Some Multiple Objective Dynamic Programs

H. W. CORLEY

Abstract—**Two examples are presented to illustrate the application of dynamic programming to Pareto optimization.**

I. INTRODUCTION

Recall the definition of Pareto optimization. Let $u = (u_1, \cdots, u_p)$, $v = (v_1, \cdots, v_p) \in D \subset R^p$. If $u_i \leq v_i$, $i = 1, \cdots, p$, and $u_j < v_j$ for some j, we write $u < v$. The point $u \in D$ is said to be a Pareto maximum of D, denoted $u \in \text{Max}^* D$, if $u \nless v$ for all $v \in D$. Pareto minima and $\text{Min}^* D$ have similar definitions.

In this brief note we present two examples illustrating the application of dynamic programming to Pareto optimization. The reason for doing so is that there have been very few examples in the literature, although various authors have studied the relation of dynamic programming to Pareto optimality. The earliest such results are due to Brown and Strauch [1] who stated a version of the principle of optimality for finite horizon models in multiplicative lattices. Mitten [2] and Sobel [3] later analyzed ordinal dynamic programs. More recently, Henig [4] has studied vector-valued dynamic programming for the viewpoint of Markov decision processes and reviewed the literature on this aspect. The examples here, however, follow the work of Klötzler [5], who gave recurrence relations under certain separability and monotonicity assumptions as well as the von Neumann–Morgenstern property. We refer to that work.

II. EXAMPLES

In both of the following examples the above assumptions are obviously satisfied. The first example is discrete and involves finding all Pareto shortest paths in a directed network, a problem of interest in its own right. The second example is continuous and illustrates how a sequential solution can simplify certain problems.

Example 1: Consider a directed network (N, A), where $N = \{1, \cdots, n\}$ is the set of n nodes, $A \subset N \times N$ is the set of arcs, and associated with each arc $(j, k) \in A$ is a vector weight $d_{jk} = (d_{jn1}, \cdots, d_{jnp}) \in R^p$. All Pareto shortest paths in (N, A) between nodes 1 and $r = 2, \cdots, n$

will be obtained here; i.e., we find

$$\text{Min}^* \left\{ \sum_{(j,k)\in\pi} d_{jk} : \pi \in P_r \right\}, \qquad r=2, \cdots, n \tag{1}$$

where P_r is the set of paths from 1 to r in (N, A) and the summation is a vector sum. We assume that A contains no loops since a loop either is not part of a Pareto shortest path or is part of an infinite number and also that $d_{jk} \neq (0, \cdots, 0)$ for $(j, k) \in A$ to avoid degenerate circuits. No restrictions are made on the signs of the components of the d_{jk}.

Problem (1) reduces to the standard shortest path problem when $p = 1$ for which Bellman [6] has used the functional equations of standard dynamic programming. To use the recurrence relations to Klötzler, let the stages correspond to the number of arcs in a path, the states to nodes, the decision variables to the next node in a path, and the stage returns to the distance to the next node. Define L_i^k to be the set of "lengths" of all Pareto paths from node 1 to node i containing k or fewer arcs. We seek L_i^{n-1}, $i = 2, \cdots, n$. An algorithm for obtaining the L_1^{n-1} is given below; the actual paths can be deduced by a substitution process. The validity of this algorithm can also be established without dynamic programming [7].

Step 1: Set $d_{ii} = (0, \cdots, 0)$, $i = 1, \cdots, n$, and $d_{ij} = (\infty, \cdots, \infty)$, $i \neq j$, if there is no arc from i to j. Set $k = 1$ and $L_i^1 = \{d_{1i}\}$, $i = 1, \cdots, n$.

Step 2: For $i = 1, \cdots, n$, set $L_i^{k+1} = \text{Min}^* \cup_{j=1}^{n} [d_{ji} + L_j^k]$, where $d_{ji} + L_j^k = \{d_{ji} + l_j^k : l_j^k \in L_j^k\}$.

Step 3: If $L_i^{k+1} = L_i^k$, $i = 1, \cdots, n$, stop. Otherwise, go to Step 4.

Step 4: If $k = n - 1$, stop. A negative circuit exists. Otherwise, set $k = k + 1$ and go to Step 2.

As an illustration, this algorithm is applied to the very simple network of Fig. 1. For Fig. 1 we easily compute the following:

$$L_1^1 = \{(0, 0)\},\ L_2^1 = \{(8, 1)\},\ L_3^1 = \{(7, 2)\},$$

$$L_4^1 = \{(1, 2)\},\ L_5^1 = \{(\infty, \infty)\};$$

$$L_1^2 = \{(0, 0)\},\ L_2^2 = \{(8, 1)\},\ L_3^2 = \{(7, 2)\},$$

$$L_4^2 = \{(1, 2)\},\ L_5^2 = \{(3, 3)\};$$

$$L_1^3 = \{(0, 0)\},\ L_2^3 = \{(8, 1)\},\ L_3^3 = \{(7, 2), (6, 3)\},$$

$$L_4^3 = \{(1, 2)\},\ L_5^3 = \{(3, 3)\};$$

$$L_1^4 = \{(0, 0)\},\ L_2^4 = \{(8, 1), (7, 4)\},\ L_3^4 = \{(7, 2), (6, 3)\},$$

$$L_4^4 = \{(1, 2)\},\ L_5^4 = \{(3, 3)\};$$

$$L_i^5 = L_i^4, \qquad i = 1, \cdots, 5, \text{ so stop.}$$

Thus, for example, the two Pareto shortest paths from node 1 to node 2 are 1-2 and 1-4-5-3-2 yielding "lengths" of (8, 1) and 7, 4), respectively.

Example 2: Consider the problem

$$\text{Max}^* \{(2x_1 - 3x_2, 4x_1 + x_2^2) : x_1 + x_2 \leq 12;\ x_1, x_2 \geq 0\}. \tag{2}$$

Let two stages correspond to the two variables and decompose the separable two-dimensional objective function into $(2x_1 - 3x_2, 4x_1 + x_2^2) = (2x_1, 4x_1) + (-3x_2, x_2^2)$. The individual stage return functions are then $(2x_1, 4x_1)$ at stage 1 and $(-3x_2, x_2^2)$ at stage 2. We let the state s_i represent the maximum amount that can be allocated to x_i at stage i, so $s_2 = 12$ and $s_1 = s_2 - x_2$. Thus

$$f_1(s_1) = \text{Max}^* \{(2x_1, 4x_1) : 0 \leqslant x_1 \leqslant s_1 : x_1 \geqslant 0\}. \tag{3}$$

Clearly in (3) the optimal $x_1^* = s_1$ for which $f_1(s_1) = \{(2s_1, 4s_1)\}$. It then follows that

$$f_2(12) = \text{Max}^* \{(-5x_2 + 24, x_2^2 - 4x_2 + 48) : 0 \leqslant x_2 \leqslant 12\}. \tag{4}$$

Since $-5x_2 + 24$ is strictly decreasing for $x_2 \geq 0$ while $x_2^2 - 4x_2 + 48$ is nondecreasing for $x_2 \geq 2$ and nonincreasing for $x_2 \leq 2$, it is easy to

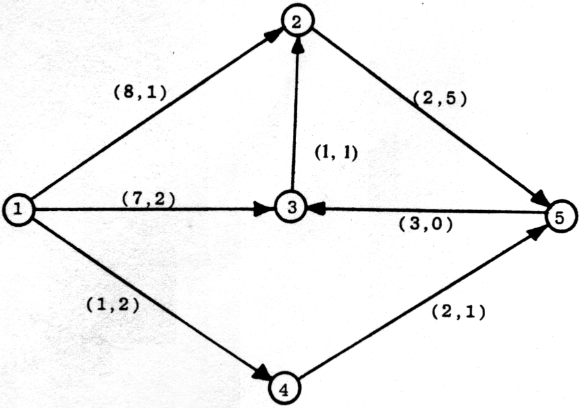

Fig. 1.

establish that (4) becomes $f_2(12) = \{(24, 48)\} \cup \{(-5x_2 + 24, x_2^2 - 4x_2 + 48 : 4 < x_2 \leq 12\}$, from which the solution to (2) is obtained with $x_1 = 12 - x_2$. This answer appears to be obtained significantly more easily here than by other methods of solving Pareto problems (for example, see [8], [9] and the references therein).

REFERENCES

[1] T. A. Brown and R. E. Strauch, "Dynamic programming in multiplicative lattices," *J. Math. Anal. Appl.*, vol. 12, pp. 364–370, 1965.

[2] L. G. Mitten, "Preference order dynamic programming," *Management Sci.*, vol. 21, pp. 43–46, 1974.

[3] M. J. Sobel, "Ordinal dynamic programming," *Management Sci.*, vol. 21, pp. 967–975, 1975.

[4] I. Mordechai Henig, "Vector-valued dynamic programming," *SIAM J. Contr. Optimiz.*, vol. 21, pp. 490–499, 1983.

[5] R. Klötzler, "Multiobjective dynamic programming." *Math. Operationsforsch. Statist., Ser. Optimization,* vol. 9, pp. 423–426, 1978.

[6] R. E. Bellman, "On a routing problem," *Quart. Appl. Math.*, vol. 16, pp. 87–90, 1958.

[7] H. W. Corley and I. D. Moon, "Shortest routes in networks with vector weights," *J. Optimiz. Theory Appl.*, to be published.

[8] H. W. Corley, "A new scalar equivalence for Pareto optimization," *IEEE Trans. Automat. Contr.*, vol. AC-25, pp. 829–830, 1980.

[9] ——, "A fixed point interpretation of Pareto optimization," *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 766–767, 1981.