^n, \Delta_{n+2}^n, \ldots\}\right)$. Here Δ_m^n represents the "total deficit" in production capacities in periods $n+1$ through m as compared to the total demand in these periods. Thus, in order to satisfy the demand in these periods, inventory y_n must be at least Δ_m^n. The quantity Δ^n is the largest of all such inventory requirements, and represents the minimum inventory needed ending period n to satisfy future demand. Thus, if $y_1, y_2, \ldots$ is an inventory schedule feasible to $(PROD)$ then it must satisfy $y_n \geq \Delta^n$ for $n = 1, 2 \ldots$. Hence we can add these "sufficient inventory" inequalities to $(PROD)$ without altering the set of feasible production-inventory schedules, and finite-horizon truncations of this modified problem are extendable (see [24] for details). This approach of adding "valid inequalities" to ensure extendability often works for planning problems where decisions in two periods are linked by some kind of an inventory variable.

Note that finite-horizon truncations of the resource allocation problem $(RES - PROC - ALL)$ are also extendable. To see this, suppose for some t-horizon feasible solution, resource type j left over at the end of period t is $y_j(t)$. This feasible solution can be extended to an infinite-horizon feasible solution by exhausting available resource $y_j(t) + b_j(t+1)$ in period $t+1$, never buying additional resources in periods $t' > t$, and choosing activity levels in periods $t' > t+1$ to entirely consume resource amounts $b_j(t')$ so that there is no resource inventory carried over to period $t'+1$. Thus Assumption 3.7 holds with $N_t = 3mt$ for $t = 1, 2, \ldots$.

Under Assumption 3.7, Lemma 3.6 characterizes shadows $F_{L_{N_n}}$ as T_{N_n}. Thus we define a sequence of finite-dimensional linear programming problems $P(L_{N_n})$ for $n = 1, 2 \ldots$ as follows:

$$P(L_{N_n}) \quad \min \sum_{i=1}^{L_{N_n}} c_i x_i, \quad x \in F_{L_{N_n}} \equiv T_{N_n}.$$

Since the integer subsequence $L_{N_n} \to \infty$ as $n \to \infty$, Theorem 2.9 implies that optimal values $V(P(L_{N_n}))$ converge to the optimal value $V(P)$ as desired when $n \to \infty$. We are now ready to present the Shadow Simplex method.

3.1 The Shadow Simplex Method

The Shadow Simplex method is an iterative procedure that runs in stages $n = 1, 2, \ldots$, where problem $P(L_{N_n})$ is solved to optimality in stage n. Theorem 2.9 implies that the optimum objective function value achieved at the end of stage n converges to the optimum objective function value of problem (P) as $n \to \infty$. We informally explain the implementation details of these stages here. Problem $P(L_{N_1})$ has N_1 constraints and L_{N_1} variables. A Phase I procedure is employed to find an extreme point $x(N_1)$ of the feasible region $T_{N_1} \equiv F_{L_{N_1}}$ of this problem. The problem is then solved by using Phase II of the standard finite-dimensional Simplex method that starts at extreme point $x(N_1)$ and stops at an optimal extreme point $x^*(N_1)$. The first stage ends here. Problem $P(L_{N_2})$, which has the first N_2 constraints and the first L_{N_2} variables is then solved in the second stage. This stage begins by implementing a Phase I procedure to find an extreme point $x(N_2)$ of the feasible region $T_{N_2} \equiv F_{L_{N_2}}$ whose first L_{N_1} components are the same as those of $x^*(N_1)$. Note that existence of such an extreme point is guaranteed by Lemma 3.2, and the Phase I procedure begins by eliminating the first L_{N_1} variables from the first N_2 constraints by substituting their values from $x^*(N_1)$. This eliminates the first N_1 constraints since variables $L_{N_1}+1$ to L_{N_2} do not appear there and leaves variables $L_{N_1}+1$ to L_{N_2} in the next L_{N_2} constraints. Problem $P(L_{N_2})$ in variables $x_1 \ldots, x_{L_{N_2}}$ is then solved by Phase II of the Simplex method that starts at $x(N_2)$ and ends at an optimal extreme point $x^*(N_2)$. The second stage ends here. This is repeated for all stages $n \geq 3$. The formal algorithmic procedure is stated below.

Algorithm 3.8. The Shadow Simplex Method
Start with $n = 1$, $L_{N_0} = 0$.

1. *Implement Stage n to solve problem $P(L_{N_n})$ as follows.*

   (a) *Eliminate variables $x_1, \ldots, x_{L_{N_{n-1}}}$ from constraints $1, 2, \ldots, N_n$.*

   (b) *Use Phase I of the standard finite-dimensional Simplex method to find an extreme point $x(N_n)$ of $P(L_{N_n})$ whose first $L_{N_{n-1}}$ components are the same as that of $x^*(N_{n-1})$. (Note that steps (a) and (b) reduce to the usual Phase I procedure when $n = 1$).*

(c) *Starting at $x(N_n)$, use Phase II of the standard finite-dimensional Simplex method to find an extreme point optimal solution $x^*(N_n)$ of $P(L_{N_n})$.*

2. *Set $n = n + 1$ and goto step 1.*

Several remarks are now in order. Every finite-dimensional extreme point visited by the Shadow Simplex method can be appended to yield an extreme point of the feasible region F of problem (P) by Lemma 3.3 (also see Remark 3.4). Thus even though it is unnecessary (and impossible) to find these continuations, one may view the above procedure as an algorithm that iterates over a countably infinite number of extreme points of F. The reason for implementing a Phase I-Phase II procedure as above is that we want $x^*(N_{n-1})$ and $x(N_n)$ to lift to the same extreme point of F. Thus it is necessary that their first N_{n-1} components be identical. The Shadow Simplex method may be viewed as a pivot selection rule at extreme points of F that guarantees convergence in value. This is a crucial observation in view of the example presented in Section 2, where naive pivot selections fail to ensure value convergence. Perhaps more importantly, in the non-degenerate case of a unique optimal solution say x^* to (P) (this must be an extreme point solution in view of Proposition 2.7), Shadow Simplex implicitly constructs a sequence of extreme points of F that converges to x^*. Observe that we successfully perform this challenging task by implementing finite computations and using finite information in every iteration. We discuss properties of the Shadow Simplex method when applied to non-stationary deterministic and stochastic dynamic programming problems.

4 Application to non-stationary infinite-horizon deterministic dynamic programming

We briefly describe a non-stationary infinite-horizon discounted deterministic dynamic programming problem whose typical "dynamic programming network" looks as illustrated in an example in Figure 3 (a). Consider a dynamic system that is observed by a decision maker at the beginning of each period $n = 1, 2, \ldots$ to be in some time-indexed state $s_n \in S_n$ where S_n is a finite set with cardinality uniformly bounded over all n. The initial state of the system is known to be s_1. The decision maker chooses an action a_n from a finite set $A(s_n)$ with cardinality uniformly bounded over all states s_n and incurs a non-negative cost $c_n(s_n, a_n) \le c < \infty$. The choice of action a_n causes the system to make a transition to some state $s_{n+1} \in S_{n+1}$. For brevity, we use state transition functions g_n and the equation $s_{n+1} = g_n(s_n, a_n)$ as a surrogate for the earlier longer statement. This procedure continues ad infinitum. Note that circles in Figure 3(a) represent states whereas arrows represent actions. Both these are numbered to facilitate discussion later. The decision maker's goal is to compute a feasible action in every possible state beginning every period so as to minimize total discounted infinite-horizon cost where the discount factor is $0 < \alpha < 1$. We consider the case where any two distinct actions feasible in a state transform the system into two distinct states in the next period. This is without loss of generality because otherwise the action with the higher immediate cost can be eliminated from consideration.

The above dynamic program can be formulated as a CILP as follows. For any $s_n \in S_n$, let $X(s_n)$ denote the set of state-action pairs (s_{n-1}, a_{n-1}) such that the system is transformed to state s_n if we choose action $a_{n-1} \in A(s_{n-1})$ in state s_{n-1} in period $n - 1$, i.e., $g_{n-1}(s_{n-1}, a_{n-1}) = s_n$. For example, in Figure 3, $X(6) = \{(2, 5), (3, 6), (4, 7)\}$. Note that for every state s_{n-1} there is at most one action a_{n-1} with this property. Let $\{\beta(s_n)\}$ be any sequence of *positive* numbers indexed by states $s_n \in S_n$ for all periods n such that $\sum_{n=1}^{\infty} \sum_{s_n \in S_n} \beta(s_n) < \infty$. Then, the non-stationary infinite-horizon dynamic programming problem is equivalent to solving the following linear program in decision variables $z(s_n, a_n)$ (see [17, 32, 41, 45]):

$$(DP) \quad \min \sum_{n=1}^{\infty} \sum_{s_n \in S_n} \sum_{a_n \in A(s_n)} c_n(s_n, a_n) z(s_n, a_n)$$

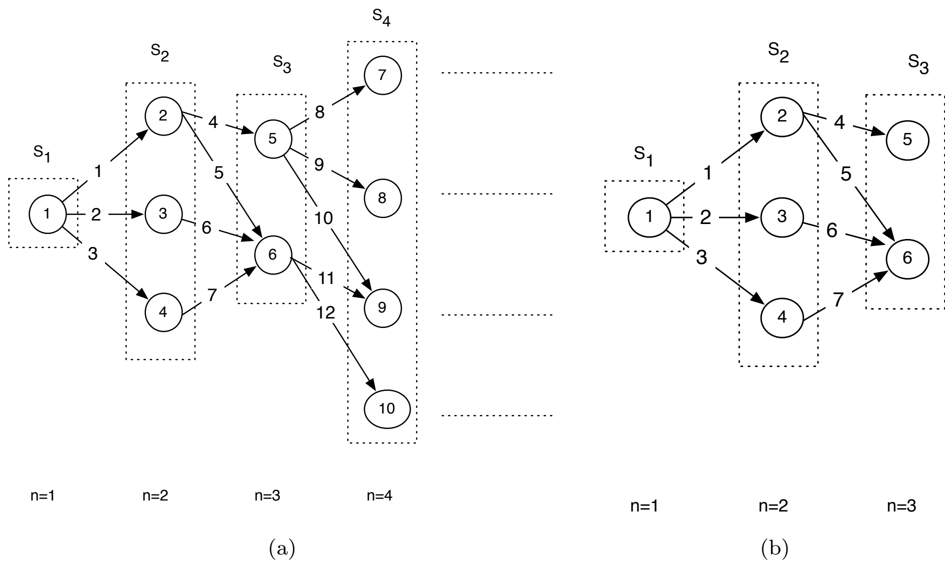

Figure 3: (a) An illustration of a non-stationary infinite-horizon discounted deterministic dynamic programming network. (b) A two-horizon truncation of the network in (a). States numbered 5 and 6 are "terminal".

$$\begin{aligned} \sum_{a_n \in A(s_n)} z(s_n, a_n) - \alpha \sum_{(s_{n-1}, a_{n-1}) \in X(s_n)} z(s_{n-1}, a_{n-1}) &= \beta(s_n), \quad s_n \in S_n, \ \forall n, \\ z(s_n, a_n) &\geq 0, \ s_n \in S_n, \ a_n \in A(s_n), \ \forall n. \end{aligned}$$

It is clear that problem (DP) above is a special case of (P). We show in Lemmas 4.1 and 4.2 that it also satisfies the required assumptions.

Lemma 4.1. *Problem* (DP) *satisfies Assumptions 2.1, 2.2, 2.3, and 2.4.*

Consider any n-horizon truncation of the infinite-horizon dynamic program described above. Refer to Figure 3 (b). Let $(s_1, a_1), (s_2, a_2), \ldots, (s_n, a_n)$ be any sequence of feasible state-action pairs in this truncation. That is, for $i = 1, \ldots, n$, $s_i \in S_i$, $a_i \in A(s_i)$ and $s_{i+1} = g_i(s_i, a_i)$ for $i = 1, \ldots, n-1$. Suppose $s_{n+1} = g_n(s_n, a_n)$. Then we call s_{n+1} a "terminal state" of the n-horizon truncation. We assume that there exists an infinite sequence of state-action pairs $\{(s_t, a_t)\}_{n+1}^{\infty}$ starting at the terminal state s_{n+1} and feasible to the original infinite-horizon problem, i.e., $s_t \in S_t$, $a_t \in A(s_t)$ and $s_{t+1} = g_t(s_t, a_t)$ for $t = n+1, \ldots$. We call this "extendability of finite-horizon strategies".

Again note that if the original formulation of the dynamic program does not have this property, it is often possible to enforce it by adding valid inequalities to the set of feasible actions especially in planning problems where the state corresponds to some type of inventory as discussed in problem $(PROD)$. More generally, it is possible to design a "big-M" approach for dynamic programs where a terminal state with no feasible continuation can be appended with a sequence of artificial state-action pairs each with "big-M" cost making the choice of these artificial actions and hence the terminal state they emerge from unattractive in the infinite-horizon problem. This approach, motivated by the "big-M" method for finding an initial basic feasible solution in finite-dimensional linear programs [11], forces extendability without loss of optimality in the infinite-horizon problem (see Section A.7 in Appendix A for more details).

The N-horizon truncation of (DP) is then given by the following finite-dimensional linear program:

$$DP(N) \quad \min \sum_{n=1}^{N} \sum_{s_n \in S_n} \sum_{a_n \in A(s_n)} c_n(s_n, a_n) z(s_n, a_n)$$

$$\sum_{a_n \in A(s_n)} z(s_n, a_n) - \alpha \sum_{(s_{n-1}, a_{n-1}) \in X(s_n)} z(s_{n-1}, a_{n-1}) = \beta(s_n), \;\; s_n \in S_n, \;\; n = 1, \ldots, N,$$

$$z(s_n, a_n) \geq 0, \; s_n \in S_n, \; a_n \in A(s_n), \;\; n = 1, \ldots, N.$$

The following lemma shows that our extendability of finite-horizon strategies assumption in the original dynamic program carries over to truncations $DP(N)$ of the CILP (DP).

Lemma 4.2. *The N-horizon truncations $DP(N)$ are extendable for all horizons N and hence Assumption 3.7 holds for (DP).*

Since all required assumptions are satisfied, Value Convergence Theorem 2.9 implies that optimal values $V(DP(N))$ converge to the optimal value $V(DP)$ as $N \to \infty$ and our Shadow Simplex method can be employed to solve (DP). Interestingly, the Shadow Simplex method for (DP) in fact reduces to a true infinite-dimensional Simplex method for reasons discussed below.

4.1 A Simplex method for problem (DP)

Our argument requires precise characterizations of extreme points of $DP(N)$ and (DP) as well as pivot operations in these problems.

Pivots at extreme points of problem DP(N)

First note that the finite-dimensional linear program $DP(N)$ is a special case of the standard linear programming formulation for finite-state discounted stochastic dynamic programs (see page 224 of [41]). It is well-known that extreme points of the feasible region of this problem are precisely feasible solutions z having the property that for every state $s_n \in S_n$ for $n = 1, \ldots, N$, $z(s_n, a_n) > 0$ for exactly one $a_n \in A(s_n)$ (Proposition 6.9.3 page 227 of [41]). This is viewed as a one-to-one correspondence between extreme points and deterministic policies [41]. Thus a pivot operation at extreme point z^1 of $DP(N)$ during execution of the Shadow Simplex method is of the form described in the next paragraph. It often helps to "visualize" this pivot operation on the relevant portion of a dynamic programming network as in Figure 4 where actions with positive z values are shown in solid arrows.

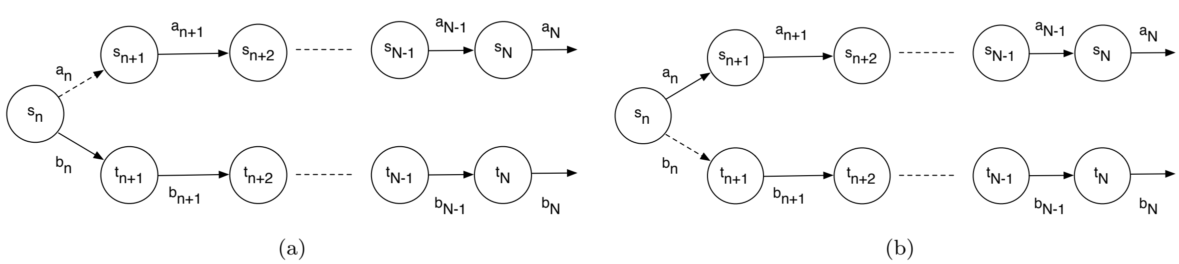

Figure 4: A $DP(N)$ pivot as described in the text. (a) Portion of the dynamic programming network at the original extreme point z^1. (b) The same portion of the dynamic programming network at the new extreme point z^2 after a pivot operation.

Choose any state $s_n \in S_n$, for some $n = 1, \ldots, N$, and an action $a_n \in A(s_n)$ such that $z^1(s_n, a_n) = 0$. Let $s_{n+1} = g_n(s_n, a_n)$. There is exactly one finite sequence of state-action pairs $\{(s_r, a_r)\}$, $r = n+1, \ldots, N$, such that $s_r \in S_r$, $a_r \in A(s_r)$, $s_{r+1} = g_r(s_r, a_r)$ and $z^1(s_r, a_r) > 0$. Select *the* action $b_n \in A(s_n)$ for which $z^1(s_n, b_n) > 0$ and let $t_{n+1} = g_n(s_n, b_n)$. There is exactly one finite sequence of state-action pairs $\{(t_r, b_r)\}$, $r = n+1, \ldots, N$, such that $t_r \in S_r$, $b_r \in A(t_r)$, $t_{r+1} = g_r(t_r, b_r)$ and $z^1(t_r, b_r) > 0$. Note that $z^1(t_r, b_r) \geq \alpha^{(r-n)} z^1(s_n, b_n)$ for $r = n+1, \ldots, N$. Set $z^2(s_n, a_n) = z^1(s_n, b_n)$ and increase $z^1(s_r, b_r)$ by an amount $\alpha^{(r-n)} z^1(s_n, b_n)$ for $r = n+1, \ldots, N$ to form the corresponding components of z^2. Set $z^2(s_n, b_n)$ to zero and reduce $z^1(t_r, b_r)$ by an amount $\alpha^{(r-n)} z^1(s_n, b_n)$ for $r = n+1, \ldots, N$ to form the corresponding components of z^2. The new solution z^2 thus formed is non-negative, satisfies the equality constraints and has the property that for each state exactly one z^2 value is positive. Thus it is a new extreme point of

$DP(N)$ and the pivot operation is complete.

Pivots at extreme points of problem (DP)

We first show that the one-to-one correspondence between extreme points and deterministic policies in finite-dimensional linear programs $DP(N)$ carries over to our CILP (DP). This essentially means that extreme points of (DP) are equivalent to its basic feasible solutions. The first, straightforward direction (Lemma 4.3) of this equivalence states that a basic feasible solution is an extreme point and its extensions hold for CILPs in general.

Lemma 4.3. *Suppose z is a feasible solution to (DP) having the property that for every state s_n, there is exactly one action $a_n \in A(s_n)$ for which $z(s_n, a_n) > 0$. Then z is an extreme point of (DP).*

We remark that the converse below is of independent interest since it often fails (and is considered one of the major pathologies) in CILPs where variable values are *not* bounded away from zero [5, 22, 42] as in (DP). Its proof outlined in Appendix A considers two cases. The first case is roughly the counterpart of lack of finite cycles in extreme points of infinite network flow problems [22, 42] and is straighforward. The second case relies on the special structure of (DP), i.e., its similarity to time-staged acyclic network flow problems (see Figure 3 for example) and that quantities $\beta(s_n)$ are positive.

Lemma 4.4. *Suppose z is an extreme point of (DP). Then z has the property that for every state s_n, there is exactly one action $a_n \in A(s_n)$ for which $z(s_n, a_n) > 0$.*

Lemmas 4.3 and 4.4 imply that a pivot operation in (DP) is conceptually identical to the pivot operation in $DP(N)$ described above, the only difference being that in (DP) it involves changing values of a countably infinite number of variables. As a result, a pivot in $DP(N)$ is a "projection" of a pivot in (DP) and a pivot in (DP) is an "extension" of a pivot in $DP(N)$. Thus the Shadow Simplex method reduces to a true Simplex method in the conventional sense for problem (DP) — moving from one extreme point to an adjacent extreme point in every iteration. We now discuss three concrete examples where this theory applies.

Production planning: Consider a (non-linear) generalization of problem $(PROD)$ where the production cost function in period n is denoted $c_n(\cdot)$ and the inventory holding cost function is denoted $h_n(\cdot)$. The goal is to find an infinite-horizon production schedule $x = (x_1, x_2, \ldots)$ that satisfies demand $(D_1, D_2, \ldots)$ subject to production and inventory warehouse capacities at minimum discounted infinite-horizon total cost $\sum_{n=1}^{\infty} \alpha^{n-1}(c_n(x_n) + h_n(y_n))$ where the inventory schedule $y = (y_1, y_2, \ldots)$ is defined by the material balance equations stated in $(PROD)$. It is easy to see that this problem is a special case of the general non-stationary infinite-horizon deterministic dynamic programming problem where the state corresponds to the inventory on hand beginning a period, and the actions correspond to feasible production quantities in that period. *Under the assumption that capacities dominate demand, i.e., $P_n \geq D_n$ for every n, finite-horizon truncations of our dynamic program are extendable. When the data do not satisfy such dominance, extendability can be forced by adding valid inequalities as discussed in $(PROD)$.*

Equipment replacement under technological change: This is the problem of deciding an equipment replacement strategy so as to minimize total purchase and maintenance costs over an infinite-horizon. Specifically, we initially have an s_1 period old piece of equipment. At the beginning of each period, we have two options - either to sell the equipment on hand and spend the money received to buy a brand new piece or to carry the equipment through one more period incurring maintenance costs. The life of an equipment is L periods. Note we have assumed the life is independent of the period of purchase for simplicity. The cost of purchasing a brand new equipment at the beginning of period n is p_n, the maintenance cost function during period n is denoted $m_n(\cdot)$ where the argument corresponds to the age of the equipment beginning period n, and finally, the salvage value function at the beginning of period n is denoted $v_n(\cdot)$ where the argument corresponds to the age of the equipment at the beginning of period n. The goal is to decide, at the beginning of each period, whether to retire the current equipment and buy a new one or to maintain the equipment through that period so as to minimize discounted infinite-horizon total cost (see [9] for example). Again, note that this is a special case of the general non-stationary infinite-horizon deterministic dynamic programming problem where the state corresponds to the age of the equipment on hand at the beginning of a period and actions correspond to buying a new equipment or keeping the current one. *Any finite-horizon equipment purchase/retire strategy can be extended to an infinite-horizon feasible sequence say for example by retiring the current equipment and buying a new one in every future period.*

Optimal exploitation of a renewable natural resource: Suppose we are initially endowed with $s_1 \in S \equiv \{0, 1, \ldots, \eta\}$ units of a natural resource, for example fish in an ocean, where η is a positive integer. If the resource remaining at the beginning of period n is s_n units, and we consume a_n of these units, then we receive a reward of $r_n(a_n)$. Moreover, the remaining $s_n - a_n$ units renew according to a function $g_n(s_n - c_n)$ during period n where the range of function $g_n(\cdot)$ is $\{0, 1, \ldots, \eta\}$ and $g_n(0) = 0$ for all n. Our goal is to find a consumption plan $a_1, a_2, \ldots$ that maximizes discounted infinite-horizon reward (see [20] for example). Again this is a special case of non-stationary infinite-horizon dynamic programming where the state corresponds to the units of resource available beginning a period, and actions correspond to the number of units consumed in that period. *Observe that any finite-horizon consumption plan can be extended to an infinite-horizon feasible plan say by consuming zero resource in all future periods.*

5 Application to non-stationary infinite-horizon Markov decision problems

Non-stationary infinite-horizon Markov decision problems also termed non-stationary infinite-horizon stochastic dynamic programs [12, 17, 32, 33, 41, 45, 53] are an important generalization of the above deterministic dynamic programming problem where the state transitions are stochastic. Given that an action $a_n \in A(s_n)$ was chosen in state s_n, the system makes a transition to state $s_{n+1} \in S_{n+1}$ with probability $p_n(s_{n+1}|s_n, a_n)$, incurring non-negative cost $c_n(s_n, a_n; s_{n+1}) \leq c < \infty$. The term Markovian policy in this context denotes a rule that dictates our choice of action in every possible state (irrespective of the earlier states visited or actions taken) over an infinite-horizon. The goal then is to find a Markovian policy that minimizes total infinite-horizon discounted expected cost when the discount factor is $0 < \alpha < 1$. Let $Y(s_n, a_n) \subseteq S_{n+1}$ denote the set of states $s_{n+1} \in S_{n+1}$ such that $p_n(s_{n+1}|s_n, a_n) > 0$. Let $c_n(s_n, a_n)$ denote the expected cost incurred on choosing actions $a_n \in A(s_n)$ in state $s_n \in S_n$. That is, $c_n(s_n, a_n) = \sum_{s_{n+1} \in Y(s_n,a_n)} p_n(s_{n+1}|s_n, a_n) c_n(s_n, a_n; s_{n+1})$. Finally, for any state $s_n \in S_n$, let $X(s_n)$ denote the set of states $s_{n-1} \in S_{n-1}$ such that there exists an action $a_{n-1} \in A(s_{n-1})$ with $p_{n-1}(s_n|s_{n-1}, a_{n-1}) > 0$. For each s_{n-1} in $X(s_n)$, we use $\mathcal{X}(s_{n-1}, s_n)$ to denote the set of actions $a_{n-1} \in A(s_{n-1})$ with $p_{n-1}(s_n|s_{n-1}, a_{n-1}) > 0$. Let $\{\beta(s_n)\}$ be a sequence of positive numbers indexed by states $s_n \in S_n$ for all periods n such that $\sum_{n=1}^{\infty} \sum_{s_n \in S_n} \beta(s_n) < \infty$. Then the non-stationary infinite-horizon Markov decision problem is equivalent to solving the following linear program in variables $z(s_n, a_n)$ (see [41, 45]) :

$$(MDP) \min \sum_{n=1}^{\infty} \sum_{s_n \in S_n} \sum_{a_n \in A(s_n)} c_n(s_n, a_n) z(s_n, a_n)$$

$$\sum_{a_n \in A(s_n)} z(s_n, a_n) - \alpha \sum_{s_{n-1} \in X(s_n)} \sum_{a_{n-1} \in \mathcal{X}(s_{n-1}, s_n)} p_{n-1}(s_n|s_{n-1}, a_{n-1}) z(s_{n-1}, a_{n-1}) = \beta(s_n),$$

$$\forall s_n \in S_n, n = 1, 2, \ldots$$

$$z(s_n, a_n) \geq 0, \ \forall s_n \in S_n, a_n \in A(s_n), \ n = 1, 2, \ldots$$

Problem (MDP) is a special case of (P). Lemmas 5.1 and 5.2 confirm that it satisfies the required assumptions.

Lemma 5.1. *Problem (MDP) satisfies Assumptions 2.1, 2.2, 2.3 and 2.4.*

Again we assume extendability of finite-horizon strategies and consider the following N-horizon truncation of (MDP) as in the (DP) case:

$$MDP(N) \min \sum_{n=1}^{N} \sum_{s_n \in S_n} \sum_{a_n \in A(s_n)} c_n(s_n, a_n) z(s_n, a_n)$$

$$\sum_{a_n \in A(s_n)} z(s_n, a_n) - \alpha \sum_{s_{n-1} \in X(s_n)} \sum_{a_{n-1} \in \mathcal{X}(s_{n-1}, s_n)} p_{n-1}(s_n | s_{n-1}, a_{n-1}) z(s_{n-1}, a_{n-1}) = \beta(s_n),$$
$$\forall s_n \in S_n, n = 1, \ldots, N$$

$$z(s_n, a_n) \geq 0,\ \forall s_n \in S_n, a_n \in A(s_n),\ n = 1, \ldots, N.$$

Lemma 5.2. *The N-horizon truncations $MDP(N)$ are extendable for all horizons N and hence Assumption 3.7 holds for (MDP).*

Value Convergence Theorem 2.9 then implies that $V(MDP(N)) \to V(MDP)$ as $N \to \infty$ and we can apply the Shadow Simplex method to solve (MDP). In the next section, we present a brief outline of our argument as to why Shadow Simplex also reduces to a true infinite-dimensional Simplex method for (MDP). The discussion is similar to the one for (DP).

5.1 A Simplex method for problem (MDP)

Again note that the finite-dimensional linear program $MDP(N)$ is a special case of the standard linear programming formulation for finite-state discounted stochastic dynamic programs (see page 224 of [41]). It is well-known [41] that a feasible solution z for this problem is an extreme point of its feasible region if and only if for every state $s_n \in S_n$, $z(s_n, a_n) > 0$ for exactly one action $a_n \in A(s_n)$ and $z(s_n, b_n) = 0$ for all other actions $b_n \in A(s_n)$. Consequently, a pivot operation is characterized as follows: at an extreme point solution z^1, select a state $s_n \in S_n$ and an action $a_n \in A(s_n)$ such that $z^1(s_n, a_n) = 0$. Let action $b_n \in A(s_n)$ be *the* action in $A(s_n)$ for which $z^1(s_n, b_n) > 0$. Then similar to the (DP) case decrease $z^1(s_n, b_n)$ to zero and increase $z^1(s_n, a_n)$ to a positive value adjusting values of other variables appropriately to construct a new extreme point z^2. By liftability of extreme points, z^1 and z^2 are both projections of extreme points of the feasible region of (MDP). Moreover, Lemmas 4.3 and 4.4 can be extended to the (MDP) case so that a feasible solution z to (MDP) is an extreme point if and only if for every state $s_n \in S_n$, $z(s_n, a_n) > 0$ for exactly one action $a_n \in A(s_n)$ and $z(s_n, b_n) = 0$ for all other actions $b_n \in A(s_n)$. Thus the extreme points of (MDP) whose projections equal z^1 and z^2 have this property. Consequently, a pivot in $MDP(N)$ is a "projection" of a pivot in (MDP) and a pivot in (MDP) is an "extension" of a pivot in $MDP(N)$. In other words, Shadow Simplex reduces to a true infinite-dimensional Simplex method for (MDP).

6 Conclusions

We showed that the Shadow Simplex algorithm performs finite computations on finite information in every iteration and implicitly constructs a sequence of infinite-dimensional extreme points that converges in value to the optimal value of the CILP at hand. This result is perhaps of independent theoretical interest since a CILP may in general have an uncountable number of extreme points. When the CILP has a unique extreme point optimal solution, the aforementioned sequence of extreme points converges to the optimal solution. In general, two consecutive extreme points in the sequence of extreme points constructed by our algorithm need not be adjacent. However, for a class of CILPs that corresponds to dynamic programs with time-indexed states, our algorithm moves through adjacent extreme points of the infinite-dimensional feasible region. This result may also be of independent interest since the feasible region of a CILP is not in general polyhedral.

Acknowledgements

Part of this work was supported by the National Science Foundation under grant DMI-0322114. The first author appreciates summer support from the University of Washington. We thank the anonymous Associate Editor and two anonymous referees whose comments improved the presentation of this manuscript as compared to an earlier version.

References

[1] Aliprantis, C. D., and Border, K. C., Infinite dimensional analysis: a hitchhiker's guide, Springer-Verlag, Berlin, (1994).

[2] Anderson, E. J., Goberna, M. A., and Lopez, M. A., Locally polyhedral linear inequality systems, *Linear Algebra and Applications*, 270, 231-253 (1998).

[3] Anderson, E. J., Goberna, M. A., and Lopez, M. A., Simplex-like trajectories on quasi-polyhedral sets, *Mathematics of Operations Research*, 26, 1, 147-162, (2001).

[4] Anderson, E. J., and Lewis, A. S., An extension of Simplex algorithm for semi-infinite linear programming, *Mathematical Programming*, 44, 247 - 269, (1988).

[5] Anderson, E. J., and Nash, P., Linear programming in infinite-dimensional spaces: theory and applications, John Wiley and Sons, Chichester, Great Britain, (1987).

[6] Anderson, E. J., and Nash, P., A continuous-time network Simplex algorithm, *Networks*, 19 (4), 395–425, (1989).

[7] Anderson, E. J., Nash, P., and Philpott, A. B., A class of continuous network flow problems, *Mathematics of Operations Research*, 7 (4), 501–514, (1982).

[8] Apostol, T., M., Mathematical Analysis, Addison Wesley, second edition, (1974).

[9] Bean, J. C., Lohmann, J. R., and Smith, R. L., A Dynamic Infinite Horizon Replacement Economy Decision Model, The Engineering Economist, 30, 99-120, (1985a).

[10] Bean, J. C., and Smith, R. L., Optimal capacity expansion over an infinite horizon, *Management Science*, 31, 1523-1532, (1985).

[11] Bertsimas, D., and Tsitsiklis, J. N., Introduction to linear optimization, Athena Scientific, Belmont, Massachusetts, (1997).

[12] Cheevaprawatdomrong, T., Schochetman, I. E., Smith, R. L., and Garcia, A., Solution and forecast horizons for infinite-horizon non-homogeneous Markov Decision Processes, *Mathematics of Operations Research*, 32 (1), 51-72, (2007).

[13] Cook, W. D., Field, C. A., Kirby, M. J. L., Infinite linear programs in games with partial information, *Operations Research*, 23, 5, 996-1010, (1975).

[14] Cross, W. P., Romeijn, H. E., Smith, R. L., Approximating extreme points of infinite dimensional convex sets, *Mathematics of Operations Research*, 23(1), (1998).

[15] Demaine, E. D., Fekete, S. P., and Gal, S., Online searching with turn cost, *Theoretical Computer Science*, 361, 2, 342-355, (2006).

[16] Denardo, E., Dynamic Programming : Models and Applications., Prentice Hall, Englewood Cliffs, NJ, (1982).

[17] Feinberg, E., Handbook of Markov Decision Processes: methods and algorithms (with A. Shwartz, editors), Kluwer, Boston, (2002).

[18] Feinberg, E. and Shwartz, A., Constrained discounted dynamic programming, *Mathematics of Operations Research*, 21, 922-945, (1996).

[19] Freidenfelds, J., Capacity extension: simple models and applications, North Holland, Amsterdam, (1981).

[20] Garcia, A., and Smith, R. L., Solving nonstationary infinite horizon dynamic optimization problems, *Journal of Mathematical Analysis and Applications*, 244, 304-317, (2000).

[21] Ghate, A. V., Markov Chains, Game Theory, and Infinite Programming: Three Paradigms for Optimization of Complex Systems, Ph. D. Thesis, Industrial and Operations Engineering, The University of Michigan, (2006).

[22] Ghate, A. V., and Smith, R. L., Characterizing extreme points as basic feasible solutions in infinite linear programs *Operations Research Letters*, 37(1), 7-10, (2009).

[23] Ghate, A. V., and Smith, R. L., Duality theory for countably infinite linear programs, working paper, January 2009.

[24] Ghate, A. V., and Smith, R. L., A short note on extendability in truncations of infinite-horizon production planning problems, working paper, January 2009.

[25] Goberna, M. A., and Lopez, M. A., Linear semi-infinite optimization, Wiley, New York, (1998).

[26] Goberna, M. A., and Lopez, M. A., Linear semi-infinite programming theory: an updated survey, *European Journal of Operations Research*, 143, 390-405, (2002).

[27] Grinold, R. C., Infinite horizon programs, *Management Science*, 18 (3), 157–170, (1971).

[28] Grinold, R. C., Finite horizon approximations of infinite horizon linear programs, *Mathematical Programming*, 12, 1–17, (1977).

[29] Grinold, R. C., Convex infinite horizon programs, *Mathematical Programming*, 25 (1), 64–82, (1983).

[30] Grinold, R. C., Infinite horizon stochastic programs, *SIAM Journal on Control and Optimization*, 24 (6), 1246-1260, (1986).

[31] Hernandez-Lerma, O., and Lasserre, J. B., Approximation schemes for infinite linear programs, *SIAM Journal of Optimization*, 8 (4), 973-988, (1998).

[32] Hernandez-Lerma, O., and Lasserre, J. B., The linear programming approach, in Feinberg, E. A., and Shwartz, A., editors, Handbook of Markov Decision Processes, Kluwer, (2002).

[33] Hopp, W. J., Bean, J. C., and Smith, R. L., A new optimality criterion for non-homogeneous Markov Decision Processes, *Operations Research*, 35, 875-883, (1987).

[34] Huang, K., Multi-stage stochastic programming models in production planning, Ph. D. Thesis, School of Industrial and Systems Engineering, Georgia Institute of Technology, (2005).

[35] Klabjan, D., and Adelman, D., Existence of optimal policies for semi-Markov decision processes using duality for infinite linear programming, *Mathematics of Operations Research*, 30 (1), 28-50, (2005).

[36] Klabjan, D., and Adelman, D., A convergent infinite dimensional linear programming algorithm for deterministic semi-Markov decision processes on borel spaces, *Mathematics of Operations Research*, 32 (3), 528–550, (2007).

[37] Luenberger, D. G., Optimization by vector space methods, John Wiley and Sons, New York, USA, (1969).

[38] Philpott, A. B., and Craddock, M., An adaptive discretization algorithm for a class of continuous network programs, *Networks*, 26, 1–11, (1995).

[39] Pullan, M. C., An algorithm for a class of continuous linear programs, *SIAM J. Control and Optimization*, 31, 1558–1577, (1993).

[40] Pullan, M. C., Convergence of a general class of algorithms for separated continuous linear programs, *SIAM Journal on Optimization*, 10, 722-731, (2000).

[41] Puterman, M. L., Markov decision processes : Discrete stochastic dynamic programming, John Wiley and Sons, New York, (1994).

[42] Romeijn, H. E., Sharma, D., and Smith, R. L., Extreme point solutions for infinite network flow problems, *Networks*, 48 (4), 209–222, (2006).

[43] Romeijn, H. E., and Smith, R. L., Shadow prices in infinite-dimensional linear programming, *Mathematics of Operations Research*, 23 (1), 239-256, (1998).

[44] Romeijn, H. E., Smith, R. L., and Bean, J. C., Duality in infinite dimensional linear programming, *Mathematical Programming*, 53, 79-97, (1992).

[45] Ross, S. M., Introduction to stochastic dynamic programming, Academic Press, New York, USA, (1983).

[46] Schochetman, I. E., and Smith, R. L., Infinite Horizon Optimization, *Mathematics of Operations Research*, 14, 559-574, (1989).

[47] Sharkey, T. C., and Romeijn, H. E., A Simplex algorithm for minimum cost network flow problems in infinite networks, *Networks*, 52 (1), 14-31, (2008).

[48] Siegrista, S., A complementarity approach to multistage stochastic linear programs, Ph.D. Thesis, University of Zurich, (2005).

[49] Taylor, A. E., and Lay, D. C., Introduction to functional analysis, Robert E. Krieger Publishing Company, Malabar, Florida, USA, (1986).

[50] Veinott, A. F. Jr., Extreme points of Leontief substitution systems, *Linear Algebra and Applications*, 1, 181–194, (1968).

[51] Veinott, A. F. Jr., Minimum concave cost solution of Leontief substitution models of multi-facility inventory systems, *Operations Research*, 17 (2), 262-291, (1969).

[52] Weiss, G., A Simplex based algorithm to solve separated continuous linear programs, *Mathematical Programming*, 115 (1), 151–198, (2008).

[53] White, D. J., Decision roll and horizon roll in infinite horizon discounted Markov Decision processes, *Management Science*, 42 (1), 37–50, (1996).

A Proofs of Technical Results

Proofs of results in the text are presented here.

A.1 Proof of Lemma 2.6

To see that the first condition is sufficient for Assumption 2.4 to hold, note that for any $x \geq 0$, $|c_j x_j| = |c_j||x_j| = |c_j|x_j \leq |c_j|u_j$ and $\sum_{j=1}^{\infty} |c_j|u_j \leq ||u||_\infty \sum_{j=1}^{\infty} |c_j| < \infty$ since $u \in l_\infty$ and $c \in l_1$. Similarly, to see that the second condition is sufficient, observe that for $x \geq 0$, $|c_j x_j| = |c_j||x_j| = |c_j|x_j \leq |c_j|u_j$ and $\sum_{j=1}^{\infty} |c_j|u_j \leq ||c||_\infty \sum_{j=1}^{\infty} |u_j| < \infty$ since $c \in l_\infty$ and $u \in l_1$.

A.2 Proof of Proposition 2.7

The proof requires three preliminary results that we now state and prove.

Lemma A.1. *The feasible region F of problem (P) is closed.*

Proof. For row i of matrix A, let $J(i)$ denote the finite (by Assumption 2.2) support set $\{j : a_{ij} \neq 0\}$. Consider sets $X^i = \{x \in R^\infty : \sum_{j \in J(i)} a_{ij}x_j = b_i\}$ for $i = 1, 2, \ldots$. Notice that $F = \bigcap_{i=1}^{\infty} X^i \bigcap \{x \in R^\infty : x \geq 0\}$. The set $\{x \in R^\infty : x \geq 0\}$ is closed. We show that sets X^i are closed for all i. Then since arbitrary intersections of closed sets are closed, F must be closed. Let $\{x^i(n)\}_{n=1}^{\infty}$ be a convergent sequence of points in X^i with limit $\bar{x}^i \in R^\infty$. For any integer n we have $\sum_{j \in J(i)} a_{ij}x^i_j(n) = b_i$. Taking limits we obtain $\lim_{n\to\infty} \sum_{j \in J(i)} a_{ij}x^i_j(n) = b_i$. Hence $\sum_{j \in J(i)} a_{ij} \left(\lim_{n\to\infty} x^i_j(n)\right) = b_i$ since $J(i)$ is finite. Thus $\sum_{j \in J(i)} a_{ij}\bar{x}^i_j = b_i$. Therefore $\bar{x}^i \in X^i$ implying X^i is closed. □

Corollary A.2. *The feasible region F of (P) is compact.*

Proof. Let $0 \leq u \in R^\infty$ be as in Assumption 2.3. Consider the set $X = \{x \in R^\infty : 0 \leq x_j \leq u_j \ \ \forall j\}$. X is compact by Tychonoff Product Theorem (Theorem 2.61 page 52 of [1]). F is closed by Lemma A.1. Assumption 2.3 implies that $F \subseteq X$. Therefore F is compact. □

Lemma A.3. *The objective function of problem (P) is continuous over its feasible region F.*

Proof. Let $\{x(n)\}_{n=1}^{\infty}$ be a convergent sequence of points in F with limit $\bar{x} \in F$. We need to show that the sequence of objective function values $\sum_{j=1}^{\infty} c_j x_j(n)$ converges to $\sum_{j=1}^{\infty} c_j \bar{x}_j$ as $n \to \infty$. Fix any $\epsilon > 0$. Let $0 \leq u \in R^\infty$ be as in Assumption 2.4. Since the series $\sum_{i=1}^{\infty} |c_i|u_i$ of non-negative summands converges by Assumption 2.4, there exists an integer K such that the tail $\sum_{i=k+1}^{\infty} |c_i|u_i < \epsilon/2$ for all $k \geq K$. Fix any such k and note that for any integer n,

$$\left|\sum_{j=1}^{\infty} c_j x_j(n) - \sum_{j=1}^{\infty} c_j \bar{x}_j\right| = \left|\sum_{j=1}^{\infty} c_j(x_j(n) - \bar{x}_j)\right| \leq \left|\sum_{j=1}^{k} c_j(x_j(n) - \bar{x}_j)\right| + \left|\sum_{j=k+1}^{\infty} c_j(x_j(n) - \bar{x}_j)\right|,$$

which is bounded above by

$$\sum_{j=1}^{k} |c_j||(x_j(n) - \bar{x}_j)| + \sum_{j=k+1}^{\infty} |c_j||(x_j(n) - \bar{x}_j)| \leq \sum_{j=1}^{k} |c_j||(x_j(n) - \bar{x}_j)| + \sum_{j=k+1}^{\infty} |c_j|u_j$$

because $0 \leq x_j(n) \leq u_j$ and $0 \leq \bar{x}_j \leq u_j$. The second term is strictly less than $\epsilon/2$. The first term is bounded above by $(\max_{1\leq j\leq k} |(x_j(n) - \bar{x}_j)|)(\sum_{j=1}^{k} |c_j|)$. The only interesting case is where $\sum_{j=1}^{k} |c_j| \neq 0$. Since the sequence $\{x(n)\}_{n=1}^{\infty}$ converges to $\bar{x}$ componentwise, there exists an integer N_k large enough such that $(\max_{1\leq j\leq k} |(x_j(n) - \bar{x}_j)|) < \frac{\epsilon}{\left(2\sum_{j=1}^{k}|c_j|\right)}$ for all $n \geq N_k$. Therefore, $|\sum_{j=1}^{\infty} c_j x_j(n) - \sum_{j=1}^{\infty} c_j \bar{x}_j| < \epsilon$ for all integers $n \geq N_k$. Hence the objective function is continuous over F. (Note that an identical proof can be reproduced to reach the stronger conclusion that the objective function is continuous over $X = \{x \in R^{\infty} : 0 \leq x_j \leq u_j \ \ \forall j\}$). □

Note that the feasible region F of (P) is nonempty (by Assumption 2.1) convex (since it is the intersection of convex sets X^i defined in Lemma A.1 with the convex set $\{x \in R^{\infty} : x \geq 0\}$) and the product topology on R^{∞} is locally convex Hausdorff (Lemma 5.74 page 206 of [1]). Existence of an extreme point optimal solution to (P) then follows directly from Corollary A.2, Lemma A.3 and a Corollary (Corollary 7.70 page 299 of [1]) of the Bauer Maximum Principle (Theorem 7.69 page 298 of [1]).

A.3 Proof of Theorem 2.9

We use Berge's Maximum Theorem (Theorem 17.31 page 570 of [1]). Let $\mathcal{I}$ denote the set of extended positive integers $\{1, 2, \ldots\} \bigcup \{\infty\}$. Also let $X = \{x \in R^{\infty} : 0 \leq x_j \leq u_j \ \ \forall j\}$ where sequence $\{u_j\}$ is as in Assumption 2.4. Recall that $F \subseteq X$ by Assumption 2.3 and also that $F_N \subseteq X$ by the definition of projections of F in Equations (1) and (2). Now define a correspondence Ψ from $\mathcal{I}$ into X as

$$\Psi(N) = F_N \ \text{ for } \ N = 1, 2, \ldots$$

$$\Psi(\infty) = F.$$

Sets F_N are non-empty for all N since F is non-empty by Assumption 2.1. Similarly, F_N is also compact for each N implying that correspondence Ψ has non-empty compact values. Moreover, it is continuous by Lemma 2.8. Now define a function $f : \mathcal{I} \times X \to R$ as

$$f(N, x) = \sum_{i=1}^{N} c_i x_i \ \text{ for } \ N = 1, 2, \ldots, \quad x \in X$$

$$f(\infty, x) = \sum_{i=1}^{\infty} c_i x_i \ \text{ for } \ x \in X.$$

Function f is continuous. To see this, fix $\epsilon > 0$ and suppose $x^k \to x$ in X and $N_k \to \infty$ as $k \to \infty$.

$$\begin{aligned} |f(\infty, x) - f(N_k, x^k)| &= \left|\sum_{i=1}^{\infty} c_i x_i - \sum_{i=1}^{N_k} c_i x_i^k\right| \leq \left|\sum_{i=1}^{\infty} c_i x_i - \sum_{i=1}^{\infty} c_i x_i^k\right| + \left|\sum_{i=1}^{\infty} c_i x_i^k - \sum_{i=1}^{N_k} c_i x_i^k\right| \\ &= \left|\sum_{i=1}^{\infty} c_i x_i - \sum_{i=1}^{\infty} c_i x_i^k\right| + \left|\sum_{i=N_k+1}^{\infty} c_i x_i^k\right| \leq \left|\sum_{i=1}^{\infty} c_i x_i - \sum_{i=1}^{\infty} c_i x_i^k\right| + \sum_{i=N_k+1}^{\infty} |c_i x_i^k| \\ &\leq \left|\sum_{i=1}^{\infty} c_i x_i - \sum_{i=1}^{\infty} c_i x_i^k\right| + \sum_{i=N_k+1}^{\infty} |c_i| u_i \ \text{ because } x^k \in X. \end{aligned}$$

Recall from the proof of Lemma A.3 that the objective function is continuous over X. Hence $\sum_{i=1}^{\infty} c_i x_i^k$ converges to $\sum_{i=1}^{\infty} c_i x_i$ because $x^k \to x$ as $k \to \infty$. Thus the first term in the above upper bound can be made smaller than $\epsilon/2$ by choosing k large enough. The second term also can be made smaller than $\epsilon/2$ for k large enough by Assumption 2.4. Therefore, there exists an integer K such that for all $k \geq K$, $|f(\infty, x) - f(N_k, x^k)| < \epsilon$. Thus f is continuous.

Now define the "value function" $m : \mathcal{I} \to R$ as follows.

$$m(N) = \min_{x \in F_N} \sum_{i=1}^{N} c_i x_i = \max_{x \in \Psi(N)} -f(N, x), \quad N = 1, 2, \ldots$$

and

$$m(\infty) = \min_{x \in F} \sum_{i=1}^{\infty} c_i x_i = \max_{x \in \Psi(\infty)} -f(\infty, x).$$

From the first part of Berge's Maximum Theorem, the value function is continuous, i.e., $m(N) \to m(\infty)$ as $N \to \infty$. This proves the first claim in Theorem 2.9. For the second claim, define the "argmax" correspondence μ from $\mathcal{I}$ into X as

$$\begin{aligned}
\mu(N) &= \{x \in \Psi(N) : f(N, x) = m(N)\} \equiv F_N^*, \quad N = 1, 2, \ldots \\
\mu(\infty) &= \{x \in \Psi(\infty) : f(\infty, x) = m(\infty)\} \equiv F^*.
\end{aligned}$$

The second claim then follows from the second part of Berge's Maximum Theorem.

A.4 Proof of Lemma 3.6

For the "if" part, let $T_N = F_{L_N}$. Then T_N is extendable by definition of F_{L_N}. For the "only if" part, suppose $T_N \neq F_{L_N}$. Then there is some $x \in T_N$ that is not in F_{L_N}. In particular, there is no $y \in F$ whose first L_N components match with x. Thus T_N is not extendable.

A.5 Proof of Lemma 4.1

We constructively show that (DP) is feasible. Every infinite-horizon feasible state by definition has at least one feasible action. We inductively construct a feasible solution to (DP) by choosing exactly one feasible action in each state in each period and setting the corresponding z variable to a positive value so as to satisfy the equality constraints. Specifically, pick $a_1 \in A(s_1)$ and set $z(s_1, a_1) = \beta(s_1)$. Set $z(s_1, a) = 0$ for all $a \in A(s_1)$ different from a_1. Suppose choosing action a_1 in state s_1 transforms the system to state s_2 in period 2. Pick $a_2 \in A(s_2)$ and set $z(s_2, a_2) = \beta(s_2) + \alpha z(s_1, a_1) = \beta(s_2) + \alpha\beta(s_1)$. Also set $z(s_2, a) = 0$ for all $a \in A(s_2)$ different from a_2. Continuing this procedure ad infinitum yields a feasible solution to (DP) satisfying Assumption 2.1. Let Θ be a uniform upper bounded on cardinalities $|S_n|$, Λ a uniform upper bound on cardinalities $|A(s_n)|$. It is easy to see that every equality constraint has at most $\Lambda + \Theta$ variables hence Assumption 2.2 holds. We now claim that for every feasible solution z to (DP) and every period $n = 1, 2, \ldots$,

$$\sum_{s_n \in S_n} \sum_{a_n \in A(s_n)} z(s_n, a_n) = \alpha^{n-1}\beta(s_1) + \alpha^{n-2} \sum_{s_2 \in S_2} \beta(s_2) + \ldots + \alpha \sum_{s_{n-1} \in S_{n-1}} \beta(s_{n-1}) + \sum_{s_n \in S_n} \beta(s_n).$$

We prove this claim by induction on n. The claim is true for $n = 1$ since $S_1 = \{s_1\}$ and $\sum_{a_1 \in A(s_1)} z(s_1, a_1) = \beta(s_1)$ from the equality constraint since $X(s_1) = \emptyset$. Suppose the claim is true for some period n. Then the equality constraint in (DP) implies that

$$\begin{aligned}
\sum_{s_{n+1} \in S_{n+1}} \sum_{a_{n+1} \in A(s_{n+1})} z(s_{n+1}, a_{n+1}) &= \sum_{s_{n+1} \in S_{n+1}} \beta(s_{n+1}) + \alpha \sum_{s_n \in S_n} \sum_{a_n \in A(s_n)} z(s_n, a_n) \\
&= \sum_{s_{n+1} \in S_{n+1}} \beta(s_{n+1}) + \alpha^n \beta(s_1) + \ldots + \alpha \sum_{s_n \in S_n} \beta(s_n),
\end{aligned}$$

where the last equality follows from the inductive hypothesis. This restores our inductive hypothesis proving the claim. Non-negativity of z then implies that

$$z(s_n, a_n) \leq \alpha^{n-1}\beta(s_1) + \alpha^{n-2} \sum_{s_2 \in S_2} \beta(s_2) + \ldots + \alpha \sum_{s_{n-1} \in S_{n-1}} \beta(s_{n-1}) + \sum_{s_n \in S_n} \beta(s_n)$$

for all $s_n \in S_n$, $a_n \in A(s_n)$ and all n. Hence Assumption 2.3 holds with

$$u(s_n, a_n) = \alpha^{n-1}\beta(s_1) + \alpha^{n-2} \sum_{s_2 \in S_2} \beta(s_2) + \ldots + \alpha \sum_{s_{n-1} \in S_{n-1}} \beta(s_{n-1}) + \sum_{s_n \in S_n} \beta(s_n).$$

Notice that components of u depend only on the time period and not on s_n and a_n (since s_1 is fixed). As a result, Assumption 2.4 holds since costs are in l_∞ as $0 \leq c_n(s_n, a_n) \leq c < \infty$ and u is in l_1. To see that $u \in l_1$ note that $\sum\limits_{n=1}^{\infty} \sum\limits_{s_n \in S_n} \sum\limits_{a_n \in A(s_n)} u(s_n, a_n)$ is bounded above as

$$\begin{aligned}
&\leq \Theta\Lambda \sum_{n=1}^{\infty} \left(\alpha^{n-1}\beta(s_1) + \alpha^{n-2} \sum_{s_2 \in S_2} \beta(s_2) + \ldots + \sum_{s_{n-1} \in S_{n-1}} \alpha\beta(s_{n-1}) + \sum_{s_n \in S_n} \beta(s_n) \right) \\
&= \Theta\Lambda \left(\beta(s_1) + \ldots + \sum_{s_n \in S_n} \beta(s_n) + \ldots \right) \left(\sum_{n=1}^{\infty} \alpha^{n-1} \right) = \left(\sum_{n=1}^{\infty} \sum_{s_n \in S_n} \beta(s_n) \right) \frac{\Theta\Lambda}{1-\alpha} < \infty.
\end{aligned}$$

Here the discount factor α, our choice of $\beta(s_n)$, and the inherent structure of our dynamic programs help us embed u in l_1. Recall that the discount factor appears in the constraints rather than with the costs in linear programming formulations of dynamic programs as in problem (DP). Thus even though the costs are discounted, the cost coefficients in the linear objective function are in l_∞, unlike say the production planning problem $(PROD)$ discussed in the paper. These same structural features will also prove helpful in deriving inequality (3) critical for our "big-M" method below.

A.6 Proof of Lemma 4.2

Let z be a feasible solution to $DP(N)$ and let s_{N+1} be any terminal state of the N-horizon truncation of our original dynamic program. Owing to our extendability of finite-horizon strategies assumption, any finite sequence of actions that terminates in s_{N+1} has an infinite-horizon feasible continuation. We append z along the state-action pairs of this continuation respecting equality constraints and non-negativity to construct a feasible solution to (DP). The detailed procedure is similar to the one used in showing that (DP) has a feasible solution and hence is omitted.

A.7 A brief outline of the "big-M" approach for dynamic programs

We modify the original non-stationary infinite-horizon dynamic program defined in Section 4 as follows. Include an artificial action $\nu(s_n)$ feasible in state $s_n \in S_n$ for $n = 1, 2, \ldots$. Similarly, include an artificial state Δ_n feasible in period n and a corresponding feasible action μ_n for $n = 2, 3, \ldots$. We set $g_n(s_n, \nu(s_n)) = \Delta_{n+1}$ and $g_n(\Delta_n, \mu_n) = \Delta_{n+1}$. See Figure 5. Let $c_n(s_n, \nu(s_n)) = c_n(\Delta_n, \mu_n) = M$ for some arbitrarily large number M. Notice that extendability of finite-horizon strategies holds in this "artifical" dynamic program. Let $\gamma(\Delta_n)$ be a sequence of positive numbers indexed by artificial states Δ_n such that $\sum\limits_{n=2}^{\infty} \gamma(\Delta_n) < \infty$. Then the CILP corresponding to the artificial dynamic program is given by

$$(DP_M) \quad \min \sum_{n=1}^{\infty} \sum_{s_n \in S_n} \sum_{a_n \in A(s_n)} c_n(s_n, a_n) z(s_n, a_n) + \sum_{n=1}^{\infty} \sum_{s_n \in S_n} My(s_n, \nu(s_n)) + \sum_{n=1}^{\infty} Mw(\Delta_n, \mu_n)$$

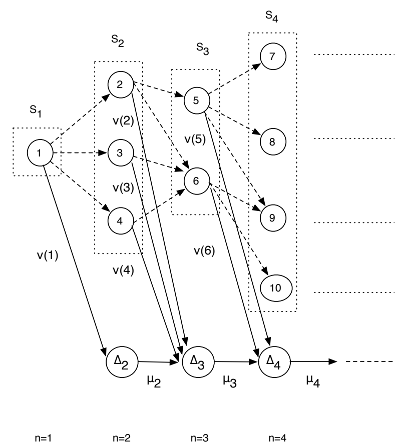

Figure 5: A portion of the dynamic programming network for the artificial dynamic program corresponding to the dynamic program in Figure 3. The artificial states and actions are shown with solid arrows for emphasis. Artificial action labels are included next to the arrows.

$$
\begin{aligned}
y(s_n,\nu(s_n)) + \sum_{a_n\in A(s_n)} z(s_n,a_n) - \alpha \sum_{(s_{n-1},a_{n-1})\in X(s_n)} z(s_{n-1},a_{n-1}) &= \beta(s_n),\ \ s_n \in S_n,\ \forall n,\\
w(\Delta_n,\mu_n) - \alpha w(\Delta_{n-1},\mu_{n-1}) - \alpha \sum_{s_{n-1}\in S_{n-1}} y(s_{n-1},\nu(s_{n-1})) &= \gamma(\Delta_n),\ \ n=2,3,\ldots\\
z(s_n,a_n)\geq 0,\ s_n\in S_n,\ a_n\in A(s_n),\ \forall n,\ \ y(s_n,\nu(s_n)) &\geq 0,\ s_n\in S_n,\ \forall n\\
w(\Delta_n,\mu_n) &\geq 0,\ n=2,3\ldots.
\end{aligned}
$$

Note that the variables in (DP_M) include the original variables z in (DP) as well as the artificial variables y and w. By replicating the proof of Lemma 4.1 one can confirm that (DP_M) satisfies Assumptions 2.1-2.4. Hence (DP_M) has an optimal solution. More importantly, all finite-horizon truncations of (DP_M) are extendable by an argument similar to Lemma 4.2 because finite-horizon strategies in the artificial dynamic program are extendable. Notice that if solution z is feasible to (DP) then it is also feasible to (DP_M) by setting $y(s_n,\nu(s_n)) = 0$ for all $s_n\in S_n$, $n=1,2,\ldots$, and $w(\Delta_n,\mu_n) = \gamma(\Delta_n) + \sum_{i=2}^{n-1}\alpha^{n-i}\gamma(\Delta_i) \equiv \hat{w}(\Delta_n,\mu_n)$ for $n=2,3,\ldots$. Suppose $\bar{z}$, $\bar{y}$, $\bar{w}$ is an optimal solution to (DP_M) and suppose $\bar{y}(s_n,\nu(s_n))>0$ for some s_n and $\nu(s_n)$. Then $\bar{w}(\Delta_n,\mu_n)\geq \hat{w}(\Delta_n,\mu_n)$ for all n. Then as in the "big-M" method for finite-dimensional linear programs (Exercise 3.26 of [11]) it is easy to prove that (DP) is infeasible. For if it is not, and if z is feasible to (DP), then $\sum_{n=1}^{\infty}\sum_{s_n\in S_n}\sum_{a_n\in A(s_n)} c_n(s_n,a_n)z(s_n,a_n) + \sum_{n=1}^{\infty} M\hat{w}(\Delta_n,\mu_n)$ is at least

$$
\sum_{n=1}^{\infty}\sum_{s_n\in S_n}\sum_{a_n\in A(s_n)} c_n(s_n,a_n)\bar{z}(s_n,a_n) + \sum_{n=1}^{\infty}\sum_{s_n\in S_n} M\bar{y}(s_n,\nu(s_n)) + \sum_{n=1}^{\infty} M\bar{w}(\Delta_n,\mu_n),
$$

owing to feasibility of z, $y=0$, $\hat{w}$ and optimality of $\bar{z}$, $\bar{y}$ and $\bar{w}$ to (DP_M) respectively. Therefore, $\sum_{n=1}^{\infty}\sum_{s_n\in S_n}\sum_{a_n\in A(s_n)} c_n(s_n,a_n)z(s_n,a_n)$ is bounded below as

$$
\geq \sum_{n=1}^{\infty}\sum_{s_n\in S_n}\sum_{a_n\in A(s_n)} c_n(s_n,a_n)\bar{z}(s_n,a_n) + \sum_{n=1}^{\infty}\sum_{s_n\in S_n} M\bar{y}(s_n,\nu_n) + \sum_{n=1}^{\infty} M\left(\bar{w}(\Delta_n,\mu_n) - \hat{w}(\Delta_n,\mu_n)\right)
$$

$$
\geq \sum_{n=1}^{\infty}\sum_{s_n\in S_n} M\bar{y}(s_n,\nu_n) + \sum_{n=1}^{\infty} M\left(\bar{w}(\Delta_n,\mu_n) - \hat{w}(\Delta_n,\mu_n)\right),
$$

where the second inequality uses non-negativity of $c_n(s_n, a_n)$ and $\bar{z}(s_n, a_n)$. Let $n \geq 1$ be the smallest period for which there exist s_n and $\nu(s_n)$ such that $\bar{y}(s_n, \nu(s_n)) > 0$. This implies that $\bar{w}(\Delta_{n+1}, \mu_{n+1}) - \hat{w}(\Delta_{n+1}, \mu_{n+1}) > \alpha\bar{y}(s_n, \nu(s_n))$. Then using an upper bound from Lemma 4.1 on the left hand side we get $\frac{c\Theta\Lambda}{1-\alpha}\left(\sum_{n=1}^{\infty}\sum_{s_n \in S_n}\beta(s_n)\right) \geq M(\bar{y}(s_n, \nu(s_n)) + \alpha\bar{y}(s_n, \nu(s_n)))$, i. e.,

$$\frac{\frac{c\Theta\Lambda}{1-\alpha}\left(\sum_{n=1}^{\infty}\sum_{s_n \in S_n}\beta(s_n)\right)}{\bar{y}(s_n, \nu(s_n)) + \alpha\bar{y}(s_n, \nu(s_n))} \geq M, \tag{3}$$

which is a contradiction since M is arbitrarily large. Thus Assumption 2.1 implies $\bar{y} = 0$. This also implies that $\bar{w} = \hat{w}$. Moreover, in that case, $\bar{z}$ is in fact optimal to (DP). In summary, (DP_M) satisfies all assumptions required for Shadow Simplex and is equivalent to (DP). This also shows that our assumption that finite-horizon strategies are extendable is without loss of generality.

A.8 Proof of Lemma 4.3

First recall that a point in a convex set is its extreme point if it cannot be expressed as a convex combination of two other distinct points in the convex set. Let z be as in the hypothesis of the lemma. Suppose by way of contradiction that z is not an extreme point. Then there exist feasible solutions x and y and a fraction $0 < \lambda < 1$ such that $z = \lambda x + (1-\lambda)y$. We show that $x = y = z$. Non-negativity of x and y implies that for all states s_n and actions $a_n \in A(s_n)$ for which $z(s_n, a_n) = 0$, $x(s_n, a_n) = y(s_n, a_n) = 0$. Then one can confirm using simple algebra starting at state s_1 and working inductively that under this restriction the equality constraints in (DP) have a unique solution, namely, z. This implies $x = y = z$.

A.9 Proof of Lemma 4.4

Suppose z is a feasible solution to (DP) and there exists a state s_n (called a "bad" state) with (at least) two actions $a_n \in A(s_n)$ and $b_n \in A(s_n)$ such that $z(s_n, a_n) > 0$ and $z(s_n, b_n) > 0$. Let $s_{n+1} = g_n(s_n, a_n)$ and $t_{n+1} = g_n(s_n, b_n)$ and note that $s_{n+1} \neq t_{n+1}$ since two distinct actions lead to two distinct states. In this proof, we consider two types of state-action sequences whose existence follows from the structure of problem (DP). The first is of the form $\{(s_r, a_r)\}_{r=n+1}^{\infty}$ starting at s_{n+1} such that for all r, $s_r \in S_r$, $a_r \in A(s_r)$, $s_{r+1} = g_r(s_r, a_r)$ and $z(s_r, a_r) > 0$. The set of all such sequences is denoted $\Omega(s_n, a_n)$. The second is of the form $\{(t_r, b_r)\}_{r=n+1}^{\infty}$ starting at t_{n+1} such that for all r, $t_r \in S_r$, $b_r \in A(t_r)$, $t_{r+1} = g_r(t_r, b_r)$ and $z(t_r, b_r) > 0$. The set of all such sequences is denoted $\Omega(s_n, b_n)$. We say that a sequence from $\Omega(s_n, a_n)$ "passes through" a particular state, or this particular state "belongs to" the sequence if it is included in a state-action pair that is in the sequence. Similarly for sequences in $\Omega(s_n, b_n)$. We also use this terminology for state-action pairs. We consider two possible cases and in both these show that it is possible to construct two distinct solutions x and y feasible to (DP) such that $z = (x+y)/2$ and hence z cannot be an extreme point to complete the proof by contrapositive.

Case 1: There exists a bad state s_n for which a sequence from $\Omega(s_n, a_n)$ passes through a state that also belongs to some sequence in $\Omega(s_n, b_n)$.

Let s_{n+k} be the first state that these two sequences have in common for some $k > 1$, that is, $s_{n+k} = t_{n+k}$, and $s_{n+j} \neq t_{n+j}$ for $j = 2, 3 \ldots, k-1$. Specifically, let $\epsilon(a_n) > 0$ be the largest amount that can be subtracted from $z(s_n, a_n)$, reducing values of $z(s_{n+1}, a_{n+1}), \ldots, z(s_{n+k-1}, a_{n+k-1})$ in order to satisfy the equality constraints in (DP) at the same time forcing these variables to be non-negative. Similarly, let $\epsilon(b_n) > 0$ be the largest amount that can be subtracted from $z(s_n, b_n)$, reducing values of $z(t_{n+1}, b_{n+1}), \ldots, z(s_{n+k-1}, b_{n+k-1})$ in order to satisfy the equality constraints in (DP) at the same time forcing these variables to be non-negative. Set $\epsilon = \min\{\epsilon(a_n), \epsilon(b_n)\}$. Let x be the feasible solution formed from z by subtracting ϵ from $z(s_n, a_n)$, reducing values of $z(s_{n+1}, a_{n+1}), \ldots, z(s_{n+k-1}, a_{n+k-1})$, adding ϵ to $z(s_n, b_n)$, and increasing values of $z(t_{n+1}, b_{n+1})$, $\ldots$, $z(t_{n+k-1}, b_{n+k-1})$ to satisfy equality constraints. Similarly, let y be the feasible solution formed from z by adding ϵ to $z(s_n, a_n)$, increasing values of $z(s_{n+1}, a_{n+1}), \ldots, z(s_{n+k-1}, a_{n+k-1})$, subtracting ϵ from $z(s_n, b_n)$, and reducing values of $z(t_{n+1}, b_{n+1}), \ldots, z(t_{n+k-1}, b_{n+k-1})$ to satisfy equality constraints. Then it is easy to check that $z = (x + y)/2$ and hence z is not an extreme point.

Case 2: There is no bad state s_n for which a sequence from $\Omega(s_n, a_n)$ and a sequence from $\Omega(s_n, b_n)$ both pass through a common state.

Consider any bad state s_n and corresponding actions $a_n \in A(s_n)$ and $b_n \in A(s_n)$ such that $z(s_n, a_n) > 0$, and $z(s_n, b_n) > 0$. For any sequence $\zeta \in \Omega(s_n, a_n)$, and any period $N \geq n+1$, ζ_N denotes the state-action pair that ζ passes through in period N and thus we can write $z(\zeta_N)$. Define $\phi^z_N(s_n, a_n) = \sum\limits_{\zeta \in \Omega(s_n, a_n)} z(\zeta_N)$.

Suppose without loss of generality that $z(s_n, a_n) \geq z(s_n, b_n)$. Note that $\phi^z_N(s_n, a_n) \geq \alpha^{(N-n)} z(s_n, a_n)$, which in turn is at least $\alpha^{(N-n)} z(s_n, b_n)$, and also that $\phi^z_N(s_n, b_n) \geq \alpha^{(N-n)} z(s_n, b_n)$. Let x be a new feasible solution formed as follows. $x(s_n, a_n) = z(s_n, a_n) - z(s_n, b_n)$ and $x(\zeta_N) = z(\zeta_N) - \epsilon(\zeta_N)$ for some $0 \leq \epsilon(\zeta_N) \leq z(\zeta_N)$ for all $\zeta_N \in \Omega(s_n, a_n)$ for all $N \geq n+1$. These $\epsilon(\zeta_N)$ are chosen so that $\phi^\epsilon_N(s_n, a_n) = \alpha^{(N-n)} z(s_n, b_n)$ satisfying (DP) equality constraints at every state that sequences ζ pass through. Moreover, $x(s_n, b_n) = 2z(s_n, b_n)$ and $x(\xi_N) = z(\xi_N) + \delta(\xi_N)$ for some $0 \leq \delta(\zeta_N) \leq z(\xi_N)$ for all $\xi_N \in \Omega(s_n, b_n)$ for all $N \geq n+1$. These $\delta(\xi_N)$ are chosen so that $\phi^\delta_N(s_n, b_n) = \alpha^{(N-n)} z(s_n, b_n)$ satisfying (DP) equality constraints at every state that sequences ξ pass through. Similarly, y is a new feasible solution formed so that $y(s_n, a_n) = z(s_n, a_n) + z(s_n, b_n)$ and $y(\zeta_N) = z(\zeta_N) + \epsilon(\zeta_N)$. Moreover, $y(s_n, b_n) = 0$ and $y(\xi_N) = z(\xi_N) - \delta(\xi_N)$. All the other components of x and y are equal to the corresponding components of z implying $z = (x + y)/2$.

A.10 Proof of Lemma 5.1

We first show constructively that (MDP) has a feasible solution. The procedure is similar to problem (DP). The first equality constraint is simply $\sum\limits_{a_1 \in A(s_1)} z(s_1, a_1) = \beta(s_1)$ since $X(s_1) = \emptyset$. Thus we arbitrarily choose one action $a_1 \in A(s_1)$ and set $z(s_1, a_1) = \beta(s_1)$. For all other actions in $a \in A(s_1)$, we set $z(s_1, a) = 0$. Then, the equality constraint corresponding to every $s_2 \in S_2$ reduces to $\sum\limits_{a_2 \in A(s_2)} z(s_2, a_2) = \alpha p_1(s_2|s_1, a_1) z(s_1, a_1) + \beta(s_2)$. We therefore choose an arbitrary action $a_2 \in A(s_2)$ and set $z(s_2, a_2) = \alpha p_1(s_2|s_1, a_1) z(s_1, a_1) + \beta(s_2)$. That is, set $z(s_2, a_2) = \alpha p_1(s_2|s_1, a_1)\beta(s_1) + \beta(s_2)$. For all other actions $a \in A(s_2)$ we set $z(s_2, a) = 0$. Continuing this way we can construct a feasible solution to (MDP) thus satisfying Assumption 2.1. Every equality constraint has at most $\Lambda + \Lambda\Theta$ variables hence Assumption 2.2 holds. Similar to problem (DP) we now claim that for any n, $\sum\limits_{s_n \in S_n} \sum\limits_{a_n \in A(s_n)} z(s_n, a_n) = \alpha^{n-1}\beta(s_1) + \alpha^{n-2} \sum\limits_{s_2 \in S_2} \beta(s_2) + \ldots + \sum\limits_{s_n \in S_n} \beta(s_n)$. We prove this by induction. The claim is true for $n = 1$ as $S_1 = \{s_1\}$ and $X(s_1) = \emptyset$ implying that $\sum\limits_{a_1 \in A(s_1)} z(s_1, a_1) = \beta(s_1)$. Now suppose it is true for some period n. Then owing to the equality constraint, $\sum\limits_{s_{n+1} \in S_{n+1}} \sum\limits_{a_{n+1} \in A(s_{n+1})} z(s_{n+1}, a_{n+1})$ equals

$$\sum_{s_{n+1} \in S_{n+1}} \beta(s_{n+1}) + \alpha \sum_{s_{n+1} \in S_{n+1}} \sum_{s_n \in X(s_{n+1})} \sum_{a_n \in \mathcal{X}(s_n, s_{n+1})} p_n(s_{n+1}|s_n, a_n) z(s_n, a_n)$$

$$= \sum_{s_{n+1} \in S_{n+1}} \beta(s_{n+1}) + \alpha \sum_{s_n \in S_n} \sum_{a_n \in A(s_n)} z(s_n, a_n) = \sum_{s_{n+1} \in S_{n+1}} \beta(s_{n+1}) + \alpha^n \beta(s_1) + \ldots + \alpha \sum_{s_n \in S_n} \beta(s_n),$$

where the last equality follows from the inductive hypothesis. This restores the inductive hypothesis proving our claim. The rest of the proof is identical to (DP) and hence is omitted.

A.11 Proof of Lemma 5.2

Similar to the (DP) case hence omitted.