VARIANTS ON DANTZIG-WOLFE DECOMPOSITION WITH APPLICATIONS TO MULTISTAGE PROBLEMS

L. Nazareth

June 1983
WP-83-61

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

Preface

The System and Decision Sciences Area at IIASA has long been involved in the investigation of decomposition procedures for solving structured LP problems. In this paper, the author studies the application of the Dantzig-Wolfe procedure to alternative representations of the structured linear program and develops a new family of methods for solving multistage, staircase structured problems. These methods are also relevant to the development of stochastic programming algorithms currently under way in the Adaptation and Optimization Project of SDS.

Andrzej P. Wierzbicki
Chairman
System and Decision
Sciences Area

Abstract

The initial representation of an LP problem to which the Dantzig-Wolfe decomposition procedure is applied, is of the essence. We study this here, and, in particular, we consider two transformations of the problem, by introducing suitable linking rows and variables. We study the application of the Dantzig-Wolfe procedure to these new representations of the original problem and the relationship to previously proposed algorithms. Advantages and disadvantages from a computational viewpoint are discussed. Finally we develop a decomposition algorithm based upon these ideas for solving multistage staircase-structured LP problems.

VARIANTS ON DANTZIG-WOLFE DECOMPOSITION WITH APPLICATIONS TO MULTISTAGE PROBLEMS

L. Nazareth

1. INTRODUCTION

Obtaining the solution to a large LP problem by solving a coordinated sequence of smaller LP problems - the Decomposition Principle of Dantzig & Wolfe, 1960 - is certainly a key concept of Linear Programming. In addition to the Decomposition Principle, a particular decomposition procedure was proposed by Dantzig & Wolfe, but as a computational technique this has enjoyed a more mixed success. Since the original paper, there have been numerous applications, variants and generalizations of the basic idea, for instance, Benders, 1962, Van Slyke & Wets, 1969, Glassey, 1971, Ho & Manne, 1974, Kallio & Porteus, see Kallio, 1975, Dantzig & Abrahamson, see Abrahamson, 1981, Nurminski, 1982, to name only a few.

The initial representation of the problem, to which the Dantzig-Wolfe procedure is applied, is clearly of the essence. We study this here, and, in particular, we consider two transformations of the problem, by introducing suitable linking rows and variables. We study the application of the Dantzig-Wolfe procedure to these new representations of the problem and the relationship to previously proposed algorithms, in Section 2. In Section 3 we discuss advantages and disadvantages from a

computational viewpoint. The reader may skip this section if he wishes, and proceed directly to Section 4, where we develop a decomposition algorithm based upon earlier ideas, for solving multistage staircase LP problems. We also reexamine this algorithm in Section 5 from the point of view of dynamic programming. No computational experience is reported here, but we hope to provide this at a later date.

In the text, equations are occasionally grouped using letters e.g. (2.1a) and (2.1b). If we subsequently refer to (2.1) we mean both (2.1a) and (2.1b).

2. APPLYING THE DECOMPOSITION PRINCIPLE TO TRANSFORMED PROBLEMS

We begin by considering the problem:

$$\text{minimize} \quad c_1x_1 + c_2x_2$$

$$A_1x_1 + A_2x_2 = b \tag{2.1a}$$

$$x_1, x_2 \geq 0$$

where A_1 and A_2 are $(m \times n_1)$ and $(m \times n_2)$ matrices respectively, and the other quantities are vectors of appropriate dimension.

The dual of (2.1a) is:

$$\text{maximize} \quad b\pi$$

$$A_1^T\pi \leq c_1 \tag{2.1b}$$

$$A_2^T\pi \leq c_2$$

Let us first review some standard ways of decomposing (2.1a) or (2.1b). For example, if we write (2.1a) as

$$\min_{x_1 \in R} \{c_1x_1 + \min_{x_2 \geq 0} [c_2x_2 | A_2x_2 = (b - A_1x_1)]\}$$

where

$$R \triangleq \{x_1 \geq 0 \mid \exists\, x_2 \geq 0 \quad \text{s.t.} \quad A_2x_2 = b - A_1x_1\}$$

we are led to Benders' decomposition procedure. This is conceptually, though of course not computationally equivalent to applying the Dantzig-Wolfe procedure to the dual (2.1b), with the constraints $A_2^T\pi \leq c_2$ defining the subproblem and the remaining constraints defining the master. Both decompositions are so standard that we do not give further details here. We shall emphasize throughout the Dantzig-Wolfe procedure, since results about the Benders' procedure follow from their dual equivalence.

Let us now consider two different sets of transformations of the problem (2.1). The first introduces new variables χ_1 and χ_2 and a *linking row*, and leads to the following reformulation:

$$\begin{aligned}
\text{minimize} \quad & c_1x_1 + c_2x_2 \\
& A_1x_1 \qquad - \chi_1 \qquad = 0 \\
& \qquad A_2x_2 \qquad - \chi_2 = 0 \\
& \qquad\qquad \chi_1 + \chi_2 = b \\
& x_1, x_2 \geq 0
\end{aligned} \tag{2.2a}$$

Its dual transformation introduces variables π_1 and π_2 and the original variable π is treated as a linking variable, leading to:

$$\begin{aligned}
\text{maximize} \quad & b\pi \\
& A_1^T\pi_1 \qquad\qquad \leq c_1 \\
& \qquad A_2^T\pi_2 \qquad \leq c_2 \\
& -\pi_1 \qquad + \pi = 0 \\
& \qquad -\pi_2 + \pi = 0
\end{aligned} \tag{2.2b}$$

If we apply the Dantzig-Wolfe procedure to (2.2b) with two subproblems defined by the constraint sets $P_1 = \{\pi_1 : A_1^T \pi_1 \leq c_1\}$ and $P_2 = \{\pi_2 : A_2^T \pi_2 \leq c_2\}$ and if π^p_{ij} , $j = 1,2,\ldots,n_{ip}$ are the extreme points of P_i , $i = 1,2$ and π^r_{ij} , $j = 1,2,\ldots,n_{ir}$ the extreme rays of P_i, then the corresponding master program is:

(2.3)

$$
\begin{array}{lrl}
\text{maximize} & & b\pi \\
& - \sum_{j=1}^{n_{1p}} \lambda_{1j}\pi^p_j - \sum_{j=1}^{n_{1r}} \mu_{1j}\pi^r_{ij} + \pi = 0 & \\
& - \sum_{j=1}^{n_{2p}} \lambda_{2j}\pi^p_{2j} - \sum_{j=1}^{n_{2r}} \mu_{2j}\pi^r_{2j} + \pi = 0 & \\
& \sum_{j=1}^{n_{1p}} \lambda_{1j} = 1 & \\
& \sum_{j=1}^{n_{2p}} \lambda_{2j} = 1 & \\
& \lambda_{ij} \, , \; \mu_{ij} \geq 0 &
\end{array}
$$

Denoting by χ_1 and χ_2 the prices corresponding to the first two constraints (these are, of course, realizations of the variables of the primal (2.2a)) then the corresponding subproblems are

(2.4)

$$
\left.
\begin{array}{ll}
\text{maximize} & \chi_i^T \pi_i \\
& A_i^T \pi_i \leq c_i
\end{array}
\right\}
\begin{array}{l}
\text{s/pi} \\
i = 1,2
\end{array}
$$

It is easily verified that this is equivalent to the *symmetric Dantzig-Wolfe decomposition* given in Nazareth, 1978, where details of the dual procedure, *symmetric Benders decomposition*, can also be found. Also for an example of the use of the transformation (2.2a) in a somewhat different algorithmic setting, see Schechtman and Granville, 1982.

The second transformation of (2.1a) that we consider introduces just a linking variable χ leading to the following equivalent problem:

$$
\begin{array}{lll}
\text{minimize} & c_1x_1 & + c_2x_2 \\
& A_1x_1 - \chi & = 0 \\
& \chi + A_2x_2 & = b \\
& x_1, x_2 \geq 0 &
\end{array}
\qquad \text{(2.5a)}
$$

Note that this can be obtained from (2.2a) by summing the last two equations and thus eliminating χ_2. Its dual introduces a *linking row* in variables π_1 and π_2 as follows:

$$
\begin{array}{lll}
\text{maximize} & b\pi_2 & \\
& A_1^T\pi_1 & \leq c_1 \\
& -\pi_1 + \pi_2 = 0 & \\
& A_2^T\pi_2 & \leq c_2
\end{array}
\qquad \text{(2.5b)}
$$

If we let the constraints corresponding to $P = \{(\chi_1, x_2) \mid \chi + A_2x_2 = b\ ,\ x_2 \geq 0\}$ define the subproblem in the Dantzig-Wolfe procedure applied to (2.5a), the corresponding master problem is:

$$
\begin{aligned}
\text{minimize} \quad & c_1 x_1 + \sum_j (c_2 x_{2j}^p)\lambda_j + \sum_j (c_2 x_{2j}^r)\mu_j \\
& A_1 x_1 - \sum_j \chi_j^p \lambda_j + \sum_j \chi_j^r \mu_j = 0 \\
& \sum_j \lambda_j = 1 \\
& \lambda_j \, , \, \mu_j \geq 0 \quad .
\end{aligned} \tag{2.6}
$$

where (χ_j^p, x_{2j}^p) and (χ_j^r, x_{2j}^r) represent extreme points and extreme rays of P.

If π is the vector of dual multipliers corresponding to the first constraint of (2.6), then the associated subproblem is:

$$
\begin{aligned}
\text{minimize} \quad & \pi\chi + c_2 x_2 \\
& \chi + A_2 x_2 = b \\
& x_2 \geq 0
\end{aligned} \tag{2.7}
$$

For an LP model that uses linking variables, see, for example, Nazareth, 1980. Within the context of the stochastic (linear) programming problem, linking variables are used for example by Wets, 1974b, 1983, to induce separability in the objective function, and by Nazareth & Wets, 1983 in an algorithmic approach utilizing generalized programming, which is obviously related to the above use of the Dantzig-Wolfe procedure. Linking variables are also used by Nurminski, 1982, who solves a more general nonlinear programming problem (LP is a special case, of course). This problem is defined, *a priori*, in terms of linking variables, and they provide the basis for an algorithmic approach that utilizes methods of nonsmooth nonlinear optimization. For LP, his algorithm, suitably extended to handle unboundedness, can in fact be interpreted in the above terms, viz. (2.6) and (2.7).

There is an alternative view of the decomposition leading to (2.6) and (2.7) which is similar to that taken in Nazareth & Wets, 1983, and yields considerable insight. Here write (2.5a) as:

$$
\begin{aligned}
&\text{minimizes} && c_1 x_1 + \psi(\chi) \\
(2.8) \qquad & && A_1 x_1 - \chi = 0 \\
& && x_1 \geq 0
\end{aligned}
$$

where

$$\psi(\chi) = \min_{x_2 \geq 0} \{c_2 x_2 \mid A_2 x_2 = b - \chi\}$$

and $\psi(\chi) = +\infty$ if $A_2 x_2 = b - \chi$, $x_2 \geq 0$ is infeasible. $\psi(\chi)$ is a *convex polyhedral function*. There is no need to prove this here since it follows from the results in Wets, 1974a, for the particular case when the resource vector b has discrete distribution with probability 1 (i.e. is deterministic).

If we apply the generalized programming method i.e. inner linearization of $\psi(x)$ combined with a column generation procedure to (2.8), (see, for example, Shapiro, 1979), then we obtain the following master program:

$$
\begin{aligned}
&\text{minimize} && c_1 x_1 + \sum_j \psi(\chi_j^p)\lambda_j + \sum_j \psi(\chi_j^r)\mu_j \\
& && A_1 x_1 + \sum_j \chi_j^p \lambda_j + \sum_j \chi_j^r \mu_j && = 0 \\
(2.9) \qquad & && \sum_j \lambda_j && = 1 \\
& && x_1 \,,\ \lambda_j \,,\ \mu_j \geq 0
\end{aligned}
$$

where χ_j^p are points where $\psi(\chi)$ is inner linearized, and $\psi(\chi_j^r)$ are lines of recession of $\psi(\chi)$.

If π is again the vector of dual multipliers corresponding to the first constraint of (2.9), then the corresponding subproblem is:

$$\text{(2.10)} \qquad \underset{\chi}{\text{minimize}} \quad [\psi(\chi) + \pi\chi]$$

and this is equivalent to:

(2.10b) $$\underset{x_2 \geq 0}{\text{minimize}} \quad [c_2 x_2 + \pi\chi \mid A_2 x_2 = b - \chi]$$

With the appropriate identification, namely, $\psi(\chi_j^p) = c_2 x_{2j}^p$, $\psi(\chi_j^r) = c_2 x_{2j}^r$, we see the equivalence of (2.6) and (2.9).

Finally, the following observation will turn out to be useful in the dynamic programming interpretation given later. If χ^* is a stationary point of (2.10), then

$$0 \in \partial[\psi(\chi^*) + \pi\chi^*]$$

where ∂ denotes the (Clark) subdifferential. Thus

(2.10c) $$-\pi \in \partial[\psi(\chi^*)]$$

and so $(-\pi)$ defines a supporting hyperplane to $\psi(\chi)$ at χ^*.

3. DISCUSSION

The revised simplex method applies naturally to (2.1a) when $m << n = (n_1 + n_2)$, since it would require an $(m \times m)$ basis matrix. When applied to (2.1b) where, for the moment, we disregard partitioning of the constraints, it would appear, at first sight, that an (n x n) basis matrix is needed. However, the cost of maintaining the inverse of such a matrix is nowhere as prohibitive as it would first seem, for the simple reason that much of it must consist of columns of the n x n identity matrix. In general, the basis matrix, when suitably permuted, is of the form $B = \left[\begin{array}{c|c} I & C \\ \hline 0 & B \end{array}\right]$ when I is an $(n - m) \times (n - m)$ identity matrix, C is $(n - m) \times m$ and B is $m \times m$. Assuming that B is invertible, $B^{-1} = \left[\begin{array}{c|c} I & -CB^{-1} \\ \hline 0 & B^{-1} \end{array}\right]$ and only B need be inverted (or factored) and updated. Using a compact basis method, the basis factorization requires $0(m^3)$ operations, basis updating $0(m^3)$, and computing updated columns, prices, and reduced costs are $0(mn)$ operations. Further saving can be effected over these rudimentary estimates, and indeed this is precisely what is achieved by the dual revised simplex method applied to (2.1a) - conceptually the equivalent of the revised simplex method applied to (2.1b).

Let us consider next the standard decompositions. With the partitioning of (2.1a), Benders' decomposition applies quite naturally. This is *conceptually* equivalent to applying the Dantzig-Wolfe procedure to (2.1b). If the first set of constraints define the master, and n_1 is reasonably small, all is fine. However if $n_1 >> m$ are $n_2 >> m$ then the benefits of a compact basis technique are much less easy to realize than in the case discussed in the previous paragraph. Since the number of extreme points of the subproblem are $0(n_2 C_m)$, the master problem will have many more columns than A_1^T and there will certainly be $n_1 \times n_1$ submatrices of the master which do not include any unit columns i.e. columns of the $n_1 \times n_1$ identity matrix that corresponds to the slack variables of the master. If the intermediate solutions on the path to the optimal solution of the master, and the optimal solution itself are such that many master constraints are slack, then every basis will contain many unit columns and compact basis techniques of the sort discussed earlier will achieve significant savings. However, the potential for difficulty remains. These sorts of considerations are precisely those that could occur when Benders' decomposition is applied to (2.1a) and $n_1, n_2 >> m$, the concern being that a large number of added constraints could be active at some point. In the subproblem in either decomposition, the dimensions are those of A_2, and as we have seen, the relative size of n_2 and m does not present a difficulty. However, because many new columns are added in the Dantzig-Wolfe master (or correspondingly constraints in the Benders' master) it is advisable to keep n_1 as small as possible relative to the other problem dimensions. Dantzig-Wolfe (or Benders) decomposition is most appropriately applied when there is a natural hierarchy in the problem with relatively few rows (or variables) defining the master. The form of the partition i.e. the relative size of n_1 and n_2 is important. Other points to note about Dantzig-Wolfe (or Benders) decomposition are that *structure is lost* in the original rows (columns) defining the master problem, and that the density of the LP matrices in the master often considerably exceeds that of A_1^T .

Turning now to the relative merits of symmetric decompositions, we see that they may be more useful when there is a need to simultaneously coordinate different models, but where there is *no clearcut hierarchical structure*. The standard and the symmetric decomposition are complementary in that one might be appropriate for a problem for which the other can be expected to encounter difficulty. Thus, in contrast to the standard decompositions, when $n_1, n_2 >> m$, the symmetric D-W master will have 2m + 2 rows, its size being determined by the number of variables in (2.2b). Note that the form of the partition i.e. the relative size of n_1 and n_2 is immaterial. Structure in the rows defining the subproblems is retained, and their sparsity pattern is unaltered. Similar comments apply to the symmetric Benders' Decomposition.

The symmetric decompositions have some disadvantages of their own. Suppose we consider a problem of the form (2.1a), with A_1 and A_2 sparse. If we applied the simplex method to (2.1a) we would employ an (m x m) *sparse* basis matrix, say B_1. When the symmetric Dantzig-Wolfe decomposition is applied to (2.1b) (or equivalently the symmetric Benders' to (2.1a)) then we have a master of the form (2.3) with 2m + 2 rows and $m + \sum_i n_{ip} + \sum_i n_{ir}$ columns, of which the last m can be expected to be in a master basis matrix, say B_2. The additional m columns of B_2 are *dense*, and B_2 will thus be much more expensive to work with than B_1, even when compact basis techniques are employed. This is a serious disadvantage when compared to the usual simplex method for those cases when it is possible to apply the latter method directly i.e. when we do not have a situation where two models must be linked through some coordination procedure. There are however some ameliorating features of models, and modifications of symmetric decomposition that help matters:

a) There are generally only a few linking rows and columns. Suppose, for example, in the symmetric Dantzig-Wolfe scheme there are only m_2 linking columns. Then the master basis will be expected to require only m_2 additional columns to those taken from its sparse portion, and the resulting algorithm may thus be reasonably efficient for this class of problems.

b) Techniques for block-angular systems could be applied to the master problem.

c) It is possible to use symmetric decomposition in combination with a relaxation of constraints. We noted earlier that compact basis techniques can be effectively used within algorithms based on the standard decomposition because, in situations when the basis is potentially large, many columns correspond to those of an identity matrix. We can induce a similar situation by modifying (2.3) as follows:

$$
\begin{aligned}
&\text{maximize} \quad b\pi \\
&\pi - \sum_{j=1}^{n_{1p}} \lambda_{1j}\pi^{p}_{1j} - \sum_{1}^{n_{1r}} \mu_{1j}\pi^{r}_{1j} = 0 \\
&\pi - \sum_{j=1}^{n_{2p}} \lambda_{2j}\pi^{p}_{2j} - \sum_{j=1}^{n_{2r}} \mu_{2j}\pi^{r}_{2j} + D\theta = 0 \\
&\sum_{j=1}^{n_{1p}} \lambda_{1j} = 1 \\
&\sum_{j=1}^{n_{2p}} \lambda_{2j} = 1 \\
&-\delta \leq \theta_j \leq \delta \quad , \quad \lambda_{ij} \, , \, \mu_{ij} \geq 0 \quad .
\end{aligned}
\tag{3.1}
$$

δ is a non-negative homotopy-like parameter which can be progressively reduced to zero and D is a diagonal matrix of row scales. In effect, we are solving a series of perturbed problems which tend to the original, and we do this by relaxing the influence of A_2 and c_2 on the *prices* in (2.1a), or equivalently the influence of the second set of *constraints* in (2.1b). Note that we do *not* thus violate primal feasibility. When δ is relatively large it is likely that only the optimal solution of s/p1 associated with (3.1) will influence its optimal solution, with other basis columns being columns of the matrix D. For computational purposes it may be preferable to work with the dual of (3.1).

Finally, let us consider the decomposition applied to (2.5a), namely (2.6) and (2.7), which we call *decomposition by tenders*, in keeping with the terminology of Nazareth & Wets, 1983. The method although price directive, has some of the features of resource directive methods (see Shapiro, 1979), since $\chi = \sum_j \lambda_j \chi_j^p + \sum_j \mu_j \chi_j^r$ at any iteration gives the allocation of resources to the set of activities associated with A_1, and $(b - \chi)$ gives the allocation to the set of activities associated with A_2. χ_j^p or χ_j^r are *tenders* and we seek, in effect a single tender associated with the optimal allocation to the two sets of activities. Decomposition by tenders applies naturally to the primal system (2.1a) when $m << n$, since m determines the size of the basis. (However a related scheme in which the Dantzig-Wolfe procedure is applied to the dual problem could be worked out which utilizes compact basis techniques more effectively than symmetric decomposition.) We can observe also that structure in A_1 and A_2 is not lost, and that the method fits within the framework of a time-staged sequence of subproblems rather than a *hierarchy* of control as in the usual Dantzig-Wolfe procedure, or a *coordinated set* of subproblems as in symmetric decomposition. The vectors χ_j^p or χ_j^r may be fairly dense vectors, but we can also expect relatively few of them in an optimal basis. Indeed all that is needed is *one* tender corresponding to the *optimal partitioning* of the resource. Overall, decomposition by tenders looks very promising, and we explore its potential for solving multistage (staircase) LP problems in the next section.

4. APPLICATIONS TO MULTISTAGE (STAIRCASE) PROBLEMS

To avoid the difficulties of notation associated with working with a t-stage problem, when t is arbitrary, we confine our description to a 3-stage problem. This is quite adequate for giving the basic ideas, and results for the general multistage problem can easily be inferred.

Consider therefore the problem:

$$
\begin{aligned}
\text{minimize} \quad & c_1x_1 + c_2x_2 + c_3x_3 \\
& A_1x_1 = b_1 \\
& B_1x_1 + A_2x_2 = b_2 \\
& B_2x_2 + A_3x_3 = b_3 \\
& x_1 , x_2 , x_3 \geq 0
\end{aligned}
\tag{4.1}
$$

In the *nested decomposition* algorithm (see Ho & Manne, 1974 and Glassey, 1971), the constraints $A_1x_1 = b_1$, $x_1 \geq 0$ define the first subproblem and the remaining constraints the master, say M/1. M/1 is in turn decomposed so that the constraints of M/1 generated by original constraints $B_1x_1 + A_2x_2 = b_2$, $x_1, x_2 \geq 0$ and a convexity row, define a new subproblem and the remaining constraints lead to a level-2 master, say M/2. M/2 is in turn decomposed and so on. *Reconstruction* of the optimal solution is necessary (see Ho, 1974) and is often a numerically taxing procedure. Dantzig & Abrahamson, see Abrahamson, 1981, have proposed working with the dual of (4.1), along with special startup procedures.

Let us now consider an approach based upon transformations of (4.1) analogous to those leading to (2.5a) and (2.5b). (4.1) then becomes

$$
\begin{array}{lllll}
\text{minimize} & c_1x_1 & + c_2x_2 & + c_3x_3 & \\
& A_1x_1 & & & = b_1 \\
& B_1x_1 - \chi_1 & & & = 0 \\
& + \chi_1 & + A_2x_2 & & = b_2 \\
& & B_2x_2 - \chi_2 & & = 0 \\
& & + \chi_2 & + A_3x_3 & = b_3 \\
& x_1 \ , & x_2 \ , & x_3 \geq 0 &
\end{array}
\tag{4.2}
$$

Now let us apply nested decomposition to (4.2), in an analogous manner to the decomposition of (2.5) leading to (2.6). We call this *multistage decomposition by tenders*. Let the constraints $\chi_2 + A_3x_3 = b_3$, $x_3 \geq 0$ define the first subproblem, the corresponding master, say M/1, is

$$
\begin{array}{lll}
\text{minimize} & c_1x_1 & \\
& A_1x_1 & = b_1 \\
& B_1x_1 - \chi_1 & = 0 \qquad (4.3a) \\
& + \chi_1 + A_2x_2 & = b_2 \\
& B_2x_2 - \sum_j \chi_2^j \lambda_{2j} & = 0 \qquad (4.3b) \\
& \sum_j \delta_{2j}\lambda_{2j} & = 1 \\
x_1 \ , \ x_2 \ , \ \lambda_{2j} \geq 0 & &
\end{array}
$$

where p_{2j} and δ_{2j} are given by

$$(4.4a) \qquad p_{2j} = (0 \mid c_3)\begin{pmatrix} \chi_2^j \\ x_3^j \end{pmatrix} = c_3 x_3^j$$

$$(4.4b) \qquad \delta_{2j} = \begin{cases} 1 & \text{if} \quad (\chi_2^j, x_3^j) \quad \text{is an extreme point} \\ 0 & \text{if} \quad (\chi_2^j, x_3^j) \quad \text{is an extreme ray} \end{cases}$$

Now if we again decompose (4.3) where the constraints (4.3b) define the subproblem, with extreme points or rays denoted by $(\chi_1^j, x_2^j, \lambda_2^j)$ then the corresponding master, say M/2 is:

$$(4.5) \qquad \begin{aligned} \text{minimize} \quad & c_1 x_1 + \sum_j p_{1j} \lambda_{1j} \\ & A_1 x_1 = b_1 \\ & B_1 x_1 - \sum_j \chi_1^j \lambda_{1j} = 0 \\ & \sum_j \delta_{1j} \lambda_{1j} = 1 \\ & x_1 \, , \, \lambda_{1j} \geq 0 \end{aligned}$$

where p_{1j} and δ_{1j} are defined by:

$$(4.5a) \qquad p_{1j} = (0 \mid c_2 \mid p_2)\begin{pmatrix} \chi_1^j \\ x_2^j \\ \lambda_2^j \end{pmatrix} = c_2 x_2^j + p_2 \lambda_2^j$$

$$\text{(4.5b)} \qquad \delta_{1j} = \begin{cases} 1 & \text{if } (\chi_1^j, x_2^j, \lambda_2^j) \text{ is an extreme point} \\ 0 & \text{if } (\chi_1^j, x_2^j, \lambda_2^j) \text{ is an extreme ray} \end{cases}$$

If $(\sigma_1, \pi_1, \rho_1)$ are the dual multipliers associated with (4.5) then pricing out the variables λ_{1j} requires us to

$$\underset{j}{\text{minimize}} \ (p_{1j} + \pi_1 \chi_1^j) = \underset{j}{\text{minimize}} \ (c_2 x_2^j + p_2 \lambda_{2j} + \pi_1 \chi_1^j)$$

This is equivalent to solving the subproblem:

$$\begin{aligned} \text{minimize} \quad & \pi_1 \chi_1 + c_2 x_2 + \sum_j p_{2j} \lambda_{2j} \\ & \chi_1 + A_2 x_2 = b_2 \\ & B_2 x_2 - \sum_j \chi_2^j \lambda_{2j} = 0 \\ & \sum_j \delta_{2j} \lambda_{2j} = 1 \\ & x_2 \ , \ \lambda_{2j} \geq 0 \end{aligned} \qquad \text{(4.6)}$$

Similarly, if $(\sigma_2, \pi_2, \rho_2)$ are the prices associated with the constraints of (4.6), then pricing out the λ_{2j} variables requires us to solve

$$\underset{j}{\text{minimize}} \ p_{2j} + \pi_2 \chi_2^j = \underset{j}{\text{minimize}} \ (c_3 x_3^j + \pi_2 \chi_2^j)$$

this requires us to solve the subproblem

$$\begin{aligned} \text{minimize} \quad & \pi_2 \chi_2 + c_3 x_3 \\ & \chi_2 + A_3 x_3 = b_3 \\ & x_3 \geq 0 \end{aligned} \qquad \text{(4.7)}$$

The three LP problems (4.5), (4.6) and (4.7) above define the decomposition scheme. We can summarize it by:

$$
\begin{aligned}
\text{minimize} \quad & \pi_{i-1}\chi_{i-1} + c_i x_i + \sum_j p_{ij}\lambda_{ij} \\
& \chi_{i-1} + A_i x_i = b_i \\
& B_i x_i + \sum_j \chi_i^j \lambda_{ij} = 0 \\
& \sum_j \delta_{ij}\lambda_{ij} = 1 \\
& x_i \ , \ \lambda_{ij} \geq 0
\end{aligned}
\tag{4.8}
$$

with $4 \geq i \geq 1$ and the end conditions $\chi_o \equiv 0$, $B_4 \equiv 0$, $p_{4j} \equiv 0$, $\chi_4^1 \equiv 0$ and the convexity row omitted when $i = 3$.

Also we have the recurrence relation defining p_{ij} as:

$$
p_{(i-1)j} = c_i x_i^j + p_i \lambda_i^j \tag{4.9}
$$

with $4 \geq i \geq 1$ and end condition $c_4 \equiv 0$, $p_4 \equiv 0$.

Note that here prices are passed *forward* (*in time* if the staircase structure reflects a time-staged model) and proposals *backward. An analogous scheme can be worked out by starting at the other end of the staircase* i.e. defining the first subproblem by the constraints $A_1 x_1 = b_1$, $B_1 x_1 - \chi_1 = 0$, $x_1 \geq 0$. *In this case prices would be passed backward (in time) and proposals forward.* See also the comments of Section 6.

Convergence of the above multistage decomposition procedure follows from the usual arguments. *It is important to note that the reconstruction of the solution is no longer necessary,* in contrast to earlier decompositions.

Finally, although our discussion was confined to a 3-stage model, results for a general t-stage model can easily be inferred.

5. INTERPRETATION IN TERMS OF DYNAMIC PROGRAMMING

In nested decomposition the number of columns in successive (unrestricted) subproblems grows exponentially, yet on the basis of experimental evidence, at least with the Ho-Manne algorithm (see, for example, Ho, 1974) nested decomposition works reasonably well. To the author, this has always been somewhat of a mystery.

There is however an alternative interpretation of nested decomposition algorithms, in particular, decomposition by tenders, that fits into the framework of *dynamic programming*. This interpretation gives some insight into the efficiency of such methods via Bellman's Principle of Optimality. (See also Birge, 1980 who gives a similar interpretation for other algorithms for solving multistage (stochastic) linear programs based upon nested Benders' algorithm and Rosen's partitioning method).

Thus, let us consider the following reformation of the multistage LP problem (4.2).

$$\psi_0 = \min\ c_1x_1 + \psi_1(\chi_1)$$

$$\begin{aligned} A_1x_1 &= b_1 \\ B_1x_1 - \chi_1 &= 0 \\ x_1 &\geq 0 \end{aligned} \qquad \text{(5.1a)}$$

where $\psi_1(\chi_1) = \min\ c_2x_2 + \psi_2(\chi_2)$

$$\begin{aligned} A_2x_2 &= b_2 - \chi_1 \\ B_2x_2 - \chi_2 &= 0 \\ x_2 &\geq 0 \end{aligned} \qquad \text{(5.1b)}$$

$$\text{where} \quad \psi_2(\chi_2) = \min \; c_3 x_3$$

$$A_3 x_3 = b_3 - \chi_2 \tag{5.1c}$$

$$x_3 \geq 0$$

Again, from more general results in the theory of stochastic programs with recourse, specialized to the deterministic problem (see Wets, 1974) the functions $\psi_i(\chi_i)$ are *convex* and *polyhedral*. We can view (5.1) as a backward dynamic programming recursion.

For example, if we *quantize* the vectors χ_i to have values $\{\chi_i^j \;, \; j = 1,2,\ldots,l_i\}$, then the backward recursion can be expressed as:

$$\psi_{i-1}(\chi_{i-1}^k) = \min_j \; c_i x_i + \psi_i(\chi_i^j)$$

$$A_i x_i = b_i - \chi_{i-1}^k \tag{5.2}$$

$$B_i x_i - \chi_i^j = 0$$

$$x_i \geq 0$$

where $i = 3,2,1$ and we assume the end conditions $\chi_0^k \equiv 0$ and $\psi_3(\chi_3^j) \equiv 0$. Finiteness of this procedure follows from Bellman's Principle of Optimality and the finiteness of the simplex method (under the usual non-degeneracy assumptions).

However, rather than using an arbitrary quantization, we can go much further by taking account of the convexity of $\psi_i(\chi_i)$ and its polyhedral structure. Thus if χ_i^j now represents either a corner point or a line of recession of $\psi_i(\chi_i)$ and if $\delta_{ij} = 1$ if χ_i^j is a corner point and 0 otherwise, then we can write (5.2) as a single minimization given by:

$$\psi_{i-1}(\chi_{i-1}^k) = \min\ c_i x_i + \sum_j \psi_i(\chi_i^j)\lambda_{ij}$$

$$A_i x_i = b_i - \chi_{i-1}^k$$

(5.3) $$B_i x_i - \sum_j \chi_i^j \lambda_{ij} = 0$$

$$\sum_j \delta_{ij}\lambda_{ij} = 1$$

(5.3) $$x_i \geq 0\ ,\ \lambda_{ij} \geq 0$$

We see that (5.3) is related to multistage decomposition by tenders, in particular compare with (5.8). To see this relationship more precisely, let us utilize the observation made at the end of Section 2, and show how the particular quantization χ_{i-1}^k is made using π_{i-1}. Thus we write (4.8) as:

(5.4) $$\text{minimize } [\tilde{\psi}_{i-1}(\chi_{i-1}) + \pi_{i-1}\chi_{i-1}]$$

with $$\tilde{\psi}_{i-1}(\chi_{i-1}) = \min\ c_i x_i + \sum_j p_{ij}\lambda_{ij}$$

$$A_i x_i = b_i - \chi_{i-1}$$

(5.5) $$B_i x_i - \sum_j \chi_i^j \lambda_{ij} = 0$$

$$\sum_j \delta_{ij}\lambda_{ij} = 1$$

$$x_i\ ,\ \lambda_{ij} \geq 0$$

Solving (5.4) is equivalent to finding χ_{i-1}^k such that

(5.6) $$-\pi_{i-1} \in \partial\tilde{\psi}_{i-1}(\chi_{i-1}^k)$$

In other words, if we solve (5.5) and seek a solution which is optimal and whose associated dual multipliers, say $(\sigma_i, \pi_i, \rho_i)$ satisfy

(5.7) $$\pi_{i-1} = \sigma_i \quad .$$

(or if the optimal solution is not unique, $\pi_{i-1} \in \{\sigma_i\}$ where where $\{\sigma_i\}$ denotes the set of such multipliers), then this is the solution to (4.8) or equivalently (5.4) with χ_{i-1} component given by χ_{i-1}^k. Now if χ_{i-1}^k was used in the quantization for (5.3), then we see the equivalence of (5.3) and (5.4) when we work with unrestricted subproblems and identify $\tilde{\psi}_{i-1}(\chi_{i-1}^k)$ with $\psi_{i-1}(\chi_{i-1}^k)$ and p_{ij} with $\psi_i(\chi_i^j)$. Note also the relationship of (4.9) to the objective function of (5.3). We see that in the dynamic programming interpretation, if we use a price-directive method for *dynamically choosing the quantization*, we obtain a procedure that is conceptually quite close to multistage decomposition by tenders. We however emphasize this correspondence only at a conceptual level, because in practical terms the two schemes differ substantially. In practice, to do a backward recursion in the standard way would be extremely expensive. Instead we work with *restricted subproblems* and develop a sequence of approximations $\tilde{\psi}_i(\chi_i)$ which are refined in a cycle through the subproblems, with prices π_1, obtained from the first subproblem seeking to match σ_2 and producing an associated χ_1^j, then π_2 from this solution seeking to match σ_3, and so on.

All of this can be brought a little more sharply into focus by looking at decomposition by tenders from yet another viewpoint, namely the application of Wolfe's generalized programming method (see, Dantzig, 1963, Chapter 24) iteratively to (5.1). The method produces successive inner linearizations of $\psi_i(\chi_i)$. If we attach the symbol 'tilde' to denote approximations i.e. $\tilde{\psi}_i(\chi_i)$ denotes an approximation to $\psi_i(\chi_i)$, then inner linearizing $\tilde{\psi}_1(\chi_1)$ we obtain:

$$
\begin{aligned}
\text{minimize} \quad & c_1 x_1 + \sum_j \tilde{\psi}_1(\chi_1^j)\lambda_{1j} \\
& A_1 x_1 = b_1 \\
& B_1 x_1 - \sum_j \chi_1^j \lambda_{1j} = 0 \qquad (5.8) \\
& \sum_j \delta_{1j}\lambda_{1j} = 1 \\
& x_1 \,,\ \lambda_{1j} \geq 0
\end{aligned}
$$

where χ_1^j represents a corner point or line of recession of $\breve{\psi}_1(\chi_1)$ and δ_{1j} has the usual meaning.

If $(\sigma_1, \pi_1, \rho_1)$ are the dual multipliers for (5.8) at optimality, then an improving candidate is obtained by solving

$$\text{(5.9a)} \qquad \text{minimize} \quad [\breve{\psi}_1(\chi_1) + \pi_1\chi_1]$$

From (5.1b) we see that (5.9a) is equivalent to solving

$$\begin{aligned} \text{minimize} \quad & \pi_1\chi_1 + c_2x_2 + \breve{\psi}_2(\chi_2) \\ & \chi_1 + A_2x_2 = b_2 \\ & B_2x_2 - \chi_2 = 0 \\ & x_2 \geq 0 \end{aligned} \qquad \text{(5.9b)}$$

$\breve{\psi}_2(\chi_2)$ is, in turn, inner linearized (5.9b) becomes

$$\begin{aligned} \text{minimize} \quad & \pi_1\chi_1 + c_2x_2 + \sum_j \breve{\psi}_2(\chi_2^j)\lambda_{2j} \\ & \chi_1 + A_2x_2 = b_2 \\ & B_2x_2 + \sum_j \chi_2^j\lambda_{2j} = 0 \\ & \sum_j \delta_{2j}\lambda_{2j} = 1 \\ & x_2 \,,\ \lambda_{2j} \geq 0 \end{aligned} \qquad \text{(5.10)}$$

and so on. A general iteration, with appropriate end conditions is therefore given as follows:

$$
\begin{aligned}
\text{minimize} \quad & \pi_{i-1}\chi_{i-1} + c_i x_i + \sum_j \tilde{\psi}_i(\chi_i^j)\lambda_{ij} \\
& \chi_{i-1} + A_i x_i = b_i \\
& B_i x_i + \sum_j \chi_i^j \lambda_{ij} = 0 \qquad (5.11) \\
& \sum_j \delta_{ij}\lambda_{ij} = 1 \\
& x_i \ , \ \lambda_{ij} \geq 0
\end{aligned}
$$

and we can immediately compare with (4.8). An effective implementation would require one to develop approximations that steer the overall process as quickly as possible to the neighborhood of an optimal solution. A pure backward iteration, which seeks to develop $\psi_i(\chi_i)$ rather than $\tilde{\psi}_i(\chi_i)$ would be wasteful as compared to a cyclic iteration. To make an analogy, suppose we were to consider the case of shortest path algorithms for directed networks whose nodes say S_{ij} , $j = 1,2,\ldots,k_i$ are defined at discrete time intervals, say $i = 0,1,2,\ldots,T$; then the above comparison, provided all costs were non-negative would not be dissimilar to a comparison between pure backward iteration and Dijkstra's labelling algorithm. Indeed, this analogy is worth exploring as a means for investigating the efficiency of nested decomposition algorithms but we will not pursue this any further in this paper.

The whole question of implementation requires much more extensive study, which we are currently undertaking. We make some further comments in the next section.

6. CONCLUDING COMMENTS, IN PARTICULAR, CONCERNING IMPLEMENTATION

Here we have sought to introduce a *family of algorithms* based upon applying Dantzig-Wolfe decomposition to transformed problems. In particular, decomposition by tenders looks quite promising for multistage problems. It can take advantage of a good set of initial allocations of the right-hand-side to stages defined by $\begin{pmatrix} A_i \\ B_i \end{pmatrix}$ which the particular LP model may have

available, and it circumvents the problem of having to reconstruct the solution, as required by the Ho-Manne algorithm. There are, of course, a number of different algorithms for multistage decomposition by tenders, determined in particular, by whether the primal or dual LP is solved, and whether this is done forward or backward (in time).

As far as implementation is concerned, an experimental version at level-1 could be based on MPL, a level-2 implementation could be developed using the subroutines described in Nazareth, 1982 and a level-3 implementation could be developed by suitably modifying the code of Ament et al, 1980. (For terminology on hierarchical implementation at different levels see Nazareth, 1982.) Also, as mentioned earlier, the algorithm of Nurminski, 1982 can be shown to be an instance of decomposition by tenders, and the experimental evidence accrued by him using the MINOS code, also looks promising.

Finally, we should mention another promising avenue of exploration, namely, the application to solving multistage stochastic programs.

Acknowledgment

It is a pleasure to thank Roger Wets for his helpful comments on various drafts of this paper.

REFERENCES

1. Abrahamson, P. (1981), A nested decomposition approach for solving staircase structured linear programs, In *Large Scale Linear Programs*, G.B. Dantzig et al (Eds.), IIASA Collaborative Proceedings Series, CP-81-S1, Laxenburg, Austria, 367-382.

2. Ament, D., J. Ho, E. Loute and M. Remmelswaal (1980), LIFT: a nested decomposition algorithm for solving lower block triangular linear programs, In *Large Scale Linear Programming*, G.B. Dantzig et al (Eds.), IIASA Collaborative Proceedings Series, CP-81-S1, Laxenburg, Austria, 383-408.

3. Benders, J.F. (1962), Partitioning procedures for solving mixed variables programming problems, *Numerische Mathematik*, 4, 238-252.

4. Birge, J.R. (1980), Solution methods for stochastic dynamic linear programs, Report SOL 80-29, Systems Optimization Laboratory, Department of Operations Research, Stanford University.

5. Dantzig, G.B. and P. Wolfe (1960), Decomposition Principle for linear programs, *Operations Research*, 8, 101-111.

6. Dantzig, G.B. (1963), *Linear Programming and Extensions*, Princeton University Press.

7. Glassey, C.R. (1971), Dynamic LP's for production scheduling, *Operations Research*, 19, 45-56.

8. Ho, J. and A. Manne (1974), Nested decomposition for dynamic models, *Mathematical Programming*, 6, 121-140.

9. Ho, J. (1974), Nested decomposition for large scale linear programs with the staircase structure, Report SOL 74-4, Systems Optimization Laboratory, Department of Operations Research, Stanford University.

10. Kallio, M.J. (1975), On large scale linear programming, Report SOL 75-7, Systems Optimization Laboratory, Department of Operations Research, Stanford University.

11. Lasdon, L. (1970), *Optimization Theory for Large Systems*, MacMillan.

12. Nazareth, L. (1978), Dually equivalent decomposition algorithms with application to solving staircase structures, Report SOL 78-30, Systems Optimization Laboratory, Department of Operations Research, Stanford University.

13. Nazareth, L. (1980), A land management model using Dantzig-Wolfe decomposition, *Management Science*, 26, 510-523.

14. Nazareth, L. (1982), Implementation aids for optimization algorithms that solve sequences of linear programs by the revised simplex method, WP-82-107, International Institute for Applied Systems Analysis, Laxenburg, Austria.

15. Nazareth, L. and R. Wets (1983), Algorithms for stochastic programs: the case of nonstochastic tenders, Report WP-83-5, International Institute for Applied Systems Analysis, Laxenburg, Austria.

16. Nurminski, E. (1982), Convergence and numerical experiments with a decomposition algorithm, WP-82-8, International Institute for Applied Systems Analysis, Laxenburg, Austria.

17. Schechtman, J. and S. Granville (1982), A market equilibrium approach to solving large scale linear programming models, Instuto de Matematica Pura e Aplicada, Rio de Janeiro, Brazil.

18. Shapiro, J.F. (1979), *Mathematical Programming: Structures and Algorithms*, Wiley.

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

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

21. Wets, R. (1974b), Stochastic Programming, (unpublished notes).

22. Wets, R. (1983), Solving stochastic programs with simple recourse, *Stochastics*, to appear.