SIAM J. APPL. MATH.
Vol. 17, No. 4, July 1969

L-SHAPED LINEAR PROGRAMS WITH APPLICATIONS TO OPTIMAL CONTROL AND STOCHASTIC PROGRAMMING*

R. M. VAN SLYKE† AND ROGER WETS‡

Abstract. This paper gives an algorithm for L-shaped linear programs which arise naturally in optimal control problems with state constraints and stochastic linear programs (which can be represented in this form with an infinite number of linear constraints). The first section describes a cutting hyperplane algorithm which is shown to be equivalent to a partial decomposition algorithm of the dual program. The two last sections are devoted to applications of the cutting hyperplane algorithm to a linear optimal control problem and stochastic programming problems.

1. Introduction. It has been observed by many authors (see, e.g., Barr [2], Gilbert [12], Rosen [21], [22], Neustadt [18], Whalen [29], [30], Zadeh [31], Pshenichniy [19]) that the techniques of mathematical programming can be utilized to solve optimal control problems. The usual approach (although others are possible, e.g., Dantzig [6], Van Slyke [23]) is to discretize the system either by finite difference approximations or by considering the system in sample data mode. If the system dynamics are linear and there are no state space constraints, various devices [6], [23] of mathematical programming can be used so that the grid size or number of sample points in the sample mode does not affect the number of equations in the associated mathematical program. This is desirable since the computational effort for solving linear programs by the simplex method depends much more on the number of equations involved than on the number of variables. However, if state space constraints are present, the number of equations can grow astronomically. This is unfortunate, especially in the common situation where the state space constraints are automatically satisfied for most time periods. If confronted with problems of this type, the following heuristic procedure suggests itself: First solve the problem without the state space constraints, then check if the solution satisfies all the state space constraints. If it violates some of these constraints, introduce only those which are violated and solve this new problem. The procedure is repeated until a feasible (and thus optimal) solution is attained. The algorithm developed in this paper formalizes the ideas of this heuristic procedure. Whenever we obtain a solution which violates some state space constraint, we generate a restriction on the *controls* (rather than on the states) which eliminates this particular solution from the feasibility region.

The algorithm can be slightly modified to solve stochastic programs with recourse, first considered by Dantzig [5] and Dantzig and Madansky [7] under the name of two-stage linear programs under uncertainty. The problem here is the following: A decision must be made before the actual values of some of the parameters of the problem are observed (it is assumed that those parameters are known in probability). Due to the lack of knowledge of the particular outcome

* Received by the editors June 27, 1966, and in revised form November 13, 1967.

† School of Engineering, University of California, Berkeley, California, 94720.

‡ Mathematics Research Laboratory, Boeing Scientific Research Laboratories, Seattle, Washington 98124.

of the random elements of the problem, discrepancies may occur which, after observing the actual values of those parameters, are to be corrected by selecting a particular recourse action (also called second-stage decision). One of the difficulties which arises when trying to solve such problems is that a particular decision and a particular outcome of the random elements may give rise to discrepancies for which there is no (feasible) recourse action. Thus, one should only select decisions such that, for every possible realization of the random elements, a (feasible) recourse action can be selected to correct the eventual discrepancies. Earlier treatments of stochastic programming ignored this difficulty [7], [25] by assuming that the structure of the stochastic program was such that this problem could not arise. It did appear that unless one made this assumption the additional constraints one had to introduce could be very large, even infinite when the random parameters had continuous distributions.

In [7] Dantzig and Madansky considered stochastic programs with finitely distributed random parameters and complete recourse; i.e., for every decision and for every outcome of random variables there exists a feasible recourse. For obvious practical reasons it seemed desirable to remove those restrictive assumptions. The last section of this paper develops an algorithm for stochastic programs which fail to satisfy the complete recourse assumption as well as the finite distribution assumption.

In [27] it was shown that for stochastic programs with recourse (random right-hand sides) the set of feasible decisions, represented by an n-vector x, is a convex polyhedral subset of $\mathscr{R}^n$; thus at most a *finite* number of linear constraints must be added to the problem to determine the set of feasible decisions. However, the characterization of the feasibility region given in [27] is not very constructive. The algorithm developed here generates these linear constraints systematically and generates only those which are violated by some optimal decision candidate, in much the same way as in the control problem with state space constraints.

The stochastic programming problem differs from the linear optimal control problem in that there is a cost associated with the recourse actions which must be accounted for. Dantzig and Madansky [7] suggest sampling to obtain the appropriate characteristics of the cost function associated with the recourse problem. However, as pointed out by Madansky,[1] the utilization of sampling can lead to inaccuracies. This, as we shall see, can be avoided by using a gradient method rather than a cutting plane method.

In § 2, an algorithm which is essentially the same as the algorithm developed by Benders [3][2] is described and a geometric interpretation is given. Section 3 exhibits the duality between this algorithm and a variant of the decomposition algorithms of Dantzig and Wolfe. The applications of this algorithm to optimal control problems with state space constraints and stochastic programs with recourse are developed in § 4 and § 5 respectively. Now, let us give a mathematical formulation of the linear program we are interested in.

[1] SIGMAP Conference on Stochastic Programming, Princeton, New Jersey, 1965.

[2] This was pointed out to us by E. Balas and J. Midler.

We give the name *L-shaped linear programs* to linear programs of the form:

Minimize

$$z = c^1x + c^2y \tag{1}$$

subject to

$$A^{11}x = b^1, \tag{1a}$$

$$A^{21}x + A^{22}y = b^2, \tag{1b}$$

$$x \geqq 0, \quad y \geqq 0,$$

where A^{11} is an $m_1 \times n_1$ matrix, A^{21} is $m_2 \times n_1$ and A^{22} is $m_2 \times n_2$. As can be seen from the applications that we have in mind, we expect that (1) has some or all of the following characteristics:

(i) The constraints $A^{21}x + A^{22}y = b^2$ are loose, in the sense that for "most" vectors x satisfying $A^{11}x = b^1$, $x \geqq 0$, there exists $y \geqq 0$ such that the constraints $A^{21}x + A^{22}y = b^2$ are satisfied.

(ii) The vector y is of little interest, and the value of c^2y is a small factor in determining the value of the optimal solution.

(iii) The constraints $A^{21}x + A^{22}y = b^2$ are numerous, possibly infinite, and are often given in an implicit manner.

Thus, in order to speed up computation and limit storage requirements, it is desirable to work mainly with the constraints (1a) and consider the constraints (1b) and the variables y only when needed.

2. A cutting plane algorithm.

2.1. Feasibility. Instead of problem (1), let us first consider the special case where $c^2 = 0$ (this corresponds to the problem arising in optimal control problems with state space constraints):

Minimize

$$z = c^1x \tag{2}$$

subject to

$$A^{11}x = b^1,$$

$$A^{21}x + A^{22}y = b^2,$$

$$x \geqq 0, \quad y \geqq 0.$$

In this case the algorithm proceeds as follows. First solve the simpler linear program:

Minimize

$$z = c^1x \tag{3}$$

subject to

$$A^{11}x = b^1,$$

$$x \geqq 0,$$

whose optimal solution we denote by $\bar{x}$. For the time being we assume that (3) is solvable.

Feasibility criterion. There exists $y \geqq 0$ such that $A^{22}y = b^2 - A^{21}\bar{x}$.

If $\bar{x}$ satisfies the feasibility criterion, then $\bar{x}$ and some y determine a feasible (and thus optimal) solution to (2). We denote by K_2 the set of all x satisfying the feasibility criterion.

If $\bar{x}$ does not satisfy the feasibility criterion, we generate a constraint involving only x which is violated by $\bar{x}$ but satisfied by any feasible solution to (2). This constraint is then added to the constraints of problem (3). This added constraint has, in a sense to be made precise later (§ 2.5), the property that it cuts deepest into the set $K_1 = \{x \mid A^{11}x = b^1, x \geqq 0\}$. The process is then repeated until an optimal solution to the augmented problem (3) satisfies the feasibility criterion. We shall show that we have to add at most a finite number of constraints to (3) in order to achieve this goal.

To determine whether $\bar{x}$ satisfies the feasibility criterion or not, we try to find a nonnegative solution, y, to

$$A^{22}y = b^2 - A^{21}\bar{x}. \tag{4}$$

This can be considered geometrically. Let pos $A^{22} = \{t | t = A^{22}y, y \geqq 0\}$ be the closed convex cone generated by the columns of A^{22}. Then $\bar{x}$ satisfies the feasibility criterion if and only if $b^2 - A^{21}\bar{x} \in \operatorname{pos} A^{22}$. If not, i.e., if

$$b^2 - A^{21}\bar{x} \notin \operatorname{pos} A^{22},$$

there is a hyperplane through the origin separating strictly $b^2 - A^{21}\bar{x}$ and pos A^{22}. Such a hyperplane, say $\{x \mid \sigma x = 0\}$, is determined by its normal σ which must satisfy $\sigma t \leqq 0$ for $t \in \operatorname{pos} A^{22}$ and $\sigma[b^2 - A^{21}]\bar{x} > 0$.

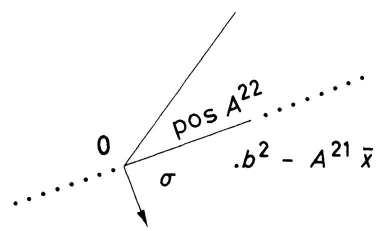

FIG. 1

The normals σ, which are needed, are generated using a slight variant of the Phase I procedure for the simplex method. We solve the following problem:

Minimize

$$w = ev^+ + ev^- \tag{5}$$

subject to

$$A^{22}y + Iv^+ - Iv^- = b^2 - A^{21}\bar{x},$$

$$y \geqq 0, \quad v^+ \geqq 0, \quad v^- \geqq 0,$$

where e is a row vector of 1's, I is an $m_2 \times m_2$ identity matrix and v^+ and v^- are m_2-vectors of variables.

Problem (5) has always an optimal solution with $w \geqq 0$. $\bar{x}$ satisfies the feasibility criterion if and only if $w = 0$ at the optimum. If at the optimum $w > 0$, then there exist dual variables σ satisfying

$$\begin{aligned} &\sigma A^{22} \leqq 0, \\ &-e \leqq \sigma \leqq e, \\ &\sigma[b^2 - A^{21}\bar{x}] = \min w > 0. \end{aligned} \tag{6}$$

Thus σ has the desired properties. In the next sections we show that the σ's generated by solving (5) are optimal in some sense and give the geometrical interpretation in more detail.

In order for x to be feasible it is clear that $b^2 - A^{21}x$ must be on the same side of the hyperplane $\{t|\sigma t = 0\}$ as pos A^{22}.

Thus, x feasible implies that

$$\sigma[b^2 - A^{21}x] \leqq 0. \tag{7}$$

We then add the constraint

$$[\sigma A^{21}]x \geqq \sigma b^2 \tag{8}$$

to the linear program (3).

It is also possible that when solving problem (3) (or even after a few additional constraints have been added) we discover that (3) is unbounded. Thus, the solution to (3) is no longer given in terms of a particular vector x, but we are given a half-line in K_1, say $\bar{x}_p + \lambda\bar{x}_c$, $\lambda \geqq 0$, on which cx decreases monotonically to $-\infty$ as λ goes to $+\infty$.

PROPOSITION 1. *If* $-A^{21}\bar{x}_c$ *and* $b^2 - A^{21}\bar{x}_p$ *belong to* pos A^{22}, *then* (2) *is unbounded. If* $-A^{21}\bar{x}_c \notin$ *pos* A^{22}, *then every solution to* (2) *must satisfy the constraint*

$$[\sigma A^{21}]x \geqq \sigma b^2, \tag{9}$$

which is violated by $\bar{x}_p + \lambda\bar{x}_c$ *for* λ *sufficiently large, where* σ *denotes the vector of optimal simplex multipliers corresponding to the optimal solution to*:

Minimize

$$\bar{w} = ev^+ + ev^- \tag{10}$$

subject to

$$A^{22}y + Iv^+ - Iv^- = -A^{21}\bar{x}_c,$$

$$y \geqq 0, \quad v^+ \geqq 0, \quad v^- \geqq 0.$$

If $-A^{21}\bar{x}_c \in$ pos A^{22} *but* $b^2 - A^{21}\bar{x}_p \notin$ pos A^{22}, *then every feasible solution to* (2) *must satisfy the constraint generated by solving the linear program* (5), *where* x *is set equal to* $\bar{x}_p$.

Proof. The conclusion is immediate if $-A^{21}\bar{x}_c$ and $b^2 - A^{21}\bar{x}_p$ belong to pos A^{22}. If $-A^{21}\bar{x}_c \notin$ pos A^{22}, then for some $\bar{\lambda}$, $b^2 - A^{21}\bar{x}_p - \lambda A^{21}\bar{x}_c \notin$ pos A^{22} for all $\lambda > \bar{\lambda}$. To see this, it suffices to observe that if the σ are the optimal simplex multipliers for (10) then $\sigma A^{22} \geqq 0$ and $\sigma A^{21}\bar{x}_c > 0$. Thus, by selecting λ sufficiently large, $\sigma(b^2 - A^{21}\bar{x}_p - \lambda A^{21}\bar{x}_c)$ can also be made arbitrarily small. Set $\bar{\lambda} = 0$ if $\sigma(b^2 - A^{21}\bar{x}_p) \leqq 0$, otherwise select $\bar{\lambda}$ such that $\sigma(b^2 - A^{21}\bar{x}_p - \bar{\lambda}A^{21}\bar{x}_c) = 0$. Then for all $\lambda > \bar{\lambda}$, σ determines a hyperplane separating pos A^{22} and

$$(b^2 - A^{21}\bar{x}_p - \lambda A^{21}\bar{x}_c).$$

It follows that every x in K_1 such that $x = (\bar{x}_p + \bar{\lambda}\bar{x}_c) + \mu\bar{x}_c$, $\mu > 0$, violates (9), which must be satisfied by every feasible solution to (2). If $-A^{21}\bar{x}_c \in \text{pos } A^{22}$ but $b^2 - A^{21}\bar{x}_p \notin \text{pos } A^{22}$, either $b^2 - A^{21}\bar{x}_p - \lambda A^{21}\bar{x}_c$ does not belong to pos A^{22} for all $\lambda \geqq 0$ or there exists $\bar{\lambda}$ such that if $\lambda > \bar{\lambda}$, $b^2 - A^{21}\bar{x}_p - \lambda A^{21}\bar{x}_c$ belongs to pos A^{22}. Now let σ denote the optimal simplex multipliers obtained from (5) by setting $x = \bar{x}_p$. If for all $\lambda \geqq 0$, $b^2 - A^{21}\bar{x}_p - \lambda A^{21}\bar{x}_c$ does not belong to pos A^{22}, the ray $\bar{x}_p + \lambda\bar{x}_c$ violates for all λ the constraint (8) so generated, and thus this particular extreme ray is eliminated from the feasible solution. On the other hand, if $b^2 - A^{21}\bar{x}_p - \lambda A^{21}\bar{x}_c$ belongs to pos A^{22} for $\lambda \geqq \bar{\lambda}$, the points

$$b^2 - A^{21}(\bar{x}_p + \bar{\lambda}\bar{x}_c) - \mu A^{21}\bar{x}_c, \qquad \mu \geqq 0,$$

satisfy the constraints, and the ray $(\bar{x}_p + \bar{\lambda}\bar{x}_c) + \mu\bar{x}_c$ has not been eliminated from the set of feasible solutions. This completes the proof.

We can thus summarize the procedure to find an optimal solution to (2) as follows:

If (3) (with or without additional constraints) is solvable with $x = \bar{x}$, we then solve (5). If $w = 0$, then $\bar{x}$ is an optimal solution for (2). Otherwise we generate a constraint of the type (8) which is then added to the constraints of (3). If (3) (with or without additional constraints) is unbounded with a direction of decrease for cx given by $x = \bar{x}_p + \lambda\bar{x}_c$, $\lambda \geqq 0$, we then solve (10), and (5) with $x = \bar{x}_p$. Let $\bar{w}$ and w denote the optimal value for (10) and (5) respectively. If $\bar{w} = w = 0$, then (2) is unbounded. If $\bar{w} > 0$, we use the optimal multipliers of (10) to generate a constraint of the type (9) which is added to the constraints of (3); if $\bar{w} = 0$ but $w > 0$, we generate a constraint of the type (8).

Clearly this process is finite since each σ corresponds to a basis for (5) (or (10)) of which there is a finite number and, moreover, no constraint can be repeated. Obviously, no constraint of type (9) will be generated after we obtain a bounded solution to (3).

It is conceivable, of course, that the number of bases of (5) or (10), corresponding to a particular σ, could be very large, so that the number of generated constraints could be large compared to the number of original constraints (1b), in which case the proposed algorithm might be inefficient. However, since we only add binding constraints which have a deepest cut property (as we shall see later) and if properties (i), (ii) and (iii) mentioned in the Introduction are satisfied, this seems unlikely.

Another useful property of the algorithm is that in adding new constraints to (3), the next iteration already has a basic solution which is infeasible only for one basic variable. The basis is the basis for the previous iteration, plus the slack variable for the added constraint. Thus, each successive x can be easily obtained by a few steps of the dual simplex method.

2.2. Optimality. We now return to our original problem (1), i.e., to the case where c^2 may be different from zero. Obviously, problem (1) is equivalent to:

Minimize

$$c^1x + \theta, \tag{11}$$

subject to

$$Q(x) \leqq \theta,$$
$$x \in K = K_1 \cap K_2,$$

where

$$Q(x) = \{\min c^2y | A^{22}y = b^2 - A^{21}x, x \geqq 0\}. \tag{12}$$

We first make the following observation.

PROPOSITION 2. *For all $x \in K_2$, $Q(x)$ is either a finite convex function or $Q(x)$ is identically $-\infty$.*

Proof. For all $x \in K_2$, the following linear program is feasible:

Minimize

$$c^2y \tag{13}$$

subject to

$$A^{22}y = b^2 - A^{21}x,$$
$$y \geqq 0.$$

Moreover, for all $x \in K_2$, (13) is unbounded if and only if the linear system $\pi A^{22} \leqq c$ is inconsistent. Thus, if (13) is unbounded for some x, it will be unbounded for all x. It remains to show that if $Q(x)$ is finite on the convex set K_2, then it is convex. Consider x^0, $x^1 \in K_2$ and $x^\lambda = (1 - \lambda)x^0 + \lambda x^1$, where $\lambda \in [0, 1]$, and let y^0, y^1 and y^λ be optimal solutions to (13) when x equals x^0, x^1 or x^λ respectively. Then,

$$(1 - \lambda)Q(x^0) + \lambda Q(x^1) = c^2[(1 - \lambda)y^0 + \lambda y^1] \geqq c^2y^\lambda = Q(x^\lambda) \tag{14}$$

since $(1 - \lambda)y^0 + \lambda y^1$ is a feasible solution (but not necessarily optimal) to (13) when x equals x^λ. This completes the proof.

PROPOSITION 3. *Suppose $Q(\bar{x})$ is finite; let $\bar{\pi}$ denote the optimal simplex multipliers corresponding to the solution of* (13) *with $x = \bar{x}$; then the linear function*

$$(\bar{\pi}A^{21})x - (\bar{\pi}b^2) \tag{15}$$

is a support of $Q(x)$.

Proof. Since $\bar{\pi}$ is optimal for (13) with $x = \bar{x}$, by the duality theory for linear programming we have that

$$\bar{\pi}(b^2 - A^{21}\bar{x}) = Q(\bar{x}). \tag{16}$$

By assumption $Q(x)$ is finite, and thus for all $x \in K_2$, $\bar{\pi}$ is a feasible solution for all duals of (13):

Maximize

$$\begin{aligned} &\pi(b^2 - A^{21}x),\\ &\pi A^{22} \leqq c^2; \end{aligned} \tag{17}$$

but $\bar{\pi}$ is not necessarily an optimal solution. Thus, again by the duality theory we have

$$\bar{\pi}(b^2 - A^{21}x) \leqq \{\max \pi(b^2 - A^{21}x) | (\pi A^{22} \leqq c^2)\} = Q(x) \quad \text{for all } x \in K_2. \tag{18}$$

This completes the proof.

Even though the following observation is not absolutely necessary for the subsequent development, it is worthwhile to note.

PROPOSITION 4. *Suppose $Q(x)$ is finite on K_2; then $Q(x)$ is a convex polyhedral function.*

Proof. By letting x range over K_2, we see that only a finite number of supports to $Q(x)$ of the type (15) can be generated, since every π corresponds to a particular basis of A^{22} and A^{22} has only a finite number of square nonsingular submatrices. Moreover, for all $x \in K_2$ there is some support of type (15) which meets $Q(x)$ at x. Thus, the upper envelope of this finite number of linear supports coincides with $Q(x)$. This completes the proof.

The process to obtain an optimal solution to (1) (or, equivalently, to (11)) is very similar to the one already described for finding a feasible solution. Suppose $\bar{x}$ is a feasible solution, i.e., $\bar{x} \in K = K_1 \cap K_2$, and (13) is solvable with $x = \bar{x}$. Let $\bar{\pi}$ be the corresponding optimal simplex multipliers. Then,

$$Q(\bar{x}) = \bar{\pi}(b^2 - A^{21}\bar{x}).$$

Moreover, by convexity of $Q(x)$ and the properties of $\bar{\pi}$ given in Proposition 3, it follows that

$$Q(x) \geqq \bar{\pi}b^2 - [\bar{\pi}A^{21}]x$$

for all x in K. Thus, a pair (x, θ) is feasible for (11) only if

$$\theta \geqq \bar{\pi}b^2 - [\bar{\pi}A^{21}]x,$$

which we can also write as

$$[\bar{\pi}A^{21}]x + \theta \geqq \bar{\pi}b^2. \tag{19}$$

On the other hand if the (x^0, θ^0) are optimal for (11) and the π^0 are the optimal simplex multipliers obtained from (13) by substituting x for x^0, we have that

$$Q(x^0) = \pi^0 b^2 - \pi^0 A^{21} x^0.$$

The optimality of x^0 implies that $cx + Q(x) \geqq cx^0 + Q(x^0)$ for all x in K. From $\theta^0 \geqq Q(x)^0$ and θ unrestricted in (11) it follows that $\theta^0 = Q(x^0)$.

These last two observations allow us to construct a finite procedure for finding an optimal solution to (1). Say (x^k, θ^k) is an optimal solution to the following linear program:

Minimize

$$c^1x + \theta \tag{20}$$

subject to

$$[\pi^l A^{21}]x + \theta \geqq (\pi^l b^2), \qquad l = 1, \cdots, k-1, \tag{20a}$$

$$x \in K_1 \cap K_2.$$

We then solve (13) with $x = x^k$. If (13) is unbounded, then (1) is unbounded. If not, let π^{k+1} denote the optimal simplex multipliers.

Optimality criterion. If

$$\theta^k = \pi^{k+1}[b^2 - A^{21}x^k], \tag{21}$$

then x^k is an optimal solution to (1).

If $\theta^k < Q(x^k)$, we add the constraint

$$[\pi^{k+1}A^{21}]x + \theta \geqq \pi^{k+1}b^2$$

to the constraints of (20), which has the effect of eliminating the point (x^k, θ^k) from the set of feasible solutions of (20). The algorithm is initiated with x^0 minimizing c^1x on K and $\theta^0 = -\infty$.

Now suppose that (20) is unbounded after at least one constraint of type (20a) has been introduced. Note that in such a case, (20) cannot be unbounded for some fixed x and $\theta = -\infty$, since θ must satisfy the constraint of type (20a). Thus, there exists some ray, say $x_p + \lambda x_c$, $\lambda \geqq 0$, on which the objective of (20) can be pushed to $-\infty$. Checking if this ray belongs to pos A^{22} has been dealt with in the previous section. If not, we generate constraints of type (8) or (9). Now suppose $b^2 - A^{21}x_p$ and $-\lambda A^{21}x_c$ belong to pos A^{22}. Let y_c be an optimal solution to the linear program:

Minimize

$$c^2y$$

subject to

$$A^{22}y = -A^{21}x_c,$$

$$y \geqq 0,$$

and let π be the corresponding vector of optimal simplex multipliers. If

$$c^1x_c + c^2y_c < 0,$$

then obviously (1) is unbounded. If $c^1x_c + c^2y_c > 0$, then x_c is not a desirable unbounded direction since letting λ go to $+\infty$ in $x_p + \lambda x_c$ would push the objective of (1) to $+\infty$. In this case adding the constraint

$$[\pi A^{21}]x + \theta \geqq [\pi b^2]$$

to (20) would eliminate the direction x_c from the desirable (optimal) solutions of (20). If $c^1x_c + c^2y_c = 0$, then no point of the ray $x_p + \lambda x_c$ will be preferable to x_p as a solution to (1); thus adding the above constraint to (20) will keep x_p in the set of feasible solutions of (20) but will eliminate the other points of the ray.

This process is obviously finite since each π corresponds to a basis of A^{22} and these are finite in number. Moreover, no π can be generated twice since this would lead to a constraint already present which could not be violated by the solution at hand. In this section we have assumed that each x generated is a feasible solution. If $x \notin K_2$, then one may have to introduce constraints of type (8) or (9) before continuing the search for an optimal solution to (1).

2.3. Summary of the algorithm.

Step 1. Solve the linear program:

Minimize

$$z = c^1x + \theta \tag{22}$$

subject to

$$A^{11}x = b^1, \tag{22a}$$

subject to

$$[\sigma^k A^{21}]x \geqq [\sigma^k b^2], \quad k = 1, \cdots, s, \tag{22b}$$

$$[\pi^k A^{21}]x + \theta \geqq [\pi^k b^2], \quad k = 1, \cdots, t, \tag{22c}$$

$$x \geqq 0.$$

Initially, $s = t = 0$. θ is set equal to $-\infty$ and is deleted from the actual computations as long as there are no constraints of type (22c). If (22) is infeasible, so is (1) and we terminate. If (22) is solvable, go to Step 2. If (22) is feasible but unbounded, go to Step 2′.

Step 2. Problem (22) is solvable. Let (x^l, θ^l) be an optimal solution to (22). Use the simplex method (Phase I, Phase II) to solve the following problem:

Minimize

$$w = c^2y \tag{23}$$

subject to

$$A^{22}y = b^2 - A^{21}x^l,$$

$$y \geqq 0.$$

If (23) is feasible, i.e., Phase I terminates with the infeasibility form different from zero, we use the multipliers so generated to construct a constraint of the form (22b). If (23) is feasible and unbounded, so is (1) and we terminate. If (23) is solvable and min $w(x^l) = \theta^l$, then x^l is optimal and we terminate. Otherwise, we use the multipliers so generated to construct a constraint of the form (22c) and return to Step 1.

Step 2′. Problem (22) is feasible but unbounded. Let $x_p^l + \lambda x_c^l$, $\lambda \geqq 0$, be a ray of unbounded decrease of c^1x. We then solve (23) with $b^2 - A^{21}x^l$ replaced by $-A^{21}x_c^l$. If this problem is infeasible (i.e., Phase I terminates with positive objective value), we use the optimal simplex multipliers to generate a constraint of type (22b). If this problem is feasible, let y_c^l be the optimal solution and π^l the associated simplex multipliers. Now solve (23) with $x^l = x_p^l$. If this new problem is infeasible, we generate a constraint (22b) as in Step 2. Otherwise, either $c^1x_c^l + c^2y_c^l < 0$ in which case (1) is unbounded and we terminate; or $c^1x_c^l + c^2y_c^l \geqq 0$, and then π^l is used to generate a constraint of type (22c) and we return to Step 1.

Finally, it is not difficult to see that if so desired (e.g., in order to keep the data related to problem (22) in the easy access memory), it is possible to remove those constraints of (22b) and (22c) which are slack, although they may be generated again and have to be re-introduced. This also necessitates a new finiteness proof which is based on the fact that, upon taking suitable account for degeneracy, the objective value $c^1x + Q(x)$ corresponding to every feasible solution to (23)—generated in Step 2—is monotonically decreasing, so there are only a finite number in which (23) has a feasible solution. On the other hand, between feasible solutions to (23) when constraints of the form (22b) are being introduced, the value of c^1x^k is monotonically increasing so that a feasible solution to (23) always occurs after a finite number of steps.

2.4. Some geometric characterizations. We have already pointed out that checking if a particular point, say $\bar{x}$, is feasible corresponds to determining if $b^2 - A^{21}\bar{x}$ belongs to the cone pos $A^{22} = \{t | t = A^{22}y, y \geqq 0\}$. Similarly, if at some stage the program (22) yields an unbounded direction, then solving (23) with $b^2 - A^{21}x^l$ replaced by $-A^{21}x_c^l$ corresponds to determining if the ray $\lambda(-A^{21}x_c^l)$, $\lambda \geqq 0$, belongs to the cone pos A^{22}. Even checking for optimality of a given pair (θ^l, x^l) can be viewed as determining if $\begin{pmatrix} \theta^l \\ b^2 - A^{21}x^l \end{pmatrix}$ belongs or does not belong to the cone

$$\operatorname{pos}\begin{pmatrix} c^2 \\ A^{22} \end{pmatrix} = \left\{ \begin{pmatrix} \tau \\ t \end{pmatrix} \middle| \tau = c^2y,\ t = A^{22}y,\ y \geqq 0 \right\}. \tag{24}$$

In this section and the following one, we limit our discussion to the case where we are checking for feasibility, i.e., $\bar{x}$ in K_2; but in view of the above observations our remarks can be adapted equally well to the other parts of the algorithm.

Suppose $x \in K_1$ but does not belong to K_2. Then solving the linear program (5) yields $w > 0$. At the same time we generate some σ, which corresponds to a particular basis of the matrix $(A^{22}, I, -I)$. The basic solution contains at least one artificial variable, i.e., a component of the vector (v^+, v^-), at positive level. Since otherwise $w = 0$ and $b^2 - A^{21}x \in \operatorname{pos} A^{22}$, we have the next proposition.

PROPOSITION 5. *Suppose the optimal solution to* (5), *with* $w > 0$, *contains exactly one artificial variable. Then,* σ *is the normal of a supporting hyperplane of* pos A^{22} *determining an* $(m_2 - 1)$*-dimensional face of* pos A^{22}.

Proof. First note that this $(m_2 - 1)$-dimensional face may be pos A^{22} itself, viz., if pos A^{22} is of dimension $m_2 - 1$. Also, by the hypothesis of this proposition the cone pos A^{22} has at least dimension $m_2 - 1$. By assumption, there are at least $m_2 - 1$ columns of A^{22} such that $\sigma A^{22}_{*j} = 0$, where A^{22}_{*j} denotes the jth column of A^{22}. Of these, $m_2 - 1$ are linearly independent since $m_2 - 1$ belong to the basis. Let $F = \{t | t = \sum_{j \in J} A^{22}_{*j} y_j,\ y \geqq 0\}$ where $J = \{j | \sigma A^{22}_{*j} = 0\}$. It now suffices to observe that $F = \text{pos } A^{22} \cap \{t | \sigma t = 0\}$, that $\{t | \sigma = 0\}$ is a supporting hyperplane of pos A^{22} and that F has dimension $m_2 - 1$ since it contains $m_2 - 1$ linearly independent points. This completes the proof.

Thus, if it is possible to obtain a solution to (5) with only one artificial variable in the optimal basis, it follows that σ determines an $(m_2 - 1)$-dimensional face of pos A^{22}. The number of deficiency 1-faces of pos A^{22} is much smaller than the number of bases of $(A^{22}, I, -I)$ (see [11]). However, it is not always possible to obtain an $(m_2 - 1)$-face of pos A^{22}. In fact as is indicated in the next proposition, it is sometimes possible to obtain solutions to (5) such that

$$\{t | \sigma t = 0\} \cap \text{pos } A^{22} = \{0\}. \tag{25}$$

PROPOSITION 6. *Suppose* $x \notin K_2$ *and* $(b^2_i - A^{21}_{i*} x)$ *is different from zero for all* i *and that, for all* j,

$$\sum_{\substack{i \\ (b^2_i - A^{21}_{i*} x) > 0}} A^{22}_{ij} > \sum_{\substack{i \\ (b^2_i - A^{21}_{i*} x) < 0}} A^{22}_{ij} \tag{26}$$

holds; then no column A^{22}_{*j} *of* A^{22} *will figure in the optimal basis of* (5). A^{21}_{i*} *denotes the* i*-th row of* A^{21}.

Moreover, one should realize that the conclusion of the above proposition depends very much on the selection of cost coefficients $+1$ for the artificial variables in the infeasibility form. In fact any set of positive numbers could be selected as cost coefficients for the infeasibility form. Thus, an obvious complement to Proposition 6 is the following corollary.

COROLLARY 1. *Suppose* $x \notin K_2$ *and* $(b^2 - A^{21}_{i*} x)$ *is different from zero for all* i, *and for all* j *and all sets of positive numbers* $\mu_1, \cdots, \mu_{m_2}$ *the relation*

$$\sum_{\substack{i \\ (b_i - A^{21}_{i*} x) > 0}} \mu_i A^{22}_{ij} > \sum_{\substack{i \\ (b_i - A^{21}_{i*} x) < 0}} \mu_i A^{22}_{ij} \tag{27}$$

holds; then no column A^{22}_{*j} *of* A^{22} *will figure in the optimal basis of* (5). A^{21}_{i*} *denotes the* i*-th row of* A^{21}.

To see that the condition (27) is not vacuous, consider

$$A^{22} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \quad \text{and} \quad b^2 - A^{21} x = \begin{pmatrix} -1 \\ -1 \end{pmatrix}. \tag{28}$$

Obviously the condition (27) is much weaker than (26) since it allows for some perturbation of the coefficients of the objective functions in (5). It also indicates how one may modify (5) in order to increase the number of the columns of A^{22} figuring in the optimal basis. This would naturally increase the dimension of the face of pos A^{22} determined by the corresponding σ. In practice, this would involve a parametric study of the linear program (5). The constraints (22b) so generated would generally be "better" than those obtained by solving (5), but whether the extra computation is justified can probably be discovered only by experience in using the algorithm.

2.5. A "deepest cut" property. As we mentioned earlier, the constraints (8) obtained by solving (5) have a deepest cut property with interesting geometrical interpretations which we now examine. The linear program (5) can be interpreted as finding the nearest point in pos A^{22} to $d = b^2 - A^{21}x$ in the sense of the l_1-norm:

Minimize

$$\|z - d\|_1 \tag{29}$$

subject to

$$z \in \text{pos } A^{22} \subset \mathscr{R}^{m_2},$$

$$d = b^2 - A^{21}\bar{x},$$

where $\|z\|_1$ denotes the l_1-norm given by $\|z\|_1 = \sum_{i=1}^{m_2} |z_i|$.

The l_1-norm is defined on the space $\mathscr{R}^{m_2}$ of column m_2-vectors. Associated with $\mathscr{R}^{m_2}$ is its dual space $(\mathscr{R}^{m_2})^*$ which may be identified with all real-valued linear functions on $\mathscr{R}^{m_2}$. As is well known, any linear function $f(z)$ in $\mathscr{R}^{m_2}$ can be represented in a one-to-one way as a matrix product $\pi \cdot x$ of an m_2-dimensional row vector π and the column vector x. We shall thus think of $(\mathscr{R}^{m_2})^*$ as a space of row vectors with the same dimension as $\mathscr{R}^{m_2}$. A hyperplane, H, passing through the origin of $\mathscr{R}^{m_2}$ can be represented in the form $H = \{z | \pi z = 0\}$ for some $\pi \neq 0$ in $(\mathscr{R}^{m_2})^*$. However, this representation is not unique since $(\beta\pi)z = 0$ determines the same hyperplane for any real number $\beta \neq 0$. To resolve this ambiguity, we specify that $\|\pi\|_* = 1$, where $\|\cdot\|_*$ is a norm defined on $(\mathscr{R}^{m_2})^*$. This norm can be defined quite naturally [9] by using $\|\cdot\|_1$ on $\mathscr{R}^{m_2}$ by means of the following relation:

$$\|\pi\|_* = \max\{\pi z | \|z\|_1 \leqq 1\}. \tag{30}$$

It is easily seen that $\|\pi\|_* = \|\pi\|_\infty = \max_i |\pi_i|$, the l_∞-norm. A given π determines also a half-space $S = \{z | \pi z \leqq 0\}$ which is bounded by H. The condition that $\|\pi\|_\infty = 1$ and the specification of which half-space is to be determined uniquely defines π.

The dual of (5) is:

Maximize

$$\sigma[b^2 - A^{21}\bar{x}] \tag{31}$$

subject to

$$\sigma A^{22} \leqq 0,$$

$$|\sigma_i| \leqq 1, \qquad i = 1, \cdots, \bar{m}.$$

Let us interpret (31) in the language developed here. σ is an m_2-dimensional row vector which is an element of $(\mathscr{R}^{m_2})^*$. It determines a half-space S by

$$S = \{x | \sigma x \leqq 0\},$$

which includes pos A^{22}. This follows from the relation $z \in \text{pos } A^{22}$, implying that $z = A^{22}y$ for some $y \geqq 0$; hence $\sigma z = (\sigma A^{22})y \leqq 0$. The relations

$$|\sigma_i| \leqq 1, \qquad i = 1, \cdots, m_2,$$

are equivalent to $\|\sigma\|_\infty \leqq 1$. Of all elements of $(\mathscr{R}^{m_2})^*$ satisfying these conditions we are to find one which maximizes $\sigma[b^2 - A^{21}\bar{x}]$. Let us now examine the geometrical interpretation of maximizing $\sigma[b^2 - A^{21}\bar{x}]$.

The distance from a point $\bar{z}$ to a hyperplane H given by $H = \{z | \sigma z = 0\}$, or equivalently from the origin to the plane $H_{\bar{z}} = \{z | \sigma z = \sigma \bar{z}\}$, can be obtained by solving the linear program:

Minimize

$$ez^+ + ez^-$$

subject to

$$\sigma z^+ - \sigma z^- = \sigma \bar{z},$$

$$z^+, z^- \geqq 0.$$

The optimal solution is obviously determined by

$$z_\nu = z_\nu^+ - z_\nu^- = \frac{1}{\sigma_\nu}\sigma\bar{z}, \qquad z_i^+ = z_i^- = 0$$

for $i \neq \nu$, where $|\sigma_\nu| = \max_i |\sigma_i|$. Thus, the distance from $\bar{z}$ to H is

$$\frac{1}{\max|\sigma_i|}|\sigma\bar{z}| = \frac{1}{\|\sigma\|_\infty}|\sigma\bar{z}|.$$

Thus, problem (31) (i.e., the dual of (5)) can be interpreted as finding $\sigma \in (\mathscr{R}^{m_2})^*$ determining a supporting hyperplane of pos A^{22} which is as far as possible from $b^2 - A^{21}\bar{x}$ in the sense of the l_1-norm. Moreover, by the duality theory of linear programming, we have that this maximum distance is equal to the l_1-distance of $b^2 - A^{21}\bar{x}$ from pos A^{22}. Thus, in terms of the l_1-norm we have generated a "deepest cut."

3. The partial decomposition algorithm. A very natural approach to L-shaped programs is via the decomposition algorithm of Dantzig and Wolfe [8]. Nonetheless, if (1) has the properties mentioned in the Introduction, the straightforward application of the decomposition algorithm to problem (1) does not take advantage of the structure of the problem.

Decomposition can, however, be advantageously applied to the dual of problem (1):

Maximize

$$w = ub^1 + vb^2 \tag{32}$$

subject to

$$uA^{11} + vA^{21} \leqq c^1,$$
$$vA^{22} \leqq c^2,$$

where decomposition is done with respect to the coefficient vectors of the variables v. The coefficient vectors of the component u are retained unmodified:

Maximize

$$w = ub^1 + \sum \lambda_k \rho_k + \sum \mu_k \gamma_k \tag{33}$$

subject to

$$\begin{aligned} uA^{11} + \sum \lambda_k R_k + \sum \mu_k T_k &\leqq c^1, \\ \sum \lambda_k \qquad\qquad &= 1, \\ \lambda_k \geqq 0, \qquad \mu_k &\geqq 0, \end{aligned}$$

where $R_k = \pi^k A^{21}$ and $\rho^k = \pi^k b^2$ for a vertex π^k of the convex polyhedron determined by $\pi A^{22} \leqq c^2$; $T_k = \sigma^k A^{21}$ and $\gamma_k = \sigma^k b^2$ for an extreme ray σ^k of the convex polyhedron $\pi A^{22} \leqq c^2$. If we now take the dual of (33) assigning dual multipliers x_j to the first n_1 inequalities and θ to the last equation, we obtain the dual problem:

Minimize

$$z = c^1 x + \theta \tag{34}$$

subject to

$$\begin{aligned} A^{11}x \qquad &= b^1, && \\ R_k x + \theta &\geqq \rho_k, && k = 1, \cdots, t, \\ T_k x \qquad &\geqq \gamma_k, && k = 1, \cdots, s, \\ x &\geqq 0; && \end{aligned}$$

or equivalently:

Minimize

$$z = c^1 x + \theta$$

subject to

$$\begin{aligned} A^{11}x \qquad &= b^1, && \\ (\sigma^k A^{21})x \qquad &\geqq \sigma^k b^2, && k = 1, \cdots, s, \\ (\pi^k A^{21})x + \theta &\geqq \pi^k b^2, && k = 1, \cdots, t, \\ x &\geqq 0, && \end{aligned}$$

which corresponds to (22). Note that the feasibility constraints (22b) correspond to the extreme rays of the polyhedron $\pi A^{22} \leqq c^2$, whereas the optimality constraints (22c) correspond to extreme points of $\pi A^{22} \leqq c^2$. The constraints generated in Step 2′ of the cutting plane algorithm correspond to columns of (33) generated during the Phase I of this partial decomposition procedure. Thus, the algorithm which we developed here can be interpreted as a dual method of the Dantzig–Wolfe decomposition algorithm.

On the other hand, let us consider the L-shaped linear program in the equivalent form:

Minimize

$$c^1x + Q(x) \tag{35}$$

subject to

$$A^{11}x = b^1,$$

$$x \in K_2, \qquad x \geqq 0;$$

then our algorithm can be interpreted as a cutting plane algorithm [4], [16]. If A^{21} and A^{22} have a finite number of rows, K_2 is a polyhedral set and Q a convex polyhedral function. The method of [4], [16] can be used to establish the convergence of our algorithm in the case where the number of rows is infinite; alternatively, the results in [23] can be used to establish convergence using the interpretation of our algorithm as the dual of a decomposition procedure.

This is simply a reflection of the fact that the cutting hyperplane methods of Cheney and Goldstein [4], Goldstein [13], and Kelley [16] on the one hand and the decomposition methods of Dantzig and Wolfe [8], the algorithm associated with Wolfe's generalized program [5], [23], and in particular Dantzig's convex programming algorithm [5] on the other hand are simply dual methods to one another.

4. Optimal control with state constraints. A rather standard optimal control problem is:

Maximize

$$q_0(T) \tag{36}$$

subject to

$$\frac{dq}{dt} = B(t)q(t) + C(t)u(t),$$

$$q(0) = q^0,$$

$$q(T) \in L = \{q = (q_0, \cdots, q_n) | q_i = q_i^T, \quad i = 1, \cdots, n\},$$

$$q(t) \in Q(t) \subset \mathscr{R}^n,$$

$$u(t) \in U(t),$$

where $U(t)$ and $Q(t)$ are closed convex polyhedral sets.[3] We consider the discrete analogue of this system:

[3] The case where $U(t)$ is not polyhedral leads to algorithms which converge but are not finite. Problems for which $U(t)$ is not polyhedral are treated in [23]. Problems for which $Q(t)$ is not polyhedral can be treated by analogous devices.

Maximize

$$q_0^N$$

subject to

$$\frac{q^{i+1} - q^i}{\Delta} = B^i q^i + C^i u^i,$$

$$q^0 = q^0,$$

$$q^N \in L,$$

$$q^i \in Q^i,$$

$$u^i \in U^i,$$

$i = 0, \cdots, N - 1$, where $\Delta = T/N$, $q^i = q(i\Delta)$, and similarly for the other functions. Since $q^{i+1} = [I + \Delta B^i]q^i + \Delta C^i u^i$, we can now solve for each q^k inductively in terms of the initial state q^0 and the control sequence $u^0, \ldots, u^{k-1}$. Thus

$$q^1 = [I + \Delta B^0]q^0 + \Delta C^0 u^0,$$

$$q^2 = [I + \Delta B^1]\{[I + \Delta B^0]q^0 + \Delta C^0 u^0\} + \Delta C^1 u^1,$$

and, in general,

$$q^{k+1} = \left\{\prod_{j=k}^{0} [I + \Delta B^j]\right\} q^0 + [I + \Delta B^k] \cdots [I + \Delta B^1]\Delta C^0 u^0 + \cdots$$
$$+ [I + \Delta B^k]\Delta C^{k-1} u^{k-1} + \Delta C^k u^k.$$

Let

(37) $$Y[j, k] = [I + \Delta B^{k-1}][I + \Delta B^{k-2}] + \cdots + [I + \Delta B^j]$$

for $j < k$, $Y[j, j] = I$ and $\Delta C^k = E^k$. Then we have

(38) $$q^k = Y[0, k]q^0 + \sum_{j=0}^{k-1} Y[j + 1, k]E^j u^j.$$

Since Q^i and U^i are closed convex polyhedral sets, we can formulate the constraints on u^j in the form $F^{(j)}u^j \geqq f^{(j)}$, $j = 0, \cdots, N - 1$, and those on the state variables q as $G^{(j)}q^j \geqq g^{(j)}$. So now we have:

Maximize

$$q_0^N$$

subject to

$$U_0 q_0^N - \sum_{j=0}^{N-1} Y[j + 1, N]E^j u^j = Y[0, N]q^0 - q^T,$$

$$F^{(j)}u^j \geqq f^{(j)}, \qquad j = 0, \cdots, N - 1,$$

and the additional constraints

$$G^{(j)}q^j \geqq g^{(j)}, \qquad j = 0, \cdots, N-1,$$

where $q^T = (0, q_1^T, \cdots, q_n^T)$. If we let $u = [u^0, \cdots, u^{N-1}]$ and $A = [-Y(1, N)E^0, -Y(2, N)E^1, \cdots, -Y(N, N)E^k]$ and $b = Y[0, k]q^0 - q^T$, we have:

Maximize

$$q_0^N \tag{39}$$

subject to

$$\begin{aligned} U_0 q_0^N + Au &= b, \\ F^{(j)}u^j &\geqq f^{(j)}, \qquad j = 0, \cdots, N-1, \\ G^j q^j &\geqq g^{(j)}, \qquad j = 0, \cdots, N-1, \end{aligned}$$

where q^j is given by (38). The approach for handling the constraints $F^{(j)}u^j \geqq f^{(j)}$ by generalized linear programming has been described in [6] and [23]. Thus for simplicity, we limit ourselves to a discussion of the constraints $G^{(j)}q^j \geqq g^{(j)}$ and assume that the constraints $F^{(j)}u^j \geqq f^{(j)}$ on the u^j's simply reduce to the requirements that they are all nonnegative. We may now simplify the maximation problem just above to read:

Maximize

$$q_0^N \tag{40}$$

subject to

$$\begin{aligned} U_0 q_0^N + Au &= b, \\ G^{(j)}q^j &\geqq g^{(j)}, \qquad j = 0, \cdots, N-1, \\ u &\geqq 0. \end{aligned}$$

It is this problem which we interpret as an L-shaped program. The correspondence is $A \sim A^{11}$, $u \sim x$, and finally the slack variables of the *implicit* constraints on u, $G^{(j)}q^j \geqq g^{(j)}$, correspond to y. In this case, $c^2 = 0$ so that second-stage feasibility is the only requirement. Frequently from the physical nature of the problem it is clear that "usually" the state constraints will not be violated, and, of course, the values of the slack variables are of no particular use so that the representation of (40) as an L-shaped program seems particularly appropriate.

To generate the cut we simply evaluate π^j by $\pi_i^j = 1$ for $[g^{(j)} - G^{(j)}q^j]_i > 0$ and $\pi_i^j = 0$ otherwise. The cut is equal to $\sum \pi^j G^{(j)} q^j \geqq \sum \pi^j g^{(j)}$, which is the sum of the infeasible equations. All that remains is to express these in terms of the u's. In other words, we wish to evaluate

$$\sum_k \pi^k G^{(k)} \left[\sum_{j=0}^{k-1} Y[j+1, k] E^j u^j \right] \tag{41}$$

and the constant term

$$\sum \pi^k[g^{(k)} - Y[0,k]q^0].$$

This will give a new constraint $A_{n+1,*}u \geqq b_{n+1}$ which must be satisfied by u, where $A_{n+1,*}$ denotes the $(n+1)$th row of A; the special structure of (41), in particular, of the $Y[j,k]$ makes possible many simplifications in determining $A_{n+1,*}$ and b_{n+1} and, in particular, the relevant quantities would be accumulated as one determines π^j, rather than determining π^j and then going back to calculate $A_{n+1,*}$ and b_{n+1}. In addition, if the state space constraints are "loose," not many of the equations would be violated.

This application is an example of an important subclass of L-shaped programs which could be called I-shaped programs. These are L-shaped programs in which the components of the y-vector are simply slack variables.

The integer programming algorithm of Gomory [14] can be considered as another example of an I-shaped program, where $A^{21}x + Iy = b^2$, or, equivalently $A^{21}x \leqq b^2$, represents the infinite number of constraints which can be added to eliminate noninteger extreme points but do not eliminate any feasible integer points.

5. Stochastic programs with recourse. A stochastic program with recourse (random right-hand sides) also known as two-stage linear programs under uncertainty [7] reads:

Minimize

$$z = cx + E_\xi\{\min qy\} \tag{42}$$

subject to

$$Ax = b, \tag{42a}$$

$$Tx + Wy \geqq \xi, \qquad \xi \text{ on } (\Xi, z, F), \tag{42b}$$

$$x \geqq 0, \quad y \geqq 0.$$

The interpretation to be given to this problem, as well as the definition of the symbols, can be found in [25] or various other papers in this area (see, e.g., [7], [15]). Problem (42) is easily recognized to be an L-shaped program with possibly an infinite number of constraints (42b) and an infinite number of y-variables. We denote by $\tilde{\Xi}$ the support of the random variable ξ, i.e., the smallest closed subset of $\mathscr{R}^{\bar{m}}$ of measure one.

We shall assume that $\tilde{\Xi}$ has a least upper bound α, such that $\alpha \in \tilde{\Xi}$ and for all i, $\xi_i \leqq \alpha_i$ for all $\xi \in \tilde{\Xi}$. If this model is viewed as the representation of a physical decision process, the assumption that for each i there exists α_i such that $\xi_i \leqq \alpha_i$ seems to be very natural. The additional assumption that $\alpha \in \tilde{\Xi}$ is somewhat more restrictive. However, this would certainly be the case if the components of ξ were independent random variables and each ξ_i had compact (or bounded above) support. Extensions and a more complete discussion of these questions can be found in [24].

From a mathematical viewpoint the assumption that for each i there exists α_i is not so appealing; but if such an upper bound does not exist, then determining if problem (42) is feasible has to be dealt with differently, as can be seen from the following proposition.

PROPOSITION 7. *Suppose for some i there is no number α_i such that $\xi_i \leqq \alpha_i$ for all $\xi \in \tilde{\Xi}$. Then* (42) *is feasible only if the lineality space of* pos $(W, -I)$ *contains $\mathscr{R}_i$, where $\mathscr{R}_i$ is the i-th component of the Cartesian product $\mathscr{R}^{\bar{m}} = \prod_{j=1}^{\bar{m}} \mathscr{R}_j$ ($\mathscr{R}_j$ denotes the real line).*

Proof. If $\mathscr{R}_i \subset \text{pos}\,(W, -I)$, then the equation $(W, -I)_i \begin{pmatrix} y \\ s \end{pmatrix} = \zeta_i$ is solvable for all ζ_i with y and s nonnegative. Otherwise, for some ζ_i the above equation is not solvable. Since ξ_i has no upper bound, for any x there exists ξ in Ξ (determining ξ_i) such that the system

$$W_i y \geqq \xi_i - T_i x, \qquad y \geqq 0 \tag{43}$$

is inconsistent. This implies that for no x the recourse (or second-stage) problem is feasible for all ξ in $\tilde{\Xi}$; thus the set of feasible solutions to (42) is empty, i.e., (42) is infeasible. This completes the proof.

If the ξ_i's are independent and for certain i's, ξ_i has no greatest upper bound, we can use the preceding proposition to determine if (42) is infeasible. If the criterion is satisfied, we can ignore those equations whenever we verify whether a given x is feasible or not. In [28] the problem of characterizing and computing (which can be easily done) the lineality space of pos $(W, -I)$ has been dealt with in detail.

In the algorithm to be described below, we shall assume that for each i, α_i exists and $\alpha \in \tilde{\Xi}$. Proposition (16) in § 2B of [26] allows us to derive constraints [26, Equation (17), p. 96] of the form

$$(\sigma T)x \geqq \sigma\alpha, \tag{44}$$

which, in view of Proposition (16) of [26], play the same role as the feasibility constraints (8) play in the L-shaped linear program. Moreover, it has been shown that the feasibility region for the decision variables x determined by the induced constraints [26, p. 92] can be represented by a finite number of linear constraints [27, Proposition 12]. In § 2.4 of this paper we have shown the relation between the feasibility constraints (8) that we introduce and the supports of the cone pos $(W, -I)$. In [27] the accent has been placed on deriving an expression in terms of a minimal number of supports of pos $(W, -I)$ determined by the rows of the polar matrix [27], rather than an arbitrary finite collection of supports. As can be seen from Proposition 5, supports of maximum dimension corresponds to obtaining a particular solution to the linear program:

Minimize

$$ev^+$$

subject to

$$Wy + Iv^+ - Iv^- = \alpha - Tx^l,$$
$$y \geqq 0, \quad v^+ \geqq 0, \quad v^- \geqq 0.$$

These observations allow us to construct an algorithm which will find feasible solutions to (42) in a finite number of steps: by requiring that x satisfy the constraints (42a) to which we add a finite number of constraints of the form (44), each constraint being generated by solving one linear program; rather than verifying if for a particular x and *for all* $\zeta \in \tilde{\Xi}$ there exists a feasible y, i.e., $y \geqq 0$, such that $Wy \geqq \xi - Tx$.

We now outline a general algorithm for solving problem (42), general in the sense that we make no assumption on the structure of the matrices (in particular W) or on the form of the distribution of the random variable ξ, except that $\tilde{\Xi}$ have a greatest upper bound. (See Proposition 7 if this is not the case.) We ignore the special cases of infeasibility and unboundedness which are to be handled as before.

Step 1. Solve the linear program:

Minimize

$$cx + \theta$$

subject to

$$\begin{aligned}
&(45a) \qquad & Ax &= b, & \\
&(45b) \qquad & (\sigma^k T)x &\geqq \sigma^k\alpha, & k = 1, \cdots, s,\\
&(45c) \qquad & (\pi^k T)x + \theta &\geqq \rho^k, & k = 1, \cdots, t,\\
& & x &\geqq 0. &
\end{aligned}$$

Initially, s and t are zero. If no constraints of the form (45c) are present, θ is set equal to $-\infty$ and is ignored in the computation. Let x^l, θ^l be an optimal solution of (45).

Step 2. Solve the linear program to find

$$w^1 = \min ev^+ \tag{46}$$

subject to

$$Wy + Iv^+ - Iv^- = \alpha - Tx^l,$$
$$y \geqq 0, \quad v^+ \geqq 0, \quad v^- \geqq 0.$$

If $w^1 = 0$, go to Step 3. If $w^1 \neq 0$, the optimal multipliers σ^l are used to generate a cut of the form (45b).

Step 3. For all ξ in $\tilde{\Xi}$, solve the linear program:

$$w^2 = \min qy \tag{47}$$

subject to

$$Wy - Is = \xi - Tx^l,$$
$$y \geqq 0, \qquad s \geqq 0.$$

Each ξ determines an optimal π, say, $\pi^l(\xi)$. We then compute

$$w^2(x^l) = E_\xi\{\pi^l(\xi)(\xi - Tx^l)\},$$

$\pi^l = E_\xi\{\pi^l(\xi)\}$ and $\rho^l = E_\xi\{\pi^l(\xi)\xi\}$. If $w^2(x^l) \leqq \theta^l$, we terminate (optimality criterion). If not, we use π^l, ρ^l to generate a new constraint of the form (45c) which we now add to our problem (45) and return to Step 1.

We should also point out that in following this procedure, it is possible to generate an infinite number of constraints of the form (45c). Nevertheless, a result of K. Murty [17] allows us to keep $\bar{m}$ (T is $\bar{m} \times n$) or less constraints of the form (45b) and (45c) at each cycle, i.e., the constraints with nonzero slack can be removed.

We have separated Step 2 of the paraphrase (in § 2.3) of the cutting plane algorithm into two parts. The reason is that, in order to generate the feasibility cuts, we need only consider the upper bound of $\tilde{\Xi}$ (not all elements of $\tilde{\Xi}$), whereas we need complete information related to the probability space $(\Xi, \mathscr{F}, F)$ in order to compute π^l and ρ^l. Even when $\tilde{\Xi}$ has finite cardinality the labor so saved should be considerable. Moreover, if $\tilde{\Xi}$ has infinite cardinality, it is difficult to perform Step 3 unless the structure of W is such that it is possible to find a closed form expression for π^l and ρ^l (e.g., see [20] and [25]). The remaining part of this section is devoted to suggesting a method to circumvent this problem. We start by describing a variant of the above algorithm.

If ξ is an absolutely continuous random variable, we can modify the algorithm as follows:

Step 1. Solve the linear program:

Minimize

$$[c - \pi^{l-1}T]x \tag{48}$$

subject to

$$A^{11}x = b,$$

$$(\sigma^k T)x \geqq \sigma^k\alpha \qquad k = 1, \cdots, s,$$

$$x \geqq 0.$$

Initially $s = 0$ and $\pi^{l-1} = 0$. Let $\bar{x}^l$ be an optimal solution to (48). Find

$$\min_{0 \leqq \lambda \leqq 1} \psi(\lambda) = c[(1 - \lambda)x^{l-1} + \lambda\bar{x}^l] + Q[(1 - \lambda)x^{l-1} + \bar{x}^l],$$

where x^{l-1} was our previous solution ($x^0 = 0$) which for $l > 1$ was used to determine π^{l-1} and the function $Q(x)$ is as defined in [26, Equation (21)]. Say $\psi(\lambda^l) \leqq \psi(\lambda)$ for $\lambda \in [0, 1]$. If $\lambda^l = 0$, we terminate with the optimal solution x^{l-1} (optimality criterion). Let $x^l = (1 - \lambda^l)x^{l-1} + \lambda^l\bar{x}^l$.

Step 2. Proceed as above.

Step 3. Proceed as above to determine π^l, and then return to Step 1.

The convergence of this algorithm can be easily verified if we observe that from Proposition (29) and Corollary (28) in [26] it follows that if ξ is a continuous random variable then $Q(x)$ is a differentiable function with gradient $-\pi^l T$ at x^l. Thus, the above algorithm can be viewed as a variant of the Frank–Wolfe algorithm [10] for finding the minimum of a convex differentiable function on a convex polyhedral set. Since their procedure requires twice-differentiability and in general $Q(x)$ will not be of class C^2, their convergence proof does not strictly apply. The necessary modifications can be found in [25, Propositions (37), (41) and (43)].

The last algorithm as well as the first one we suggested, to solve stochastic programs with recourse, relies on the possibility of performing Step 3. If $\tilde{\Xi}$ does not have finite cardinality, this seems to be nearly impossible. However, one could exploit a suggestion of Dantzig and Madansky [7] which consists in sampling the distribution of ξ and solve Step 3 for some finite sample. This would naturally result in approximated values for π^l and ρ^l. As has already been pointed out in the Introduction, this approach would generate a constraint of type (45c) which would not necessarily be a support of the function $Q(x)$, and could very well eliminate the optimal solution (42) from the set of feasible solutions to (45). This inconvenience is completely eliminated if we follow the second procedure since all the constraints present in (48) never involve any approximation process.

We are however still left with two problems; First to solve Step 3 for a large (possibly very large) number of values of ξ in $\tilde{\Xi}$. Second, the resulting $\tilde{\pi}^l$ will not, in general, determine the gradient of $Q(x)$ at x^l, and thus the convergence properties of the algorithm are changed. This second problem will be the object of another paper in which various sampling techniques are examined and the convergence properties of the algorithm are established. We now show how to obtain the approximate value $\tilde{\pi}^l$ for π^l from a specific sample. Let $\xi^1, \cdots, \xi^N$ be a sample of size N obtained from the distribution of ξ. Our purpose is to solve (in Step 3) the N linear programs of the form:

Minimize

$$qy \tag{49}$$

subject to

$$Wy - Is = \xi^k - Tx^l, \qquad k = 1, \cdots, N,$$

$$y \geqq 0, \qquad s \geqq 0.$$

Since we are performing Step 3, $x^l \in K$; and since $\xi \in \tilde{\Xi}$, it follows that (49) is feasible for all ξ^k. Moreover if (49) is unbounded for some ξ^k, it is unbounded for all $\xi \in \tilde{\Xi}$; thus (42) is unbounded (see Proposition 2). Let us assume (49) is solvable. Let $\pi^l(\xi)$ denote the optimal simplex multipliers associated with solving (49) for a particular ξ. Then

$$\tilde{\pi}^l = \sum_{k=1}^{N} \frac{1}{N} \pi^l(\xi^k). \tag{50}$$

In the appendix of [26], we have reviewed the properties of the function $Q(t) = \{\min qy | Wy = t, y \geqq 0\}$; in particular, we shall use the fact that if $\{\pi(t)\}$

is the mapping determining the optimal simplex multipliers, then there exists a function $\pi(t)$ in $\{\pi(t)\}$ piecewise constant on pos W. In particular, if $W^{(i)}$ is an optimal basis corresponding to a particular value of t, then $W^{(i)}$ is also an optimal basis for all $t \in \text{pos } W^{(i)}$. Let $q^{(i)}$ be the subvector of q corresponding to $W^{(i)}$; then $\pi(t) = q^{(i)}W^{(i)-1}$ determines an optimal vector of multipliers for all $t \in \text{pos } W^{(i)}$. Applying this result to (49) we see that $\pi^l(\xi)$ is also optimal for all ξ^j such that $(\xi^j - Tx^l) \in \text{pos } W(\xi)$, where $W(\xi)$ is the optimal basis obtained from solving (49) for some fixed ξ. To determine if $(\xi^j - Tx^l) \in \text{pos } W(\xi)$ it is sufficient to verify that

$$W(\xi)^{-1}(\xi^j - Tx^l) \geqq 0.$$

This can be easily done since $W(\xi)^{-1}$ is available from the final optimal tableau.

We now give an algorithmic procedure to find $\tilde{\pi}^l$, as defined by (50).

Step a. Select an unbiased sample of size N from the distribution of ξ, say $\xi^1, \cdots, \xi^N$. Compute $\zeta^k = \xi^k - Tx^l, k = 1, \cdots, N$. By (ζ^j) we denote the set of available ζ^j and set $L = N$.

Step b. Select some ζ in $\{\zeta^j\}$ and set $\zeta = \zeta^k$ (initially $k = 1$) and solve the linear program:

Minimize

$$qy$$

subject to

$$Wy - Is = \zeta^k,$$

$$y \geqq 0.$$

Let $\pi(\zeta^k)$ be the optimal simplex multipliers and $W(\zeta^k)$ be the corresponding optimal basis.

Step c. Let $n(k)$ be the number of vectors ζ^j in the set $\{\zeta^j\}$ such that

$$W(\zeta^k)^{-1}\zeta^j \geqq 0. \tag{51}$$

Set $L = L - n(k)$; and if $L > 0$, return to Step b with $k = k + 1$ and delete from the set $\{\zeta^j\}$ those ζ^j which satisfied (51). If $L = 0$, terminate with

$$\tilde{\pi}^l = \frac{1}{N}\sum_k n(k)\pi(\zeta^k).$$

In returning to Step b it is suggested to select ζ (in the remaining set of $\{\zeta^j\}$) such that ζ fails to satisfy (51) only in a minimum number of components (if possible, one). Thus the previous basis would be the optimal basis for the new ζ^k up to very few dual simplex steps.

A few experiments have been made on an IBM 7094 (with a not nearly optimal code). We have selected $N = 3000$ and 5000, and $10 \leqq \overline{m} \leqq 40$ ($\overline{m}$ is the numbers of rows of W). In each case the computation of $\tilde{\pi}^l$ took never more than twice the time required to solve one linear program of the same size. In the same vein, a number of experiments have been conducted by Balintfy and Prekopa for random linear programs. In their manuscript [1] they show that numerous "tricks" can be performed to improve sampling procedures.

REFERENCES

[1] J. BALINTFY AND A. PREKOPA, *Simulation of basis stability in stochastic linear programs*, Computer Systems Research, Tulane University, New Orleans, Louisiana, 1965.

[2] ROBERT O. BARR, *Computation of optimal controls by quadratic programming on convex reachable sets*, Doctoral thesis, University of Michigan, Ann Arbor, 1966.

[3] J. F. BENDERS, *Partition procedures for solving mixed-variables programming problems*, Numer. Math., 4 (1962), pp. 238–252.

[4] E. W. CHENEY AND A. A. GOLDSTEIN, *Newton's method of convex programming Tchebycheff approximation*, Ibid., 1 (1959), pp. 253–268.

[5] G. B. DANTZIG, *Linear Programming and Extensions*, 2nd ed., Princeton University Press, Princeton, 1965.

[6] ———, *Linear control processes and mathematical programming*, SIAM J. Control, 4 (1966), pp. 56–60.

[7] G. B. DANTZIG AND ALBERT MADANSKY, *On the solution of two-stage linear programs under uncertainty*, Proc. Fourth Berkeley Symposium on Mathematical Statistics, vol. 1, University of California, Berkeley, 1961, pp. 165–176.

[8] G. B. DANTZIG AND PHILIP WOLFE, *The decomposition algorithm for linear programs*, Econometrica, 29 (1961), pp. 767–778.

[9] W. FLEMING, *Functions of Several Variables*, Addison-Wesley, Reading, Massachusetts, 1965.

[10] M. FRANK AND P. WOLFE, *An algorithm for quadratic programming*, Naval Res. Logist. Quart., 3 (1956), pp. 95–110.

[11] D. GALE, *On the number of faces of a convex polytope*, Canad. J. Math., 16 (1964), pp. 12–17.

[12] E. G. GILBERT, *An iterative procedure for computing the minimum of a quadratic form on a convex set*, SIAM J. Control, 4 (1966), pp. 61–80.

[13] A. A. GOLDSTEIN, *Convex programming and optimal controls*, Ibid., 3 (1965), pp. 142–146.

[14] R. E. GOMORY, *An algorithm for integer solutions to linear programs*, Recent Advances in Mathematical Programming, McGraw-Hill, New York, 1963, pp. 269–302.

[15] P. KALL, *Qualitative Aussagen zu einigen Problemen der stochastischen Programmierung*, Z. Wahrscheinlichkeitstheorie und Verw. Gebiete, 6 (1966), pp. 246–272.

[16] J. E. KELLEY, *The cutting-plane method for solving convex programs*, this Journal, 8 (1960), pp. 703–712.

[17] K. G. MURTY, *Two-stage linear program under uncertainty: A basic property of the optimal solution*, Z. Wahrscheinlichkeitstheorie und Verw. Gebiete, 10 (1968), pp. 284–288.

[18] L. NEUSTADT, *Discrete time optimal control systems*, Nonlinear Differential Equations and Nonlinear Mechanics, J. P. LaSalle and S. Lefshetz, eds., Academic Press, New York, 1963, pp. 267–283.

[19] B. N. PSHENICHNIY, *Linear optimal control problems*, SIAM J. Control, 4 (1966), pp. 577–593.

[20] A. PREKOPA, *On the probability distribution of the optimum of a random linear program*, Ibid., 4 (1966), pp. 211–222.

[21] J. B. ROSEN, *Iterative solution of nonlinear optimal control problems*, Ibid., 4 (1966), pp. 223–244.

[22] ———, *Optimal control and convex programming*, Non-Linear Programming, J. Abadie, ed., North-Holland, Amsterdam, 1967, pp. 289–302.

[23] R. M. VAN SLYKE, *Mathematical programming and optimal control*, Rep. ORC68-21, Operations Research Center, University of California, Berkeley, 1968.

[24] ROGER WETS, *Stochastic Programming: A Survey. I*, In preparation.

[25] ———, *Programming under uncertainty: The complete problem*, Z. Wahrscheinlichkeitstheorie und Verw. Gebiete, 4 (1966), pp. 316–339.

[26] ———, *Programming under uncertainty: The equivalent convex program*, this Journal, 14 (1966), pp. 89–105.

[27] ———, *Programming under uncertainty: The solution set*, this Journal, 14 (1966), pp. 1143–1151.

[28] R. WETS AND C. WITZGALL, *Algorithms for frames and lineality spaces of cones*, J. Res. Nat. Bur. Standards Sect. B, 71 (1967), pp. 1–7.

[29] B. H. WHALEN, *On linear programming and optimal control*, Doctoral thesis, University of California, Berkeley, 1962.

[30] ———, *On linear programming and optimal control*, IRE Trans. Automatic Control, 7 (1962), pp. 45–46.

[31] L. A. ZADEH, *A note on linear programming and optimal control*, Ibid., 7 (1962), p. 46.