JOURNAL OF MATHEMATICAL ANALYSIS AND APPLICATIONS **166**, 175–191 (1992)

Computing Efficient Steady State Policies for Deterministic Dynamic Programs, I

JAMES FLYNN

Department of Quantitative Business Analysis, Cleveland State University, Cleveland, Ohio 44115

Submitted by E. Stanley Lee

Received October 1, 1990

1. INTRODUCTION

The computer storage and computing time requirements of dynamic programming can be excessive when the dimension of the state space exceeds three (Bellman's "curse of dimensionality"). One approach to this type of difficulty is to search for *efficient heuristics*, i.e., suboptimal decision rules that perform well and are not too demanding computationally. This paper deals with the heuristic *steady state policy* (SSP), i.e., a policy that moves the system from a set of states S_0 to a *target state* s^* and then keeps it there. SSPs are sometimes appropriate for deterministic, stationary, dynamic programming models, where the objective is to optimize the total undiscounted costs over an infinite or long finite planning horizon; [11, Sect. 1] discusses advantages of SSPs.

The adverse effect of a suboptimal policy π on the total costs need not be serious. To gauge the damage, we use the *opportunity cost*. Section 2 defines this, roughly, as the asymptotic difference between the N-stage costs under π and the optimal N-stage costs, thus measuring how suboptimal π is over a long horizon. Policies with *finite opportunity cost are average-cost optimal*. The opportunity cost reflects their transient performance. It provides *information* about whether one should seek a better policy.

In [11], the author obtains general sufficient conditions for the existence of an SSP with finite opportunity cost. Given these conditions, one can determine such an SSP by, first, solving a *mathematical program* to find s^* and, second, constructing a *feasible guidance rule* that moves the system to s^* from some set S_0. This SSP generally does not optimize transient costs. Nevertheless, using it as a *first-order approximation* can produce dramatic computational savings, sometimes allowing one to handle problems that would ordinarily be intractable (see [11, 19] and the models cited in Section 5). Section 5 covers the first-order approximation in detail.

0022-247X/92 $3.00

Sections 6 through 8 study second-order improvements that reduce the transient costs of a given first-order approximation π. Section 6 shows that for $s \in S_0$, the opportunity cost of π can be expressed as the sum of three nonnegative terms: the first, $O_G(\pi, s)$, is the penalty associated with the guidance rule; the second, $O(s, s^*)$, the penalty for visiting s^* from state s; the third, $\mathrm{OC}(s^*)$, the opportunity cost of keeping the system at s^*. Think of $\mathrm{OC}(s^*)$ as the *tail* penalty. Upper bounds on each term are available. Sections 7 and 8, respectively, examine improvements in the guidance rule and the tail.

Let Ω denote the set of SSPs with the same s^* and S_0 as the first-order approximation. Section 7 studies the problem of finding a $\hat{\pi} \in \Omega$ with $O_G(\hat{\pi}, s) = 0$, for $s \in S_0$ (i.e., a $\hat{\pi}$ in Ω with minimum opportunity cost). This reduces to a shortest route problem in a network with nonnegative arc lengths whose node set is the *original* state space. Exact solutions are difficult to obtain for large finite S and need not exist for infinite S. The latter part of Sect. 7 considers methods for finding approximate solutions.

Given a *finite* horizon N, one can reduce the total costs by allowing the system to quit s^* towards the end of the horizon and switch to a policy that is optimal for the remaining stages, thus mimicking turnpike behavior [5, 7] (Ref. [10, Sects. 1 and 6] compares turnpike theorems and steady state policies). The savings produced by such tail improvements can approach $\mathrm{OC}(s^*)$. Section 8 provides bounds on the N-stage cost and relative error.

Given appropriate smoothness assumptions, one can apply nonlinear programming and optimal control techniques when making second-order improvements (see the discussion at the end of Sect. 7). A companion paper [16] applies our results to multi-dimensional examples from [9]. Using nonlinear programming methods, that paper obtains relative errors of under 0.6 percent for 5-dimensional models with planning horizons $N \geqslant 20$. (Small problems of this size were easily solved on a microcomputer using the software package GINO [21]). This suggests that SSPs might be appropriate for medium sized planning horizons.

The rest of this paper is organized as follows. Section 2 presents our model; Section 3 summarizes material needed from [15]; and Section 4 obtains results on bounds.

2. The Model

This paper considers deterministic dynamic programs that are characterized by the following objects: S, a nonempty compact subset of R^n; A, a nonempty compact subset of R^m; $A(\cdot)$, a mapping from S into the family of nonempty subsets of S, whose graph, $D \equiv \{(a, s) : a \in A(s) \text{ and } s \in S\}$, is closed and thus compact in R^{m+n}; $c(\cdot, \cdot)$, a bounded, real lower semicontinuous function on D; and $t(\cdot, \cdot)$, a continuous mapping from D into S. (These continuity and compactness assumptions hold trivially when S and A are finite; however, the main applications of our results are to models with convex S.) Periodically, at stages $M = 1, 2, \ldots$, one observes a *state* $s \in S$ and selects an *action* $a \in A(s)$. The result is an *immediate cost* $c(a, s)$ and a *transition* at stage $M + 1$ to a new state $t(a, s)$. A *policy* is any rule that specifies for each *initial state* $s \in S$ a sequence of *feasible* actions: $a_1 \in A(s)$, $a_2 \in A(t(a_1, s)), \ldots$. The problem is to control the system over a prescribed horizon, which may be finite or infinite.

Let N be a positive integer. For any policy π and any $s \in S$, let $V_\pi^N(s)$ denote the total undiscounted N-stage costs under π when the initial state is s. Similarly, let $V_*^N(s)$ denote the corresponding optimal quantity, i.e., $V_*^N(s) \equiv \min_{\tilde\pi} V_{\tilde\pi}^N(s)$. Call π *N-stage optimal* at s if $V_\pi^N(s) = V_*^N(s)$. It is convenient to define $V_\pi^0(s) \equiv 0$ and $V_*^0(s) \equiv 0$.

Our main concern is with long horizons. Define π to be *average-cost optimal* at $s \in S$ if $\limsup_N [V_\pi^N(s) - V_*^N(s)]/N = 0$ (see [12]). This criterion can be underselective, since it depends only on the tail. Think of $[V_\pi^N(s) - V_*^N(s)]$ as the penalty for using π for N-stages when the initial state is s. Following [11, 12], define the *opportunity cost* of π at s as

$$\mathrm{OC}(\pi, s) = \limsup_N [V_\pi^N(s) - V_*^N(s)], \qquad (s \in S). \tag{2.1}$$

The policy π is said to have *finite opportunity cost* at $s \in S$ if $\mathrm{OC}(\pi, s) < \infty$. If π has finite opportunity cost, π is average-cost optimal. The opportunity cost refects the transient performance of average-cost optimal policies. It provides *information* about how much better one might do with a different policy; $\mathrm{OC}(\pi, s)$ is the smallest number with the property: For each $\varepsilon > 0$, the possible decreases in $V_\pi^N(s)$ are bounded by $[\mathrm{OC}(\pi, s) + \varepsilon]$ for N sufficiently large. For more details about the opportunity cost, see [12, Sect. 4].

If for some $N \geqslant 0$, the system can reach $s' \in S$ in N stages when the initial state is $s \in S$, then s' is said to be *accessible* from s. (Every state is accessible from itself.) Denote by $\mathrm{ACS}(s)$ the set of states accessible from s. For any subset S_0 of S, let $\mathrm{ACS}(S_0)$ denote the set consisting of each state that is accessible from some $s \in S_0$.

3. Optimal Steady States and Excessive Functions

This section summarizes definitions and results needed from [15].

3.1. *Optimal Steady States*

Intuitively, s^* is an *optimal steady state* (OSS) if one can stay in s^* while incurring finite opportunity cost. Let $D_f = \{(a, s) : (a, s) \in D$ and

$t(a, s) = s\}$, and let S_f equal the *projection* of D_f on S. Both D_f and S_f are compact. For $s \in S_f$, let $c_f(s) = \min\{c(a', s): a' \in A(s)$ and $t(a', s) = s\}$ and $A_f(s) = \{a' : a' \in A(s),\ t(a', s) = s$, and $c(a', s) = c_f(s)\}$. Suppose $s \in S_f$ and $N \geqslant 1$. Clearly, $V_*^{N+1}(s) \leqslant c_f(s) + V_*^N(s)$.This implies that $Nc_f(s) - V_*^N(s)$ is nondecreasing in N. Define

$$\mathrm{OC}(s) = \lim_N Nc_f(s) - V_*^N(s), \qquad \text{for} \quad s \in S_f. \tag{3.1}$$

Note that $\mathrm{OC}(s)$ equals the opportunity cost of staying in state s through the choice of a minimum cost $a \in A_f(s)$. As in [15], define s to be an OSS if $s \in S_f$ and $\mathrm{OC}(s)$ is finite. The following is necessary for s^* to be an OSS:

$$c_f(s^*) = \min_{s' \in \mathrm{ACS}(s^*) \cap S_f} c_f(s'). \tag{3.2}$$

3.2. *Lagrangian Saddle-Points and Excessive Functions*

One can sometimes (see Sect. 3.3) find an OSS by solving the *constrained optimization problem*: Compute (a^*, s^*) where

$$(a^*, s^*) \in D_f \qquad \text{and} \qquad c(a^*, s^*) = \min\{c(a, s) : (a, s) \in D_f\}. \tag{3.3}$$

Define the *Lagrangian function*, $L(a, s, w) = c(a, s) + w[t(a, s) - s]$, for $(a, s) \in D$ and $w \in R^n$. Call $(a^*, s^*, w^*) \in D \times R^n$ a *Lagrangian saddle-point* (LSP) if $L(a^*, s^*, w) \leqslant L(a^*, s^*, w^*) \leqslant L(a, s, w^*)$, for all $(a, s) \in D$ and $w \in R^n$. This is equivalent to

$$\begin{aligned} &(a^*, s^*) \in D_f \\ \text{and} \qquad & \\ &c(a, s + w^* t(a, s) \geqslant w^* s + c(a^*, s^*), \qquad \text{for} \quad (a, s) \in D. \end{aligned} \tag{3.4}$$

If (a^*, s^*, w^*) is an LSP, then (3.3) holds and $c(a^*, s^*) = c_f(s^*)$, so $s^* \in S_f$ and $a^* \in A_f(s^*)$. Call w^* a *saddle-point multiplier* if (a^*, s^*, w^*) is an LSP for some (a^*, s^*). Clearly if w^* is a saddle-point multiplier and (a^*, s^*) satisfies (3.3), then (a^*, s^*, w^*) is an LSP. Ref. [15] generalizes (3.4) as follows. For $s^* \in S_f$, let $E(s^*)$ denote the set of extended-real valued functions W on S that satisfy

$$\begin{aligned} &W(s^*) = 0 \\ \text{and} \qquad & \\ &c(a, s) + W(t(a, s)) \geqslant W(s) + c_f(s^*), \qquad \text{for} \quad (a, s) \in D. \end{aligned} \tag{3.5}$$

Define W to be *excessive at* $s^* \in S_f$ if $W \in E(s^*)$. Call W *excessive* if $W \in E(s^*)$ for some $s^* \in S_f$. See [15] for a detailed study of excessive functions. By Theorem 6.1 of that paper, s^* is an OSS if and only if some $W \in E(s^*)$ is bounded above on ACS(s^*).

3.3. *Examples of Excessive Functions*

For $w^* \in R^n$ and $s^* \in S_f$, define the function

$$w^*_{s^*}(s) = w^*s - w^*s^*, \qquad \text{for} \quad s \in S. \tag{3.6}$$

Since S is compact, $w^*_{s^*}$ is bounded. Clearly, if (a^*, s^*, w^*) is an LSP, then $w^*_{s^*} \in E(s^*)$. (Incidentally, this and Theorem 4.1 below imply s^* is an OSS when (a^*, s^*, w^*) is an LSP.)

By [15, Sect. 6], when s^* is an OSS, $E(s^*)$ has a *greatest element* $W^{\mathrm{G}}_{s^*}$ and a *least element* $W^{\mathrm{L}}_{s^*}$. Their definitions arise naturally from the problem of finding for each $M \geqslant 1$ an optimal M-stage path between two arbitrary states in S. For $M \geqslant 1$, and $s, s' \in S$, let

$$T_M = \{\langle (a_j, s_j)\rangle_{j=1}^M : (a_j, s_j) \in D,\ 1 \leqslant j \leqslant M, \\ \text{and} \qquad s_{j+1} = t(a_j, s_j),\ 1 \leqslant j < M\}, \tag{3.7a}$$

$$T_M(s, s') = \{\langle (a_j, s_j)\rangle_{j=1}^M \in T_M : s_1 = s \text{ and } s' = t(a_M, s_M)\}, \tag{3.7b}$$

$$V^M(s, s') = \begin{cases} \min\left\{\sum_{j=1}^M c(a_j, s_j) \colon \langle (a_j, s_j)\rangle_{j=1}^M \in T_M(s, s')\right\}, \\ \qquad \text{if} \quad T_M(s, s') \neq \varnothing, \\ \infty, \qquad \text{if} \quad T_M(s, s') = \varnothing. \end{cases} \tag{3.8}$$

Clearly, T_M and $T_M(s, s')$ are compact sets in D^M. This and a continuity argument justify the "min" in (3.8). The set $T_M(s, s')$ consists of all M-stage, feasible, state-action paths from s to s'; $V^M(s, s')$ is the cost of an optimal path in $T_M(s, s')$. Suppose $s^* \in S_f$. Define

$$W^{\mathrm{G}}_{s^*}(s) = \lim_M [V^M(s, s^*) - Mc_f(s^*)], \qquad \text{for} \quad s \in S, \tag{3.9}$$

$$W^{\mathrm{L}}_{s^*}(s) = \lim_M [Mc_f(s^*) - V^M(s^*, s)], \qquad \text{for} \quad s \in S, \tag{3.10}$$

$$V^{\infty}_{s^*}(s) = \liminf_N [V^N_*(s) - V^N_*(s^*)], \qquad \text{for} \quad s \in S. \tag{3.11}$$

(Ref. [12, Lemma 6.1] justifies the "lim" in (3.9) and (3.10)). The next lemma lists properties of $W^G_{s^*}$, $W^{\mathrm{L}}_{s^*}$, and $V^{\infty}_{s^*}$ needed in this paper (see [15, Lemma 5.1 and Theorems 6.1 and 6.2]).

LEMMA 3.1. *If s^* is an OSS, then the following hold for $s \in S$:*

(a) *The functions $W^{G}_{s^*}$, $V^{\infty}_{s^*}$, and $W^{L}_{s^*}$ belong to $E(s^*)$.*

(b) *If $W \in E(s^*)$, then $W^{L}_{s^*}(s) \leqslant W(s) \leqslant W^{G}_{s^*}(s)$.*

(c) $V^M(s, s^*) - Mc_f(s^*) \downarrow W^{G}_{s^*}(s)$ *and* $Mc_f(s^*) - V^M(s^*, s) \uparrow W^{L}_{s^*}(s)$.

(d) $W^{G}_{s^*}(s) < \infty$ *iff* $s^* \in ACS(s)$; $W^{L}_{s^*}(s) > -\infty$ *iff* $s \in ACS(s^*)$.

(e) $W^{L}_{s^*}(s) \leqslant \sup_{s' \in S} W^{L}_{s^*}(s') = \lim_M [Mc_f(s^*) - V^M_*(s^*)] < \infty$.

4. EXCESSIVE FUNCTIONS AND BOUNDS

Using excessive functions, one can construct lower bounds on the N-stage costs and upper bounds on the opportunity cost (see Theorems 4.1 and 5.1). For $s^* \in S_f$, $W \in E(s^*)$, $K \geqslant 0$, and the subset S_0 of S, define

$$U^K(W, s^*, S_0) = \sup_{s \in \mathrm{ACS}(S_0)} [W(s) - V^K_*(s) + Kc_f(s^*)]. \tag{4.1}$$

The next theorem follows easily from the arguments for [12, Theorem 5.1].

THEOREM 4.1. *Suppose S_0 is a subset of S, $s^* \in S_f \cap S_0$, and $W \in E(s^*)$ is bounded above on $ACS(S_0)$.*

(a) $U^K(W, s^*, S_0) < \infty$, *for* $K \geqslant 0$;
furthermore, $U^K(W, s^*, S_0)$ *is nonincreasing in* K.

(b) $V^N_*(s) \geqslant W(s) + Nc_f(s^*) - U^K(W, s^*, S_0)$,
for $N \geqslant K$ *and* $s \in ACS(S_0)$. (4.2)

(c) $Mc_f(s^*) - V^M_*(s^*) \uparrow OC(s^*) \leqslant U^K(W, s^*, S_0)$,
for $M \geqslant 0$ *and* $K \geqslant 0$. (4.3)

(d) s^* *is an OSS.*

Part (a) ensures that the accuracy of the bounds in (4.2) and (4.3) is nondecreasing in K. (The same holds for the bounds in Theorem 5.1.) Of course the work also increases with K. Because of (4.3), information about $V^M_*(s^*)$ can aid in selecting K (see Section 8). The following theorem characterizes the limiting behavior of $U^K(W, s^*, S_0)$ as K approaches ∞.

THEOREM 4.2. *Assume the hypotheses of Theorem 4.1.*

(a) $OC(s^*) = U^K(W, s^*, S_0)$, *for* $K \geqslant 0$ *when* $W = W^{L}_{s^*}$.

(b) $\lim_K U^K(W, s^*, S_0) \geqslant \sup_{s \in ACS(S_0)} [W(s) - V_{s^*}^{\infty}(s)] + OC(s^*)$. (4.4)

(c) *Equality holds in* (4.4) *when*

$$V_*^N(s) - V_*^N(s^*) \text{ converges uniformly to } V_{s^*}^{\infty}(s).$$

Proof. By Lemma 3.1(a, d, and e), $W_{s^*}^{\mathrm{L}} \in E(s^*)$ is bounded above. Hence, Theorem 4.1(c) implies $\mathrm{OC}(s^*) \leqslant U^K(W_{s^*}^{\mathrm{L}}, s^*, S_0)$, while (3.1) and Lemma 3.1(d and e), imply $U^0(W_{s^*}^{\mathrm{L}}, s^*, S_0) = \mathrm{OC}(s^*)$. To finish (a), apply Theorem 4.1(a). Next, (4.1) implies that for $K \geqslant 0$,

$$U^K(W, s^*, S_0) = \sup_{s \in \mathrm{ACS}(S_0)} \{W(s) - [V_*^K(s) - V_*^K(s^*)]\} + Kc_f(s^*) - V_*^K(s^*). \tag{4.5}$$

Using Theorem 4.1(a), (3.11), and (3.1), one can show that $\lim_K U^K(W, s^*, S_0) \geqslant W(s) - V_{s^*}^{\infty}(s) + \mathrm{OC}(s^*)$, for $s \in \mathrm{ACS}(S_0)$. Take "$\sup_s$" to get (b). For (c), let $K \to \infty$ in (4.5). Uniform convergence lets one interchange "$\sup_s$" and "$\lim_K$" to get equality in (4.4). ∎

Part (a) suggests that larger K are less worthwhile when W is close to its lower bound $W_{s^*}^{\mathrm{L}}$. Note that $S_0 = S$ and $W = w_{s^*}^*$ in [16], which does computations for small K using

$$U^0(w_{s^*}^*, s^*, S) = \max_{s \in S} w_{s^*}^*(s), \tag{4.6}$$

$$U^K(w_{s^*}^*, s^*, S) \equiv \max \left\{ w_{s^*}^*(s_1) - \sum_{j=1}^{K} c(a_j, s_j) + Kc_f(s^*) \colon \langle (a_j, s_j) \rangle_{j=1}^K \in T_K \right\}, \quad \text{for } K \geqslant 1 \tag{4.7}$$

(see Sect. 3.3). Those computations find that on the average, $K = 1$ does much better than $K = 0$, but larger K yield only marginal improvements. Finally, note that (c) needs uniform convergence: pointwise convergence is not enough.

5. Steady State Policies and First-Order Approximations

Additional definitions

For infinite horizon problems, the policies usually studied are *stationary policies*, i.e., policies defined by a mapping $\pi: S \to A$ such that $\pi(s) \in A(s)$, $s \in S$, and the rule: When in state s, choose action $\pi(s)$. In contrast, this paper focuses on *steady state policies*, a class of policies—including both stationary and nonstationary ones—studied extensively in [11]. Under such a policy, there is a subset S_0 of states from which the system heads towards a "steady state." Formally, a steady state policy (SSP) π is specified by the following entities: the *target state* $s^* \in S_f$; the *domain* S_0, a subset of S which contains s^*; and the *guidance rule* which directs the system from each $s \in S_0$ to s^* in a finite number of stages. On reaching the target s^*, the system stays there by continually choosing a *stabilizing action* $a^* \in A_f(s^*)$. For $s \neq s^*$, let $N(\pi, s)$ equal the number of stages that elapse *before* the system reaches s^* from s. Let $N(\pi, s^*) = 0$. Say π has *finite opportunity cost* if it has finite opportunity cost at each $s \in S_0$. Define $\Omega(s^*, S_0)$ as the set of SSPs with domain S_0 and target s^*.

The definition of SSP in [11] does not require either (i) the same target for each $s \in S_0$, or (ii) $a^* \in A_f(s^*)$. Clearly (ii) does not sacrifice optimality. One can circumvent (i) by breaking S_0 into smaller subsets and making each the domain of some SSP. The next theorem describes verifiable conditions under which an SSP has finite opportunity cost together with bounds. (The bound in (5.2) is essentially the same as the bound in [11, Theorem 4.1] when $K = 0$. For additional information about the bounds, see Section 4.)

THEOREM 5.1. *Let π be an SSP with domain S_0 and target state s^*. Let W be excessive at s^* and finite and bounded above on $ACS(S_0)$. Then π has finite opportunity cost and s^* is an OSS. Furthermore, the following hold for $s \in S_0$, $K \geqslant 0$, and $N \geqslant \max\{N(\pi, s), K\}$:*

$$V_\pi^N(s) - V_*^N(s) \leqslant V_\pi^{N(\pi,s)}(s) - N(\pi,s)\, c_f(s^*) - W(s) + U^K(W, s^*, S_0) < \infty, \tag{5.1}$$

$$OC(\pi, s) \leqslant V_\pi^{N(\pi,s)}(s) - N(\pi, s)\, c_f(s^*) - W(s) + U^K(W, s^*, S_0) < \infty. \tag{5.2}$$

Proof. Theorem 4.1 implies s^* is an OSS and $U^K(W, s^*, S_0) < \infty$, for $K \geqslant 0$. Fix $s \in S_0$, $K \geqslant 0$, and $N \geqslant \max\{N(\pi, s), K\}$. Clearly, $V_\pi^N(s) = V_\pi^{N(\pi,s)}(s) + [N - N(\pi, s)]\, c_f(s^*)$. Hence $V_\pi^N(s) - V_*^N(s) = [V_\pi^{N(\pi,s)}(s) - N(\pi, s)\, c_f(s^*)] + [N c_f(s^*) - V_*^N(s)]$. This, $W(s) > -\infty$, and Theorem 4.1 imply (5.1). Let $N \to \infty$ in (5.1) to get (5.2). ▮

COROLLARY 5.1. *Under the hypotheses of Theorem 5.1, π is average-cost optimal on S_0, so*

$$c_f(s^*) = \min_{s' \in ACS(S_0) \cap S_f} c_f(s'). \tag{5.3}$$

Note (5.3) holds and $a^* \in A_f(s^*)$ when (a^*, s^*) satisfies (3.3). This suggests the following approach to finding a π satisfying Theorem 5.1. First, solve (3.3) for the target state s^* and stabilizing action a^*. Second, construct a guidance rule that directs the system from some subset S_0 to s^*. According to Theorem 5.1, π must have finite opportunity cost, provided that some $W \in E(s^*)$ is finite and bounded above $\mathrm{ACS}(S_0)$.

Recall from Sect. 3.3 that when (a^*, s^*, w^*) is an LSP, (a^*, s^*) satisfies (3.3) and $w^*_{s^*}(s) \equiv w^*s - w^*s^*$ $(s \in S)$ is excessive at s^* and bounded on S. Under mild conditions, LSPs exist for the *growth models* of [10, 17] and the *fractional flow model* of [9, 13, 14]. Both models have convex S. For the growth models, one can obtain LSPs after solving a *convex programming problem* (c is convex, t is affine, D is convex); Ref. [10], pp. 191–198, describes how to construct a guidance rule under general assumptions, where S_0 contains almost all of S. For the flow models, one loses convexity (both c and t can be nonconvex, A can be finite); however, one can compute LSPs using algorithms like ones for n-state *Markov decision processes*; Ref. [13, Theorem 6.1(c)] shows how to construct a guidance rule where $S_0 = S$ under general assumptions which hold for [9].

The previous models have multidimensional state spaces. Using π that satisfy Theorem 5.1 as *first-order approximations* can reduce computations dramatically. Sections 6 through 8 examine the problem of making *second-order improvements* that reduce the transient costs of such π; Ref. [16] applies these results to the flow model of [9].

6. Second-Order Improvements

Under the hypotheses of Theorem 5.1, one can separate making second-order improvements on π into two subproblems. The first concerns the cost of the guidance rule directing the system to the target s^*, the second, the cost of staying at s^*. Theorem 6.1 justifies this decomposition and provides bounds. Sections 7 and 8 study the two subproblems in detail.

THEOREM 6.1. *Let π, S_0, s^*, and W satisfy the hypotheses of Theorem 5.1.*

(a) *$W^{\mathrm{G}}_{s^*}$ and $V^{\infty}_{s^*}$ are excessive at s^* and finite on S_0.*

(b) *For $s \in S_0$, one can express $OC(\pi, s)$ uniquely as the sum of three nonnegative terms,*

$$OC(\pi, s) = O_{\mathrm{G}}(\pi, s) + O(s, s^*) + OC(s^*), \tag{6.1}$$

such that

$$\inf_{\hat{\pi} \in \Omega(s^*, S_0)} O_{\mathrm{G}}(\hat{\pi}, s) = 0 \qquad \text{and} \qquad O(s^*, s^*) = 0, \tag{6.2}$$

viz.,

$$O_G(\pi, s) = V_\pi^{N(\pi,s)}(s) - N(\pi, s)\, c_f(s^*) - W_{s^*}^G(s), \tag{6.3}$$

$$O(s, s^*) = W_{s^*}^G(s) - V_{s^*}^\infty(s). \tag{6.4}$$

(c) $$O_G(\pi, s) \leqslant V_\pi^{N(\pi,s)}(s) - N(\pi, s)\, c_f(s^*) - W(s) < \infty, \quad \textit{for} \quad s \in S_0. \tag{6.5}$$

Furthermore, the left-hand inequality holds as equality at $s = s^*$.

(d) *If* $s \in S_0 \cap ACS(s^*)$, *then for* $M \geqslant 1$,

$$O_G(\pi, s) + O(s, s^*) \leqslant V_\pi^{N(\pi,s)}(s) + V^M(s^*, s) - (N(\pi, s) + M)\, c_f(s^*). \tag{6.6}$$

Think of the first term in (6.1), $O_G(\pi, s)$, as the penalty associated with the guidance rule. (The other terms do not depend on how the system moves to s^*.) A policy $\hat{\pi}$ minimizes $OC(\hat{\pi}, s)$ over $\Omega(s^*, S_0)$ if $O_G(\hat{\pi}, s) = 0$. Section 7 examines the problem of finding $\hat{\pi} \in \Omega(s^*, S_0)$ that drive $O_G(\hat{\pi}, s)$ towards 0. The second term, $O(s, s^*)$, is the penalty for paying a visit to s^* when the initial state is s. (Note that $O(s^*, s^*) = 0$.) There is no attempt to save this cost by selecting a better s^* (see Remark 6.1). Since $O_G(\pi, s)$ is nonnegative, (6.6) provides an upper bound on $O(s, s^*)$. The third term, $OC(s^*)$, is the opportunity cost of keeping the system at s^*. Theorem 4.1(c) bounds this. Think of $OC(s^*)$ as the *tail* penalty. When N is finite, one can reduce the total N-stage costs by having the system quit s^* towards the end of the horizon and switch to an optimal policy. By (4.3), the saving *can* approach $OC(s^*)$ when N is large. Section 8 covers this.

Remark 6.1. Some comments on the problem of selecting a better target than s^*. By Corollary 5.1, one cannot improve on the opportunity cost of π by changing the target when (5.3) uniquely determines s^*, e.g., when s^* is the only state satisfying (3.3). On the other hand, if another $s' \in ACS(S_0) \cap S_f$ were to achieve the minimum in (5.3), it might also satisfy $O(s, s') + OC(s') < O(s, s^*) + OC(s^*)$ for some $s \in S_0$, and it would be possible to improve the opportunity cost at s by selecting target s' (given also appropriate changes in the guidance rule). In principle, one could require of s^* and S_0 that s^* minimize $O(s, s^*) + OC(s^*)$ for each $s \in S_0$. We do not do this.

We now turn to the proof of Theorem 6.1, which requires the following lemma.

LEMMA 6.1. *Let* π *be an SSP with domain* S_0 *and target state* s^*, *and let* $s \in S_0$. *Then* $V_{s^*}^\infty(s) < \infty$ *and*

$$OC(\pi, s) = [V_\pi^{N(\pi,s)}(s) - N(\pi,s)\, c_f(s^*)] - V_{s^*}^\infty(s) + OC(s^*). \tag{6.7}$$

Hence π has finite opportunity cost if and only if s^* is an OSS and $|V_{s^*}^{\infty}(s)| < \infty$, $s \in S_0$.

Proof. Fix $s \in S_0$ and $N \geqslant N(\pi, s)$. Now, $V_\pi^N(s) = V_\pi^{N(\pi,s)}(s) + [N - N(\pi, s)]\, c_f(s^*)$. Hence, $V_\pi^N(s) - V_*^N(s) = [V_\pi^{N(\pi,s)}(s) - N(\pi, s)\, c_f(s^*)] - [V_*^N(s) - V_*^N(s^*)] + [Nc_f(s^*) - V_*^N(s^*)]$. Since $s^* \in \mathrm{ACS}(s)$, $[V_*^N(s) - V_*^N(s^*)]$ is bounded above in N, so (3.11) implies $V_{s^*}^{\infty}(s) < \infty$. This, (2.1), and (3.1) imply (6.7). The rest follows from (6.7) and the definition of OSS. ∎

Proof of Theorem 6.1. For this proof, assume $s \in S_0$. Also, let $\Omega = \Omega(s^*, S_0)$. Theorem 5.1 and Lemma 6.1 imply s^* is an OSS and $V_{s^*}^{\infty}(s)$ is finite. By Lemma 3.1(a, b, and d), $W_{s^*}^{\mathrm{G}}$ and $V_{s^*}^{\infty}$ are excessive at s^* and $W_{s^*}^{\mathrm{G}}(s)$ is finite, giving as (a). For (b), if $\mathrm{OC}(\pi, s)$ is the sum of three terms where the first two satisfy (6.2), then (6.7) implies the third must be $\mathrm{OC}(s^*)$. Furthermore, this, (6.1), and (6.2) imply $O(s, s^*) = \inf_{\hat{\pi} \in \Omega} \mathrm{OC}(\hat{\pi}, s) - \mathrm{OC}(s^*)$ and $O_{\mathrm{G}}(\pi, s) = \mathrm{OC}(\pi, s) - \inf_{\hat{\pi} \in \Omega} \mathrm{OC}(\hat{\pi}, s)$, proving uniqueness. Hence, assume (6.3) and (6.4). Theorem 5.1 and (6.7) then imply (6.1). Using (6.3), (3.8), and Lemma 3.1(c), one can easily prove (6.2), which also gives us $O_{\mathrm{G}}(\pi, s) \geqslant 0$. Now (6.4), (a), and Lemma 3.1(b) imply $O(s, s^*) \geqslant 0$. Finally, (3.1) ensures that $\mathrm{OC}(s^*) \geqslant 0$, finishing (b). For (c), Lemma 3.1(b) implies $W(s) \leqslant W_{s^*}^{\mathrm{G}}(s)$. This, (a), and (6.3), prove (6.5). Also $W(s^*) = W_{s^*}^{\mathrm{G}}(s^*) = 0$ ensure equality at $s = s^*$. For (d), add (6.3) and (6.4) to obtain, $O_{\mathrm{G}}(\pi, s) + O(s, s^*) = V_\pi^{N(\pi,s)}(s) - N(\pi, s)\, c_f(s^*) - V_{s^*}^{\infty}(s)$. Lemma 3.1(a, b, and c) implies $-V_{s^*}^{\infty}(s) \leqslant -W_{s^*}^{\mathrm{L}}(s) \leqslant V^M(s^*, s) - Mc_f(s^*)$, for $M \geqslant 1$. This and the previous equality prove (6.6). ∎

7. Guidance-Rule Improvements

Throughout this section assume unless stated otherwise that the following holds.

Assumption 7.1. (i) S_0 is a subset of S which contains the state $s^* \in S_f$; (ii) s^* is accessible from each $s \in S_0$; (iii) $W \in E(s^*)$ is finite and bounded above on $\mathrm{ACS}(S_0)$.

Define $\Omega = \Omega(s^*, S_0)$. By (i) and (ii), $\Omega \neq \varnothing$. By Theorem 5.1, the opportunity cost of every SSP in Ω is finite. This section considers the problem of finding an SSP in Ω with minimum opportunity cost. After transforming the costs, we show this is equivalent to finding a *shortest route* from each $s \in S_0$ to s^* in a network with *nodes* S and *nonnegative arc lengths.* For finite S, the shortest route problem has an optimal solution, which one can determine using Dijkstra's algorithm and its extensions when S is not too large. Exact solutions, however, need not exist for infinite S and are often impossible to obtain for finite S whose dimension n exceeds three ("curse of dimensionality"). The latter part of this section deals with the problem of finding approximate solutions for large S.

Given $s \in S_0$, call π *optimal among* Ω at s if $\pi \in \Omega$ and $\mathrm{OC}(\pi, s) \leqslant \mathrm{OC}(\hat{\pi}, s)$, for $\hat{\pi} \in \Omega$, or equivalently, $O_G(\pi, s) = 0$ (See Theorem 6.1(b)). Call $\pi \in \Omega$ optimal among Ω if the latter holds for all $s \in S_0$. Theorem 7.2 proves such π exist provided S is finite. Of course, given $\varepsilon > 0$ and $s \in S_0$, there always exist π that are ε-optimal among Ω at s, i.e., $\pi \in \Omega$ with $\mathrm{O_G}(\pi, s) \leqslant \varepsilon$. Such policies are important for the applications cited in Section 5, where S is multidimensional and convex.

The bound on $O_G(\pi, s)$ in Theorem 6.1(c) can help one to decide how far to proceed when computing approximations. Next, we introduce a transformation which incorporates this bound in the costs (see Theorem 7.1) and makes them nonnegative. Let D_W denote the set of $(a, s) \in D$ with both $W(s)$ and $W(t(a, s))$ finite. (If $W = w^*_{s^*}$ as in (3.6), then $D_W = D$.) Let

$$\bar{c}(a, s) = \begin{cases} \infty, & \text{for } (a, s) \in D - D_W, \\ c(a, s) - c_f(s^*) + W(t(a, s)) - W(s), & \text{for } (a, s) \in D_W. \end{cases} \tag{7.1}$$

By (3.5), $\bar{c}$ is nonnegative. Lemma 7.1 describes other properties of $\bar{c}$. For $s \in S$, let

$$z(s) = \inf\left\{ \sum_{j=1}^{M} \bar{c}(a_j, s_j) \colon 1 \leqslant M < \infty \text{ and } \langle (a_j, s_j) \rangle_{j=1}^{M} \in T(s, s^*) \right\}, \tag{7.2}$$

where

$$T(s, s^*) = \bigcup_{M=1}^{\infty} T_M(s, s^*) \tag{7.3}$$

(see (3.7)); $T(s, s^*)$ contains all feasible, state-action paths from s to s^*. Note that $z(s) = \infty$ if $T(s, s^*) = \varnothing$. Of course, Assumption 7.1 ensures that $T(s, s^*) \neq \varnothing$ when $s \in S_0$.

LEMMA 7.1. *Suppose s^* is an OSS, $W \in E(s^*)$, $s \in S$, and $W(s) > -\infty$:*

(a) $$\sum_{j=1}^{M} \bar{c}(a_j, s_j) = \sum_{j=1}^{M} c(a_j, s_j) - Mc_f(s^*) - W(s),$$

for $\langle (a_j, s_j) \rangle_{j=1}^{M} \in T(s, s^*)$. (7.4)

(b) $$z(s) = W^{G}_{s^*}(s) - W(s). \tag{7.5}$$

Proof. Suppose $\langle(a_j, s_j)\rangle_{j=1}^M \in T(s, s^*)$. Then, $s_1 = s$, $s_{j+1} = t(a_j, s_j)$, $1 \leqslant j < M$, and $s^* = t(a_M, s_M)$. Using (7.1), it is easy to verify (7.4), assuming $|W(s_j)| < \infty$, for $1 \leqslant j \leqslant M$. Fix $1 \leqslant j \leqslant M$. By hypothesis, $W(s_1) > -\infty$. By (3.5) and an induction argument, $W(s_j) > -\infty$. Lemma 3.1(b and d) and $s^* \in \text{ACS}(s_j)$ imply $W(s_j) \leqslant W_{s^*}^{\text{G}}(s_j) < \infty$, finishing (a). Using (a), the definition of $z(s)$, (3.8), Lemma 3.1(c), and routine arguments, one can prove (7.5). ∎

The next theorem follows immediately from Lemma 7.1 and Theorem 6.1(b and c).

THEOREM 7.1. *Suppose Assumption* 7.1 *holds and* $s \in S_0$. *If* $\pi \in \Omega$ *uses* $\langle(a_j, s_j)\rangle_{j=1}^M \in T(s, s^*)$ *to guide the system from* s *to* s^*, *then* $\sum_{j=1}^M \bar{c}(a_j, s_j)$ *coincides with the upper bound on* $O_{\text{G}}(\pi, s)$ *in* (6.5) *and* $\sum_{j=1}^M \bar{c}(a_j, s_j) = O_{\text{G}}(\pi, s) + z(s)$.

By Theorem 7.1, constructing a π optimal among Ω means finding for each $s \in S_0$, a path in $T(s, s^*)$ attaining the infimum in (7.2). This reduces to finding a shortest route from each $s \in S_0$ to s^* is a network with *nodes* S, *directed arcs* $S_D \equiv \{(s, s'):(a, s) \in D$ and $t(a, s) = s'\}$, and *nonnegative length* $l(s, s') \equiv \min\{\bar{c}(a, s): a \in A(s)$ and $t(a, s) = s'\}$. (That is, $l(s, s') \equiv \min\{c(a, s): a \in A(s)$ and $t(a, s) = s'\} - c_f(s^*) + W(s') - W(s)$.) When S is finite, there exists a shortest acyclic route. Since the lengths are nonnegative, this must be a shortest route. Arguing along these lines, one can prove the following.

THEOREM 7.2. *Suppose S is finite and Assumptions* 7.1 *holds. Then there exists an SSP π^* that is optimal among Ω. Furthermore, there exists a π^* that is also stationary.*

Two examples appear below. The first shows that Theorem 7.2 does not extend to infinite S; the second shows that one needs a transformation like (7.1) to get an equivalent shortest route problem with nodes S and nonnegative arc lengths—even when c is nonnegative.

EXAMPLE 7.1. Let $S = \{1/2^n: n = 0, 1, ...\} \cup \{0\}$, $A = \{0, 1/2\}$, $D = A \times S$, $c(a, s) \equiv (1 - 2a)s$, and $t(a, s) \equiv as$. Clearly, $(a^*, s^*, w^*) = (1/2, 0, 0)$ is an LSP. Also, one can move from any s to $s^* = 0$ with $a = 0$. Hence, s^*, $S_0 = S$, and $W(s) \equiv 0$ meet our requirements. Let $s \in S$. By choosing a^* for M stages, one can move from s to $(1/2^M)s$ at *zero* cost. Since $a = 0$ moves $(1/2^M)s$ to s^* at cost $(1/2^M)s$, we have $z(s) \equiv 0$. But $c(a, s) > 0$ unless $s = s^*$ or $a = a^*$. Thus, no element of $T(s, s^*)$ can achieve the infimum in (7.2) when $s \neq s^*$.

EXAMPLE 7.2. Let $S=A=\{0,1,2\}$, $D=\{(0,0),(0,1),(2,1),(0,2)\}$, $c(0,0)=2$, $c(0,1)=10$, $c(2,1)=0$, $c(0,2)=11$, and $t(a,s)\equiv a$. Clearly, $c(a,s)+2a\geqslant 2s+c(0,0)$, for $(a,s)\in D$, so $(a^*,s^*,w^*)=(0,0,2)$ is an LSP. For $W(s)\equiv 2s$: $\bar{c}(0,0)=0$, $\bar{c}(0,1)=6$, $\bar{c}(2,1)=0$, and $\bar{c}(0,2)=5$. There are two routes from 1 to $s^*=0$: $1\rightarrow 0$ and $1\rightarrow 2\rightarrow 0$. Given arc lengths $\bar{c}(a,s)$, the former has length 6 and the latter length 5. Given arc lengths $c(a,s)$, however, the order is reversed, $1\rightarrow 0$ having length 10 and $1\rightarrow 2\rightarrow 0$ having length 11.

The nonnegativity of the arc lengths allows one to use Dijkstra's algorithm and its extensions to solve our shortest route problem when S is finite. (See Denardo [8, Chap. 2] or Ahuja *et al.* [3]; Ref. [3, pp. 334–38] discusses the shortest route literature). Unfortunately, the computing time and computer storage requirements can become excessive when the dimension $n\geqslant 4$, and one may have to settle for an approximate solution. Morin [22] surveys the available computational techniques.

One approach, used in [16], is to find for a finite sequence of M a minimum cost M-stage path from a given state $s\in S_0$ to s^*. The advantage is that given certain smoothness assumptions, one can employ nonlinear programming and optimal control methods to solve the M-stage problems (see the discussion below). For $M\geqslant 1$ and $s\in S$, define

$$z_M(s)=\begin{cases}\min\left\{\sum_{j=1}^{M}\bar{c}(a_j,s_j)\colon \langle(a_j,s_j)\rangle_{j=1}^{M}\in T_M(s,s^*)\right\}, & \\ \qquad\text{if}\quad T_M(s,s^*)\neq\varnothing & \\ \infty, \qquad\text{if}\quad T_M(s,s^*)=\varnothing. & \end{cases} \tag{7.6}$$

Note that according to Lemma 7.1, $\sum_{j=1}^{M}\bar{c}(a_j,s_j)$ and $\sum_{j=1}^{M}c(a_j,s_j)$ differ by $[Mc_f(s^*)+W(s)]$ when $\langle(a_j,s_j)\rangle_{j=1}^{M}\in T_M(s,s^*)$. This, the lower semicontinuity of c, and compactness justify the "min" in (7.6). Incidentally, for the purpose of finding a path in $T_M(s,s^*)$ that attains the minimum in (7.6), the objective function can be either $\sum_{j=1}^{M}\bar{c}(a_j,s_j)$ or $\sum_{j=1}^{M}c(a_j,s_j)$. Example 7.2 above proves that the analogous statement about (7.2) is false.

For $M\geqslant 1$ and $s\in S$, one can extend any path $\langle(a_j,s_j)\rangle_{j=1}^{M}$ in $T_M(s,s^*)$ to a path in $T_{M+1}(s,s^*)$ with the same $\bar{c}$-cost: Simply let $(a_{M+1},s_{M+1})=(a^*,s^*)$, where $a^*\in A_f(s^*)$. Clearly $\bar{c}(a^*,s^*)=0$, which implies $\sum_{j=1}^{M+1}\bar{c}(a_j,s_j)=\sum_{j=1}^{M}\bar{c}(a_j,s_j)$. One can thus construct a feasible solution to an $(M+1)$-stage problem from an optimal solution to an M-stage one. (Determining a feasible solution is the first step in many nonlinear programming and optimal control algorithms.) Incidentally, this construction implies

$$z_M(s)\downarrow z(s), \qquad \text{for}\quad s\in S. \tag{7.7}$$

If π and s satisfy Theorem 7.1, then $O_G(\pi, s) = z_M(s) - z(s)$ and, furthermore, $z_M(s)$ is an upper bound on $O_G(\pi, s)$. (By Lemma 7.1, how tight this bound is depends on how close $W_{s^*}^G(s)$ is to $W(s)$.) One can use $z_M(s)$ as a basis for deciding when to stop the computations.

Under appropriate smoothness assumptions, (7.6) becomes a classical discrete-time, deterministic, constrained optimal control problem, and one can attempt to solve it using nonlinear programming and optimal control methods [4, 6, 24, 26], which include methods of feasible directions [25], general reduced gradient methods [1, 2], and differential dynamic programming [20, 23, 27]. These methods are capable of dealing with sizable problems. However, they need not produce a global optimum, unless one imposes convexity assumptions (which, as discussed in Section 5, hold for the growth model of [10, 17], but not for the flow model of [9, 13, 14]). This disadvantage may not always be serious, since $z_M(s)$ provides an upper bound on how much better a different solution can be.

8. Tail Improvements

Suppose the horizon is finite and equals N and π satisfies Theorem 5.1. By allowing the system to leave the target s^* and follow an M-stage optimal policy in the last M stages, one can reduce the total N-stage costs under π by $Mc_f(s^*) - V_*^M(s^*)$. By Theorem 4.1(c), the cost savings approach $\mathrm{OC}(s^*)$ for large M. For this tail improvement to be practical, M should be small relative to N. Computing $V_*^1(s^*)$ is often easy. The bound on $\mathrm{OC}(s^*)$ in (4.3) contains information that is helpful when choosing M.

Under appropriate smoothness assumptions, one can use nonlinear programming and optimal control methods when calculating $V_*^M(s^*)$ (see the last paragraph of the previous section). Applying nonlinear programming techniques, [16] solves

$$V_*^M(s^*) \equiv \left\{ \sum_{j=1}^{M} c(a_j, s_j) : \langle (a_j, s_j) \rangle_{j=1}^{M} \in \bigcup_{s \in S} T_M(s^*, s)) \right\} \tag{8.1}$$

for small M. To reduce the work, that paper constructs an initial feasible solution $\langle (a_j', s_j') \rangle_{j=1}^{M}$ to an M-stage problem from an optimal solution $\langle (a_j, s_j) \rangle_{j=1}^{M-1}$ to an $(M-1)$-stage problem, i.e., $(a_j', s_j') = (a_{j-1}, s_{j-1})$, $2 \leqslant j \leqslant M$, and $(a_1', s_1') = (a^*, s^*)$, where $a^* \in A_f(s^*)$. The next theorem follows easily from Theorem 5.1.

Theorem 8.1. *Let* π, S_0, s^*, *and* W *satisfy the hypotheses of Theorem* 5.1. *Let* $s \in S_0$, $0 \leqslant K \leqslant N$, *and* $1 \leqslant M \leqslant N - N(\pi, s)$. *Let* π' *be a* *nonstationary policy that selects the same decisions as π in stages 1 through $N-M$ and then switches to an optimal M-stage policy at stage $N-M+1$. Then the following holds:*

$$V_{\pi}^{N}(s)-V_{*}^{N}(s)\leqslant V_{\pi}^{N(\pi,s)}(s)-(N(\pi,s)+M)\,c_f(s^*)$$
$$+V_{*}^{M}(s^*)-W(s)+U^{K}(W,s^*,S_0). \qquad (8.2)$$

Remark 7.1. Inequality (4.2) provides a lower bound on $V_{*}^{N}(s)$. If this lower bound is positive, then its reciprocal multiplied by the upper bound in (8.2) is an upper bound on the *relative error*, $|[V_{\pi}^{N}(s)-V_{*}^{N}(s)]/V_{*}^{N}(s)|$. Note that this bound is valid even when one chooses not to make tail improvements, i.e., when $M=0$.

REFERENCES

1. J. ABADIE AND J. CARPENTER, Generalization of the Wolfe reduced gradient method to the case of nonlinear constraints, *in* "Optimization" (R. Fletcher, Ed.), pp. 37–47, Academic Press, New York, 1969.
2. J. ABADIE, Application of the GRG algorithm to nonlinear control problems, *in* "Integer and Nonlinear Programming" (J. Abadie, Ed.), North-Holland, Amsterdam, 1970.
3. R. AHUJA, T. MAGNANTI, AND J. ORLIN, Network flows, *in* "Handbooks in Operations Research and Management Science" (G. Nemhauser, A. Rinnoy Kan, and M. Todd, Eds.), Vol. I, pp. 211–369, North-Holland, New York, 1989.
4. M. BAZARRA AND M. SHETTY, "Nonlinear Programming: Theory and Algorithms," Wiley, New York, 1979.
5. E. BURMEISTER AND A. DOBELL, "Mathematical Theories of Economic Growth," McMillan Co., New York, 1970.
6. M. CANON, C. CULLUM, AND E. POLAK, "Theory of Optimal Control and Mathematical Programming," McGraw–Hill, New York, 1970.
7. D. CARLSON AND A. HAURIE, "Infinite Horizon Optimal Control," Springer-Verlag, New York/Berlin, 1987.
8. E. DENARDO, "Dynamic Programming," Prentice–Hall, Englewood Cliffs, New Jersey, 1982.
9. J. FLYNN, Retaining productive units: a dynamic programming model with a good steady state solution, *Management Sci.* **21** (1975), 754–764.
10. J. FLYNN, Steady state policies, dynamic programming, and optimal economic growth, *in* "Dynamic Programming and Its Applications" (M. Puterman, Ed.), pp. 173–200, Academic Press, New York, 1978.
11. J. FLYNN, Steady state policies for deterministic dynamic programs, *SIAM J. Appl. Math.* **37** (1979), 128–147.
12. J. FLYNN, On Optimality criteria for dynamic programs with long finite horizons, *J. Math. Anal. Appl.* **76** (1980), 202–208.
13. J. FLYNN, A dynamic programming model of fractional flows with application to maintenance and replacement problems, *Math. Oper. Res.* **6** (1981), 405–419.
14. J. FLYNN, An algorithm for a dynamic programming model of fractional flows, *Math. Oper. Res.* **13** (1988), 711–722.
15. J. FLYNN, Optimal steady states, excessive functions, and deterministic dynamic programs, *J. Math. Anal. Appl.* **144** (1989), 586–594.

16. J. FLYNN, "Computing Efficient Steady State Policies for Deterministic Dynamic Programs. II," submitted, *J. Math. Anal. Appl.*, 1991. Working paper, 1990.
17. D. GALE, Optimal development in a multisector economy, *Rev. Econom. Stud.* **34** (1967), 1–18.
18. K. HINDERER, Estimates for finite-stage dynamic programs, *J. Math. Anal. Appl.* **55** (1976), 207–238.
19. D. JAQUETTE AND G. NELSON, Optimal wage rates and force composition in military manpower planning, *Management Sci.* **23** (1976), 257–266.
20. D. JACOBSON AND D. MAYNE. "Differential Dynamic Programming," Elsevier, New York, 1970.
21. J. LIEBMAN, L. LASDON, L. SCHRAGE AND A. WAREN, "Modeling and Optimization with GINO," Scientific Press, Palo Alto, 1986.
22. T. MORIN, Computational advances in dynamic programming, *in* "Dynamic Programming and Its Applications" (M. Puterman, Ed.), pp. 53–90, Academic Press, New York, 1978.
23. K. OHNO, A new approach to differential dynamic programming for discrete time systems, *IEEE Trans. Automat. Control* **23** (1978), 37–47.
24. E. POLAK, "Computational Methods in Optimization," Academic Press, New York, 1971.
25. E. POLAK, A survey of methods of feasible directions for the solution of optimal control problems, *IEEE Trans. Automat. Control* **17** (1972), 591–596.
26. E. POLAK, An historical survey of computational methods in optimal control, *SIAM Rev.* **15** (1973), 553–584.
27. S. YAKOWITZ, The stagewise Kuhn–Tucker condition and differential dynamic programming, *IEEE Trans. Automat. Control* **31** (1986), 25–30.