AD-A066 441 STANFORD UNIV CALIF SYSTEMS OPTIMIZATION LAB F/G 12/1
ERROR PROPAGATION AND SOLUTION RECONSTRUCTION IN NESTED DECOMPO--ETC(U)
DEC 78 L NAZARETH N00014-75-C-0267
UNCLASSIFIED SOL-78-27 NL

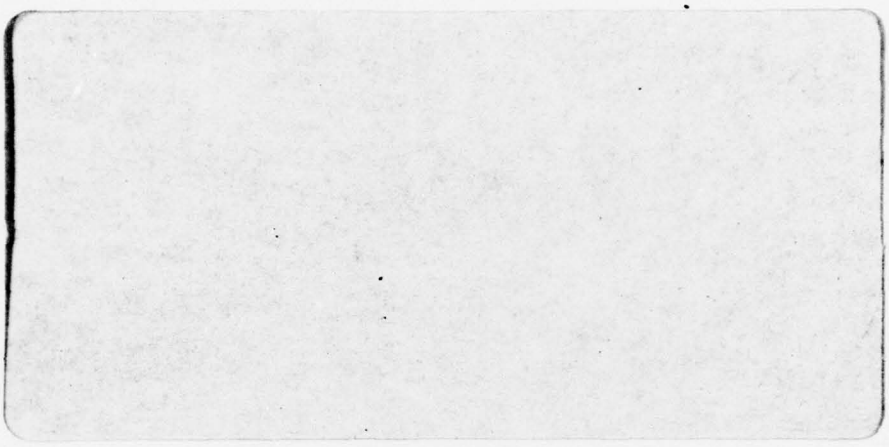
| OF |
AD A066441

END
DATE FILMED
5-79
DDC

LEVEL II

12

Systems
Optimization
Laboratory



Department of Operations Research
Stanford University
Stanford, CA 94305

LEVEL II

12

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH
Stanford University
Stanford, California
94305

6 ERROR PROPAGATION AND SOLUTION RECONSTRUCTION IN NESTED DECOMPOSITION.

by

10 L. Nazareth

9 TECHNICAL REPORT, SOL-78-27
December 1978

11 Dec 78

12 34p.

15 Research and reproduction of this report were partially supported by the Office of Naval Research Contract N00014-75-C-0267; the National Science Foundation Grants MCS76-20019 A01 and ENG77-06761 A01; and the Department of Energy Contract EY-76-S-03-0326 PA #18.

408 765

LB

Abstract

We study some of the numerical properties of the nested decomposition algorithm of Ho and Manne. In particular we seek to show how well developed theory in the area of computational linear algebra, due primarily to J.H. Wilkinson, carries over to linear programming and yields useful insight into the behavior of algorithms in this area.

Acknowledgment

My grateful appreciation to Professor G.B. Dantzig and the many others who helped make my visit to the Systems Optimization Laboratory an interesting and valuable learning experience.

My thanks also to the Applied Mathematics Division, Argonne National Laboratory, who jointly with the Systems Optimization Laboratory supported my appointment at Stanford University.

Special thanks to M. Aganagic and J. Ho for their help during the course of this research.

ERROR PROPAGATION AND SOLUTION RECONSTRUCTION IN NESTED DECOMPOSITION

by

L. Nazareth

1. Introduction

The idea of a nested decomposition algorithm dates back to the fundamental paper of Dantzig and Wolfe [1960]. Many authors have since contributed to its development including Dantzig [1963, Ch. 23], Cobb and Cord [1967], Glassey [1973], Ho and Manne [1974], Ho [1974a, 1974b], Kallio [1975]. The major credit for the development of this algorithm goes to Ho [1974a,b], who put it on a sound algorithmic footing, implemented the algorithm and showed it to be a workable technique for solving LP problems with staircase structire, of the form:

$$\begin{aligned} \text{minimize} \quad & \sum_{t=1}^{T} \underline{c}_t \underline{x}_t \\ \text{subject to} \quad & A_1 \underline{x}_1 = \underline{d}_1 \\ & B_{t-1} \underline{x}_{t-1} + A_t \underline{x}_t = \underline{d}_t \\ & \underline{x}_t \geq 0 \qquad \text{for} \quad t = 1,\ldots,T \end{aligned} \tag{1.1}$$

where $\underline{x}_t$ is $n_t \times 1$, A_t is $m_t \times n_t$, $B_t = m_{t+1} \times n_t$, $\underline{c}_t$ is $1 \times n_t$, and $\underline{d}_t$ is $m_t \times 1$ in dimension. We shall call the above problem LP.

In this paper we look at Ho's algorithm from a numerical standpoint. This paper is organized into five sections as follows:

In Section 2 we provide background by briefly describing the algorithm and relating practical experience obtained by M. Aganagic* [1977] who extended the code of Ho and ran it on a version of the PILOT model of Dantzig and Parikh [1975]. It is this experimentation which motivated the research described here.

In Section 3 we give examples which illustrate numerical difficulties which can arise, and we discuss certain numerical properties of this algorithm.

In Section 4 we use some of the results of backward error analysis and perturbation theory of computational linear algebra, Wilkinson [1965], in order to study the propagation of error from stage to stage.

In Section 5 we examine further the reconstruction of the solution in nested decomposition, in the presence of numerical error. We propose an alternative method for doing this.

Finally in Section 6 we make some recommendations about the implementation of nested decomposition and describe some lessons we have learned from our research, about the development of structured LP algorithms and codes.

Our main aim in this paper is to look at LP algorithms based upon the decomposition principle, from a numerical standpoint and to show that such algorithms have some very interesting numerical properties. A complete investigation is, of course, well beyond the scope of this paper.

---

*Ph.D. student, Operations Research Department, Stanford University, California.

2. Background

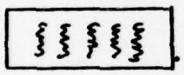
2.1. Brief description of algorithm

We assume familiarity with the nested decomposition method for solving systems of the form (1.1) as described in Ho [1974a]. We summarize this by the schemata shown in Fig. 1, for a four-stage problem.

Notes on Figure 1.

(a) We distinguish extreme points and corresponding proposals by superscripts, to emphasize that we are dealing with vectors.

(b) The matrix of generated proposals is denoted by Q_j. In order to emphasize the distinction between a master problem and the corresponding restricted master problem we append the symbol



(c) The convexity row is denoted by .

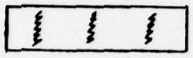
(d) The objective row for a subproblem comes from the pricing out of proposals generated by the subproblem, as explained in the right-hand column.

2.2. Reconstruction of the solution

This has again been described in detail in Ho [1974a]. He gives two methods. In the first method (Method I), all extreme points developed for each subproblem are kept. Let $(\underline{\lambda}_4^*, \underline{x}_4^*)$ be the optimal solution of S/P 4. In Fig. 1,

$$\begin{pmatrix} \underline{\lambda}_3^* \\ \underline{x}_3^* \end{pmatrix} = \sum_j \begin{pmatrix} \underline{\lambda}_3^j \\ \underline{x}_3^j \end{pmatrix} \lambda_{4j}^*, \quad \begin{pmatrix} \underline{\lambda}_2^* \\ \underline{x}_2^* \end{pmatrix} = \sum_j \begin{pmatrix} \underline{\lambda}_2^j \\ \underline{x}_2^j \end{pmatrix} \lambda_{3j}^* \quad \text{and} \quad \underline{x}_1^* = \sum_j \underline{x}_1^j \lambda_{2j}^* .$$

In the second method (Method II), it is unnecessary to keep all extreme points. Instead they can be regenerated, if necessary, in the following reconstruction algorithm. Let us again consider the four-stage problem. Fix $\underline{x}_4$ at $\underline{x}_4^*$ and treat the problem as a three-stage decomposition in variables $\underline{x}_1$, $\underline{x}_2$, and $\underline{x}_3$. Following Figure 1, the last stage would then be of the form



$$\left.\begin{array}{lll} Q_3 & A_3 = \underline{b}_3 & \\ & \sim\sim\sim = 1. & \\ & B_3 = \underline{b}_4 - A_4\underline{x}_4^* & \end{array}\right\} \text{S/P 3}$$

$$\left.\begin{array}{ll} \lambda_3 & x_3 \end{array}\right\} \text{Variables}$$

and the first and second stage would be as in Fig. 1. When this problem is solved $\underline{x}_3$ will be known, and let us say its optimal value is $\underline{x}_3^*$. We can now fix $\underline{x}_3$ and $\underline{x}_4$ at their optimal values, i.e. the last set of equations effectively drops out. We now solve the first three sets of equations as a two-stage decomposition in variables $\underline{x}_1$ and $\underline{x}_2$. This process is continued until the optimal value of all variables is known.

Note: Reconstructed solutions are not, in general, basic solutions for the original LP.

FIGURE 1. NESTED DECOMPOSITION SCHEMATA

	$A_1 \quad = \underline{b}_1$ $B_1 \; A_2 \quad = \underline{b}_2$ $B_2 \; A_3 \quad = \underline{b}_3$ } MASTER 1 ROWS $B_3 \; A_4 = \underline{b}_4$ $\underline{c}_1 \; \underline{c}_2 \; \underline{c}_3 \; \underline{c}_4$ } COST ROW $\underline{x}_1 \; \underline{x}_2 \; \underline{x}_3 \; \underline{x}_4$ } VARIABLES	Initial LP
$\underline{x}_1^j$ and $\lambda_2 = \sum_j \underline{x}_1^j \lambda_{2j}$ $\underline{q}_2^j = B_1 \underline{x}_1^j$ $p_{2j} = \underline{c}_1 \underline{x}_1^j$	$A_1 \quad = \underline{b}_1$ } S/P 1 $Q_2 \; A_2 \quad = \underline{b}_2$ $\quad \quad = 1$ } MASTER 2 $B_2 \; A_3 \quad = \underline{b}_3$ $B_3 \; A_4 = \underline{b}_4$ $p_2 \; \underline{c}_2 \; \underline{c}_3 \; \underline{c}_4$ } COST ROW $\lambda_2 \; \underline{x}_2 \; \underline{x}_3 \; \underline{x}_4$ } VARIABLES	As below with S/P 3 replaced by S/P 2 and S/P 2 replaced by S/P 1.
$\begin{pmatrix} \lambda_2^j \\ \underline{x}_2^j \end{pmatrix}$ and $\begin{pmatrix} \lambda_2 \\ \underline{x}_2 \end{pmatrix} = \sum_j \begin{pmatrix} \lambda_2^j \\ \underline{x}_2^j \end{pmatrix} \lambda_{3j}$ $\underline{q}_3^j = (0 \mid B_2) \begin{pmatrix} \lambda_2^j \\ \underline{x}_2^j \end{pmatrix}$ $p_{3j} = (p_2, \underline{c}_2)^T \begin{pmatrix} \lambda_2^j \\ \underline{x}_2^j \end{pmatrix}$	$Q_2 \; A_2 \quad = \underline{b}_2$ $\quad \quad = 1$ } S/P 2 $Q_3 \; A_3 \quad = \underline{b}_3$ $\quad \quad = 1$ } MASTER 3 ROWS $B_3 \; A_4 = \underline{b}_4$ $p_3 \; \underline{c}_3 \; \underline{c}_4$ } COST ROW $\lambda_3 \; \underline{x}_3 \; \underline{x}_4$ } VARIABLES	When optimizing S/P 3 the pricing out of λ_3 variables defines the objective row of S/P 2 as $\min_j (p_{3j} - \pi_3 q_3^j)$ $= \min_j p_2^T \lambda_2^j + (c_2 - \pi_3 B_2)^T \underline{x}_2^j$
$\begin{pmatrix} \lambda_3^j \\ \underline{x}_3^j \end{pmatrix}$ and $\begin{pmatrix} \lambda_3 \\ \underline{x}_3 \end{pmatrix} = \sum_j \begin{pmatrix} \lambda_3^j \\ \underline{x}_3^j \end{pmatrix} \lambda_{4j}$ $\underline{q}_4^j = (0 \mid B_3) \begin{pmatrix} \lambda_3^j \\ \underline{x}_3^j \end{pmatrix}$ $p_{4j} = (p_3, \underline{c}_3)^T \begin{pmatrix} \lambda_3^j \\ \underline{x}_3^j \end{pmatrix}$	$Q_3 \; A_3 \quad = \underline{b}_3$ } S/P 3 $Q_4 \; A_4 = \underline{b}_4$ } MASTER 4 = S/P 4 $\quad \quad = 1$ $p_4 \; \underline{c}_4$ } COST ROW $\lambda_4 \; \underline{x}_4$ } VARIABLES	Pricing out λ_4 variables gives the objective row for S/P 3 as $\min_j (p_{4j} - \pi_4 q_4^j)$ $= \min_j p_3^T \lambda_3^j + (c_3 - \pi_4 B_3)^T \underline{x}_3^j$

2.2. Practical experience

By extending the code of Ho and running it on a version of the PILOT Model, of Dantzig and Parikh [1975], M. Aganagic [1977] obtained a great deal of valuable experience with the algorithm. He found

- that master/subproblems could become badly scaled
- that in certain cases, propagation of error from stage to stage was substantial
- that the technique for reconstructing the solution could fail.

However the code used was an experimental one, and the LP problem itself had coefficients which differed widely in magnitude. It is therefore unclear whether these difficulties are inherent in the nested decomposition algorithm, or whether some of them are the result of numerical instabilities in the code (associated with use of the product form of the inverse) or inherent in the LP problem being solved. Hence the analysis described here.

3. Examples and Discusion

We now discuss each of the difficulties mentioned in Section 2.2, mainly through numerical examples.

3.1. Scaling.

An LP problem which is well scaled for the simplex algorithm, can be badly scaled for the decomposition algorithm. Consider the problem

$$\text{minimize} \quad x + y$$

$$\left.\begin{array}{rl} \text{subject to} & x + \varepsilon y \geq 0 \\ & x - \varepsilon y \geq 0 \\ & x - \varepsilon y \leq 1 \\ & x + \varepsilon y \leq 1 \end{array}\right\} \text{S/P}$$

$$\left.\begin{array}{r} y \leq 1 \\ -y \leq 1 \end{array}\right\} \text{Master}$$

where ε is small.

If the simplex algorithm is applied to this problem, every basis is well-conditioned (i.e has a reasonable condition number), and small variations in the coefficients will produce small variations in the extreme points of the problem.

Suppose however we decompose this LP into a master and subproblem as indicated above. The extreme points of the subproblem are

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1/2 \\ 1/2\varepsilon \end{pmatrix}, \begin{pmatrix} 1/2 \\ -1/2\varepsilon \end{pmatrix}$$

and the master problem becomes

$$\begin{aligned} (1/2\varepsilon)\lambda_3 + (-1/2\varepsilon)\lambda_4 + s_1 &= 1 \\ (-1/2\varepsilon)\lambda_3 + (1/2\varepsilon)\lambda_4 + s_2 &= 1 \\ \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 &= 1 \\ \lambda_i \geq 0, \quad s_1, s_2 \geq 0 & \end{aligned}$$

This problem has basis matrices with a large condition number $\approx O(1/\varepsilon)$ and the structural (non-slack) columns differ widely in magnitudes, so that their reduced costs can give misleading information about the value of introducing a particular column into the basis. In this sense, see Tomlin [1975], the master is badly scaled.

3.2. Error propagation

Consider the two-stage problem

$$\begin{aligned} A_1\underline{x}_1 &= \underline{b}_1 \quad \} \text{ S/P rows} \\ B_1\underline{x}_1 + A_2\underline{x}_2 &= \underline{b}_2 \quad \} \text{ Master rows} \\ \underline{x}_1, \underline{x}_2 &\geq 0 \end{aligned} \qquad (3.2a)$$

Let $\underline{x}_1^j$ be the true extreme points of the S/P and $\tilde{\underline{x}}_1^j$ the computed extreme points. If a stable algorithm is used, then by the backward error analysis of Wilkinson [1965], each $\tilde{\underline{x}}_1^j$ is the exact solution of a perturbed system

$$(A_1^j + E^j)\tilde{\underline{x}}_1^j = \underline{b}_1 \tag{3.2b}$$

where A_1^j is the basis matrix corresponding to $\tilde{\underline{x}}_1^j$ and E^j is a small perturbation. For example, if we assume that all elements of A_1^j satisfy $|(A_1^j)_{\ell m}| \leq 1$, that partial pivoting in the LU factorization of A_1^j is used and that $|U_{\ell m}| \leq 1$, then $\|E^j\|_\infty \leq 3 \cdot n \cdot 2^{-t_1}$, when t_1 is the number of bits in a floating point word (see Wilkinson [1965]).

Note that $\underline{x}_1^j$ can be extremely different from $\underline{x}_1^j$, this being determined by the condition number of A_1^j. We can show by perturbation theory, Wilkinson [1965], that

$$\frac{\|\underline{x}_1^j - \tilde{\underline{x}}_1^j\|}{\|\underline{x}_1^j\|} \leq \frac{k_j(\|E^j\|/\|A_1^j\|)}{(1 - k_j\|E^j\|/\|A_1^j\|)} \quad \text{where} \quad k_j = \|A_1^j\|\|(A_1^j)^{-1}\|$$

If the exact solution to (3.2a) is $\underline{x}_t = \sum_j \lambda_j \underline{x}_1^j$, $\sum_j \lambda_j = 1$, and the computed solution $\underline{x}_c = \sum_j \tilde{\lambda}_j \tilde{\underline{x}}_1^j$, $\sum_j \tilde{\lambda}_j = 1$, then

$$\|\underline{x}_c - \underline{x}_t\| = \|\sum \tilde{\lambda}_j (A_1^j + E^j)^{-1}\underline{b}_1 - \sum \lambda_j (A_1^j)^{-1}\underline{b}_1\|$$

assuming $(A_1^j + E^j)$ and A_1^j are invertible,

$$= \|\sum \tilde{\lambda}_j (I + A_1^{j^{-1}} E^j)^{-1}(A_1^j)^{-1}\underline{b}_1 - \sum \lambda_j (A_1^j)^{-1}\underline{b}_1\|$$

Now, for any square matrix X, $(I + X)^{-1} = (I - X + X^2 - X^3 + \cdots)$, provided $\|X\| < 1$. If we assume, therefore, that $\|(A_1^j)^{-1}E^j\| < 1$

$$\|\underline{x}_c - \underline{x}_t\| = \|\sum(\tilde{\lambda}_j - \lambda_j)A_1^{j^{-1}}\underline{b}_1 + \sum \tilde{\lambda}_j(-(A_1^j)^{-1}E_j)(I-(A_1^j)^{-1}E^j+\cdots)(A_1^j)^{-1}\underline{b}_1\|$$

$$\leq \|\sum_j(\tilde{\lambda}_j - \lambda_j)(A_1^j)^{-1}\underline{b}_1\| + \|\sum_j \tilde{\lambda}_j((A_1^j)^{-1}E^j)(I + (A_1^j)^{-1}E^j)^{-1}(A_1^j)^{-1}\underline{b}_1\|$$

Assume further that $\tilde{\lambda}_j \neq 0 \iff \lambda_j \neq 0$.

$$\|\underline{x}_c-\underline{x}_t\| \leq \| \sum_j\left(\frac{\tilde{\lambda}_j}{\lambda_j} - 1\right)\lambda_j A_1^{j^{-1}}\underline{b}_1\| + \| \sum_j(\lambda_j(A_1^j)^{-1}\underline{b}_1)\left(\frac{\tilde{\lambda}_j}{\lambda_j}\right)(A_1^j)^{-1}E^j(I+(A_1^j)^{-1}E^j)^{-1}\|$$

$$\frac{\|x_c-x_t\|}{\|x_t\|} \leq \sum_j\left|\left(\frac{\tilde{\lambda}_j}{\lambda_j} - 1\right)\right| + \frac{\sum_j|\tilde{\lambda}_j/\lambda_j|\ \|A_1^j\|\|(A_1^j)^{-1}\|(\|E^j\|/\|A_1^j\|)}{\sum_j(1 - \|A_1^j\|\|(A_1^j)^{-1}\|(\|E^j\|/\|A_1^j\|)}$$

This bound is not an encouraging one, if $\|A_1^j\|\|(A_1^j)^{-1}\|$ is large or $\tilde{\lambda}_j$ differs substantially from λ_j.

Let us however now adopt the viewpoint of backward error analysis. Ignoring, for the moment, errors in optimizing the master (see Section 4) then we are, in effect, <u>exactly</u> solving a system of the form (3.2b), in which each element of A_1 has a small uncertainty with a known bound.

<u>NOTE</u> that since E^j varies with A_1^j, we cannot have A_1 perturbed by a fixed matrix, i.e. by a <u>static</u> perturbation. We shall use the term <u>dynamic perturbation</u> to distinguish our case.

Now, if we assume that the solution of the LP is stable, then we would expect $\|\underline{x}_c - \underline{x}_t\|$ to be small. This implies a correlation of error, which the following example exhibits:

$$
\begin{array}{lrcl}
\text{minimize} & & x_4 & \\
\text{subject to} & x_1 + (1+\varepsilon_1)x_3 & = a_1 & \left.\begin{array}{l}\\ \end{array}\right\} \text{S/P} \\
& x_1 + x_2 + (1+\varepsilon_2)x_3 & = a_2 & \\
& x_3 + x_4 & = a_3 & \left.\begin{array}{l}\\ \end{array}\right\} \text{Master} \\
& x_4 & = a_4 & \\
& x_i \geq 0 & &
\end{array}
\qquad (3.2c)
$$

There are only two feasible extreme points of S/P, and these are

$$
x_1^1 = \begin{pmatrix} a_1 \\ a_2 - a_1 \\ 0 \end{pmatrix} \quad \text{and} \quad x_1^2 = \begin{pmatrix} a_1 - (1+\varepsilon_1)k \\ 0 \\ k \end{pmatrix}
$$

when $k = (a_2 - a_1)/(\varepsilon_2 - \varepsilon_1)$ and where we assume that ε_1 and ε_2 are small, $\varepsilon_2 > \varepsilon_1$ and $a_2 > a_1 > a_2(1 + \varepsilon_1)/(1 + \varepsilon_2)$. The master problem is:

$$
\begin{array}{lrcl}
\text{minimize} & & x_4 & \\
\text{subject to} & \left(\dfrac{a_2 - a_1}{\varepsilon_2 - \varepsilon_1}\right)\lambda_2 + x_4 & = a_3 & \\
& & x_4 & = a_4 \\
& \lambda_1 + \lambda_2 & = 1 & \\
& \lambda_1, \lambda_2, x_4 \geq 0 & &
\end{array}
$$

This has the solution

$$x_4 = a_4$$

$$\lambda_2 = (a_3 - a_4)/k$$

$$\lambda_1 = 1 - (a_3 - a_4)/k$$

where we assume $a_3 > a_4 > 0$ and chosen so that $\lambda_1, \lambda_2 > 0$. The optimal solution $\underline{x}_{opt}$ is given by $\lambda_1 \underline{x}_1^1 + \lambda_2 \underline{x}_1^2$, which becomes

$$\underline{x}_{opt} = \begin{bmatrix} a_1 - (1+\varepsilon_1)(a_3-a_4) \\ (a_2-a_1) - (a_3-a_4)(\varepsilon_2-\varepsilon_1) \\ a_3-a_4 \\ a_4 \end{bmatrix}$$

Solving the system (3.2c) as an LP also yields the unique (optimal) solution and the associated 4×4 basis is <u>well conditioned</u>. This implies that x_{opt} is relatively insensitive to small changes in the matrix or r.h.s. elements. Suppose that $a_1, a_2, \varepsilon_1, \varepsilon_2$ are obtained by truncating $\hat{a}_1, \hat{a}_2, \hat{\varepsilon}_1$ and $\hat{\varepsilon}_2$ which are not machine representable numbers. Suppose also that the above derivation is carried out with $\hat{a}_1, \hat{a}_2, \hat{\varepsilon}_1$ and $\hat{\varepsilon}_2$ used in place of a_1, a_2, ε_1, and ε_2, and denote the quantities corresponding to k, λ_1 and λ_2 by $\hat{k}, \hat{\lambda}_1$ and $\hat{\lambda}_2$. Then the quantity k which determines $\underline{x}_1^2$ can be drastically different from $\hat{k} = (\hat{a}_2-\hat{a}_1)/(\hat{\varepsilon}_2-\hat{\varepsilon}_1)$. This is because the basis which determines $\underline{x}_1^2$ is ill-conditioned. The corresponding quantities $\hat{\lambda}_1$ and $\hat{\lambda}_2$ will also be quite different from λ_1 and λ_2. However the errors in λ_1 and λ_2 are <u>correlated</u> with the errors in $\underline{x}_1^1$ and $\underline{x}_1^2$, so that $\underline{x}_{opt}$ does not change drastically.

3.3. Reconstruction

Finally we give an example which illustrates difficulties associated with reconstruction of the solution. We wish to show that Method 2 of Section 2.2 can result in numerical instability.

Consider the problem:

$$\begin{aligned} &\text{maximize} \quad x_4 \\ &\text{subject to} \quad \left.\begin{aligned} x_1 \quad + (1+\epsilon_1)x_3 \quad &= b_1 \\ x_2 + (1+\epsilon_2)x_3 \quad &= b_2 \end{aligned}\right\} \text{S/P} \\ &\qquad\quad \left.\begin{aligned} x_1 + x_2 + 2(1+\epsilon_3)x_3 + x_4 &= b_3 \end{aligned}\right\} \text{Master} \\ &\qquad\quad x_i \geq 0 \end{aligned} \tag{3.3a}$$

There are only two feasible extreme points and these are given by

$$\underline{x}_1^1 = \begin{pmatrix} b_1 \\ b_2 \\ 0 \end{pmatrix}, \quad \underline{x}_1^2 = \begin{pmatrix} b_1 - b_2(1+\epsilon_1)/(1+\epsilon_2) \\ 0 \\ b_2/(1+\epsilon_2) \end{pmatrix} \tag{3.3b}$$

where we assume that b_1 and b_2 are chosen so that $\underline{x}_1^1, \underline{x}_1^2 \geq 0$. All bases corresponding to these are well-conditioned. The master problem is given by

$$\text{maximize} \qquad x_4$$

$$\text{subject to} \quad (b_1 + b_2)\lambda_1 + \left[b_1 - \frac{b_2(1+\epsilon_1)}{(1+\epsilon_2)} + \frac{2(1+\epsilon_3)b_2}{(1+\epsilon_2)} \right] \lambda_2 + x_4 = b_3 \qquad (3.3c)$$

$$\lambda_1 + \qquad\qquad \lambda_2 = 1$$

$$\lambda_1, \lambda_2, x_4 \geq 0$$

Assume $(2\epsilon_3 - \epsilon_1) < \epsilon_2$. Then the coefficient of λ_2 < coefficient of λ_1 and so the optimal solution is given by $\lambda_1 = 0$, $\lambda_2 = 1$ and x_4 determined by (3.3c). Then the optimal solution, by Method I of Section 2.2,

$$\underline{x}_{opt} = \begin{bmatrix} b_1 - b_2(1+\epsilon_1)/(1+\epsilon_2) \\ 0 \\ b_2/(1+\epsilon_2) \\ b_3 - \left(b_1 + \dfrac{b_2(1 + 2\epsilon_3 - \epsilon_1)}{(1 + \epsilon_2)} \right) \end{bmatrix}$$

(The basis corresponding to variables x_1, x_3 and x_4 in the original LP is <u>well-conditioned</u> and the solution x_{opt} can be alternatively obtained from this basis.) If Method II (cf. Section 2.2) is however used, we must then solve the system

$$\begin{bmatrix} 1 & 0 & 1+\epsilon_1 \\ 0 & 1 & 1+\epsilon_2 \\ 1 & 1 & 2(1+\epsilon_3) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_1 + \dfrac{(1 + 2(\epsilon_3-\epsilon_1))b_2}{(1 + \epsilon_2)} \end{bmatrix}$$

This is an ill-conditioned problem, and thus numerical error could result in the reconstructed solution being substantially different from the true well-conditioned optimal solution, even possibly infeasible. We return to the problem of reconstruction in Section 5.

4. Analysis of Error Propagation

In this section we use some of the results in Wilkinson [1965], Chap. 4 on backward error analysis and perturbation theory of linear systems of equations, in order to study the propagation of error from stage to stage. A complete error analysis is beyond the scope of this paper.

4.1. In backward error analysis one seeks to show that the computed solution x_c of a problem, say P, is the exact solution of a problem obtained by perturbing P. In the nested decomposition algorithm, a typical restricted subproblem is of the form (see Fig. 1)

$$\left.\begin{array}{rl} m_t \text{ rows:} & Q_t\underline{\lambda}_t + A_t\underline{x}_t = \underline{b}_t \\ \text{convexity rows:} & \underline{e}_t^T\underline{\lambda}_t = 1 \\ & \underline{\lambda}_t,\ \underline{x}_t \geq 0 \end{array}\right\} S/P - t \qquad (4.1a)$$

where $Q_t = B_{t-1}X^c_{t-1}$, and X^c_{t-1} is the matrix whose columns are computed extreme points and rays of the previous subproblem $S/P - (t-1)$. $\underline{e}_t^T$ is a row vector whose j'th element is 1 if the corresponding column of X^c_{t-1} is an extreme point and 0 if it is an extreme ray. A typical basis of $S/P - t$ consists of a set of $m_t + 1$ columns, and is of the form

$$\mathcal{B} = \left[\begin{array}{c|c} B_{t-1}\ \hat{X}^c_{t-1} & \hat{A}_t \\ \hline \hat{\underline{e}}_t^T & 0 \end{array}\right] \qquad (4.1b)$$

where $\hat{X}^c_{t-1}$, $\hat{A}_t$ and $\hat{\underline{e}}^T_t$ denote columns of X^c_{t-1}, A_t and $\underline{e}^T_t$.

Computed extreme points and rays for the next subproblem S/P - (t + 1) are exact solutions of

$$(\mathcal{B} + \delta\mathcal{B})\underline{x}^c_t = \begin{pmatrix} \underline{b}_t \\ 1 \end{pmatrix} \tag{4.1c}$$

e.g. if (i) $|\mathcal{B}_{ij}| \leq 1$

(ii) partial pivoting in the LU factorization of $\mathcal{B}$ is used (4.1d)

(iii) $|U_{ij}| \leq 1$

Then $\|\delta\mathcal{B}\|_\infty \leq 3\,(m_t + 1)2^{-t_1}$, see Wilkinson [1965] and Section 3.2. Under what circumstances can we cast errors in $\mathcal{B}$ back into the data of the original problem and into the convexity row?

Let us partition $\delta\mathcal{B}$ in the same way that $\mathcal{B}$ is partitioned in (4.1b). Then

$$\delta\mathcal{B} = \left[\begin{array}{c|c} \delta\mathcal{B}_{11} & \delta\mathcal{B}_{12} \\ \hline \delta\mathcal{B}_{21} & \delta\mathcal{B}_{22} \end{array}\right] \tag{4.1e}$$

The errors $\delta\mathcal{B}_{12}$ can be attributed to $\hat{A}_t$, in (4.1b) and $[\delta\mathcal{B}_{21}|\delta\mathcal{B}_{22}]$ to the convexity row in (4.1b). In addition let us seek δB_{t-1} such that

$$(B_{t-1} + \delta B_{t-1})\hat{X}^c_{t-1} = B_{t-1}\hat{X}^c_{t-1} + \delta \mathcal{B}_{11}$$

or

$$\delta B_{t-1}\hat{X}^c_{t-1} = \delta \mathcal{B}_{11}$$

A strong assumption which ensures this is that every matrix $\hat{X}^c_{t-1}$ is a feasible basis has full column rank and that

$$\hat{X}^{c+}_{t-1} = (\hat{X}^{cT}_{t-1}\ \hat{X}^c_{t-1})^{-1}\ \hat{X}^c_{t-1} \quad \text{satisfies} \quad \|\hat{X}^{cT}_{t-1}\| \leq k\ ,$$

where Z^+ denotes the generalized inverse of Z. Then

$$\|\delta B_{t-1}\|_\infty \leq \|\delta \mathcal{B}_{11}\|_\infty \|\ \hat{X}^{c+}_{t-1}\|_\infty \leq 3k(m_t + 1)2^{-t}$$

when (4.1d) holds.

Note that δB_{t-1} varies with $\hat{X}^c_{t-1}$. We can however say that when the above assumptions hold, the computed solution is the exact solution obtained by nested decomposition, applied to an LP (1.1), whose matrices A_t have a dynamic perturbation bounded by $3(m_t + 1)2^{-t_1}$, and whose matrices B_t have a dynamic perturbation bounded by $3 \cdot k \cdot (m_{t+1} + 1)2^{-t_1}$. Further each convexity row in the nested decomposition has a dynamic perturbation bounded by $3 \cdot (m_t + 1) \cdot 2^{-t_1}$.

The above assumptions are very restrictive. Less restrictive assumptions would require $\delta \mathcal{B}_{11}$ to be cast into errors in the data of previous stages. Also in order to circumvent having to explicitly introduce error into the convexity row, we could force it to be satisfied, by using a method similar to that used in the GUB algorithm, Dantzig and Van Slyke [1967].

4.2. Perturbation theory complements backward error analysis, see Wilkinson [1965]. We study the following questions:

Define LP^1 by perturbing each matrix A_t by δA_t and each B_t by δB_t in (1.1), where δA_t or δB_t are small fixed (static) perturbations. If the matrices Q_t in the decomposition of LP are in consequence perturbed to Q_t^1 we ask: Will each column of Q_t^1 be close to the corresponding columns of Q_t?

We make the following strong assumptions:

Assumption 4.2: Every basis $\mathcal{B}_t$ of S/P - t (see (4.1a)) is well-conditioned and

$$\|\mathcal{B}_t\|\|\mathcal{B}_t^{-1}\| \leq c \tag{4.2c}$$

Assumption 4.3: The subproblems in the decomposition of LP^1 match those in the decomposition of LP in the following sense: each has the same number of columns; furthermore, if a basis $\mathcal{B}_t$ is feasible for SP - t in the nested decomposition of LP, the basis $\mathcal{B}_t'$ for the corresponding set of columns in SP^1 - t, in the decomposition of LP^1, is also feasible.

Given a basis $\mathcal{B}_t$ for subproblem t of LP, let $\mathcal{B}_t' \triangleq \mathcal{B}_t + \delta\mathcal{B}_t$ for some $\delta\mathcal{B}_t$ be the corresponding basis for the subproblem of LP^1. Let the two feasible solutions be x_t and $x_t + \delta x_t$, respectively.

Thus

$$\mathcal{B}_t x_t = b_t$$

$$(\mathcal{B}_t + \delta\mathcal{B}_t)(x_t + \delta x_t) = b_t$$

As in Section 3.2,

$$\frac{\|\delta x_t\|}{\|x_t\|} \leq \frac{c\|\delta \mathcal{B}_t\|/\|\mathcal{B}_t\|}{(1 - c\|\delta \mathcal{B}_t\|/\|\mathcal{B}_t\|)} \tag{4.2b}$$

where c is defined in Assumption 4.2.

The proposal transmitted to SP - (t+1) is $q_{t+1} = B_t x_t$ and to SP^1 - (t+1) is $q^1_{t+1} = (B_t + \delta B_t)(x_t + \delta x_t)$. So

$$\|q^1_{t+1} - q_{t+1}\| \leq \|\delta B_t x_t\| + \|B_t \delta x_t\|$$

$$\frac{\|q^1_{t+1} - q_{t+1}\|}{\|q_{t+1}\|} \leq \left[\frac{\|\delta B_t\|\|x_t\|}{\|B_t x_t\|} + \frac{\|B_t\| c(\|\delta \mathcal{B}_t\|/\|\mathcal{B}_t\|)\ \|x_t\|}{(1 - c(\|\delta \mathcal{B}_t\|/\|\mathcal{B}_t\|))\ \|B_t x_t\|} \right] \tag{4.2c}$$

If we assume that each $q_{t+1}(q^1_{t+1})$ in a basis $\mathcal{B}_{t+1}$ $(\mathcal{B}^1_{t+1})$ is of the same order of magnitude, the left hand side of (4.2c) is also an estimate of $\|\delta \mathcal{B}_{t+1}\|/\|\mathcal{B}_{t+1}\|$.

If we therefore let $\|\delta \mathcal{B}_1\|/\|\mathcal{B}_1\| \leq 2^{-t_1}$, and note that $\|B_t x_t\| \leq \|B_t\|\|x_t\|$ so that

$$\frac{c(\|\delta \mathcal{B}_t\|/\|\mathcal{B}_t\|)}{(1 - c(\|\delta \mathcal{B}_t\|/\|\mathcal{B}_t\|))}$$

underestimates the 2nd term in (4.2c), then it is readily seen by induction that the leading term in the right hand side of (4.2c) contains the factor

$(c^{t}{}_{2}{}^{-t}{}_{1})$. This suggests that the columns of Q_t could depart substantially from the columns of Q_t when t is large. Note however that the solution of LP^1 will be very close to the solution of LP if the optimal basis is stable. Again, as in Section 3.2, we have the effect of correlation of errors.

5. *Reconstruction of the Solution*

In Section 2.3 we showed an example of how numerical difficulties can arise with reconstruction of the solution. In this section we discuss further difficulties associated with solution reconstruction.

5.1. Returning to the two-stage problem (3.2a) of Section 3.2, suppose now that computed proposals $\tilde{\underline{x}}_1^j$, which are exact solutions of (3.2b), are such that $\|E^j\|$ is no longer small--e.g. where an unstable algorithm is used to solve the system of equations $A_1^j \underline{x}_1^j = \underline{b}_1$. In this case, the optimal solution, say $(x_1^*, x_2^*) \geq 0$, of a linear programming problem LP^1, obtained by dynamically perturbing (3.2a), may be such that $\nexists$ no vector $x_1 > 0$ for which $(\underline{x}_1, \underline{x}_2^*)$ is feasible for LP (3.2a). (Recall that the (dynamic) perturbation is determined by the pattern of decomposition.) Thus method II of Section 2.2, which solves directly the LP

$$\begin{aligned} &\text{minimize} \quad \underline{c}_1^T \underline{x}_1 \\ &\text{subject to} \quad \begin{pmatrix} A_1 \\ B_1 \end{pmatrix} \underline{x}_1 = \begin{pmatrix} \underline{b}_1 \\ \underline{b}_2 - A_2 \underline{x}_2^* \end{pmatrix} \qquad (5.1a) \\ &\qquad\qquad \underline{x}_1 \geq 0 \, . \end{aligned}$$

could fail. A similar situation could arise when the assumptions of Section 4.2 do not hold, and it is not possible to account for computational errors in the master problem in terms of small (dynamic) perturbations in the original data A_1, A_2, B_1. Note that Method I will succeed, in the sense that it reconstructs a solution which is feasible for the perturbed LP. Thus in the

presence of numerical error Method II does not adequately reproduce Method I. Similar arguments also hold for a t stage problem. It is also clear from these arguments how crucial it is to use a stable implementation of the simplex method for solving the subproblems.

The second point we would like to make is that loss of feasibility occurs in a particular way, as we now show. Consider a three-stage problem of the form (1.1). As summarized in Fig. 1, the true extreme points of the first stage are denoted by $\underline{x}_1^j$, and we assume that the corresponding subproblem is bounded. The true extreme points of the second stage also assumed bounded are

$$\begin{pmatrix} \underline{\lambda}_2^j \\ \underline{x}_2^j \end{pmatrix},$$

and we shall assume that these are deliberately perturbed to

$$\begin{pmatrix} \underline{\tilde{\lambda}}_2^j \\ \underline{\tilde{x}}_2^j \end{pmatrix},$$

with $\sum_k \tilde{\lambda}_{2k}^j = 1 \; \forall \; j$. The third stage master is then given by

$$\begin{aligned} \sum_j (0 | B_2) \begin{pmatrix} \underline{\tilde{\lambda}}_2^j \\ \underline{\tilde{x}}_2^j \end{pmatrix} \lambda_{3j} + A_3 \underline{x}_3 &= \underline{b}_3 \\ \sum_j \lambda_{3j} &= 1 \\ \lambda_{3j}, \; \underline{x}_3 &\geq 0 \end{aligned} \qquad (5.1b)$$

Let the optimal solution obtained by solving (5.1b) exactly be

$$\begin{pmatrix} \lambda_3^* \\ \underline{x}_3^* \end{pmatrix} .$$

Then by Method I

$$\begin{pmatrix} \lambda_{-2}^* \\ \underline{x}_2^* \end{pmatrix} = \sum_j \begin{pmatrix} \tilde{\lambda}^j_{-2} \\ \underline{\tilde{x}}_2^j \end{pmatrix} \lambda_{3j}^*$$

and

$$\underline{x}_1^* = \sum_j \underline{x}_1^j \lambda_{2j}^*$$

Note now that (x_1^*, x_2^*, x_3^*) satisfies stage 1 and stage 3 constraints exactly. Only stage 2 constraints are violated. (We have assumed that $\sum_k \tilde{\lambda}_{2k}^* = 1$, again demonstrating the values of solving subproblems by a GUB type algorithm.) Error has however propogated to stage 3 in the sense that allocation of resources to each stage is affected and hence so is the computed optimal solution. However feasibility of stage 3 is retained.

5.2. In the light of the above two observations, we seek a method of reconstruction which

(1) uses the same pattern of decomposition as the Phase 1 and 2 procedures, and avoids numerical difficulties discussed earlier.

(2) is able to reconstruct at any point, not just at optimality.

(3) accurately mirrors Method I, but gets around having to save all

extreme points generated. We accept it to be reasonable that if a proposal $B_i \underline{x}_{-1}^j$ is saved, then the corresponding $\underline{x}_{-1}^j$ is also saved. When $B_i \underline{x}_{-1}^j$ is thrown away, then the corresponding $\underline{x}_{-1}^j$ can be dispensed with.

(4) is able to utilize a strategy akin to that used in a standard simplex method, when the optimal values of a set of variables say $\underline{x}_j^*$, are known.

We suggest the following strategy:

Method III:

Step 1: Let $\lambda_t^* \leftarrow \lambda_m^*$ and $t \leftarrow m$

Step 2: At stage t use λ_t^* to compute $\underline{x}_{t-1}^*$ by

$$\underline{x}_{t-1}^* = \sum_j \lambda_{tj}^* \underline{x}_{t-1}^j$$

(Recall (3) above.)

If $(t-1) = 1$, then STOP.

Step 3: Omit stage t and all subsequent stages, put upper and lower bounds on $\underline{x}_{t-1}$, constraining these variables to be close to $\underline{x}_{t-1}^*$, and solve the (t-1) stage problem for λ_{t-1}^* using nested decomposition.

Step 4. $t \leftarrow (t-1)$ and GOTO Step 2.

The major difference in Phase 3 between Method III and Method II is that Method III uses the same pattern of decomposition as the one used in Phase 2. The price one pays for this is the added information that must be saved, as discussed under (3) above in this Section.

6. Conclusions

We conclude by recounting some lessons we have learned from this effort for the implementation of large scale LP algorithms, and some specific recommendations for nested decomposition.

(a) This effort has benefited substantially from the numerical experiences of M. Aganagic [1977] on a real life model, and we reiterate the importance of gathering such practical experience. Indeed this is one of the points emphasized in Dantzig and Parikh [1977], who discuss how the activities of algorithms and model development complement each other.

(b) at the same time it is clear that the experimental information gathered will be suspect, unless the experimental implementation is sound and is designed to fail gracefully, i.e. it does not have to deal with all possible cases, but it must at least provide reasonably good clues when the algorithm encounters difficulties. This is the theme of Nazareth [1978a], where a set of software aids for developing good experimental implementations is described.

(c) our research demonstrates that a fairly substantial reformulation of the nested decomposition algorithm may be necessary before setting about developing a robust user oriented version, which is capable of handling a diverse set of LP applications.

(d) because error propagation may be an inherent problem we have been prompted to look at other decompositions, see Nazareth [1978b].

(e) It is clear that some extremely interesting numerical problems arise in nested decomposition. Our results are only a small contribution, and we feel that a more complete error analysis and perturbation theory, which we now plan to undertake, would be a substantial contribution.

There is also a definite need for a sound experimental implementation, in order to carry out further experimentation. This code should, for example,

(i) use stable factoring and updating techniques as described in Reid [1976].

(ii) use the method of reconstruction described in Section 5.

(iii) use the techniques of structured programming, since the data flow in the algorithm is quite complex. Data flow and numeric considerations should be separated when possible.

(iv) be able to start from an arbitrary feasible (not necessarily basic) solution, either user supplied or developed from an intermediate reconstruction.

(v) use a technique similar to that in the GUB algorithm, Dantzig and Van Slyke [1967] with LU factorization of the basis, see Tomlin [1974], to ensure that convexity rows are closely satisfied.

REFERENCES

Aganagic, M. (1977). *Private communication.*

Cobb, R.H. and J. Cord (1967). "Decomposition approaches for solving linked programs" in Proc. of the Princeton Symposium on Mathematical Programming, Ed. H. W. Kuhn, Princeton University Press, Princeton, N.J., pp. 37-49.

Dantzig, G.B. (1963). Linear Programming and Extensions, Princeton University Press, Princeton, N.J.

Dantzig, G.B. and S. C. Parikh (1975). "On a pilot linear programming model for assessing physical input on the economy of a changing energy picture," Systems Optimization Laboratory, Rept SOL 75-14, Department of Operations Research, Stanford University.

Dantzig, G. B. and S. Parikh (1977). "At the interface of modelling and algorithms research," Systems Optimization Laboratory Rept SOL 77-29, Department of Operations Research, Stanford University.

Dantzig, G.B. and R. M. Van Slyke (1967). "Generalized Upper Bounding Techniques," J. Comput. System. Sci. 1, pp. 213-226.

Dantzig, G.B. and P. Wolfe (1960). "Decomposition Principle for Linear Programs," Operations Research 8, pp. 101-111.

Glassey, C.R. (1973). "Nested decomposition and multi-stage linear programs," Manag. Sci., 20, pp. 282-292.

Ho, J. K. and A. S. Manne (1974). "Nested decomposition for dynamic models," Math. Prog. 6, pp. 121-140.

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

Ho, J. K. (1974b). "Optimal design of multistage structures: a nested decomposition approach," Systems Optimization Laboratory, Rept SOL 74-2, Department of Operations Research, Stanford University.

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

Madsen, O.B. (1977). "Solution of LP problems with staircase structure, Rep. 26/1977, IMSOR, Technical University of Denmark, DK-2800 Lyngby, Denmark.

Nazareth, L. (1978a). "Modules to aid the implementation of LP algorithms," Systems Optimization Laboratory, Rept. SOL 78-28, Department of Operations Research, Stanford University.

Nazareth, L. (1978b). "Dually equivalent decomposition algorithms with application to solving staircase structures," Systems Optimization Laboratory, Rept. SOL78-30. Department of Operations Research, Stanford University.

Reid, J. K. (1976). "FORTRAN Subroutines for handling sparse linear programming bases," AERE Report R8269, Harwell, Oxfordshire, England.

Tomlin, J.A. (1974). "Generalized upper bounds and triangular decomposition in the simplex method," Operations Research 22, No. 3, pp. 664-668.

Tomlin, J.A. (1975). "On scaling LP problems," Math. Prog. Study 4, pp. 146-166.

Wilkinson, J.H. (1965). The Algebraic Eigenvalue Problem, Oxford University Press, Oxford.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER SOL 78-27	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ERROR PROPOGATION AND SOLUTION RECONSTRUCTION IN NESTED DECOMPOSITION		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER SOL 78-27
7. AUTHOR(s) L. Nazareth		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0267
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Operations Research -- SOL Stanford University Stanford, CA 94305		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR-047-143
11. CONTROLLING OFFICE NAME AND ADDRESS Operations Research Program -- ONR Department of the Navy 800 N. Quincy Street, Arlington, VA 22217		12. REPORT DATE December 1978 13. NUMBER OF PAGES 29
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This report has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Nested Decomposition Method
Computational Linear Algebra
Ho and Manne Algorithm

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

We study some of the numerical properties of the nested decomposition algorithm of Ho and Manne. In particular we seek to show how well developed theory in the area of computational linear algebra, due primarily to J.H. Wilkinson, carries over to linear programming and yields useful insight into the behavior of algorithms in this area.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)