JOURNAL OF MATHEMATICAL ANALYSIS AND APPLICATIONS **86**, 665–674 (1982)

Monotonicity and the Principle of Optimality

THOMAS L. MORIN

School of Industrial Engineering, Purdue University, West Lafayette, Indiana 47907

Submitted by A. O. Esogbue

This paper explores some of the theoretical and algorithmic implications of the fact that the Monotonicity Assumption does not ensure either the validity of the Principle of Optimality or the discovery of all optimal solutions in finite dynamic programs, even though it is sufficient to ensure the validity of the functional equations. A slightly stronger assumption is introduced to resolve these problems. Our analysis is illustrated with some extremely simple examples.

1. INTRODUCTION

Bellman's [1] characterization of dynamic programming through the use of the Principle of Optimality and Mitten's [12] subsequent introduction of the Monotonicity Assumption as a sufficient condition for the validity of the functional equations provide a basis for the formal study of finite dynamic programs. This research has been further advanced by [2–4, 6–10, 17, 19–22], among others. However, as we shall soon seen, the subtlety of the initial developments can easily elude the unwary.

Bellman's statement [1, p. 83] of the Principle of Optimality is that "an optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision." We will find it convenient to work with the following simple implication of Bellman's Principle of Optimality.

PRINCIPLE OF OPTIMALITY. *An optimal solution must contain optimal (partial) solutions.*

That is, any portion of an optimal solution must also be optimal. Notice that, as pointed out by Porteus [17], these are necessary conditions for optimal *solutions* (and policies[1]), whereas the functional equations are both

[1] Unlike [1, 10] and others, we distinguish *policies* which specify decisions for all states from *solutions* which specify decisions only for those states through which the process evolves.

0022-247X/82/080665-10$02.00/0

necessary and sufficient for the optimal *values*. Therein lies one of the subtle distinctions to be explored in this paper. For example, this observation makes it clear that the common statement that the functional equation is a mathematical transliteration (or translation or statement) of the Principle of Optimality is simply not correct.

Our analysis will be confined to deterministic finite (stage or horizon) dynamic programs in which the set of minimum cost paths (solutions) from the initial state to the set of final states is to be determined.This is a generalization of the prototype shortest path problem for a directed acyclic graph and is introduced in Section 2 using the formalism of [10, 14, 15]. The simple examples of Section 3 demonstrate that the Monotonicity Assumption is *not* sufficient to ensure either the validity of the Principle of Optimality or the discovery of all optimal solutions by dynamic programming. (While these revelations may surprise some readers, the former was apparently known at least as early as 1971 [3]—see also [2, 10, 19, 22].) In Section 4 we introduce a stronger assumption which is sufficient to ensure the validity of both the Principle of Optimality and the functional equations, and specify a broad class of cost functions which satisfy the strong monotonicity assumption. Fortunately, the class includes additive and multiplicative costs (the latter for positive factors). The paper concludes with a discussion in Section 5.

2. Finite Dynamic Programs

A *finite dynamic program* $\mathscr{D}$ is a quadruple (Ω, D, t, h), in which Ω is the finite nonempty state space; D is the finite nonempty set of decisions; $t: A \to \Omega$, where $A \subseteq \Omega \times D$, is the transition mapping;[2] and $h: \mathbf{R} \times A \to \mathbf{R}$, where $\mathbf{R}$ is the set of real numbers, is the cost function. Let $y_0 \in \Omega$ be the initial state, $\Omega_F \subset \Omega$ be the set of final states and assume that return to the initial state is not possible. Then, $t(y', d)$ is the state that is reached when decision $d \in D$ is applied at state $y' \in \Omega$ and $h(\xi, y', d)$ is the cost of reaching state $t(y', d)$ by an initial sequence of decisions (partial solution) which reaches state y' at cost $\xi \in \mathbf{R}$ and is then extended by decision d. Let $\xi_0 \in \mathbf{R}$ be the cost incurred in the initial state $y_0 \in \Omega$. Recall that our objective is to reach Ω_F from y_0 at minimum cost. A solution algorithm for $\mathscr{D}$ will find both this minimum cost f^* and a subset of the set Δ^* of optimal decision sequences (solutions) which achieve this cost. Following [13, 18],

[2] Notice that the domain A of the transition mapping is a subset of $\Omega \times D$. This eliminates consideration of nonsensical decisions. That is, if $D(y') \subseteq D$ is the set of admissible decisions at state y', then $t(y', d)$ would only be defined for $d \in D(y')$ with $D(y') = \varnothing \ \forall y' \in \Omega_F$. Likewise, we let $B \subseteq \Omega \times \Delta$ to eliminate consideration of nonsensical sequences of decisions.

we call an algorithm *strong* if it finds *all* optimal solutions Δ^* and *weak* otherwise.

Before discussing the role of the functional equations in an algorithm for $\mathscr{D}$ it is necessary to define the solution space Δ and to extend the domains of both the transition mapping and the cost function. Let Δ be the set of all finite sequences of individual decisions from $\mathscr{D}$, and let $B \subseteq \Omega \times D$ (see footnote 2). The domains of both the transition mapping and the cost function can be inductively extended from A and $\mathbf{R} \times A$ to B and $\mathbf{R} \times B$, respectively. Then for any decision sequence $\delta = \delta_1 \delta_2 \in \Delta$, $t(y', \delta) = t(t(y', \delta_1), \delta_2)$ is the state that is reached when decision sequence δ is applied at state $y' \in \Omega$ and $h(\xi, y', \delta) = h(h(\xi, y', \delta_1), t(y', \delta_1), \delta_2)$ is the cost of reaching state $t(y', \delta)$ by a decision sequence which reaches y' at cost ξ and is then extended by δ.

We will also find it convenient to define some subsets of Δ. For any state $y' \in \Omega$, let $\Delta(y')$ denote the set of decision sequences which cause a transition from state y_0 to y', i.e., $\Delta(y') = \{\delta \in \Delta \mid t(y_0, \delta) = y'\}$. Also define $\Delta(\Omega_F) = \bigcup_{y' \in \Omega_F} \Delta(y')$. For $y' \in \{\Omega - \Omega_F\}$, $\Delta(y')$ is the set of *feasible* (*partial*) *solutions* for state y' and $\Delta(\Omega_F)$ is the set of *feasible solutions* for $\mathscr{D}$. For each $y' \in \Omega$, let $f(y')$ denote the minimum cost[3] of reaching y', i.e., $f(y') = \min\{h(\xi_0, y_0, \delta) \mid \delta \in \Delta(y')\}$, and let $f^* = \min\{f(y') \mid y' \in \Omega_F\}$. Then for $y' \in \{\Omega - \Omega_F\}$, $\Delta^*(y') = \{\delta \in \Delta(y') \mid h(\xi_0, y_0, \delta) = f(y')\}$ is the set of *optimal* (*partial*) *solutions* for state y' and $\Delta^* = \{\delta \in \Delta(\Omega_F) \mid h(\xi_0, y_0, \delta) = f^*\}$ is the set of *optimal solutions* for $\mathscr{D}$.

Following [10], we can derive the following *functional equations* of dynamic programming

$$f(y_0) = \xi_0,$$

and

$$f(y) = \min_{(y', d) \in \Omega \times D} \{h(f(y'), y', d) \mid t(y', d) = y\} \qquad y \in \{\Omega - \{y_0\}\}. \quad (1)$$

In the following section we will discuss conditions on h to ensure the validity of the functional equations. We will see that these are *not* the same as conditions which ensure both that the functional equations yield a strong algorithm (which finds all optimal solutions) for $\mathscr{D}$ or that the following Principle of Optimality is valid.

Principle of Optimality. *If* $\delta = \delta_1 \delta_2 \in \Delta^*$, *where* $\delta_1 \in \Delta(y)$, *then* $\delta_1 \in \Delta^*(y)$.

[3] We will sometimes refer to $\{f(y) \mid y \in \Omega\}$ as the set of *optimal values*.

3. Monotonicity, Functional Equations and the Principle of Optimality

Mitten [12] appears to have been the first to present a condition on the cost function which suffices to ensure the validity of the functional equations of dynamic programming. This monotonicity assumption (MA) can be restated for finite dynamic programs as follows:

Assumption 1 (MA). *For any* $\xi_1, \xi_2 \in \mathbf{R}$, $\delta \in \Delta$ *and* $y' \in \Omega$

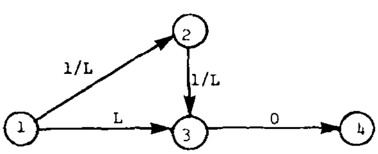
$$\xi_1 \leqslant \xi_2 \Rightarrow h(\xi_1, y', \delta) \leqslant h(\xi_2, y', \delta).$$

We will term any finite dynamic program in which h satisfies the MA *monotone*. It is easily established [10, Theorem 1] that if $\mathscr{D}$ is monotone, then the functional equations are valid. However, the following simple example shows that the MA is not sufficient to ensure either the validity of the Principle of Optimality or that the functional equations yield a strong dynamic programming algorithm which finds all optimal solutions.

Example 1. Given a directed acyclic graph $\mathscr{G} = (V, E)$ in which the length of a path is equal to the product of the arc lengths on the path, find the set $\{\tau^*\}$ of shortest paths from node $1 \in V = \{1, 2, \ldots, N\}$ to node $N \in V$. That is, we wish to find $\{\tau^*\}$ so as to

$$\min_{\tau \in T} \prod_{(i,j) \in \tau} c_{ij},$$

where T is the set of all feasible paths from node 1 to node N and $c_{ij} \geqslant 0$ is the length of arc $(i, j) \in E$. Here $\Omega = V$, $D = E$, $t(i, (i, j)) = j \ \forall\, i \in V$ and $(i, j) \in E$, $h(\xi, i, (i, j)) = \xi \cdot c_{ij} \ \forall\, \xi \in \mathbf{R}$, $i \in V$, and $(i, j) \in E$, with $\xi_0 = 1$ (the ficticious length of node 1). Consider the simple graph



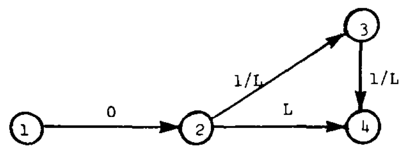
where L is some arbitrarily large real number. The MA is satisfied yet one of the optimal paths (solutions) $\delta = \{(1, 3), (3, 4)\}$ contains the suboptimal (in fact, disastrous!) subpath (partial solution) $\{(1, 3)\}$. That is, $\{(1, 3), (3, 4)\} \in \Delta^*$ but $\{(1, 3)\} \notin \Delta^*(3) = (\{(1, 2), (2, 3)\})$.

Thus, the Principle of Optimality is violated. However, the functional equations

$$f(1) = 1,$$
$$f(j) = \min\{c_{ij} \cdot f(i) \mid (i, j) \in E\}, \qquad j = 2,..., N,$$

are valid and these values are achieved by the other optimal path (solution) $\{(1, 2), (2, 3), (3, 4)\}$ which also has length 0. Clearly, the (valid) functional equations cannot be termed mathematical transliterations (or translations or statements) of the (invalid) Principle of Optimality.

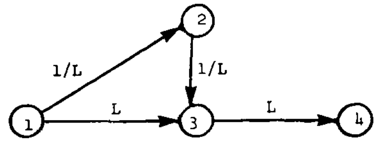
To see that Bellman's original statement of the Principle of Optimality also does not hold, simply flip the graph around, yielding



Taking 1 as the initial state and $(1, 2)$ as the initial decision results in state 2. Clearly the choice of $(2, 4)$ as the remaining decision is disastrously suboptimal with regard to state 2 yet $\{(1, 2),(2, 4)\}$ is an optimal path (solution) and $\{(1, 2), (2, 4), (3, 4), \phi\}$ is an optimal policy. ∎

The problem is that the MA admits the pathological possibility of having $\xi_1 < \xi_2$ but $h(\xi_1, y', \delta) = h(\xi_2, y', \delta)$, i.e., h can be monotone and not (strictly) isotone [11]. This allows for the possibility of an optimal solution containing suboptimal partial solutions thereby contradicting the Principle of Optimality and (contrary to [8, Theorem 6.1] as noted in [9]) making it impossible to find such solutions with the (weak) dynamic programming algorithm. Other cost functions which exhibit this phenomenon involve the binary operations min and max (see [3, Problem 7, p. 27] for an example involving the latter). A simple example follows:

EXAMPLE 2. Consider the problem of finding the set of paths from node $1 \in V = \{1, 2,..., N\}$ to node N in a directed acyclic graph $\mathscr{G} = (V, E)$ for which the maximum arc length is minimized. Here, $h(\xi, i, (i, j)) = \max(\xi, c_{ij})$ and $\xi_0 = 0$. Consider the simple graph



Again the MA is satisfied yet one of the optimal paths (solutions) $\{(1,3), (3,4)\}$ contains the suboptimal subpath (partial solution) $\{(1,3)\}$. ∎

Of course, as pointed out by one of the referees, it is always possible to augment the space of a monotone (or even a nonmonotone) dynamic program so that the Principle of Optimality holds on the augmented state space. That is, appending a binary variable to each state (node) in Example 1, which indicates whether or not an arc cost of zero has occurred on a path to that state, permits the construction of a strong dynamic programming algorithm on the augmented state space which would yield all optimal solutions (paths). However, this would not be computationally attractive, since it involves a doubling of the number of states. Moreover, as we shall see in the next section, such a scheme amounts to nothing more than transforming a monotone dynamic program into a *strictly* monotone dynamic program defined on an augmented state space.

About the only implication the MA has for optimal solutions is to ensure that there exists an optimal solution which contains optimal partial solutions. Similar existence results are termed versions of the Principle of Optimality by Denardo [2, 3], Sniedovich [22], and Sobel [23]—see also [10, Lemma 1]. All such optimal solutions (since there may be more than one) can be reconstructed from the dynamic programming algorithm since they achieve the values determined by the functional equations. That is, the functional equations yield a weak algorithm. However, we can only find *all* optimal solutions with such a weak dynamic programming algorithm if either there is only one solution in Δ^* or all solutions in Δ^* attain the values specified by the functional equations. Moreover, verification of such conditions is a nontrivial task which might even require that we solve $\mathscr{D}$ via some other means.

To ensure the validity of the Principle of Optimality and the subsequent discovery of all optimal solutions with dynamic programming, we need to introduce a slightly stronger assumption.

4. Strictly Monotone Dynamic Programs

To rule out the possibility of having $\xi_1 < \xi_2$ but $h(\xi_1, y', \delta) = h(\xi_2, y', \delta)$ we introduce the following *strict monotonicity assumption.*[4]

Assumption 2 (SMA). *For any* $\xi_1, \xi_2 \in \mathbf{R}$, $\delta \in \Delta$ *and* $y' \in \Omega$

$$\xi_1 < \xi_2 \Rightarrow h(\xi_1, y', \delta) < h(\xi_2, y', \delta)$$

[4] A similar condition is satisfied by the type M loss function introduced in [24].

We will term any finite dynamic program in which h satisfies the SMA *strictly monotone*. The SMA *is* sufficient to ensure the Principle of Optimality as shown below:

THEOREM 1. *If* $\mathscr{D}$ *is strictly monotone and* $\delta = \delta_1 \delta_2 \in \Delta^*$, *where* $\delta_1 \in \Delta(y')$, *then* $\delta_1 \in \Delta^*(y')$.

Proof. Assume to the contrary that $\delta_1 \notin \Delta^*(y')$, i.e., $\exists \hat{\delta} \in \Delta(y')$ such that

$$h(\xi_0, y_0, \hat{\delta}) < h(\xi_0, y_0, \delta_1).$$

Now, since δ_1, $\hat{\delta} \in \Delta(y')$, we have

$$t(t(y_0, \delta_1), \delta_2) = t(t(y_0, \hat{\delta}), \delta_2).$$

Hence,

$$\hat{\delta}\delta_2 \in \Delta(\Omega_F).$$

By the SMA, we have

$$h(\xi_0, y_0, \hat{\delta}) < h(\xi_0, y_0, \delta_1) \Rightarrow h(\xi_0, y_0, \hat{\delta}\delta_2) < h(\xi_0, y_0, \delta_1 \delta_2)$$

which contradicts $\delta_1 \delta_2 \in \Delta^*$ and completes the proof. ∎

Notice that this also implies that if $\mathscr{D}$ is strictly monotone, then optimal partial solutions themselves must also contain optimal partial solutions. That is, if $\delta_1 = \hat{\delta}_1 \delta$, then invoking Theorem 1 again with $\hat{\delta}_1$ and $\delta\delta_2$ in the roles of δ_1 and δ_2, respectively, we obtain that $\hat{\delta}_1 \in \Delta(y'')$ implies $\hat{\delta}_1 \in \Delta^*(y'')$. So, if $\hat{\delta}_1 \delta \in \Delta^*(y')$, where $\hat{\delta}_1 \in \Delta(y'')$, then $\hat{\delta}_1 \in \Delta^*(y'')$.

We could also generalize Theorem 1 in the following manner. For any y', $y \in \Omega$ and $\hat{\delta} \in \Delta$ such that $t(y', \hat{\delta}) = y$ the SMA implies that if $\hat{\delta} \in \delta \in \Delta^*$, then $h(\xi, y', \hat{\delta}) = \min_{\delta'} \{h(\xi, y', \delta') \mid t(y', \delta') = y\}$ for any $\xi \in \mathbf{R}$. That is, if the SMA is satisfied, then an optimal solution is also optimal between any of the states it enters (or any portion of an optimal solution is optimal). This is Denardo's [2, 3] first version of the Principle of Optimality. Notice that, although it includes ours as the special case in which $\hat{\delta} = \delta$, it does not impart additional algorithmic insight.

Since the SMA $\Rightarrow$ MA, invocation of the result [10, Theorem 1] on monotone dynamic programs establishes the validity of the functional equations for strictly monotone dynamic programs.

PROPOSITION 1. *If $\mathscr{D}$ is strictly monotone, then the functional equations* (1) *are valid.*

Following [4], we can specify a broad class of cost functions which satisfy the SMA by introducing the concept of a strictly isotonic, associative symmetric binary operator $\circ$. Notice that the term *strictly* isotonic is used, as opposed to the usual isotonic [11], to emphasize the preservation of *strict* inequalities. That is, we will call $\circ$ strictly isotone if for $a, b, c \in \mathbf{R}$, we have

$$a < b \Rightarrow a \circ c < b \circ c.$$

Notice that this, together with the trichotomy law for real numbers, [5, 2.2.1] implies that $a = b \Rightarrow a \circ c = b \circ c$.

Then it follows that the general separable cost function

$$h(\xi_0, y_0, e) = \xi_0,$$
$$h(\xi, y', \delta) = \xi_0 \circ \pi(y', \delta), \qquad y' \in \{\Omega - \{y_0\}\}, \quad \delta \in \Delta,$$

where ξ_0, $\xi \in \mathbf{R}$, π: $\Omega \times \Delta \to \mathbf{R}$, $e \notin \Delta$ is the null policy, and $\circ$ is any strictly isotonic associative symmetric binary operator, satisfies the SMA.

Fortunately, since $\mathbf{R}$ is an ordered ring[5], addition and multiplication (of positive reals) are strictly isotonic [11, Proposition V.4]. Hence, the most common separable cost functions, the additive and the multiplicative (the latter on $\mathbf{R}_+^* = (0, \infty)$), satisfy the SMA. Consequently, both the Principle of Optimality and the functional equations are valid, allowing us to find all optimal solutions by dynamic programming.

5. DISCUSSION

The problems resulting from the insufficiency of the Monotonicity Assumption to ensure the validity of the Principle of Optimality can be resolved by the introduction of the Strict Monotonicity Assumption. Fortunately, most of the cost functions of interest involve addition or multiplication (of positive reals) and thus satisfy *both* the MA and SMA. In these cases the Principle of Optimality holds and we are assured that all optimal solutions will be found by the (strong) dynamic programming algorithm. This is not always the case for cost functions which satisfy the MA but not the SMA, as we have seen for cost functions involving multiplication of nonnegative reals or the binary operators min and max.

[5] Notice that this is not the case if we allow costs to be unbounded, i.e., addition and multiplication (even for positive reals) are not strictly isotonic on $\bar{\mathbf{R}}$, the extended real line. To see this, simply replace the 0 with $+\infty$ in the first graph of Example 1.

There will still be situations, however, in which the MA is satisfied but the SMA is not where it is possible to find all optimal solutions; namely if either the optimal solution is unique or all optimal solution achieve the values determined by the functional equations (although verification of such situations may be a nontrivial task). Even failing this, we are still assured of finding at least one optimal solution with the (weak) dynamic programming algorithm and any such optimal solution will contain optimal partial solutions. We also indicated that it is always possible, at the cost of considerable additional computations and storage, to transform a monotone (or even a nonmonotone) dynamic program into a strictly monotone dynamic program by appropriately augmenting the state space.

The examples also demonstrated that the functional equations are *not* mathematical transliterations (or translations or statements) of the Principle of Optimality. Instead, they are necessary and sufficient conditions on the optimal values for the states. The functional equations could be viewed as mathematical transliterations of the following Optimality Condition: A set of values is optimal if and only if it has the property that whatever the (initial) state and decision are, the (remaining) value is optimal with regard to the state resulting from that (first) decision.

Finally, we consider the ramifications of our results for stochastic dynamic programs and infinite (stage or horizon) dynamic programs. Our analysis extends immediately to both cases *mutatis mutandis*. In fact [22] presents examples similar to ours. However, many of the issues raised here would be moot if we were primarily interested in the existence and construction of a single (stationary) optimal policy.

Acknowledgments

The author gratefully acknowledges useful discussions with G. B. Hazen, K. S. Nagaraj, D. M. Preklas, and J. J. Talavage and the helpful comments of A. O. Esogbue, T. Ibaraki, and the referees. This research was partially supported by the National Science Foundation under Grant ENG-7614396 and the Department of the Interior under Grant OWRT B-109-IND.

References

1. R. Bellman, "Dynamic Programming," Princeton Univ. Press, Princeton, N.J., 1957.
2. E. V. Denardo, Dynamic programming, *in* "Handbook of Operations Research" (J. J. Moder and S. E. Elmaghrahy, Eds.), pp. 586–606, Van Nostrand–Reinhold, New York, 1978.
3. E. V. Denardo, "Dynamic Programming: Models and Applications," Prentice–Hall, Englewood Cliffs, N.J., 1982.
4. E. V. Denardo and L. G. Mitten, Elements of sequential decision processes, *J. Indust. Engrg.* **18** (1967), 106–112.

5. J. DIEUDONNÉ, "Foundations of Modern Analysis," Academic Press, New York, 1960.
6. S. E. ELMAGHRABY, The concept of "state" in discrete dynamic programming, *J. Math. Anal. Appl.* **29** (1970), 523–557.
7. K. HINDERER, "Foundations of Non-Stationary Dynamic Programming with Discrete Time Parameter," Springer-Verlag, Berlin, 1970.
8. T. IBARAKI, Solvable classes of discrete dynamic programming, *J. Math. Anal. Appl.* **43** (1973), 642–693.
9. T. IBARAKI, "Kumiawase Saitekika no Riron" ("Theory of Combinatorial Optimization"), Institute of Electronics and Communication Engineers of Japan, Tokyo, 1979.
10. R. M. KARP AND M. HELD, Finite-state processes and dynamic programming, *SIAM J. Appl. Math.* **15** (1967), 693–718.
11. S. MACLANE AND G. BIRKHOFF, "Algebra," Macmillan, New York, 1967.
12. L. G. MITTEN, Composition principles for synthesis of optimal multistage processes. *Oper. Res.* **12** (1964), 610–619.
13. L. G. MITTEN AND A. R. WARBURTON, "Implicit enumeration procedures," Working Paper No. **251**, Faculty of Commerce and Business Administration, University of British Columbia, Vanvouver, Canada, 1973.
14. T. L. MORIN, Computational advances in dynamic programming, *in* "Dynamic Programming and Its Applications" (M. L. Puterman, Ed.), pp. 53–90, Academic Press, New York, 1979.
15. T. L. MORIN AND R. E. MARSTEN, Branch-and-bound strategies for dynamic programming, *Oper. Res.* **24** (1976), 611–627.
16. G. L. NEMHAUSER, "Introduction to Dynamic Programming," Wiley, New York, 1966.
17. E. PORTEUS, An informal look at the principle of optimality, *Management Sci.* **21** (1975), 1346–1348.
18. D. M. PREKLAS AND T. L. MORIN, A unifying framework for discrete optimization, presented at the Joint National ORSA/TIMS Meeting, Colorado Springs (1980).
19. M. SNIEDOVICH, "A formal look at the principle of optimality and the dynamic programming algorithm," Research Report 77-WR-1, Department of Civil Engineering, Princeton University, Princeton, N.J., 1977.
20. M. SNIEDOVICH, A new look at dynamic programming, presented at the International Symposium on Extremal Methods and Systems Analysis, University of Texas, Austin, Tex. (1977).
21. M. SNIEDOVICH, Dynamic programming and the principle of optimality: A systematic approach, *Adv. in Water Resources* **1** (1978), 183–190.
22. M. SNIEDOVICH, Dynamic programming and principles of optimality, *J. Math. Anal. Appl.* **65** (1978), 586–606.
23. M. J. SOBEL, Ordinal dynamic programming, *Management Sci.* **21** (1975), 967–975.
24. S. J. YAKOWITZ, "Mathematics of Adaptive Control Processes," American Elsevier, New York, 1969.