ALGORITHMS FOR STOCHASTIC PROGRAMS: THE CASE OF NONSTOCHASTIC TENDERS

Larry Nazareth
Roger J.-B. Wets

January 1983
WP-83-5

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
2361 Laxenburg, Austria

ABSTRACT

We consider solution strategies for stochastic programs whose deterministic equivalent programs take on the form: Find $x \in R^n$, $\chi \in R^m$ such that $x \geq 0, Ax = b$, $Tx = \chi$ and $z = cx + \Psi(\chi)$ is minimized.

We suggest algorithms based upon (i) extensions of the revised simplex method, (ii) inner approximations (generalized programming techniques), (iii) outer approximations (min-max strategies). We briefly discuss implementation and associated software considerations.

ALGORITHMS FOR STOCHASTIC PROGRAMS: THE CASE OF NONSTOCHASTIC TENDERS

Larry Nazareth and Roger J.-B. Wets

1. INTRODUCTION

We report on some approaches to solving certain classes of stochastic programming problems. The main purpose is to provide a basis for discussion with a view to identifying promising algorithms that could eventually be realized as software.

The subclass of stochastic programs (with recourse) that we have in mind, and to which we refer as having nonstochastic tenders, arise as models for the following decision process. An (optimal) decision vector x must be selected when some of the parameters of the problem are only known in probability, i.e. only in a statistical sense, the actual cost depending in part on how well a transformation of x, $\chi = Tx$ matches a random demand or recourse vector p.

We think of χ as a *tender*, *nonstochastic* if the transformation T does not depend on the (unknown) values of the random parameters. For example, stochastic programs with simple recourse and fixed technology matrix are of this type. As we shall see in Section 2, for stochastic (linear) programs, the equivalent deterministic program can then be expressed as:

(1.1) Find $x \in R^n$, $\chi \in R^m$ such that

$$Ax = b \ , \ Tx = \chi \ , \ x \geq 0 \ ,$$

and $z = cx + \Psi(\chi)$ is minimized .

The algorithms that we analyze could be viewed as procedures for convex programs of the type (1.1) that seek to take advantage of the special structure, and to some extent that view is certainly correct. In fact we expect that the suggested techniques will also be efficient whenever nonlinear optimization problems can be cast in the form (1.1). However, because stochastic programming problems present computational challenges of their own, it is their specific properties that are always in the background of our solution strategies. For example, our title is intended to suggest that the major task of the solution procedure is the finding of optimal or nearly optimal tenders.

In Section 2, we review briefly the properties of stochastic programs that will be used in the design of algorithmic procedures. In Section 3, we examine the issue of what information can be made available and its cost, and we also exhibit some important special cases when the objective and the underlying distribution functions are such, that the equivalent deterministic programs can be conveniently and inexpensively specified. We

then turn to the three main methods that we consider here. They are based upon

(i) extensions of the revised simplex method,

(ii) inner approximations (generalized programming techniques),

(iii) outer approximations (min-max strategies).

In order to give the essence of each solution strategy, we consider first, in Section 4, a very simple case, viz., equivalent linear programming formulations for finding the minimum of a convex piecewise linear function of one variable. In Sections 5, 6 and 7 we go into each approach as it applies to our class of stochastic programming problems. Each section is organized along similar lines as follows: the case of simple recourse is considered in detail, extensions to problems with complete recourse are briefly outlined and finally some comments are made related to software choices and implementation.

2. STOCHASTIC PROGRAMS WITH RECOURSE: NONSTOCHASTIC TENDERS

We consider stochastic (linear) programs of the type

(2.1) find $x \in R^{n_1}$ such that

$Ax = b$, $x \geq 0$

and $z = E\{c(w)x + Q(x,w)\}$ is minimized ,

where Q is calculated by finding for given decision x and event w, an optimal recourse $y \in R^{n_2}$, viz.

(2.2) $Q(x,w) = inf_{y \in C}[q(y,w) \mid Wy = p(w) - Tx]$.

Here $A(m_1 \times n_1)$, $T(m_2 \times n_1)$, $W(m_2 \times n_2)$ and $b(m_1)$ are given (fixed) matrices, $c(\cdot)(n_1)$ and $p(\cdot)(m_2)$ are random vectors, $y \mapsto q(y,\cdot): R^{n_2} \mapsto R$ is a random finite-valued convex function and C is a convex polyhedral subset of R^{n_2}, usually $C = R^{n_2}_+$. Because W is nonstochastic one refers to (2.1) as having *fixed recourse*. Tenders are nonstochastic because T is fixed. (Strictly speaking nonstochastic tenders allow for the possibility of having W random. However because of the computational intractability of that case, it will not be considered here.) With

$$c = E\{c(w)\} \text{ and } Q(x) = E\{Q(x,w)\}$$

we obtain the equivalent deterministic form of (2.1):

(2.3) find $x \in R^{n_1}$ such that

$Ax = b$, $x \geq 0$

and $z = cx + Q(x)$ is minimized .

We assume that the random elements of the problem are such that all quantities introduced are well-defined, with $Q(x)$ finite, unless

$$\text{Prob.}\{w \mid (p(w) - Tx) \notin W(C)\} > 0$$

where $W(C) = \{t = Wy \mid y \in C\}$, i.e. there is no feasible recourse with positive probability, in which case $Q(x) = +\infty$. Detailed conditions have been made explicit in [1]; extensions to the multistage case have been provided by P. Olsen [2], consult also [3] for some results in the nonconvex case.

As background to the algorithmic development, we review the basic properties of (2.3), proofs and further details can be found in [1]; see also [4] for a compact treatment for stochastic programs with *complete*

recourse, i.e. when $W(C) = R^{m_2}$ and thus Q is everywhere finite.

2.4. *PROPERTIES. The function* Q *is lower semicontinuous and convex. It is Lipschitz if for (almost) all* w, $y \longmapsto q(y,w)$ *is Lipschitz. Also the set*

$$K_2 = \{x \mid Q(x) < +\infty\}$$

is a convex polyhedron that can be expressed as

$$K_2 = \{x \mid Dx \geq d\}$$

for some matrix D *and vector* d. *Moreover if the distribution of the random elements of the problem is absolutely continuous then* Q *is differentiable relative to* K_2.

Because $q(\cdot,w)$ is Lipschitz rather than linear, the assertion about Q being Lipschitz does not follow directly from Theorem 7.7 of [1] but can be gathered from its proof, or see [5], for example.

In the case of nonstochastic tenders it is useful to consider another representation of the deterministic equivalent program. Let

$$\chi := Tx \quad ,$$

$$\psi(\chi,w) = \inf_{y \in C} [q(y,w) \mid Wy = p(w) - \chi]$$

and

$$\Psi(\chi) = E\{\psi(\chi,w)\} \quad .$$

Problem (2.3) is then cast in the form (1.1):

(2.5) find $x \in R^{n_1}$, $\chi \in R^{m_2}$ such that

$Ax = b$, $Tx = \chi$, $x \geq 0$

and $z = cx + \Psi(\chi)$ is minimized.

This program, more exactly the function Ψ, exhibits the same properties as those listed for Q under Properties 2.4. In particular it is finite for all χ such that $\chi = Tx$ and $x \in K_2$. Including these constraints explicitly in the formulation of the problem, we get

(2.6) find $x \in R^{n_1}$, $\chi \in R^{m_2}$ such that

$z = cx + \Psi(\chi)$ is minimized, and

$$Ax = b \ ,$$

$$Dx \geq d \ ,$$

$$Tx - \chi = 0 \ ,$$

$$x \geq 0,$$

i.e. a convex program with Ψ finite on the feasible region. In what follows we shall assume that the constraints $Dx \geq d$ have been incorporated in the constraints $Ax = b$, $x \geq 0$, so that they will no longer appear explicitly, and that Ψ is finite on

$$\{\chi = Tx \mid Ax = b \ , x \geq 0\} \quad .$$

Stochastic programs of this type are said to have relatively complete recourse [1, Section 6], a situation which is always obtained if the (induced) constraints, determining K_2, are incorporated in the original constraints.

When $W = I$ and $C = R^{m_2}$, there is really no need to solve an optimization problem to know the optimal recourse and its associated cost. It is uniquely determined by the relation

$$y = p(w) - \chi \ , \ Tx = \chi$$

and

$$\Psi(\chi) = E\{q(p(w) - \chi, w)\}$$

The stochastic program is then said to be with *simple recourse*, which clearly implies complete recourse: $K_2 = R^{m_1}$. Determining the value of Ψ at χ depends then on our capability of performing the multidimensional integration. Usually, the cost-function will be separable. However, if there is dependence between some of the components of the $p(\cdot)$-vector and the cost depends on the joint realizations, then one must necessarily resort to this more general form. Assuming that the integral is well-defined, we have that the subdifferential of Ψ is given by

$$\partial\Psi(\chi) = -E\{\partial_y q(p(w) - \chi, w)\}$$

where $\partial_y q(\cdot, w)$ denotes the subdifferential with respect to the first variable. It is easy to see that if the convex function $y \longmapsto q(y, w)$ is differentiable, then so is Ψ. The function Ψ is also differentiable if the measure is absolutely continuous. If the random variables are independent, then the multidimensional integration to obtain the value of Ψ or its gradient is reduced to a number of simple integrals on R^1. This also occurs when there is separability.

If in addition to simple recourse, the recourse costs are *separable* , i.e. for all w

$$q(y, w) = \sum_{i=1}^{m_2} q_i(y_i, w) \quad ,$$

then

$$\Psi(\chi) = E \sum_{i=1}^{m_2} q_i(p_i(w) - \chi_i, w)$$

$$= \sum_{i=1}^{m_2} Eq_i(p_i(w) - \chi_i, w)$$

$$= \sum_{i=1}^{m_2} \Psi_i(\chi_i)$$

Thus (2.5) becomes a convex *separable* program:

(2.7) find $x \in R^{n_1}$, $\chi \in R^{m_2}$ such that

$Ax = b$, $Tx = \chi$, $x \geq 0$.

and $cx + \sum_{i=1}^{m_2} \Psi_i(\chi_i)$ is minimized .

This latter optimization problem possesses many properties. Those that are directly relevant to our further development are summarized here below.

2.8 PROPERTIES. *For $i = 1,\ldots,m_2$, the functions Ψ_i are convex, finite-valued and thus continuous. If the random elements have a discrete distribution, the Ψ_i are piecewise linear when the $q_i(\cdot,w)$ are piecewise linear. On the other hand, if the marginals of $\{(q_i(\cdot,w), p_i(w)), i = 1,\ldots,m_2\}$ are absolutely continuous then the Ψ_i are differentiable. Moreover, if problem* (2.7) *is solvable it admits an optimal solution with no more than $m_1 + m_2$ positive entries in the x-vector.*

These properties are derived in [6], (see also [7]), except the last assertion which was obtained by Murty [8] in a somewhat modified context; a very simple proof appears in [9].

A version of (2.7) which has received a lot of attention, because of its direct amenability to efficient computational schemes and the many applications that can be cast in this form, is when q_i is itself independent of w and piecewise linear with respect to y. More precisely $q_i(\cdot)$ is given by

(2.9)
$$q_i(y_i,w) = \begin{cases} -q_i^+ y_i & \text{if} \quad y_i \le 0 \\ q_i^- y_i & \text{if} \quad y_i \ge 0 \end{cases},$$

with $q_i = q_i^+ + q_i^- \ge 0$, yielding the convexity of $y_i \longmapsto q_i(y_i,w)$. In this case the function Ψ_i takes on a form particularly easy to describe. This is done in the next section.

3. AVAILABILITY OF INFORMATION ABOUT THE OBJECTIVE

The exact evaluation of Q or its gradient for general probability distribution μ, function q and recourse matrix W, might be prohibitively expensive, if at all possible. The difficulties come from two directions:

(i) for each w, having to evaluate $Q(x,w)$ which involves solving a minimization problem, and

(ii) having to perform the multidimensional integration

$$Q(x) = \int Q(x,w)\mu(dw) \quad .$$

For simple recourse, the evaluation of $Q(x,w)$, or equivalently $\psi(\chi,w)$, because T is fixed, is easy since the recourse is uniquely determined. When the recourse costs are also separable, the multidimensional integration is reduced to m_2 separate 1-dimensional integrals. With T fixed, it takes the form:

$$\Psi(\chi) = \sum_{i=1}^{m_g} \Psi_i(\chi_i) = \sum_{i=1}^{m_g} \int q_i(p_i(w) - \chi_i, w) F_i(dw)$$

where F_i is the marginal distribution function of the random elements appearing in this expression, and the integral $\int$ is a Lebesque-Stieltjes integral. The subgradients of the convex function Ψ are then the (Cartesian) product of the subgradients of the Ψ_i which are themselves

$$\partial\Psi_i(\chi_i) = -\mathrm{cl} \int \partial_y q_i(p_i(w) - \chi_i , w) F_i(dw) \quad . \tag{3.1}$$

at least when the problem satisfies the regularity conditions suggested at the beginning of Section 2. In general $\partial_y q(p_i(w) - \chi_i , w)$ is multivalued, in fact closed convex valued, and the integral is then also a closed convex set. In particular if q is piecewise linear as in (2.9), we get the following expression:

$$\partial\Psi_i(\chi_i) = [q_i F_i^-(\chi_i) - q_i^+ , q_i F_i(\chi_i) - q_i^+] \quad , \tag{3.2}$$

where $q_i = q_i^+ + q_i^-$,

$$F_i^-(z) = \mathrm{Prob.}[p_i(w) < z] \ ,$$

and

$$F_i(z) = \mathrm{Prob.}[p_i(w) \leq z] \quad .$$

We just note in passing that this implies that Ψ_i is differentiable whenever the distribution function F_i is continuous. In general, we have the two following representations for Ψ_i

$$\Psi_i(\chi_i) - q_i^+ \bar{p}_i = (q_i F_i^-(\chi_i) - q_i^+)\chi_i - q_i \int_{\zeta < \chi_i} \zeta dF_i(\zeta) \tag{3.3}$$

$$= (q_i F_i(\chi_i) - q_i^+)\chi_i - q_i \int_{\zeta \leq \chi_i} \zeta dF_i(\zeta)$$

where $\bar{p}_i = E\{p_i(\cdot)\}$. If the distribution of $p_i(\cdot)$ is discrete, say with possible values

$$p_{i1}, p_{i2}, \ldots, p_{i,k_i} \quad ,$$

with $p_{il} < p_{i,l+1}$, and with associated probabilities

$$f_{i1}, f_{i2}, \ldots, f_{i,k_i} \quad ,$$

the function Ψ_i is piecewise linear. With $\sum_{l=0}^{-1} = 0$, we have

$$F_i^-(\chi_i) = \sum_{l=1}^{\tau'-1} f_{il} \quad \text{where} \quad \tau' = \min\,[k_i\,,\, \inf(t \mid p_{it} \geq \chi_i)] \quad ,$$

and

$$F_i(\chi_i) = \sum_{l=1}^{\tau-1} f_{il} \quad \text{where} \quad \tau = \min\,[k_i\,,\, \inf\,(t \mid p_{it} > \chi_i)].$$

Also

$$\int_{\zeta<\chi_i} \zeta dF_i(\zeta) = \sum_{l=1}^{\tau'} p_{il} f_{il} \quad ,$$

and

$$\int_{\zeta\leq\chi_i} \zeta dF_i(\zeta) = \sum_{l=1}^{\tau} p_{il} f_{il} \quad ;$$

Note that $\tau' = \tau$ unless $\chi_i = p_{it}$ for some $t = 1,\ldots,k_i$ and then $\tau' = \tau - 1$.

For $l = 0,\ldots,k_i$, we set

$$s_{il} = (\sum_{t=1}^{l} f_{it})q_i - q_i^+ \quad ,$$

and

$$e_{il} = q_i^+\bar{p}_i - q_i(\sum_{t=1}^{l} p_{it} f_{it}) \quad .$$

We thus get

$$(3.4) \qquad \Psi_i(\chi_i) = \sup_{l=0,\ldots,k_i} (s_{il}\chi_i + e_{il}) \quad .$$

Observe that for any value of χ_i the supremum is attained by at most 2 linear forms. As we shall see in the subsequent sections, both (3.3) and (3.4) yield useful representations for Ψ_i when developing algorithmic procedures for problems involving functions of this type. Still another representation of Ψ_i can be exploited in an algorithmic context. Here the value of Ψ_i is obtained as the solution of an optimization problem parametrized by χ_i. Let

$$d_{il} = p_{i,l+1} - p_{il} \quad , \quad \text{for} \quad l = 1,\ldots,k_{i-1} \quad ,$$

and

$$d_{i0} = p_{i1} \quad .$$

Then

$$(3.5) \qquad \Psi(\chi_i) - q_i^+\bar{p}_i = \inf \sum_{l=0}^{k_i} s_{il}\, y_{il}$$

$$\text{subject to} \quad \sum_{l=0}^{k_i} y_{il} = \chi_i \quad ,$$

$$y_{i0} \le d_{i0} \quad ,$$

$$0 \le y_{il} \le d_{il} \quad , \; l = 1,\ldots,k_{i-1}$$

$$0 \le y_{i,k_i} \quad .$$

To verify (3.5) it suffices to use the fact that the coefficients s_{il} , $l = 0,\ldots,k_i$ are strictly increasing and that consequently $y_{il} > 0$ only if

$y_{i0} = d_{i0}$ and for $0 < t < l$, all y_{it} are at their upper bounds. Details are worked out in [10, Proposition 1].

These expressions derived for Ψ_i taken in conjunction with the methods of Section 4 contain the germ of different algorithmic procedures embedded in them.

Before we turn to this, and in order not to lose sight of the fact that we are also interested in a more general class of problems, not simply stochastic programs with simple recourse with piecewise linear separable cost structure, we also describe a more general case. Suppose

$$\psi(\chi,w) = \inf_{y \geq 0} [qy \mid Wy = p(w) - \chi]$$

and χ is such that $\psi(\chi,w)$ is finite for all possible $p(w)$, i.e. the linear program defining $\psi(\chi,w)$ is feasible and bounded with probability 1. Then parametric analysis, in particular the Basis Decomposition Theorem [1], shows that there is a (simplicial) decomposition of the sample space of $p(\cdot)$ (= the activity space),

$$S = \{S_h \subset R^{m_2}, h = 1,\ldots,l\}$$

such that if $p(w) \in S_h$ then

$$\psi(\chi,w) = q_{(h)} W_{(h)}^{-1} (p(w) - \chi) \quad , \tag{3.6}$$

and with co denoting the convex hull,

$$\partial\psi(\chi,w) = - co \{q_{(h)} W_{(h)}^{-1} \mid p(w) \in S_h\} \quad , \tag{3.7}$$

where $W_{(h)}$ is an invertible submatrix of W and $q_{(h)}$ the subvector of q corresponding to the columns of $W_{(h)}$. Let S' be any partition generated by S. Then

$$\Psi(\chi) = \sum_h \int_{S_h'} q_{(h)} \, W_{(h)}^{-1} \, (p(w) - \chi) P(dw)$$

and

$$\partial\Psi(\chi) = \sum_h \int_{S_h'} \partial\psi(\chi, w) P(dw) \quad .$$

adding to the second term the normal cone to K_2 at χ if χ is on the boundary of K_2. (We really only need the above, the rest being taken care of through the constraints.) The potential use of the preceding formulas depends very much on how accurate one needs to be. Multidimensional integration over convex polyhedral cones can really only be approached through sampling methods, cf. [11],[12] and the references given therein.

If $p(\cdot)$ is discretely distributed by which we mean here that it takes on a finite number of possible values, with

$$P[p(w) = p_k] = f_k \, , \, k = 1,\ldots,N \quad ,$$

then the above formulas become simply sums, viz.

$$\Psi(\chi) = \sum_h \sum_{\{k \,|\, p_k \in S_h'\}} q_{(h)} \, W_{(h)}^{-1} \, (p_k - \chi) f_k \tag{3.8}$$

and a similar expression for $\partial\Psi$. To actually compute the above we can proceed via a sort of parametric analysis that we now describe. We refer to it as a ***bunching*** procedure. Let

$$\zeta_k = p_k - \chi \, , \, k = 1,\ldots,N \quad ,$$

and suppose we have solved the linear program

(3.9) find $y \geq 0$ such that $Wy = \zeta_1$ and qy is minimized,

with optimal basis $W_{(1)}$ and associated subvector $q_{(1)}$ of q. S_1', bunch 1, is defined by

$$S_1' = \{p_k \mid W_{(1)}^{-1}\ \zeta_k \geq 0\}$$

While constructing this set, identify those $\zeta_k \notin S_1'$ such that the vector $W_{(1)}^{-1}\ \zeta_k$ has the fewest number (and smallest) negative elements. Let ζ_{k_2} be such a vector. We find the optimal solution and a corresponding basis $W_{(2)}$ of the linear program (3.9) with ζ_{k_2} replacing ζ_1, by dual simplex pivoting, starting with the old basis $W_{(1)}$. The second bunch S_2' is given by

$$S_2' = \{p_k \notin S_1' \mid W_{(2)}^{-1}\ \zeta_k \geq 0\} \quad .$$

We continue in this fashion until all p_k have been bunched. Alternative procedures can be devised taking further advantage of the combinatorial structure of decompositions, how to do this so as to minimize the work involved needs further investigation. In any case with the above we obtain the value of Ψ at χ, as well as a subgradient of Ψ at χ, viz.

$$\sum_h \sum_{\{k \mid p_k \in S_h'\}} (q_{(h)}\ W_{(h)}^{-1}) f_k \in \partial\Psi(\chi) \quad .$$

Observe that $q_h\ W_{(h)}^{-1}$, the vector of simplex multipliers, remains constant on S_h'. With

$$P_h = P[p_k \in S_h']$$

the above becomes

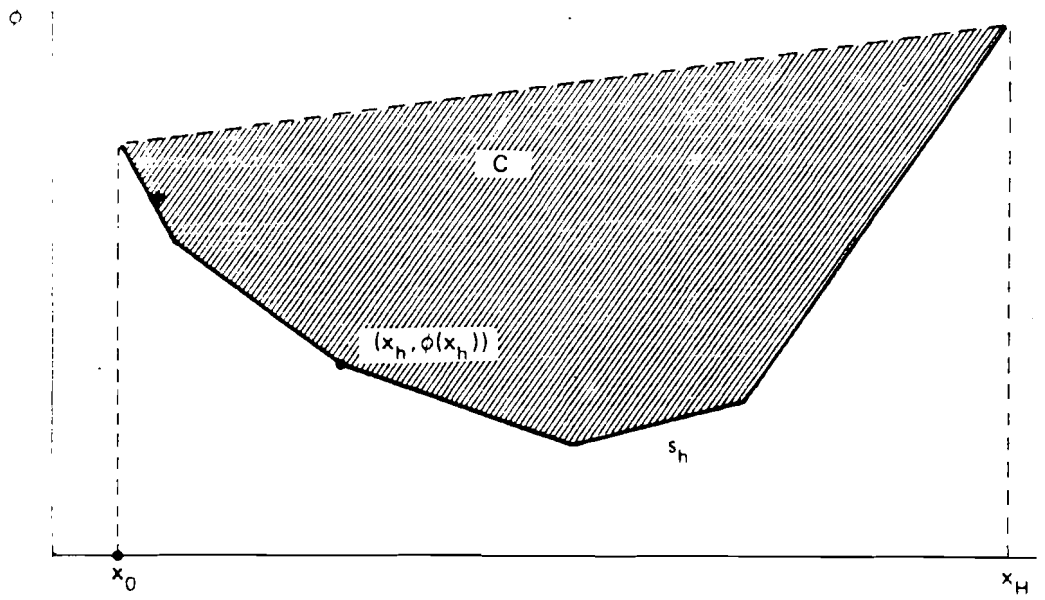
$$\sum_h q_{(h)}\ W_{(h)}^{-1}\ P_h \in \partial\Psi(\chi)$$

This formula for a subgradient of Ψ is, in fact, independent of the form of

the distribution of $p(\cdot)$, the problem being always the evaluations of P_h for a partitioning scheme constructed in the manner described above.

4. AN ILLUSTRATION OF EACH ALGORITHMIC APPROACH

We consider the very simple problem of finding the unconstrained minimum of a 1-dimensional finite piecewise linear convex function φ defined on $[x_0, x_H]$ by reformulating the problem as an equivalent linear program. (This function φ could of course be minimized by some 1-dimensional search procedure or simply by a sort of the slopes to find where they change sign, but this is not our real concern here.) There are at least three ways of formulating this equivalent linear program. Each contains the germ of a more general solution strategy considered in later sections.



4.1 Figure: The function φ

The coordinates of the breakpoints of φ are denoted by

$$(x_h, \varphi(x_h)) \ , \ h = 0,\ldots,H$$

and with slopes

$$s_h \text{ for } x \in [x_{h-1} \ , \ x_h] \ , \ h = 1,\ldots,H \quad .$$

The convexity of φ implies that

(4.2) $$s_1 \leq s_2 \leq \cdots \leq s_H \quad .$$

With

$$\alpha_h = x_h - x_{h-1} \text{ and } e_h = \varphi(x_h) - s_h x_h \ ,$$

the line segment on $[x_{h-1} \ , \ x_h]$ takes the form

(4.3) $$\varphi_h(x) := s_h x + e_h$$

The bounded variable method

If we introduce a new variable y_h for each interval $[x_{h-1} \ , \ x_h]$ for any given value of x it is easily verified that on $[x_0 \ , \ x_H]$,

(4.4) $$\varphi(x) = \varphi(x_0) + \min \left[\sum_{h=1}^{H} s_h y_h \,|\, x = x_0 + \sum_h y_h \ , \ 0 \leq y_h \leq \alpha_h \ , \ h = 1,\ldots,H \right]$$

The assumption of convexity and hence (4.2) is of course crucial, since this means that y_1 is preferred to y_2, and y_2 to y_3 and so on in the minimization of (4.4). Hence at the optimum point in (4.3), $y_h > 0$ implies

$$y_1 = \alpha_1 \, , \, y_2 = \alpha_2, \ldots, y_{h-1} = \alpha_{h-1} \quad .$$

Minimizing φ on $[x_0 \, , \, x_H]$ is equivalent to solving the following linear program:

(4.5) find $y_h \in [0 \, , \, \alpha_h]$, for $h = 1,\ldots,H$,

such that $z = \sum_{h=1}^{H} s_h y_h$ is minimized.

The optimal x^* is determined by

$$x^* = x_0 + \sum_h y_h^*$$

where $(y_h^* \, , \, h = 1,\ldots,H)$ is the optimal solution of (4.5).

Inner approximation

Referring to Figure 4.1, any point $(x \, , \, \alpha)$ in the shaded region C, i.e. with $\alpha \geq \varphi(x)$, can be written as a convex combination of the extreme points

$$(x_h \, , \, \varphi(x_h)) \, , \, h = 0,\ldots,H$$

For any given x it follows that

(4.6) $$\varphi(x) = \min \left[\sum_{h=0}^{H} \lambda_h \varphi(x_h) \,|\, \sum \lambda_h x_h = x \, , \, \sum \lambda_h = 1 \, , \, \lambda_h \geq 0 \right]$$

and thus minimizing φ on $[x_0 \, , \, x_H]$ is equivalent to solving the following linear program

(4.7) find $\lambda_h \geq 0 \, , \, h = 1,\ldots,H \, , \, \sum_{h=1}^{H} \lambda_h = 1$,

such that $z = \sum_{h=0}^{H} \lambda_h \varphi(x_h)$ is minimized.

The optimal x^* is determined by

$$x^* = \sum_{h=0}^{H} \lambda_h^* x_h$$

where $(\lambda_h^*, h = 0,\ldots,H)$ is the solution of (4.7). For an arbitrary finite convex function ρ with $\rho(x_h) = \varphi(x_h)$, the function φ can be viewed as an inner linearization of ρ.

Outer approximation

Since φ is piecewise linear, we have that

(4.8) $$\varphi(x) = \max_{h=1,\ldots,H} \varphi_h(x)$$

where the functions φ_h are defined by (4.3). When φ is expressed in this form, finding its minimum consists in solving the minimax problem:

$$\min_{x\in[x_0,x_H]} \max_{h=1,\ldots,H} \varphi_h(x) \quad .$$

This is equivalent to solving the following linear program

(4.9) $\vartheta \in R$, and $x \in [x_0, x_H]$ such that

$\vartheta \geq s_h x + e_h$,

and $z = \vartheta$ is minimized.

The methods of inner and outer approximation rely on dual representations of the epigraph of φ, but one should note that the linear programs (4.7) and (4.9) are not dual linear programs.

Each of the above three problem manipulations delineates an approach to solving problems of the type (1.1). We consider each one in this more general setting in the next three sections.

5. EXTENSIONS OF THE REVISED SIMPLEX METHOD

An algorithm for solving stochastic programs with simple recourse with separable piecewise linear recourse costs and discrete random variables has been proposed in [10], a write-up and computer code has been provided by Kallberg and Kusy [13], and computational experience is reported in [14], [15]; cf. also [16]. From (3.4) we know that

$$\Psi(\chi) = \sum_{i=1}^{m_2} \Psi_i(\chi_i)$$

with each Ψ_i *polyhedral* with the slopes s_{ih} and $s_{i0} < s_{i1} < \cdots < s_{i,k_i}$. We can clearly apply the bounded variable method of Section 4 to the m_2-functions Ψ_i. Using (3.5) we obtain the following linear program, the analog of (4.5):

(5.1) find $0 \le x_j$, $j = 1,\ldots,n_1$,

and for $i = 1,\ldots,m_2$,

$0 \le y_{il} \le d_{il}$, $l = 0,\ldots,k_{i-1}$, and $y_{i,k_i} \ge 0$;

such that

$A_i x = b_i$, $\quad i = 1,\ldots,m_1$,

$T_i x - \sum_{l=0}^{k_i} y_{il} = p_{i0}$, $i = 1,\ldots,m_2$,

and $z = cx + \sum_{i=1}^{m_2} \sum_{l=0}^{k_i} s_{il} y_{il}$ is minimized.

where A_i and T_i are the i-th rows of A and T, and for $i = 1,\ldots,m_2$,

$$d_{i0} = p_{i1} - p_{i0} \quad ,$$

with $p_{io} \in)-\infty , p_{i1}($ chosen so that for the optimal solution x^* , $T_i x^* > p_{io}$

is guaranteed.

In seeking the solution of (5.1) using the revised simplex method for linear programs with simple upper bounds, let us assume that we have in hand a nondegenerate basic feasible solution say $(\tilde{x},\tilde{y}_{il}, i = 1,\ldots,m_2, l = 0,\ldots,k_i)$ which yields the values of the functions Ψ_i , $i = 1,\ldots,m_2$ at a point $\tilde{\chi} = T\tilde{x}$ as follows: for each $i = 1,\ldots,m_2$,

$$\Psi_i(\tilde{\chi}_i) = \Psi_i(p_{i0}) + \sum_{h=0}^{k_i} s_{ih}\tilde{y}_{ih} \quad ,$$

i.e. the $\tilde{y}_{il}$ actually solve the program (3.5) for $\chi_i = \tilde{\chi}_i$. It follows that if

$$0 < \tilde{y}_{il} < d_{il}$$

then

$$\tilde{y}_{ih} = d_{ih} \qquad \text{for } h < l \quad ,$$

$$\tilde{y}_{ih} = 0 \qquad \text{for } h > l \quad .$$

In [10] such a solution is called a *perfect* basic solution. The fact that the optimal solution is of that type and that one can pass from a perfect basic solution to another can be argued as follows: Let $(\sigma,\pi) \in R^{m_1} \times R^{m_2}$ be the simplex multipliers associated with the solution at hand. To find the variable to be entered into the basis at the next iteration, we compute $\bar{s}_{ih}$, the reduced cost (the component of the reduced gradient corresponding to the current basis) associated with the variable y_{ih}, viz.

$$\bar{s}_{ih} = s_{ih} + \pi_i$$

The $\{\bar{s}_{ih}$, $h = 0,\ldots,k_i\}$ are increasing, thus for each i among all variables at their lower bound $(h > l)$, the variable $y_{i,l+1}$ is the one that yields

potentially the greatest (marginal) improvement. Similarly, among all variables at their upper bound ($h < l$) the best candidate for decrease is $y_{i,l-1}$. It is also readily established that when either $y_{i,l+1}$ or $y_{i,l-1}$ is introduced into the basis then y_{il} will move to its appropriate bound and leave the basis. Thus the new basis will also be perfect. This property is not affected by exchanges between x-variables and y_{ih}-variables, unless degenerate cases are mishandled. A potential difficulty is that the algorithm could go through a great number of steps and associated basis changes if Ψ_i has many pieces. This can partially be overcome by an acceleration procedure that in one sweep makes a number of basis changes involving variables $\{y_{ih}\ , h = 0,\ldots,k_i\}$ for a given i, see [10]. This algorithm can thus be regarded as an application of the bounded variable revised simplex method with an acceleration step; in [10] it is also shown how to exploit the structure of the problem to obtain a good starting basis.

An extension of the above approach that seeks to avoid the difficulties associated with introducing bounded variables, and has also the advantage of greater generality permitting the distribution F_i to be arbitrary and $q_i(\cdot,w)$ to be nonlinear, is to handle the variable χ_i explicitly rather than implicitly in terms of a basic variable y_{il}. This method, discussed in [9], can be viewed as an extension of the convex simplex method to problems with nonlinear *nonsmooth* objective functions. Let us briefly review the convex simplex method as it applies to the problem:

(5.2) find $x \geq 0$ such that $Ax = b$, and $f(x)$ is minimized.

Here $f: R^n \to R$ is a smooth (continually differentiable) function and A is

$m \times n$. In contrast to the case when f is linear, we cannot work only with basic solutions $Ax = b$, i.e. with no more than m-elements of x larger than 0. At the start of a cycle of the convex simplex method, assume that x is a feasible solution with x_B the m-variables associated with a basis B. The remaining variables x_N define the non-basic variables with associated matrix N. For convenience let us assume that the basic variables are the first m variables, i.e. $x = (x_B , x_N)$, $A = [B \mid N]$, and

$$x_B = \bar{b} - B^{-1}Nx_N \quad .$$

We use this relation to eliminate x_B from (5.2). The reduced gradient of f is then given by

$$\bar{c}(x) = \nabla f(x) - \nabla f(x)_B \, B^{-1}A \quad , \tag{5.3}$$

where

$$\nabla f(x)_B = \left[\frac{\partial f}{\partial x_1}(x),\ldots,\frac{\partial f}{\partial x_m}(x)\right] \quad .$$

Schematically the *convex simplex method* proceeds as follows:

Step 0. Find $\hat{x}$ a basic feasible solution of $Ax = b$, $x \geq 0$.

Step 1. Select as basic variables the m components of $\hat{x}$ largest in magnitude such that the associated matrix B is a basis.

Compute $\bar{c}(\hat{x})$ using (5.3).

Step 2. With δ_j denoting a change in variable x_j that does not violate feasibility, identify the improving variables as follows:

$$\bar{c}_j < 0 \text{ and } \delta_j > 0 \text{ , or}$$

$$\bar{c}_j > 0 \text{ , } \hat{x}_j > 0 \text{ and } \delta_j < 0 \quad .$$

Choose a "best" candidate, say $j = s$, with associated column A^s. If none exists, the problem is solved.

Step 3. Changing $\hat{x}_s$ by δ_s corresponds to a step to a new point $\bar{x}$ given by

$$\bar{x}_B = \hat{x}_B - \delta_s(B^{-1}A^s)$$

$$\bar{x}_s = \hat{x}_s + \delta_s$$

$$\bar{x}_j = \hat{x}_j \text{ , for } j \neq s \text{ and } \hat{x}_j \text{ nonbasic.}$$

With $d = \bar{x} - \hat{x}$, find

$$\bar{\eta} \in \arg\min [f(\hat{x} + \eta d) | \hat{x} + \eta d \geq 0] \quad .$$

Assume $\bar{\eta}$ exists. Otherwise the original problem (5.2) could be unbounded or one needs to work with ε-approximates of $\bar{\eta}$.

Define $\hat{x}_{(new)} = \hat{x}_{(old)} + \bar{\eta} d$ and return to Step 1.

In [9] this overall scheme is used to design a method for solving (2.7), i.e. (convex) programs of the type

$$\text{find } x \in R^{n_1}, \chi \in R^{m_2} \text{ that minimize } cx + \sum_{i=1}^{m_2} \Psi_i(\chi_i)$$

$$\text{such that } Ax = b \text{ , } Tx = \chi \text{ and } x \geq 0 \quad ,$$

with the Ψ_i not necessarily differentiable. Let

$$\partial\Psi_i(\chi_i) = \{v \,|\, \check{c}_i \leq v \leq \hat{c}_i\} \quad .$$

Let B be a basis, i.e. a $m \times m$- submatrix of

$$\begin{bmatrix} A & 0 \\ T & -I \end{bmatrix}$$

$m = m_1 + m_2$, and (σ,π) the associated multipliers defined by

(5.4) $\qquad (\sigma,\pi) = c_B B^{-1}$

where

$(c_B)_k = c_j$ if the k-th basic variable is x_j,

$(c_B)_k = \hat{c}_i$ if the k-th basic variable is χ_i

Reduced subgradients are computed for the nonbasic variables as follows:

$$\bar{c}_j = c_j - \sigma A^j - \pi T^j \text{ if } j \leq n_1$$

and for the other nonbasic χ_i-variables

$$[\underline{c}_i \, , \bar{c}_i] = [\check{c}_i + \pi_i \, , \hat{c}_i + \pi_i]$$

In Step 2 of the algorithm, the "best" candidate is chosen by

(5.5) $\qquad s = \arg \min_{j,i} [\bar{c}_j \, , j = 1,\ldots n_1 \, ; \bar{c}_i \, , i = 1,\ldots,m_2 \, ; -\underline{c}_i \, , i = 1,\ldots,m_2]$.

The remaining operations remain the same, but note that finding $\bar{\eta}$ in Step 3 is greatly simplified because of the separability of the objective function.

The next logical step is to consider algorithmic procedures for a class of problems whose nonlinear features can still be relegated to the unconstrained optimization of some nonlinear function on a subspace, but this time with nonseparable objective possibly also nonsmooth, such as for problems of type (2.6).

Let us first examine the problems raised by nondifferentiability and suppose that $\partial\Psi(\chi)$ is given in terms of a finite number of vectors, say $g^1,\ldots,g^k$, i.e. $\partial\Psi(\chi)$ is a polytope. A vector g that can play the role of the

gradient in this case is the solution of the following quadratic program [17, 18, 19]:

(5.6) find $\lambda \in R_+^k$ such that $||v||^2$ is minimized.

and $\sum_{i=1}^{k} \lambda_i = 1$, $v = \sum_{i=1}^{k} \lambda_i g^i$

We shall refer to the solution of (5.6) as the gradient of Ψ at χ. A variant of the convex simplex method with this definition of gradient would naturally lead to a corresponding notion of reduced gradient. When $||\cdot||$ denotes Euclidean norm, Ψ is separable and $\partial\Psi_i(\chi_i) = [\check{c}_i , \hat{c}_i]$, (5.6) gives:

For $i = 1,\ldots,m_2$,

(5.7) find $v_i \in [\check{c}_i , \hat{c}_i]$ such that $|v_i|$ is minimized.

Let us again denote the solution by $g = (g_1,\ldots,g_{m_2})$. The components of the reduced gradient corresponding to the variables χ_i are thus

$$\bar{c}_i = g_i + \pi_i \text{ for } i = 1,\ldots,m_2 \quad ,$$

where (σ , π) are as in (5.4), the simplex multipliers associated with the basis B. For the components corresponding to the x_j variables we have as before

$$\bar{c}_j = c_j - \sigma A^j - \pi T^j \text{ for } j \leq n_1 \quad .$$

The selection rule for the incoming variable is similar to (5.5) used in the convex simplex method implementation, viz.,

(5.8) $s = \arg\min [\bar{c}_j , j = 1,\ldots,n_1 ; \bar{c}_i = g_i + \pi_i , i = 1,\ldots,m_2]$

and *produces the same incoming variable*, as can easily be verified, provided naturally that the vector π is unambiguously defined. In both cases, we recognize optimality through the condition $\bar{c}_s \geq 0$.

When $\Psi(\chi)$ is not separable we can continue to define the reduced gradient through (5.6) and use (5.8). A further development is to minimize in a subspace of non-basic variables leading to a reduced gradient method for nondifferentiable optimization.

Again, let us briefly review the reduced gradient method for problems of the type (5.2), i.e.

find $x \geq 0$ such that $Ax = b$ and $f(x)$ is minimized.

for any basic feasible solution say x', the constraints $Ax = b$ are always active. Suppose in addition that a number of the constraints $x_j \geq 0$, $j = 1,\ldots,n$ are also active:

$$x_j' = 0 \text{ for } j \in J \subset \{1,\ldots,n\}$$

whereas for $j \notin J$ they are inactive, $x_j' > 0$. We thus have partitioned the constraints as follows:

$$x_j > 0 \quad \text{for } j \notin J \quad ,$$

$$\left.\begin{array}{ll} x_j = 0 & \text{for } j \in J \\ & \\ Ax = b & \end{array}\right\} \text{ active}$$

The normals to the active constraints are the columns of

$$N = [A^T \, , \, I^{(J)}] \quad .$$

where $I^{(J)}$ consists of the columns I^j of the $(n \times n)$-identity matrix with

$j \in J$. Let us denote by Z a matrix with linearly independent columns such that

$$N^T Z = 0\,, \quad \text{lin. span } Z \cup \text{lin. span } N^T = R^n$$

If we consider as inactive all $x_j \geq 0$ corresponding to a given basis B as well as all other nonzero variables, we have that $|J| = n - (m + s)$. Now assuming N to be of full rank, we have

$$\text{rank } Z = n - (m + n - (m + s)) = s$$

Under the usual nondegeneracy assumptions, if we ignore the inactive constraints in the neighbourhood of x' problem (5.2) becomes

find $x \in R^n$ such that $N^T x = \binom{b}{0}$ and $f(x)$ is minimized.

Let us make a transformation of variables

$$x = x' + Zy$$

where $y \in R^s$. If $f(x)$ transforms to $\tilde{f}(y)$ the preceding program becomes

(5.9) find $y \in R^s$ such that $\tilde{f}(y)$ is minimized.

There are no constraints since they become

$$N^T x' + N^T Z y \equiv \binom{b}{0}$$

Also

(5.10) $$\nabla \tilde{f}(y) = \tilde{g} = Z^T g \text{ where } g = \nabla f(x)$$

and

(5.11) $\nabla^2 \tilde{f}(y) = \tilde{H} = Z^t H Z$ where $H = \nabla^2 f(x)$.

An unconstrained optimization algorithm utilizing $\tilde{g}$ and $\tilde{H}$ (or suitable approximations to these quantities) combined with an active (index) set strategy for revising Z, gives the usual reduced gradient method, see for example [20]. The above discussion is, of course, clearly related to the one given earlier for the convex simplex method. In particular compare (5.10) with (5.3). Note that if we ignore the zero components of $\bar{c}(x)$ in (5.3), namely, those corresponding to the basic variables, and when $|J| = n - m$ then (5.10) and (5.3) are alternative expressions for the same quantity. The matrix Z plays a crucial role in the implementation of the reduced gradient method, and it has a special structure. If we partition A as follows:

$$A = [B \mid S \mid M]$$
$$\quad m, s, n-m-s$$

and

$$W := B^{-1}S \quad ,$$

then

$$Z^T = [-W^T \mid I \mid 0]$$

This particular form of Z is exploited in MINOS which to date is the most effective implementation of nonlinear techniques for solving large problems of type (5.2), see [21]. The method is particularly effective on problems that are linear with respect to a large number of variables and nonlinear only with respect to a few variables. Then the dimensionality s of the unconstrained minimum can be kept relatively small. This is based

on the following observation [21].

5.12 PROPOSITION. *If problem* (5.2) *is solvable and there is an optimal solution involving only* t *of the nonlinear variables, then there exists an optimal solution at which the number of inactive constraints is less than or equal to* $m + t$.

Turning now to the application of these ideas to stochastic programs with complete recourse, let us consider our original problem (2.1) with nonstochastic tenders, i.e. with the equivalent deterministic form, cf. (2.6):

(5.13) find $x \geq 0$ such that $Ax = b$, $Tx = \chi$

and $z = cx + \Psi(\chi)$ is minimized.

Note that Proposition 5.12 is precisely the counterpart of Murty's result [8, 9] which asserts that (5.13) admits an optimal solution (x^* , χ^*) with no more than $(m_1 + m_2)$-positive entries in the x-vector. Thus there is an optimal solution to (5.13) with no more than m_2 nonbasic variables with positive value. If we have simple recourse with separable recourse costs and the marginals of the random variables $\{(q_i(w) , p_i(w) , i = 1,...,m_2\}$ are absolutely continuous, then the Ψ_i are differentiable, see Properties 2.8 and (3.2), and the reduced gradient method is directly applicable. In general however the functions Ψ_i are nonsmooth; for simple recourse and discretely distributed random variables, the subgradient at χ_i is given by

$$\partial\Psi_i(\chi_i) = \sum_{l=1}^{k_i} f_l \partial q_i(p_i(w_l) - \chi_i , w_l)$$

as follows from (3.1), where f_l is the probability associated with event w_l , $l = 1,...,k_i$. In this case the gradient could be computed using (5.6)

and the reduced gradient $\tilde{g}$ by means of (5.10). This would then give a very natural extension of MINOS to handle this class of stochastic (linear) programs. (In practice one would extend the scheme so as to compute ε-approximates, to ensure convergence.)

For the more general case of nonseparable objective function (5.13) this approach would still apply provided the gradient can be computed using (5.6). Indeed the question of what information about the subdifferential of the function can be povided becomes even more pressing when we are outside the case of simple recourse. Very often it is necessary to resort to an approximation scheme that would provide upper and lower bounds for the solutions [22, Section 3] or accept the fact that the gradient can only be estimated such as in the methods of stochastic quasi-gradient [23]. The approach that has been suggested above, based on adaptation and extension of MINOS, can also be pursued here with naturally some adjustments. We sketch out some of the possibilities in order to high-light the new obstacles that need to be overcome but also to stress the fact that there is a natural continuation of this approach that provides solution procedures for more sophisticated stochastic programming problems.

Recall that $\Psi(\chi) = E\{\psi(\chi,w)\}$ and

$$\psi(\chi,w) = \inf_{y \in C} [q(y,w) \mid Wy = p(w) - \chi]$$

in this more general case, see Section 2. We consider only the linear case, i.e. when

$$\psi(\chi,w) = \inf_{y \geq 0} [q(w)y \mid Wy = p(w) - \chi]$$

$$= \sup_{\pi}[\pi(p(w) - \chi) \mid \pi W \leq q(w)]$$

For stochastic programs with complete recourse (that are bounded), $\psi(\cdot,w)$ is finite for all (possible) w and

$$\partial\Psi(\chi) = E\{\partial\psi(\chi,w)\}$$

where

$$\partial\psi(\chi,w) = \{-\pi \in R^{m_2} \mid \pi \in \arg\max_{\pi W \leq q(w)} [\pi(p(w) - \chi)]\} \quad .$$

If the random variables have a discrete distribution with $p(\cdot)$ and $q(\cdot)$ taking on the values

$$\{(p^l,q^l)\ ,\ l = 1,\ldots,L\}$$

with probabilities f_l , $l = 1,\ldots,L$, then a "gradient" as defined by (5.6) is obtained by solving the program:

find $v \in R^{m_2}$ such that $||v||^2$ is minimized,

where $v = \sum_{l=1}^{L} f_l \pi^l$ and $\pi^l W \leq q^l$, $\pi^l(p^l - \chi) \geq \psi(\chi,(q^l,p^l))$.

To solve this program efficiently we need to take advantage of its special structure, use the fact that for most l there is only a unique π^l that satisfies the inequalities and that for many l , π^l will be determined by the same basis of (W,I), and so on.

In general, when the random variables are not discretely distributed or when there are too many possible values for the discretely distributed random variables it may not be possible to obtain complete information about $\Psi(\chi)$ or $\partial\Psi(\chi)$. We are then reduced to accepting approximates. There is at present no theory that allows us to deal directly with this case. What is needed is to extend the subgradient techniques, such as [17, 18,

19] and in particular [24], with appropriate reduced gradient calculations to handle this case. The convergence proofs could be derived by relying on the framework provided by the study of nonlinear programming methods in the presence of noise [25].

The question of how approximate the calculation of $\Psi(\chi)$ and $\partial\Psi(\chi)$ should be is still very much open to much deeper investigation. The method of stochastic quasi-gradients [23] advocates the use of a single sample point, say (p^s, q^s), to obtain

$$\partial\psi(\chi^s, (q^s, p^s))$$

as an estimate to $\partial\Psi(\chi^s)$. However, this estimate is only used to slightly change χ^s, rather than as an adjusted cost function as in the extensions of the revised simplex method discussed here.

We conclude this section by making some *comments about implementation*. As mentioned already earlier, the most natural vehicle for implementing the algorithms described above is the MINOS Code of Murtagh and Saunders [21]. But even for the case of simple recourse with discretely distributed random variables several augmentations of the code are necessary, including

1. Design and implementation of a standardized input format. A very natural situation is for a linear programming model specified in MPS format to be later modified to permit some demands and costs to be stochastic. Thus a standardized input would be based on a combination of

(a) MPS format for specifying c, the LP matrices A, T, the right-hand sides b and bounds. In particular, we could easily allow for bounds $l \leq x \leq u$ in place of $x \geq 0$,

(b) a specification section, like SPECS in MINOS to describe the rows of A and T that correspond to the technology matrix, the cost functions q_i (linear or nonlinear), and the distribution functions F_i (piecewise constant, piecewise linear).

2. Routines to compute the "gradient", e.g. as in (5.6) and (5.10) suitably extended to ensure convergence.

3. Modification of the line search procedure to the nonsmooth case, see [26].

4. Design of a suitable output routine for interpreting the results in statistical terms.

MINOS is also a natural vehicle for incorporating techniques for solving more general problems as discussed above. An alternative starting point is the code of Nguyen and Bihain, see [24], which already handles nonsmooth objective functions but does not have all the linear programming features of MINOS that are bound to play an important role in the efficiency and stability of the method used to solve problems of type (1.1).

6. INNER APPROXIMATION

The algorithms we consider next use inner approximation of the type discussed in Section 4, see (4.7). After a general discussion of the algorithm, we consider first how it applies to problems with simple recourse and, as in Section 5, see how to extend the approach to more general classes of stochastic programs (with nonstochastic tenders).

The resulting algorithm is in effect the generalized programming technique, attributed by Dantzig [27, Chapter 24] to P. Wolfe. Here we apply it to problems of type (1.1) taking advantage of the special structure and of the form of $\Psi(\chi)$. As a means to obtain error bounds, Williams [28] already suggested an approach of this nature, but apparently it has not been exploited as a general solution technique.

The algorithm as it applies to (1.1) or equivalently (2.5) can be summarized as follows:

Step 0. Find a feasible solution of $Ax^0 = b$, $x^0 \geq 0$

Set $\chi^0 = Tx^0$.

Choose $\chi^1, \ldots, \chi^\nu$ (a selection of tenders, $\nu \geq 0$).

Step 1. Solve the linear program:

(6.1) Minimize $cx + \sum_{l=0}^{\nu} \lambda_l \Psi(\chi^l) = z$

subject to $Ax = b$

$$Tx - \sum_{l=0}^{\nu} \lambda_l \chi^l = 0$$

$$\sum_{l=0}^{\nu} \lambda_l = 1$$

$x \geq 0$, $\lambda_l \geq 0$ for $l = 0, \ldots, \nu$.

Let $(\sigma^\nu , \pi^\nu , \vartheta^\nu)$ be the (optimal) multipliers associated with the solution of (6.1).

Step 2. Find $\chi^{\nu+1} \in \arg\min [\Psi(\chi) + \pi^\nu \chi]$

If $\Psi(\chi^{\nu+1}) + \pi^\nu \chi^{\nu+1} \geq \vartheta^\nu$, stop: optimal.

Otherwise return to Step 1 with $\nu = \nu + 1$

We have assumed here that for all π^ν generated in Step 1, the function $\chi \longmapsto (\Psi(\chi) - \pi^\nu\chi)$ attains its minimum. There are naturally regularity conditions for stochastic programs that will guarantee this [1, 28], but mostly we have done so to simplify the presentation and interpretation of the algorithm. Note that both upper and lower bounds for the infimum are available. Let z^ν denote the optimal value of z, and $(\lambda_l^\nu, l = 0,\ldots,\nu)$ the optimal values of the λ variables in (6.1). Then

$$(6.2) \qquad z^\nu + [\Psi(\chi^{\nu+1}) + \pi^\nu\chi^{\nu+1}] - \sum_{l=0}^{\nu} \lambda_l^\nu[\Psi(\chi^l) + \pi^\nu\chi^l] \leq z^* \leq z^\nu$$

where z^* is the optimal value of the original program. The second inequality follows from the fact that (6.1) is an inner approximation, whereas the first one follows from Step 2 which implies that

$$-\pi^\nu\chi + [\Psi(\chi^{\nu+1}) + \pi^\nu\chi^{\nu+1}] \leq \Psi(\chi) \quad .$$

Adding cx and taking inf on both sides with respect to (x,χ) on the set $\{x \geq 0 \,|\, Ax = b\ ,\ Tx = \chi\}$ yields the desired inequality, it suffices to observe that the first one of these two minimization problems admits for optimal solution the pair $(x^\nu, \sum_{l=0}^{\nu} \lambda_l^\nu\chi^l)$ with $(x^\nu, \lambda_l^\nu, l = 0,\ldots,\nu)$ the optimal solution of (6.1). Thus

$$0 \leq z^\nu - z^* \leq \operatorname*{Max}_{k=0,\ldots,\nu} [(\Psi(\chi^{\nu+1}) + \pi^\nu\chi^{\nu+1}) - \sum_{l=0}^{\nu} \lambda_l^\nu(\Psi(\chi^l) + \pi^\nu\chi^l)]$$

We interpret the algorithm as the search for a particular (optimal) tender χ^*. It is easy to see that if χ^* is part of the collection $\chi^0, \ldots, \chi^\nu$, then solving (6.1) will yield the optimal x^*. One reason for believing that this approach holds promise is that in practice, one should be able to initialize the algorithm with a good choice of tenders $\chi^0, \ldots, \chi^\nu$. The

subsequent iterations can then be viewed as refinements of the original guesses. A line of further research is to find effective strategies for choosing initial tenders.

The convergence of the algorithm, with the following assumptions

(6.3) all tenders are retained, as part of (6.1),

(6.4) complete information is available about the function values of Ψ so that Step 2 can be carried out exactly,

has been proved by Dantzig [27, Chapter 24]. Further, the algorithm applied to the convex program (1.1) is equivalent to a cutting plane algorithm applied to its dual. We can thus translate the results about retention of cuts [29, 30] into retention of tenders. In particular in our case, they imply that all tenders not associated with a basic variable λ_l can be dropped at the next iteration, without affecting the convergence proof of the algorithm.

A large number of tenders could be generated, although this is very unlikely in practice, especially if a good set of initial tenders is used. From a theoretical standpoint however and for reasons of sound implementation, it is worth examining the question of which tenders should be retained to enhance convergence. At iteration ν with the multipliers $(\sigma^\nu, \pi^\nu, \vartheta^\nu)$ we have for all tenders χ^l, $l = 0,\ldots,\nu$,

(6.5) $$\Psi(\chi^l) + \pi^\nu \chi^l \geq \vartheta^\nu$$

At the next iteration, a tender $\chi^{\nu+1}$ is developed (several tenders could equally well be formed) and we need to resolve (6.1) with respect to $\chi^0, \ldots, \chi^{\nu+1}$. Suppose prior to the commencement of the next iteration

$\nu + 1$, a subset of tenders

$$\{\chi^l \ , l \in L\} \subset \{\chi^0, \ldots, \chi^{\nu+1}\}$$

must be found such that the optimal solution of (6.1) is unaffected. Since the (optimal) multipliers

$$(\sigma^{\nu+1} \ , \pi^{\nu+1} \ , \vartheta^{\nu+1})$$

are unknown at this stage, we formulate this problem as

(6.6) given any $(\sigma \ , \pi \ , \vartheta)$ and a fixed index k

find $L \subset \{0,\ldots,\nu + 1\}$ such that

$\Psi(\chi^l) + \pi\chi^l \geq \vartheta$ forall $l \in L$

implies $\Psi(\chi^k) + \pi\chi^k \geq \vartheta$.

Let us write

$$D^l = \begin{bmatrix} \Psi(\chi^l) \\ \chi^l \\ -1 \end{bmatrix} , b = \begin{bmatrix} \Psi(\chi^k) \\ \chi^k \\ -1 \end{bmatrix} \text{ and } I^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

with $D = [D^0,\ldots,D^{\nu+1}]$.

6.7 PROPOSITION. *A sufficient condition that*

(6.8) $\Psi(\chi^l) + \pi\chi^l \geq \vartheta$ for all $l \in L$

implies

(6.9) $\Psi(\chi^k) + \pi\chi^k \geq \vartheta$

is that

$$b \in \text{pos}\,[I^1 \, ; D^l \ , l \in L] \quad .$$

i.e. that b belongs to the positive hull of or equivalently the convex cone, generated by the columns of D corresponding to L and I^1.

Proof. To say that $b \in \text{pos}\,[I^1 ; D^l , l \in L]$ is to say that the linear system

$$b = \sum_{l \in L} D^l y_l + I^1 \alpha , \alpha \geq 0 ; y_l \geq 0 , l \in L$$

is solvable. Thus the system

$$(\lambda,\pi,\sigma)D^l \geq 0 , l \in L ; \lambda \geq 0 \text{ and } (\lambda , \pi , \vartheta) \cdot b < 0$$

is not solvable, as follows from Farkas Lemma. Using now the definitions of D^l and b, we see that this implies that for a choice of variables $(\lambda = 1 , \pi , \vartheta)$ satisfying (6.8), we necessarily must satisfy (6.9).

The question raised in (6.6) can thus be translated into finding a minimum number of generators, i.e. a *frame*, for the convex poyhedral cone

$$\text{pos}\,[I^1 ; D^l , l = 0,\ldots,\nu + 1] \quad .$$

An algorithm for doing this is described by Wets and Witzgall [31]. Note also that it may be worthwhile to also eliminate tenders that have not been utilized in the solution of (6.1) on several prior iterations.

The use of this algorithm in the context of stochastic programming makes assumption (6.4) nontrivial. Even in the case of simple recourse, situations can arise when $\Psi(\chi)$ cannot be calculated exactly or the cost of calculating it could be excessive. For example, if $q(y,w)$ is nonlinear in y and the dependence on w is not simple (e.g. linear), the cost of evaluating

$$E\{q(p(w) - \chi , w)\}$$

could be very large. A similar situation could arise even after approximating the distribution functions by piecewise constant or piecewise linear distributions. In this case the generalized linear programming approach must be revised to include noisy functions and the question of convergence, theoretical and practical, still needs further investigation.

In the case of simple recourse with separable cost the evaluation of the function Ψ presents no serious challenge since

$$\Psi(\chi) = \sum_{i=1}^{m_2} \Psi_i(\chi_i)$$

and each Ψ_i defined on R is given by a 1-dimensional integral, viz.

$$\Psi_i(\chi_i) = \int q_i(p_i(w) - \chi_i \ , w) F_i(dw)$$

with the subgradient given by (3.1). Special forms of q_i and F_i lead to even simpler representation for Ψ such as (3.3). Even more explicit is the expression obtained in (3.4) and (3.5) in the case of piecewise linear recourse costs and piecewise constant distributions; for piecewise linear recourse costs and piecewise linear distributions see [32, §3], for even more detailed expressions for specific distributions consult [33], [34]. Note also that in this case Step 2 of the algorithm consists in finding for $i = 1,\ldots,m_2$, $\chi_i^{\nu+1}$ such that

$$-\pi_i^{\nu} \in \partial\Psi_i(\chi_i^{\nu+1}) \quad ,$$

where the subgradient is given by (3.1). Again, in many cases it is possible to use the special forms of q_i and F_i to find efficient solutions procedurs for the preceding relation. For example, in the situation covered by (3.3), the above becomes: find $\chi_i^{\nu+1}$ such that

$$\frac{q_i^+ - \pi_i^\nu}{q_i} \in [F_i^-(\chi_i^{\nu+1}) \, , \, F_i(\chi_i^{\nu+1})] \quad .$$

It thus suffices to have a bracketing routine for finding the point at which the monotone function F_i passes through the value $(q_i^+ - \pi_i^\nu)/q_i$.

In the more general case, when it is not feasible to compute the value of Ψ at χ exactly, see Section 3, there are basically two strategies available. The first one is to accept inaccurate evaluations of Ψ, view them as noisy observations of Ψ and rely on a convergence in probability argument [25]. How to design an efficient and reliable algorithm that proceeds in this fashion has not been investigated yet.

The second approach is to proceed by approximations. By this we mean replace the original problem (2.1) by an approximate one, solve the approximating problem, obtain if possible bounds using this approximating solution and repeat the process with a refinement of the approximation if the bounds are not sufficiently tight. The subject of approximations, specially via discreteization of the random variables, is reviewed in [22, Section 3] and will not be taken up here. We only want to raised some of the questions that need to be resolved before such a scheme could be made operational:

(i) How should the initial approximation be designed so as to obtain with minimal computational effort a "good" approximate of the solution?

(ii) How to improve (refine) the approximation so as to "maximize" the resulting improvement?

(iii) How to blend in, these successive approximations with the steps of the algorithm?

As is clear from the preceding discusion the implementation of this algorithm requires different special versions for each step depending on the form of the objective and the distribution functions. We are currently utilizing the subroutines of Nazareth, described in [35], to develop an experimental implementation. XMP [36] of Marsten and also MINOS [21] could provide suitable vehicles for implementation.

7. OUTER LINEARIZATION

The third class of algorithms that we consider is based upon the outer approximation approach described in Section 4, see (4.9). We deal with this technique somewhat more briefly because for problems with simple recourse it appears at this time to be more limited in scope, whereas for more general classes of stochastic programs this approach is very close to the L-shaped algorithm [37] which has already been studied extensively in the stochastic programming setting [38], [22, Section 2], [39].

Consider first simple recourse with separable recourse cost, i.e. with the objective of the equivalent deterministic program of the type

$$cx + \sum_{i=1}^{m_2} \Psi_i(\chi_i) \quad ,$$

and let us assume that complete information is available about values and derivatives of Ψ_i. Consistent with (4.8), we assume that we have the following representation for each Ψ_i:

$$\Psi_i(\chi_i) = \max_{j \in J_i} \Psi_{ij}(\chi_i)$$

where each Ψ_{ij} is a convex differentiable function and J_i a finite set of indices. For each χ_i, the value of $\max_{j \in J_i} \Psi_{ij}(\chi_i)$ is attained for a finite set $A(\chi_i) \subset J_i$ known as the *active set*, i.e. we are dealing with stochastic programs whose equivalent deterministic forms have (possibly) nondifferentable objectives with explicitly known subdifferentials.

Problem (2.7) can thus be stated as

(7.1) find $x \in R_+^{n_1}$, $\chi \in R^{m_2}$ and $v \in R^{m_2}$ such that

$$Ax = b,$$

$$Tx - \chi = 0 \quad ,$$

$$-\Psi_{ij}(\chi_i) + v_i \geq 0, \text{ for } j \in J_i \text{ , } i = 1,\ldots,m_2$$

and $cx + \sum_{i=1}^{m_2} v_i$ is minimized.

When $A(\chi_i)$ and $\partial\Psi_i(\chi_i)$ are known explicitly, the functions Ψ_{ij} can be obtained systematically and as needed.

In the method of succesive linear approximation, see [40] for example, with differentiable Ψ_i we solve a sequence of problems of the form

(7.2) find $x \in R_+^{n_1}$, $\chi \in R^{m_2}$ and $v \in R^{m_2}$ such that

$$Ax = b$$

$$Tx - \chi = 0$$

$$-\nabla\Psi_i(\chi_i^l) \cdot \chi_i + v_i \geq (\Psi_i(\chi_i^l) - \nabla\Psi_i(\chi_i^l) \cdot \chi_i^l)$$

$$\text{for } l = 1,\ldots,\nu \text{ , } i = 1,\ldots,m_2$$

and $cx + \sum_{i=1}^{m_2} v_i$ is minimized.

The next approximation is obtained by linearization of the Ψ_i at $\chi_i^{\nu+1}$ where $\chi^{\nu+1}$ is the optimal value for χ in (7.2). (It is common to use an additional constraint that restricts the step size.)

When second order differentiables (or good approximates thereof) can be computed such as when the recourse costs and marginal distribution functions are piecewise linear [32] or other cases dealt with in the beginning of Section 3, one could proceed via quadratic approximations, as proposed for nonlinear programming by Wilson [41] and Han [42]. Good reviews of both approaches can be found in [20], [43] and [44], see also [47].

In the more general situation Ψ is not necessarily separable and subgradients can only be calculated approximately, one could consider a cutting plane algorithm. This would involve solving a sequence of problems of the form

(7.3) find $x \in R_+^{n_1}$, $\chi \in R^{m_2}$ and $v \in R$ such that

$$Ax = b$$

$$Tx - \chi = 0$$

$$-\pi^l \chi + v \geq (\Psi(\chi^l) - \pi^l \chi^l) \, , \; l = 1,\ldots,\nu$$

and $cx + v$ is minimized.

Here $\pi^l \in \partial\Psi(\chi^l)$ for $l = 1,\ldots,\nu$. The solution of (7.3) yields a new tender $\chi^{\nu+1}$, we then need to compute $\pi^{\nu+1}$ and $\Psi(\chi^{\nu+1})$. This defines a new constraint to be added to (7.3). More sophisticated strategies based upon utilizing higher order information are given by Womersley [45].

When the equation $Tx = \chi$ is used to delete χ from the formulation of (7.3) then we are precisely in the L-shaped format [22, Section 2] for which we already have experimental codes [38]. The code [46] to solve nonlinear programming problems by successive approximations using quadratic programs should also be studied for implementation in this setting. Because of the second order information necessary to carry out the steps, it only appears possible to use it for a special class of stochastic programs with simple recourse.

REFERENCES

[1] R. Wets, Stochastic programs with fixed recourse: the equivalent deterministic program, *SIAM Review* 16(1974), 309-339.

[2] P. Olsen, Multistage stochastic programming with recourse: the equivalent deterministic program, *SIAM J. Control Optim.* 14(1976), 495-517.

[3] J.-B. Hiriart-Urruty, Conditions necessaires d'optimalite pour un programme stochastique avec recours, *SIAM J. Control Optim.* 16(1978), 125-130.

[4] P. Kall, Stochastic programming, *European J. Operations Research*, 10(1982), 125-130.

[5] R. Wets, Stochastic programs with recourse: a basic theorem for multistage problems, *Z. Wahrscheinlichkeitstheorie verw. Geb.* 21(1972), 201-206.

[6] D. Walkup, and R. Wets, Stochastic programs with recourse: special forms, in *Proceedings of the Princeton Symposium on Mathematical Programming*, ed. H. Kuhn, Princeton University Press, Princeton, 1970, 139-162.

[7] W. Ziemba, Stochastic programs with simple recourse, in *Mathematical Programming in Theory and Practice*, eds. P. Hammer and G. Zoutendijk, North-Holland, Amsterdam, 1974, 213-274.

[8] K. Murty, Linear programming under uncertainty: a basic property of the optimal solution, *Z. Wahrscheinlichkeitstheorie verw. Gebiete*, 10(1968), 284-288.

[9] R. Wets, A statistical approach to the solution of stochastic programs with (convex) simple recourse, in *Generalized Lagrangian in Systems and Economic Theory*, ed. A. Wierzbicki, International Institute for Applied Systems Analysis Proceedings Series, Pergamon Press, Oxford 1983.

[10] R. Wets, Solving stochastic programs with simple recourse, *Stochastics* (1983).

[11] I. Deak, Three digit accurate multiple normal probabilities, *Num. Math.*35(1980), 369-380.

[12] J. Birge and R. Smith, New Monte-Carlo procedures for numerical integration,Tech. Report, University of Michigan, 1982

[13] J. Kallberg and M. Kusy, Code instruction for S.L.P.R., a stochastic linear program with simple recourse, Tech. Report, University of British Columbia,May 1976.

[14] J. Kallberg, R. White and W. Ziemba, Short term financial planning

under uncertainty, *Management Sci.*, 28(1982), 670-682.

[15] M. Kusy and W. Ziemba, A bank asset and liability manangement model, Tech.Report, University of British Columbia, 1981.

[16] H.Cleef, A solution procedure for the two-stage stochastic program with simple recourse, *Z. Operations Research*, 25(1981), 1-13.

[17] V. Demyanov, Algorithms for some minimax problems, *J. Computer and Syst. Sci.*, 2(1968), 342-380.

[18] C. Lemarechal, Nonsmooth optimization and descent methods, Research Report 78-4, International Institute for Applied Systems Analysis, Laxenburg, 1978.

[19] P. Wolfe, A method of conjugate subgradients for minimizing nondifferentiable functions, in *Nondifferentiable Optimization, Mathematical Programming Study*, 3(1975), 145-173.

[20] R. Fletcher, *Practical Methods of Optimization, Vol.2: Constrained Optimization*, J. Wiley, Chichester, 1981.

[21] B. Murtagh and M. Saunders, Large-Scale linearly constrained optimization, *Mathematical Programming*, 14(1978), 41-72.

[22] R. Wets, Stochastic programming: solution techniques and approximation schemes, Working Paper WP-82-84, International institute for Applied Systems Analysis, Laxenburg, 1982; to appear in *Mathematical Programming 1982: The State-of-the-Art*, Springer Verlag, Berlin 1983.

[23] Yu. Ermoliev, Stochastic quasigradient methods and their application in systems optimization, IIASA Working Paper 81-2, Laxenburg 1981; to appear in *Stochastics*.

[24] A. Bihain, G.R.S.G. - A general purpose N.D.O. Code, Tech. Report University of Namur, 1982.

[25] B. Poljak, Nonlinear programming methods in the presence of noise, *Mathematical Programming*, 14(1978), 87-97.

[26] C. Lemarechal, A view of line-searches, in *Optimization and Optimal Control*, eds. A. Auslender, W. Oettli and J. Stoer, Springer Verlag Lecture Notes in Control and Information Sciences 30, Berlin 1981, 59-78.

[27] G. Dantzig *Linear Programming and Extensions*, Princeton University Press 1963.

[28] A.C. Williams, Approximation formulas for stochastic linear programming, *SIAM J. Appl. Math.*, 14(1966), 668-677.

[29] D. Topkis, Cutting plane methods without nested constraint sets, *Operations Research* 18(1970), 403-413.

[30] B.C. Eaves and W. Zangwill, Generalized Cutting plane algorithms, *SIAM J. Control*, 9(1971), 529-542.

[31] R. Wets and C. Witzgall, Algorithms for frames and lineality spaces of cones, *J. of Research of National Bureau Standards*, Section B, 71B(1967), 1-7.

[32] R. Wets, Solving stochastic programs with simple recourse II, *Proceed. Johns Hopkins Symposium on Systems and Information*, 1975, 1-6.

[33] B. Hansotia, Some special cases of stochastic programs with recourse, *Operations Research*, 25(1977), 361-363.

[34] B. Hansotia, Stochastic linear programs with simple recourse: the

equivalent deterministic progam for the normal, exponential and Erlang case, *Naval Res. Logist. Quarterly*, 27(1980), 257-272.

[35] L. Nazareth, Implementation aids for optimization algorithms that solve sequences of linear programs by the revised simplex method, Working Paper WP-82-107, IIASA, Laxenburg 1982.

[36] R. Marsten, The design of the XMP linear programming library, Tech. Report 80-2, University of Arizona 1980.

[37] R. Van Slyke and R. Wets, L-shaped linear program with applications to optimal control and stochastic linear programs, *SIAM J. on Appl. Math.*, 17(1969), 638-663.

[38] J. Birge, Decomposition and partitioning methods for multistage stochastic linear programs, Tech. Report 82-6, University of Mchigan 1982.

[39] J. Birge, The relationship between the L-shaped method and dual factorization for stochastic linear programming, Tech. Report, University of Michigan 1982.

[40] K. Madsen and H. Schjaer-Jacobsen, Linearly constrained minimax optimization, *Mathematical Programming*, 14(1978), 208-223.

[41] R. Wilson, A simplicial algorithm for concave programming, Doct. Thesis, Harvard 1963.

[42] S.P. Han, Superlinearly convergent variable metric algorithms for general nonlinear programming problems, *Mathematical Programming*, 11(1976), 263-282.

[43] M.J.D. Powell, Algorithms for nonlinear constraints that use Lagrangian functions, *Mathematical Programming*, 14(1978), 224-248.

[44] A. Wierzbicki, Lagrangian functions and nondifferentiable optimization. In *Progress in Nondifferentiable Optimization*, E. Nurminski (Ed.), IIASA Collaborative Proceedings, Series CP-82-58 (1982), 173-213.

[45] R. Womersley, Numerical methods for structured problems in nonsmooth optimization, Doct. Thesis, University of Dundee 1981.

[46] K. Schittkowski, *Nonlinear Programming Codes*, Springer Verlag Lecture Notes in Economics and Mathematical Systems, Berlin 1980.

[47] S. Han, Variable metric method for minimizing a class of nondifferentiable functions, *Mathematical Programming*, 20(1981), 1-12.