# ON PARALLEL PROCESSORS DESIGN FOR SOLVING STOCHASTIC PROGRAMS

Roger Wets

October, 1985
WP-85-67



INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
2361 Laxenburg, Austria

# On Parallel Processors Design for Solving Stochastic Programs

Roger Wets*

University California - Davis and IIASA, Austria

Abstract

A design based on parallel processing is laid out for solving (multistage) stochastic programs. Because of the very special nature of the decomposition used here, one could rely on hard-wired micro-processors that would be extremely simple in design and fabrication, and would reduce the time required to solving stochastic programs to that needed for solving deterministic linear programs of the same size (ignoring the time required to design the parallel decomposition).

---

*Supported in part by the National Science Foundation

We deal with the following class of stochastic programs with recourse:

$$\text{find} \quad x \in R^{n_1} \quad \text{such that} \quad Ax = b, \quad x \geq 0,$$

$$\text{and} \quad z := cx + E\{Q(x,\underset{\sim}{\xi})\} \quad \text{is minimized,}$$

where

$$Q(x,\xi) := \inf_{y \in R^{n_2}_+} \{qy \mid Wy = h - Tx\},$$

and

$$\xi := (h_1,\ldots,h_{m_2}; t_{11},\ldots,t_{1,n_1}; t_{21},\ldots, ; t_{m_2 1},\ldots,t_{m_2 n_1}),$$

are the random elements of the problem. The matrices and vectors $c(n_1)$, $A(m_1 \times n_1)$, $b(m_1)$, $q(n_2)$ and $W(m_2 \times n_2)$ are deterministic in this model. Let $P$ denote the probability measure defined on the support $\Xi \subset R^N$ $(N = m_2 + m_2 \times n_1)$ of the random vector $\underset{\sim}{\xi}$. We assume that for all $x \in K_1$

$$K_1 := \{x \mid Ax = b, \; x \geq 0\}$$

the expectation $E\{Q(x,\underset{\sim}{\xi})\}$ is finite. This means that we are dealing with a stochastic program with relatively complete recourse, either because it was originally given to us in that form or we have generated enough of the induced constraints to reduce it to a problem of that form. We also assume that the problem is solvable (feasible and bounded) and stable (the Kuhn-Tucker conditions are sufficient and necessary for optimality). For a

review of the terminology, and a discussion of the conditions used here, consult Wets (1974).

We restrict our attention to problems of this type, but the scheme that we suggest in the ensuing development applies (with modifications) to much more general classes of problems. If we do not have relatively complete recourse, then an additional step must be introduced in the algorithm to generate feasibility cuts such as in Step 2 of the L-shaped algorithm, for a review see Wets (1986). If we are dealing with more than a two-stage linear program, or equivalently--see Wets (1972)--with a nonlinear Lipschitzian function $q(y)$ for the objective of the recourse problem (replacing also $y \in R^{n_2}_{+}$ with $y \in D$, a polyhedral convex set), we can still rely on the principles laid out here for the decomposition of the recourse problem, but instead of separable linear approximates one needs to use separable nonlinear approximates. The details of the nonlinear case, as it applies to multistage problems, will be included in another article.

The first part deals with approximating general linear programs by very simple linear programs. This is the basis for the parallel decomposition of the recourse problem. The second part deals with the use of this decomposition to solve stochastic programs. Finally, we discuss its implementation in parallel processing machines.

We present here a scheme based on a "primal" parallel decomposition of the recourse function of a two-stage stochastic program; it relies on ray function approximates. A related scheme based on a "dual" representation of the recourse function is also under study, see the comments at the end of Section 1.

## 1. APPROXIMATIONS FOR LINEAR PROGRAMS

In Birge and Wets (1985), we introduced simple recourse approximates for general recourse problems to calculate upper bounds for stochastic problems. It is this type of approximations that will be used to obtain a "parallel" decomposition of the recourse problem. To begin with let us consider the following function:

$$\psi(t) := \inf_{y \in R^{n_2}_+} \{qy \mid Wy = t, \quad y \geq 0\}. \tag{1.1}$$

This is a sublinear function (positively homogeneous and convex), proper ($\psi(t) > -\infty$, $\psi(0) = 0$) as follows from the assumptions we have made about the stochastic program, and finite on the convex polyhedral cone

$$\text{pos } W := \{t \mid t = Wy, \quad y \geq 0\} \tag{1.2}$$

Note that the assumption of relatively complete recourse means that for all $x \in K_1$,

$$h - Tx \in \text{pos } W$$

for all $\xi = (h; T_1, \ldots, T_{m_2})$ in $\Xi$. Not only is the effective domain of $\psi$ polyhedral, so is actually also epi $\psi$ its epigraph, this means that we can find a representation of $\psi$ in terms of a finite collection of rays that determine a frame for its epigraph. Let

$$\{\binom{\alpha^{\ell}}{t^{\ell}},\ \ell = 1,\ldots,L\}$$

be a finite collection of vectors in $R^{m_2+1}$ such that

$$\text{epi } \psi = \text{pos } [\binom{\alpha^{\ell}}{t^{\ell}},\ \ell = 1,\ldots,L] \tag{1.3}$$

$$= \{\binom{\alpha}{t} \in R^{m_2+1} \,|\, \binom{\alpha}{t} = \Sigma_{\ell=1}^{L}\ \mu_{\ell}\binom{\alpha^{\ell}}{t^{\ell}},\ \mu_{\ell} \geq 0\}.$$

Then, it is easy to see that

$$\psi(t) = \inf \{\alpha \,|\, \binom{\alpha}{t} = \Sigma_{\ell=1}^{L}\ \binom{\alpha^{\ell}}{t^{\ell}}\mu_{\ell},\quad \mu_{\ell} \geq 0\} \tag{1.4}$$

This representation of $\psi$ brings us to the following construction of approximates of $\psi$. Let

$$\{\binom{\alpha^{s}}{t^{s}} \in R^{m_2+1},\quad s = 1,\ldots,S\}$$

be any finite collection of vectors in $R^{m_2+1}$ such that for all $s = 1,\ldots,S$,

(1.5) $$\binom{\alpha^s}{t^s} \in \text{epi } \psi$$

i.e., $\alpha^s \geq \psi(t^s)$. Then

(1.6) $$\psi(t) \leq \inf \{\alpha \mid \binom{\alpha}{t} = \Sigma_{s=1}^{S} \binom{\alpha^s}{t^s}\mu_s, \quad \mu_s \geq 0\}.$$

By choosing $\{\binom{\alpha^s}{t^s}, s = 1,\ldots,S\}$, we have built another sublinear function that majorizes $\psi$. If this new sublinear function is to be a reasonable approximate of $\psi$, at least on the rays:

$$\{t \mid t = \lambda\, t^s, \quad \lambda \geq 0\}, \quad s = 1,\ldots,S,$$

we should choose

(1.7) $$\alpha^s := \psi(t^s) = \inf \{qy \mid Wy = t^s, \quad y \geq 0\}.$$

A second condition, that needs to be satisfied in order to be able to use this new sublinear function as an approximation for $\psi$ is that the collection of vectors $\{\binom{\alpha^s}{t^s}, s = 1,\ldots,S\}$ be rich enough. This means basically that one would need to choose this collection so that it determines, or nearly determines a frame for the epigraph of $\psi$. If we proceed in that fashion, ignoring even the work necessary to find such a

frame, it would not be any easier to determine the value of $\psi(t)$ by solving the linear program (1.1) that defines it, or the linear program (1.4) that yields its dual representation.

The parallel decomposition that we introduce here is based on sublinear majorization of $\psi$, but instead of using on one such approximate, we rely on a whole collection -- each one corresponding to a very simple linear program -- that can be combined to yield the value, or an approximate value, of $\psi(t)$.

Let $D = [D^1,\ldots,D^{m_2}]$ be a square invertible matrix, i.e., a linear basis of $R^{m_2}$. Then

$$\text{pos } [D, -D] = R^{m_2}, \tag{1.8}$$

i.e., the columns of $D$ and $-D$ determine a positive linear basis of $R^{m_2}$, i.e., every vector in $R^{m_2}$ can be obtained as a positive linear combination of the column vectors of $D$ and $-D$, and none of the vectors in $D$ or $-D$ is a positive linear combination of the others. For $j = 1,\ldots,m_2$, let

$$\delta_j^+ := \inf [qy \mid Wy = D^j, \quad y \geq 0\} = \psi(D^j) \tag{1.9}$$

and

$$\delta_j^- := \inf [qy \mid Wy = D^j, \quad y \geq 0\} = \psi(-D^j). \tag{1.10}$$

If $D^j$ (or $-D^j$) does not belong to pos $W$, then we set $\delta_j^+$ (or $\delta_j^-$) $= \infty$; this will not affect the operations that are needed to be performed when working with the stochastic program (in view of our relatively complete recourse condition). Let

$$(1.11) \qquad \psi_D(t) := \inf \{\alpha \mid \binom{\alpha}{t} = \Sigma_{j=1}^{m_2} \binom{\delta_j^+}{D^j}\mu_j^+ + \Sigma_{j=1}^{m_2} \binom{\delta_j^-}{-D^j}\mu_j^-, \quad \mu_j^+ \geq 0,\ \mu_j^- \geq 0\}.$$

From our earlier remarks, it follows that

$$(1.12) \qquad \psi \leq \psi_D$$

with

$$(1.13) \qquad \psi(t) = \psi_D(t) \quad \text{whenever} \quad t = \pm D^j, \quad j = 1,\ldots,m_2.$$

Moreover notice that for any given $t$, finding $\psi_D(t)$ is particularly easy, indeed we have that

$$(1.14) \qquad \psi_D(t) = \Sigma_{j=1}^{m_2} \psi_D^j(t)$$

where

$$(1.15) \qquad \psi_D^j(t) := \inf \{\delta_j^+ \mu_j^+ + \delta_j^- \mu_j^- \mid \mu_j^+ - \mu_j^- = (D^{-1}t)_j,\ \mu_j^+ \geq 0,\ \mu_j^- \geq 0\}$$

$$= \begin{cases} \delta_j^+ (D^{-1}t)_j & \text{if } (D^{-1}t)_j \geq 0, \\ -\delta_j^- (D^{-1}t)_j & \text{if } (D^{-1}t)_j \leq 0 \end{cases}$$

Finding the value of $\psi_D(t)$ requires:

(i) premultiplying $t$ by the inverse of $D$,

(ii) by a sign check determining the value of $\psi_D^j$,

(iii) adding up the resulting values (1.14).

The subgradients of $\psi_D(t)$ are equally easy to determine. Indeed, we have that

$$\partial\psi_D(t) = \{\pi = \sigma D^{-1} \mid \sigma_j \in \Sigma^j,\ j=1,\ldots,m_2\} \tag{1.16}$$

and

$$\Sigma^j = \begin{cases} -\delta_j^- & \text{if } (D^{-1}t)_j < 0 \\ [-\delta_j^-, \delta_j^+] & \text{if } (D^{-1}t)_j = 0 \\ \delta_j^+ & \text{if } (D^{-1}t) > 0 \end{cases} \tag{1.17}$$

In addition to the operations indicated here, there is also the work required to choose a matrix $D$, and compute the slopes

$$\{(\delta_j^+, \delta_j^-),\ j = 1,\ldots,m_2\}$$

of the function $\psi$ in the directions $D^j$ and $-D^j$. But once that work is done, and $D^{-1}$ is available, then finding the value of $\psi_D(t)$ for a large number of possible values of $t$ is extremely easy.

However $\psi_D$ might not be a sufficiently good approximate of $\psi$ so that it can be used as a substitute for $\psi$. To improve the approximation we could use not just one function $\psi_D$, but a collection of functions

$$\{\psi_{D(\nu)} \quad \nu = 1,\ldots,N\}$$

where each matrix $D(\nu)$ is a square invertible matrix. For each $\nu$, we have that $\psi \leq \psi_D(\nu)$, so that $\psi \leq \inf_\nu \psi_{D(\nu)}$. In fact we can do somewhat better. Since $\psi$ is convex, its epigraph is convex, and $\psi \leq \psi_{D(\nu)}$ implies that the epigraph of $\psi_{D(\nu)}$ is contained in the epigraph of $\psi$. Thus,

$$\text{epi } \psi \supset \text{co } (\text{epi } \psi_{D(\nu)}, \ \nu = 1,\ldots,N) =: C$$

where co denotes convex hull. Now let

$$\text{co } \psi_{D(\nu)}(t) = \inf [\alpha | \binom{\alpha}{t} \in C],$$

then

$$\psi \leq \text{co } \psi_{D(\nu)} \leq \inf_\nu \psi_{D(\nu)}. \tag{1.18}$$

If by * we denote conjugation, i.e.,

$$\psi^*(u) = \sup_{t \in R^{m_2}} [u\, t - \psi(t)],$$

from the definitions and (1.18), it follows that

$$(1.19) \qquad \psi^*(u) \geq \sup_{\nu=1,\ldots,N} \psi^*_{D(\nu)}(u) = (\text{co } \psi_{D(\nu)})^*(u).$$

Each function $\psi^*_{D(\nu)}$ is the indicator function of a generalized rectangle, indeed

$$(1.20) \qquad \psi^*_{D(\nu)}(u) = \begin{cases} 0 & \text{if } -\delta_j^- \leq u[D(\nu)]^j \leq \delta_j^+ \text{ , } j = 1,\ldots,m_2, \\ +\infty & \text{otherwise.} \end{cases}$$

Using this in conjunction with (1.19), it implies.

$$(1.21) \qquad \psi^*(u) = \begin{cases} 0 & \text{only if for all } \nu = 1,\ldots,N \\ & \qquad -\delta_j^- \leq u[D(\nu)]^j \leq \delta_j^+ \text{ , } j = 1,\ldots,m_2, \\ +\infty & \text{otherwise.} \end{cases}$$

Note that we can always choose a collection $\{D(\nu),\ \nu = 1,\ldots,N\}$ such that

$$\psi(t) = \inf_\nu \psi_{D(\nu)}$$

or equivalently.

$$\psi^*(u) = \sup_\nu \psi^*_{D(\nu)}.$$

For example, if $W$ is of full rank, then we could choose for the $\{D(\nu),\ \nu = 1,\ldots,N\}$ all square invertible submatrices of $W$. Practically, of course, we would not proceed in that fashion, we would choose an appropriate subcollection, or even a collection that would not necessarily be connected with the submatrices of $W$.

The problem of choosing a parallel decomposition, or equivalently a collection of matrices $\{D(\nu),\ \nu = 1,\ldots,N\}$, that yields a "good" approximation of $\psi$ and involves a relatively small number $N$ of functions $\psi_{D(\nu)}$ is very much an open question, and needs much further investigation. One possibility is to construct the matrices $D(\nu)$ in sequence, so that each one in turn will reduce as much as possible the error of the approximation. One such procedure is laid out in what follows.

To begin with we could always take $D(1) = I$ and compute using formulas (1.9) and (1.10), the corresponding vectors $\delta^+(1)$ and $\delta^-(1)$. Let

$$V(\nu) := \cup_{\ell=1}^{\nu} \{D^j(\ell),\ -D^j(\ell);\ j = 1,\ldots,m_2\}$$

i.e., all the columns that form the matrices $D(1),\ldots,D(\nu)$ that are already in our collection, and let $e(\nu) :=$ {corresponding vector of cost coefficients}, i.e., with this notation we have

$$e_j(\nu) = \inf \{qy \mid Wy = V^j(\nu),\ y \geq 0\}.$$

Let us also assume that the matrices $D(\ell)$ have been chosen so that

$$\text{co } \psi_{D(\nu)} = \inf_\nu \psi_{D(\nu)} \tag{1.22}$$

and that $\psi$ is inf-compact, (i.e., with compact level sets), as would usually be the case for well-posed problems, cf. Wets 1973. Then, to compare $\psi$ and $\text{co } \psi_{D(\nu)}$, we see how well

$$\text{lev}_1\ \psi := \{t \mid \psi(t) \leq 1\}$$

matches up with

$$\text{lev}_1\ (\text{co } \psi_{D(\nu)}) := \{t \mid \text{co } \psi_{D(\nu)}\ (t) \leq 1\}.$$

In view of the assumptions we have made, we have that

$$\text{lev}_1\ \psi = \{t \mid t = \Sigma_{j=1}^{m_2} \lambda_j\ q_j^{-1}\ W^j,\ \Sigma_{j=1}^{m_2} \lambda_j = 1,\ \lambda_j \geq 0\}$$

$$= \text{co } \{(q_j^{-1}\ W^j),\ j = 1,\ldots,m_2\}$$

and

$$\text{lev}_1(\text{co } \psi_{D(\nu)}) = \text{co } \{(e_j(\nu)^{-1}\ V^j(\nu)),\ j = 1,\ldots,m(\nu)\}$$

As mentioned in the Introduction, there is a dual scheme that could also be used to obtain a parallel decomposition of the recourse function (1.1). Indeed, since $\psi$ is sublinear and polyhedral:

$$\psi(t) = \sup \{a^k t \mid k = 1,\ldots,p\} \tag{1.24}$$

i.e., is the sup of a collection of linear functions of $t$. It can be shown that each $a^k$ can be identified with the multipliers associated to some basis of $W$ (for some $t \in \text{pos } W$), see Wets (1974). Assuming that we have obtained these vector $\{a^k, k = 1,\ldots,p\}$, we could then find the values of $\psi(t)$, for any $t \in \text{pos } W$, by calculating in parallel the values $a^k t$ for $k = 1,\ldots,p$ and take their supremum.

These two parallel decompositions of $\psi$ could of course be combined, in practice. However, finding a good dual representation of $\psi$ may be more demanding than obtaining the parallel decomposition that we have outlined first.

## 2. APPLICATION TO STOCHASTIC PROGRAMMING

We show here how to use the parallel decomposition of linear programs outlined in Section 1. To simplify the presentation let us assume, that the stochastic program.

$$\begin{aligned} &\text{find } x \in R^{n_1} \text{ such that } Ax = b,\ x \geq 0, \\ &\text{and } z = cx + E\{Q(x,\tilde{\xi})\} \text{ is minimized} \end{aligned} \tag{2.1}$$

where

$$(2.2) \qquad Q(x,\xi) = \inf_{y \in R_+^{n_2}} \{qy \mid Wy = h - Tx\}$$

and

$$\xi = (h, T_1, \ldots, T_{m_2})$$

a random vector with support $\Xi \subset R^N$ and distribution $P$, is a stochastic program with <u>complete recourse</u>, i.e.,

$$\text{pos } W = R^{m_2},$$

which means that $\psi$ as defined by (1.1) is finite everywhere and thus there will be no need to introduce feasibility cuts. Let us also assume, that $\psi$ is inf-compact, as is to be expected in practice (excluding some degenerate cases).

Before we actually begin with solving the stochastic program (2.1), we design the parallel decomposition of the linear program that determines the recourse function. To fix the ideas, let us consider an example in $R^2$. Let

$$W = \begin{bmatrix} -3 & -1 & 2 & 3/2 & -3/2 \\ 1 & 2 & 1 & -1/2 & -1 \end{bmatrix}, \quad q = [1, 1, 1, 1, 1]$$

then pos $W = R^{m_2}$, and $\psi$ is compact. Suppose that we consider the collection

$$D(1) = \begin{bmatrix} -1 & -1 \\ 0 & 1 \end{bmatrix}, \quad D(2) = \begin{bmatrix} -1 & 0 \\ 1 & 1 \end{bmatrix}, \quad D(3) = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, \quad D(4) = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

then

$$\delta^+(1) = [4/9,\ 3/5] \quad , \qquad \delta^-(1) = [3/5,\ 14/9]$$

$$\delta^+(2) = [3/5,\ 3/5] \quad , \qquad \delta^-(2) = [14/9,\ 4/3]$$

$$\delta^+(3) = [3/5,\ 4/5] \quad , \qquad \delta^-(3) = [4/3,\ 10/9]$$

$$\delta^+(4) = [4/5,\ 3/5] \quad , \qquad \delta^-(4) = [10/9,\ 4/9]$$

and comparing $\text{lev}_1\ \psi$ and $\text{lev}_1(\text{co}\ \psi_{D(\nu)})$ we obtain Figure 1.

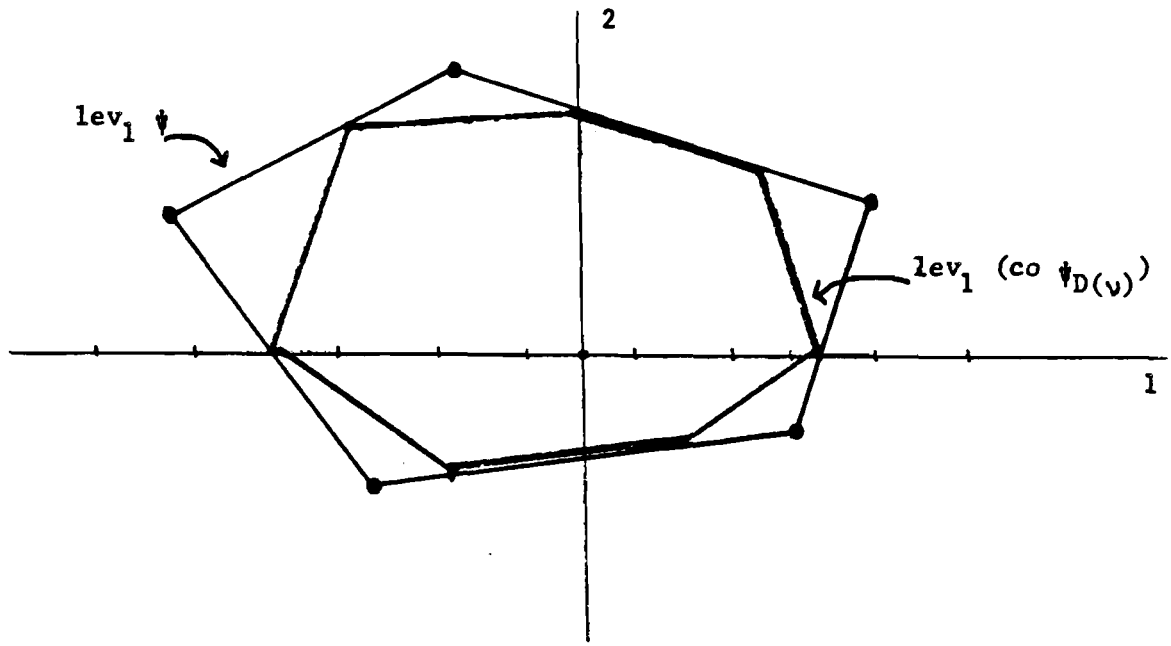


Figure 1. Comparing $\psi$ and its approximate.

It is easy to see that a slightly richer collection of matrices $\{D(\nu)\}$ will lead to a parallel decomposition that closely approximates $\psi$. But

even if we work with this one the error should not be too large. The maximum error occurs when $t$ is a positive multiple of

$$t = \begin{pmatrix} -3 \\ -1 \end{pmatrix}, \quad \text{then} \quad \psi(t) = 1 \quad \text{and} \quad \text{co}\ \psi_{D(\nu)}(t) = 2.378.$$

Substituting for $D(1)$, the two matrices

$$D(1) = \begin{bmatrix} -1 & -3 \\ 0 & 1 \end{bmatrix}, \qquad D(1') = \begin{bmatrix} -3 & -1 \\ 1 & 1 \end{bmatrix}$$

reduces substantially the error between $\psi$ and $\text{co}\ \psi_{D(\nu)}$. The maximum error then occurs when the vector $t$ is a multiple of

$$t = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \quad \text{then} \quad \psi(t) = 1 \quad \text{and} \quad \text{co}\ \psi_{D(\nu)}(t) = 7/5.$$

Replacing $D(4)$, by the two matrices

$$D(4) = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} \quad \text{and} \quad D(4') = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}$$

would make the error between $\psi$ and $\text{co}\ \psi_{D(\nu)}$ inconsequential.

Now suppose that the analysis has led us to the collection $\{D(k), K = 1,\ldots,\nu\}$ to form the basis of our parallel decomposition. Let us also suppose that by sampling or using a discrete approximation to $P$ we have reduced the stochastic program to one where only a finite number of possible values of $\underset{\sim}{\xi}$ need to be examined, say

$$\{\xi^{\ell} = (h^{\ell}, T^{\ell}),\ \ell = 1,\ldots,L\}$$

For given $\hat{x}$, we construct the collection

$$\{t^{\ell} = h^{\ell} - T^{\ell}\,\hat{x},\ \ell = 1,\ldots,L\}$$

To find $(\inf_k \psi_{D(k)})\,(t^{\ell})$ and $\partial(\text{co}\ \psi_{D(k)})\,(t^{\ell})$ we use the formulas given by (1.15) and (1.17). For each $k = 1,\ldots,\nu$, we calculate $D^{-1}(k)\ t$, and set

$$\psi_{D(k)}(t^{\ell}) = \Sigma_{j=1}^{m_2}\ \psi^{j}_{D(k)}(t^{\ell})$$

cf. (1.15). Then, we simply identify the index $\hat{k}$ that yields the inf of these values $\{\psi_{D(k)}(t^{\ell})\}$. This fixes the value of $\inf_k \psi_{D(k)}(t^{\ell})$ and from there, with the help of (1.17) we obtain the subdifferential by setting

$$\partial(\text{co}\ \psi_{D(\nu)})\,(t^{\ell}) = \sum_{j=1}^{m_2}\ (\Sigma^{j}(\hat{k})).$$

Thus we have (upper) approximates for

$$Q(\xi^{\ell},\ \hat{x}) \leq \text{co}\ \psi_{D(\nu)}(t^{\ell})$$

$$\partial_x Q(\xi^{\ell},\ \hat{x}) \supseteq -T^{\ell}(\partial\ \text{co}\ \psi_{D(\nu)}(t^{\ell}))$$

If the probabilities $\{p_\ell, \ell = 1,\ldots,L\}$ are associated to the points $\{\xi^\ell, \ell = 1,\ldots,L\}$, it follows that

$$E\{Q(\underset{\sim}{\xi}, \hat{x})\} \geq -\Sigma_{\ell=1}^{L} p_\ell (\text{co } \psi_{D(\nu)}(t^\ell))$$

and

$$\partial_x E\{Q(\underset{\sim}{\xi}, \hat{x})\} \geq -\sum_{\ell=1}^{L} p_\ell T^\ell (\text{co } \psi_{D(\nu)}(t^\ell))$$

This is all the information that we need to work with decomposition type methods, such as the L-shaped procedure Wets (1986), to solve stochastic programs of type (2.1).

## 3. PARALLEL PROCESSING IMPLEMENTATION

G. Dantzig (1985) suggested that parallel processors be used for solving stochastic programs of type (2.1) or more precisely, multistage versions of that model. For the problem considered in Section 2, his suggestions correspond to reserving a processor for solving for each $t^\ell$, the linear program

$$\text{(3.1)} \qquad \text{find } y \in R^{m_2} \text{ such that } Wy = t^\ell$$
$$\text{and } w_\ell = qy \text{ is minimized}$$

or more exactly its dual.

$$\text{(3.2)} \qquad \text{find } \pi \in R^{m_2} \text{ such that } \pi W \leq q$$
$$\text{and } v_\ell = \pi t^\ell \text{ is maximized.}$$

We then form a weighted combination of the values and solutions of (3.2) to obtain the quantities needed in the implementation of decomposition type methods for solving the stochastic program, see the end of Section 2. This procedure would thus require L parallel processors, all equipped to solve general linear programs.

The parallel decomposition of the recourse problem however would allow us to use extremely simple processors as a matter of fact we can actually build hardware that would solve simultaneously a very large number of these trivial linear programs giving us for each $t^{\ell}$ almost instantaneously the exact or an approximate solution of (3.2). A double array of such circuits would thus allow us to process in parallel the vectors $\{t^{\ell}, \ell = 1,\ldots,L\}$ in about the time it takes to solve one of these simple linear programs, cf.(1.15). The implication is that construction of the optimality cuts in the L-shaped method could be reduced to an insignificant operation, where as up to now this was the operation that was the real stumbling block in obtaining fast solutions to stochastic linear programs.

Naturally there is some set-up time required in designing the parallel decomposition of the recourse problem, but this can also be done systematically using bunching techniques, see Section 4 of Wets (1986). Indeed what we need to find are the coefficients $\delta_j^+$ and $\delta_j^-$ to associate to a collection of linear programs that differ only in the right-hand sides, cf. (1.9) and (1.10).

## 4. CONCLUSION

We have shown that with the appropriate design of a parallel micro-processor we could reduce the time required for solving stochastic programs

to essentially the same time than that required for solving deterministic linear programs of the same type, ignoring the set-up time required for designing the parallel decomposition of the recourse problem.

For the application of stochastic programming in the modeling of decision making under uncertainty, this approach allows us to study the stability of the solution under perturbations of the probability measure associated to the realizations $\{\xi^\ell = (h^\ell, T^\ell), \ell = 1,\ldots,L\}$, (there is of course no need to redesign the parallel decomposition of the recourse problem). Thus if the $\xi^\ell$ are various scenarios that we consider as future possibilities, we can then make an analysis of the solution as we change the probability of these events, without having to "mix" (?) the solutions that would be obtained by considering each scenario individually (without reference to other possibilities).

# REFERENCES

Birge, J. and R. Wets, (1985), Designing approximation schemes for stochastic optimization problems, in particular stochastic programs with recourse, Mathematical Programming Study.

Dantzig, G. (1985), Lecture given at the Boston Mathematical Programming Symposium, Cambridge, Mass., August.

Wets, R. (1972), Stochastic programs with recourse: a basic theorem for multistage problems, Z. Wahrscheinlichkeitstheorie verw. Gebiete, 201-206.

Wets, R. (1974), Stochastic programs with fixed recourse: the deterministic equivalent problem, SIAM Review, 16, 309-339.

Wets, R. (1973), On inf-compact mathematical programs, Springer Verlag Lecture Notes in Computer Sciences, 3, 426-436.

Wets, R. (1986), Large scale linear programming techniques in stochastic programming, in Numerical Methods for Stochastic Optimization, eds. Y. Ermoliev and R. Wets, Springer Verlag.