NBER WORKING PAPER SERIES

FACTORING LP BLOCK-ANGULAR BASES

Wm. Orchard-Hays*

Working Paper No. 61

COMPUTER RESEARCH CENTER FOR ECONOMICS AND MANAGEMENT SCIENCE
National Bureau of Economic Research, Inc.
575 Technology Square
Cambridge, Massachusetts 02139

November 1974

Preliminary: not for quotation

*NBER Computer Research Center. Research supported in part by National Science Foundation Grant GJ-1154X3 to the National Bureau of Economic Research, Inc.

Abstract

A factorization of the basis for any block-angular LP model is presented, and its inverse is shown to be readily maintainable as piecemeal product-forms plus possible additional columns. Straightforward rules for piecemeal transformation of full rows and columns are given.

Contents

1. Environment

The form of a general block-angular LP model for decomposition algorithms is as follows:

$$A_o X_o + \sum_{p=1}^{P} A_p X_p = b_o$$

$$B_p X_p = b_p, \; p=1,\ldots, P$$

where the A_p and B_p are matrices of row-order m_o and m_p ($p=0,1,\ldots,P$ for A_p, $p=1,\ldots,P$ for B_p), X_p are n_p-order columns matching the column orders of the A_p (and B_p for $p > 0$), and b_p are m_p-order columns of constants.

A_o and each B_p are assumed to contain full m_p-order identity matrices with corresponding logical (slack) variables in the X_p columns. Except as explicitly noted, it is unnecessary to distinguish logical and structural variables in the present discussion. One free logical in

$$X_o = \{X_o^1, X_o^2, \ldots, X_o^{n_o}\}$$

is to be maximized. For simplicity, we will take this to be X_o^1 when necessary to distinguish it, i.e., the functional is the top row.

When necessary to be precise, the m_p-order identity matrix will be denoted by I_p, but usually I will stand for the identity of whatever order is required. Occasssionally I_k is used to dentoe the k-order identity.

In the present discussion, the total model is of only minor interest, the basis for some solution to such a model being the focus of attention. Hence the same letters as above will be used for substructures of a basis, with no additional notation since additional marks will be required for other purposes. Thus, in the sequel, A_o and B_p stand for square, nonsingular matrices unless modified by "the full".

2. A General Basis

Almost the entire difficulty in partitioning a block-angular model for computation is due to the fact that a general basis has a more complicated structure than the entire model. The most general structure required for a basis is as follows:

$$\begin{bmatrix} (A_{oo} & T_{o1} & \cdots & T_{oP}) & A_1 & \cdots & A_p & \cdots & A_P \\ (0 & T_{11} & \cdots & 0) & B_1 & & & & \\ & \vdots & & & & \ddots & & & \\ (0 & 0 \, .. \, T_{pp} & \cdots & 0) & & & B_p & & \\ & \vdots & & & & & & \ddots & \\ (0 & 0 \, ... & 0 & T_{PP}) & & & & & B_P \end{bmatrix}$$

where A_{oo} is k_o columns from the full A_o, T_{op} are k_p columns from the full A_p with T_{pp} the corresponding columns from the full B_p.

The columns of A_{oo} and of each T_{oP} are all linearly independent so that the matrix

$$(A_{oo} \; T_{o1} \; \cdots \; T_{oP})$$

is nonsingular. Obviously then,

$$\sum_{p=o}^{P} k_p = m_o$$

3. Factoring of D and D^{-1}

Let

$$B = \begin{bmatrix} I_o & A_1 & \cdots & A_p & \cdots & A_P \\ & B_1 & & & & \\ & & \ddots & & & \\ & & & B_p & & \\ & & & & \ddots & \\ & & & & & B_P \end{bmatrix}$$

The matrix B is readily factored into P matrices of the form

$$\begin{bmatrix} I_o & \cdots & A_p & \cdots & 0 \\ & I & & & \\ & & \ddots & & \\ & & & B_p & \\ & & & & \ddots \\ & & & & & I \end{bmatrix}$$

which are completely commutative and have inverses of the same form. Hence the inverse of B can be computed piecemeal and the pieces multiplied together in any order. Since P=2 encompases all possible cases, that is, any results can be applied recursively, we illustrate the above statements for P=2, which in fact constitutes a proof.

$$\begin{bmatrix} I & A_1 & \\ & B_1 & \\ & & I \end{bmatrix}^{-1} = \begin{bmatrix} I & -A_1 B_1^{-1} & \\ & B_1^{-1} & \\ & & I \end{bmatrix}, \qquad \begin{bmatrix} I & & A_2 \\ & I & \\ & & B_2 \end{bmatrix}^{-1} = \begin{bmatrix} I & & -A_1 B_2^{-1} \\ & I & \\ & & B_2^{-1} \end{bmatrix}$$

$$\begin{bmatrix} I & A_1 & \\ & B_1 & \\ & & I \end{bmatrix} \begin{bmatrix} I & & A_2 \\ & I & \\ & & B_2 \end{bmatrix} = \begin{bmatrix} I & & A_2 \\ & I & \\ & & B_2 \end{bmatrix} \begin{bmatrix} I & A_1 & \\ & B_1 & \\ & & I \end{bmatrix} = \begin{bmatrix} I & A_1 & A_2 \\ & B_1 & \\ & & A_2 \end{bmatrix}$$

Let $-A_p B_p^{-1} = \bar{A}_p$. Then substituting $\bar{A}_p$ for A_p in the above shows that the inverses commute and multiply together in exactly the same way. Hence handling of B and B^{-1} poses no problem at all. Note that $-A_p B_p^{-1}$ is computed automatically by the product form of inverse. In fact, the above factorization is merely a generalization of a special case of the product form of inverse. This gave rise to the name "block-product form of inverse" in an ealier decomposition algorithm.[4]

We must now seek a matrix E such that either EB=D or BE=D. A little experimentation should convice the reader that the form EB leads to more complications than it resolves. Hence, we adopt the form BE=D.

Let E be partitioned exactly as D with blocks denoted by E_{pq}. Let

$$E_{op} = 0 \quad \text{for } p > 0$$

$$E_{pp} = I_p \quad \text{for } p > 0$$

$$E_{pq} = 0 \quad \text{for } q \neq p > 0$$

$$E_{po} = B_p^{-1} (0 \ldots T_{pp} \ldots 0) = (0 \ldots \bar{T}_{pp} \ldots 0) \quad \text{for } p > 0$$

$$\begin{aligned} E_{oo} &= (A_{oo}\ T_{o1} \ldots T_{oP}) - \sum_{p=1}^{P} A_p B_p^{-1} (0 \ldots T_{pp} \ldots 0) \\ &= (A_{oo}\ T_{o1} \ldots T_{oP}) + \sum_{p=1}^{P} \bar{A}_p (0 \ldots T_{pp} \ldots 0) \\ &= (A_{oo}\ T_{o1} \ldots T_{oP}) - \sum_{p=1}^{P} A_p (0 \ldots \bar{T}_{pp} \ldots 0) \end{aligned}$$

Note that the second term in E_{oo} can be computed in either of two ways, whichever is more convenient.

Again illustrating with P=2, it is apparent that the above definitions of E_{pq} satisfy the equation BE=D.

$$\begin{bmatrix} I & A_1 & A_2 \\ & B_1 & \\ & & B_2 \end{bmatrix} \begin{bmatrix} ((A_{oo}\ T_{o1}\ T_{o2}) - A_1\ (0\ \bar{T}_{11}\ 0) - A_2\ (0\ 0\ \bar{T}_{22})) & & \\ (0\ \bar{T}_{11}\ 0) & I & \\ (0\ 0\ \bar{T}_{22}) & & I \end{bmatrix}$$

$$= \begin{bmatrix} (A_{oo} & T_{o1} & T_{02}) & A_1 & A_2 \\ (0 & T_{11} & 0) & B_1 & \\ (0 & 0 & T_{22}) & & B_2 \end{bmatrix} = D$$

Therefore $D^{-1} = E^{-1}\ B^{-1}$. Since E is of the form

$$\begin{bmatrix} E_{oo} & & \\ E_{10} & I & \\ E_{20} & & I \end{bmatrix}$$

its inverse is of the form

$$\begin{bmatrix} E_{oo}^{-1} & & \\ -E_{10}\ E_{oo}^{-1} & I & \\ -E_{20}\ E_{oo}^{-1} & & I \end{bmatrix}$$

The difficulty is thus reduced to computing E_{oo}^{-1}. Although this looks somewhat formidable, further simplications are possible.

4. Recombining Second Order Factors

Consider a single factor of B^{-1} in the form

$$\begin{bmatrix} I_o & \bar{A}_1 \\ 0 & B_1^{-1} \end{bmatrix} = D_1^{-1}$$

and an E_{oo} with all T-columns replaced with unit vectors, the positions in all lower blocks being zero, i.e.

$$(A_{oo} \quad I_{m_o - k_o})$$

Suppose some column T_1 from the full $\{A_1, B_1\}$ is to be introduced to replace the (k_o+1) - st unit vector. The column T must first be transformed by D_1^{-1}, as follows:

$$D_1^{-1}\, T_1 = \begin{bmatrix} I & \bar{A}_1 \\ & B_1^{-1} \end{bmatrix} \begin{bmatrix} T_{o1} \\ T_{11} \end{bmatrix} = \begin{bmatrix} T_{o1} + \bar{A}_1 \; T_{11} \\ B_1^{-1} \; T_{11} \end{bmatrix} = \begin{bmatrix} T_{o1} - A_1 \; \bar{T}_{11} \\ \bar{T}_{11} \end{bmatrix}$$

The result is exactly the two subcolumns which should go into the new E_{oo} and E_{10}. The new E_{oo}^{-1} can be computed in product form, as follows:

Let $\bar{T}_{o1} = T_{o1} - A_1 \bar{T}_{11}$

$\alpha = E_{oo}^{-1} \; \bar{T}_{o1}$

η_{o1} = the eta column formed from α by pivoting on α^{k_o+1}

η_{11} = the extension to η_{o1} using $\bar{T}_{11}$

$F(\eta_{o1})$ = the elementary column matrix containing η_{o1} in column k_o+1

Then

$$\hat{E}_{oo}^{-1} = ((F(\eta_{o1})\ A_{oo}^{-1})\ \eta_{o1}\ I_{m_o-k_o-1}) = (\hat{A}_{oo}^{-1}\ \ \eta_{o1}\ I)$$

(Note: The notation A_{oo}^{-1} is imprecise since A_{oo} is not square. What is meant is the part of E_{oo}^{-1} corresponding to A_{oo}. The remaining unit columns are unchanged.)

Now $\quad -\hat{E}_{10}\ \hat{E}_{oo}^{-1} = -\ (0\ldots\bar{T}_{11}\ldots)\ \hat{E}_{oo}^{-1}$ which is merely*

row k_o+1 of $\hat{E}_{oo}^{-1}$ repeated m_1 times scaled by the elements $-\bar{T}_{11}^{i}$. Call this highly singular matrix $(\hat{E}^{-1})_{10}$. To see its effect in subsequent transformations, suppose some general column $S=\{S_o,S_1,S_2\}$ is to be transformed into terms of the new basis or, more briefly, "updated". First

$$D^{-1}S = \begin{bmatrix} I & \bar{A}_1 & \bar{A}_2 \\ & B_1^{-1} & \\ & & B_2^{-1} \end{bmatrix} \begin{bmatrix} S_o \\ S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} S_o + \bar{A}_1\ S_1 + \bar{A}_2\ S_2 \\ B_1^{-1}\ S_1 \\ B_2^{-1}\ S_2 \end{bmatrix} = \begin{bmatrix} \bar{S}_o \\ \bar{S}_1 \\ \bar{S}_2 \end{bmatrix}$$

Note that $\bar{S}_o$ is simply the upper parts of the updates by D_1^{-1} and D_2^{-1} added to S_o. Next,

$$\hat{E}^{-1}\ \bar{S} = \begin{bmatrix} \hat{E}_{oo}^{-1} & & \\ (\hat{E}^{-1})_{10} & I_1 & \\ 0 & & I_2 \end{bmatrix} \begin{bmatrix} \bar{S}_o \\ \bar{S}_1 \\ \bar{S}_2 \end{bmatrix} = \begin{bmatrix} \hat{E}_{oo}^{-1}\ \ \bar{S}_o \\ (E^{-1})_{10}\ \bar{S}_o+\bar{S}_1 \\ \bar{S}_2 \end{bmatrix} = \begin{bmatrix} \bar{\bar{S}} \\ \bar{\bar{S}}_1 \\ \bar{\bar{S}}_2 \end{bmatrix}$$

*Such a product of a column on the left multiplied by a row on the right is often called an <u>outer product</u>, as it is in the sequel.

$\bar{\bar{S}}_o$ is merely another update to $\bar{S}_o$ and $\bar{\bar{S}}_2 = \bar{S}_2$. But what of the term $(\hat{E}^{-1})_{10}\ \bar{S}_o$? It is merely

$$- \bar{\bar{S}}_o^{k_o+1} \quad \bar{T}_{11} \qquad\qquad \text{(scalar times } m_1\text{-order column)}$$

Hence the computation and recording of either η_{11} or $(\hat{E}^{-1})_{10}$ is unnecessary. Only $\bar{T}_{11}$ need be kept. Furthermore, since it applies only to the p=1 segment, it can be kept with the p=1 block. If there were several such columns, they would all be additive (subtractive) with multipliers from $\bar{\bar{S}}_o$. If $(\hat{E}^{-1})_{20}$ were not void, it would apply in the same way to $\bar{\bar{S}}_2$.

This appears to be about as complete a factorization if D^{-1} as is possible. Note that all inverse factors except the $\bar{\bar{T}}_{pp}$ can be carried in product form in the usual manner, in fact this is advantageous for computing the terms $\bar{A}_p\ S_p$. And even the application of the $\bar{T}_{pp}$ is only a slight variation on usual product-form updating.

Note also that if a new $\hat{T}_1$ replaces the original T_1, the upper part can be incorporated in the product form of E_{oo}^{-1} and the lower part simply replaces $\bar{T}_{11}$. It must be understood, however, that if any B_p^{-1} changes, the $\bar{T}_{pp}$ must be updated, which is another reason for carrying them with block p. Moreover, there is an effect on E_{oo}^{-1} which is rather more complicated. This will be taken up in Section 6.

5. Row Updates with the Factored Inverse

Suppose a general row $R=(R_o,R_1,\ldots,R_p)$ must be transformed. In point of fact, the simplex method and most of its variations almost never update a general row, except the Phase 1 feasibility form. In a decomposition model, feasible solutions to the subproblems must be found independently anyway, so that even a feasibility form would be nonzero only in R_o. In dual pricing, the denominator form would be nonzero in some block R_p and zero elsewhere. However, we may as well look at the general case since the special cases will be apparent.

In order to compute RD^{-1}, we first compute

$$RE^{-1} = (R_o,R_1,R_2)\begin{bmatrix} E_{oo}^{-1} & & \\ (E^{-1})_{10} & I_1 & \\ (E^{-1})_{20} & & I_2 \end{bmatrix}$$

$$= ((R_oE_{oo}^{-1} + R_1(E^{-1})_{10} + R_2(E^{-1})_{20}),\ R_1,\ R_2)$$

Since R_1 and R_2 are unchanged we can concentrate on R_o. The first term is merely the usual backward transformation if E_{oo}^{-1} is kept in product form. If $R_1=R_2=0$, as would often be the case, that is the end of it, but suppose not. However, even then, there is a further effect only when both R_p and $(E^{-1})_{po}$ are nonzero. Suppose R_1 and $(E^{-1})_{10}$ are both nonzero; what is their product? Recall that $(E^{-1})_{10}$ is the <u>outer</u> product of $\bar{\bar{T}}_{11}$ and $(E_{oo}^{-1})^{k_o+1}$ or, more generally, the r-th row of E_{oo}^{-1}, or the sum of several such outer products for $r_1,r_2,\ldots$ For each such r, let

$$f_r = R_1\ \bar{T}_{11},\ \text{an } \underline{\text{inner}} \text{ product over } m_1 \text{ elements}\quad .$$

Then

$$R_1(E^{-1})_{10} = \sum_r f_r \pi^r$$

where

$$\pi^r \text{ is the r-th row of } E_{oo}^{-1}$$

This is readily computed by adding each f_r to the r-th element of R_o before computing

$$R_o E_{oo}^{-1}$$

Let $\bar{\bar{R}}_o$ be the product so computed. Then we must compute

$$\bar{R} = (\bar{\bar{R}}_o, R_1, R_2) B^{-1}$$

For p=1, this gives

$$(\bar{\bar{R}}_o, R_1, R_2) \begin{bmatrix} I & \bar{A}_1 & \\ & B_1^{-1} & \\ & & I \end{bmatrix} = (\bar{\bar{R}}_o, (\bar{\bar{R}}_o \bar{A}_1 + R_1 B_1^{-1}), R_2)$$

and this applied to block p=2 gives

$$(\bar{\bar{R}}_o, (\bar{\bar{R}}_o \bar{A}_1 + R_1 B_1^{-1}), R_2) \begin{bmatrix} I & & \bar{A}_2 \\ & I & \\ & & B_2^{-1} \end{bmatrix} = (\bar{\bar{R}}_o, (\bar{\bar{R}}_o \bar{A}_1 + R_1 B_1^{-1}), (\bar{\bar{R}}_o \bar{A}_2 + R_2 B_2^{-1}))$$

$$= (\bar{R}_o, \bar{R}_1 . \bar{R}_2)$$

Note that $\bar{R}_o = \bar{\bar{R}}_o$ and that $\bar{R}_p$, $p > 0$, are of the same form, merely the usual backward transformation for $(\bar{R}_o, R_p)$ post-multiplied by D_p^{-1} .

Thus the $\bar{R}_p$ can be computed piecemeal provided all the $\bar{T}_{pp}$ are accessible first to compute $\bar{R}_o$.

6. Effect of Change in D_p^{-1} on E_{oo}^{-1}

It has thus far been shown that both row and column updates are readily performed with the factored inverse and that changes in E^{-1} are easily accounted for. However, a change in a D_p^{-1} is not as simple when $\bar{T}_{pp}$ exist. First of all, the $\bar{T}_{pp}$ must be updated, as previously noted, but this is not different from one eta-update on any set of m_p-order columns. We must now investigate the effect on E_{oo}^{-1}.

Consider again the situation arrived at with $\hat{E}_{oo}^{-1}$ in Section 4, and suppose the next change of basis occurs in D_1, i.e. in $\{A_1, B_1\}$. Insofar as D_1^{-1} is concerned, this is handled in normal fashion with an additional eta-column. But $\hat{E}_{oo}$ and $\hat{E}_{oo}^{-1}$ were computed on the basis of the original D_1 and are no longer valid. Drop the hats on $\hat{E}_{oo}$ and $\hat{E}_{oo}^{-1}$ and consider them the current E_{oo} and E_{oo}^{-1}. We now have $\hat{D}_1$ and $\hat{D}_1^{-1}$. Let

$$\hat{D}_1^{-1}\ T_1 = \begin{bmatrix} T_{o1} - \hat{A}_1 \ T_{11} \\ T_{11} \end{bmatrix} = \begin{bmatrix} \hat{T}_{o1} \\ \hat{T}_{11} \end{bmatrix}$$

Then the (k_o+1)-st column of E_{oo} has changed from $\bar{T}_{o1}$ to $\hat{T}_{o1}$ and must be accounted for in E_{oo}^{-1}. This is done as usual by computing

$$\hat{\alpha} = E_{oo}^{-1}\ \hat{T}_{o1}$$

and pivoting on $\hat{\alpha}^{k_o+1}$ to form a new $\hat{\eta}_{o1}$ which is added to the product form for E_{oo}^{-1} to give (a new) $\hat{E}_{oo}^{-1}$. Furthermore, this must be done for all T-columns from block p=1 if several are in effect.

The question arises as to whether $\hat{\alpha}^{k_o+1}$ might not vanish so that pivoting is not possible. We glossed over this question in the first place but presumably the column T_1 was selected to pivot in position k_o+1 becasue that α-element was nonzero. The change in $\{A_1,B_1\}$, however, was determined on the basis of some updated element from the full B_1 being nonzero and has no obvious implication for $\hat{T}_{o1}$ and $\hat{\alpha}$. One answer is that since, in a global sense, we are simply multiplying nonsingular matrices to form a nonsingular product, no factor can become singular, but this argument is incomplete and vague.

Let S be the column from the full $\{A_1,B_1\}$ which replaced the r-th column ($r > m_o$) of D_1 to give $\hat{D}_1$. To make this selection, the column S had to be updated and made into an α-column. Let us carry out this calculation, again with P=2.

$$D^{-1} S = \begin{bmatrix} I & \bar{A}_1 & \bar{A}_2 \\ & B_1^{-1} & \\ & & B_2^{-1} \end{bmatrix} \begin{bmatrix} S_o \\ S_1 \\ 0 \end{bmatrix} = \begin{bmatrix} S_o + \bar{A}_1 S_1 \\ B_1^{-1} S_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \bar{S}_o \\ \bar{S}_1 \\ \bar{S}_2 \end{bmatrix}$$

$$E^{-1} \bar{S} = \begin{bmatrix} E_{oo}^{-1} & & \\ (E^{-1})_{10} & I & \\ (E^{-1})_{20} & & I \end{bmatrix} \begin{bmatrix} \bar{S}_o \\ \bar{S}_1 \\ 0 \end{bmatrix} = \begin{bmatrix} E_{oo}^{-1} \bar{S}_o \\ (E^{-1})_{10} \bar{S}_o + \bar{S}_1 \\ (E^{-1})_{20} \bar{S}_o \end{bmatrix} = \begin{bmatrix} \bar{\bar{S}}_o \\ \bar{\bar{S}}_1 \\ \bar{\bar{S}}_2 \end{bmatrix}$$

Now $\bar{\bar{S}}_1^r \neq 0$ since it was selected to pivot on. Assuming only one $\bar{T}_{11}$ in position k_o+1, then

$$\bar{\bar{S}}_1^r = \bar{S}_1^r - \bar{\bar{S}}_o^{k_o+1} \bar{T}_{11}^r \neq 0$$

Hence not both terms are zero and they are not equal. But we have uncovered another question: How do we know that $\bar{S}_1^r \neq 0$ so the pivot can be made in D_1? Let us answer this question first.

Suppose $\bar{S}_1^r=0$. Then S should not replace the r-th column in D_1 but the T_1 column in E since $\bar{\bar{S}}_o^{k_o+1} \neq 0$. We have already seen how to accomplish this. But of course this is not to the purpose, so after replacing T_1 with S, we must replace the r-th column of D_1 with T_1. We are assured that this is possible since $\bar{T}_{11}^r$, the second pivot, is also nonzero.

Now suppose $\bar{S}_1^r \neq 0$. The $\hat{\alpha}^{k_o+1}$ we initially started to investigate is simply $\bar{S}_o^{k_o+1}$ and what makes it nonzero? Suppose it is zero. Then S does not depend on T_1 and vice-versa so E_{oo}^{-1} need not be updated.* This seems like a nice answer but there is a catch.

Let us now assume there are several T-columns from block p=1, say in positions $t_1, t_2, \ldots$ Then

$$\bar{\bar{S}}_1^r = \bar{S}_1^r - \sum_t \bar{\bar{S}}_o^t \bar{T}_t^r$$

Now if $\bar{S}_1^r=0$, we can select any $\bar{\bar{S}}_o^t \neq 0$ for the interchange, say the largest magnitude. But if $\bar{S}_1^r \neq 0$ and the sum is zero, it is not necessarily true that S is independent of all T_t. Furthermore if the sum is not zero but some $\bar{\bar{S}}_o^t$ are zero, they may not remain zero as individual updates to E_{oo}^{-1} are made. Hence the updating of E_{oo}^{-1} may be order-dependent. (A similar phenomenon occurs in updating GUB bases which are a special case of decomposition. The situation is much messier in general block-angular models.)

*Note that if $\bar{\bar{S}}^{k_o+1}=0$, $\bar{\bar{S}}_1=\bar{S}_1$, that is, $\bar{T}_{11}$ does not enter into the calculation and hence $\bar{T}_{o1}$ does not change.

7. Cross-Block Exchanges

There are two remaining cases to consider: a column from the full A_o replacing a T-column, and a column from block q replacing one from block p. We will take up the latter first.

The calculation of $\bar{\bar{S}}$ in the previous section showed that although $S_2=0$, $\bar{\bar{S}}_2$ might not be zero and the pivot might be selected from this segment. However, in this case, for some r-index in p=2 and some t-index in p=0,

$$\bar{\bar{S}}_2^r = \bar{\bar{S}}_o^t \, \bar{\bar{T}}_{t2}^r \neq 0$$

Hence column S from block 1 replaces column T_t from block 2 in E. The old $\bar{T}_{t2}$ must be dropped and $\bar{S}_1$ added to the T-columns for p=1. The E_{oo}^{-1} is updated in standard fashion. There is no trouble with zero values.

Now suppose some column S from the full A_o is to replace some T-column. In this case,

$$\bar{\bar{S}} = \begin{bmatrix} E_{oo}^{-1} & S_o \\ (E^{-1})_{10} & S_o \\ (E^{-1})_{20} & S_o \end{bmatrix}$$

since $S_1=S_2=0$ and $\bar{S}_o=S_o$. The update of E_{oo}^{-1} is standard, i.e. $\bar{\bar{S}}_o$ is a regular α-column pivoting in some position $r \leq m_o$. The $\bar{T}_{pp}$ is merely dropped from the set for block p.

We thus have the rather surprising result that inter-block exchanges are simpler than intra-block exchanges, except for p=0. (A column from the full A_o replacing another in E_{oo} is a standard operation, just like the last case above.)

8. Summary of Basis-Change Cases

We summarize here for more convenient reference the various basis-change cases analyzed in prior sections. The designation (p,q) indicated (in, out) with respect to blocks.

A. Case (o,o)

Some S from the full A_o replaces another in E_{oo}.

Standard LP update.

B. Case (o,p)

Same as Case (o,o) except the outgoing T-column must have its $\bar{T}_{pp}$ dropped from the set p.

C. Case (p,o)

Compute $\bar{T}_{op}$ and $\bar{T}_{pp}$ with D_p^{-1}. Use $\bar{T}_{op}$ as the entering column in $\hat{E}_{oo}$ and add $\bar{T}_{pp}$ to the set of T-columns for block p.

D. Case (p,q)

Can only occur as a change in E with one or more columns $\bar{T}_{qq}$ in effect. Drop the outgoing $\bar{T}_{qq}$ from the q set and then proceed as in Case (p,o).

E. Case (p,p)

1. $\bar{S}_p^r \neq 0$ where r is basis index of outgoing column.
   Use $\{\bar{S}_o, \bar{S}_p\}$ as the α-column to update D_p^{-1} to $\hat{D}_p^{-1}$.
   If $(E^{-1})_{po} = 0$, done. ($(E^{-1})_{po} = 0$ if and only if the set of $\bar{T}$-columns for block p is empty.) If not, proceed as follows:

Update all T_{pp} to reflect the change in D_p.
Each one has a position index t. For each $\bar{\bar{S}}_o^t \neq 0$, use $\bar{\bar{S}}_o$ as an entering column in E_{oo} pivoting on position t. This may have to be done recursively until all t are processed. (Note that $\bar{\bar{S}}_o$ itself changes with each such update.)

2. $\bar{S}_p^r = 0$

In this case, $\sum_t \bar{\bar{S}}_o^t \bar{T}_t^r \neq 0$. Select, say,

$$\max_t \left[\bar{\bar{S}}_o^t\right]$$

as the t of interest. Treating T_t as an outgoing column, do Case (p,q). (Actually q=p but this is immaterial.) Now treating T_t as an incoming column (in place of S which replaced it in E), do step 1. above.

9. A Skeletal Decomposition Algorithm

Of the several algorithms which have been developed for block-angular decomposition mdoels, the best-known and, probably for that reason, the most successful have been those based on the Dantzig-Wolfe principle. However, D-W algorithms have often proved unsatisifacotry in practice although the generality of approach is sometimes indispensible.

The concept of partitioning is not usually associated with D-W algorithms and, in fact, all algorithms are sometimes regarded as falling into two classes: D-W or Generalized LP, and partitioning schemes. But this is inaccurate.

D-W algorithms must deal with subproblems and a master or derived problem just as any others do. The proper distinction is whether or not factorization of the basis inverse is employed. In D-W algorithms, factorization of B^{-1} is implicitly used but no particular point is made of it. Nevertheless, factorization is an outgrowth of Dantzig's old idea of a pseudo-basis. The GUB algorithm of Dantzig and Van Slyke[5] when implemented with product-form is a special case of complete factorization, or, more properly, complete factorization of a special case of a block-angular model.

The Beale decomposition scheme,[3] produced before computers were adequate, used a form of pseudo-basis and what amounted to factorization. The block-product algorithm developed by this writer[4] used factorization essentially as described in the preceding sections, though in more tortuous forms, combined with a parametric RHS approach. Unfortunately, most readers focused on the parametric aspects rather than the factorization. Also, the computer implementations of the algorithm (of which there were two with a third variant reportedly under development) fell into obscurity for nontechnical reasons. Consequently, factorization as such is not well known. However, the excellent performance of GUB algorithms in recent years ought to recommend more attention to it. Furthermore, it is not antithetical to other concepts but may be helpful to their successful implementation. Any reasonable algorithm must, in this writer's opinion, employ the factorization of B^{-1}. Indeed, this is virtually the <u>raison d'etre</u> for decomposing block-angular models. The use of E^{-1}, while more complicated, avoids many of the numerical problems of standard D-W algorithms and the associated slow-convergence properties.

A skeletal algorithm is outlined below. Points at which a user's own variation are easily incorporated will be noted. The reason for such a skeletal algorithm is to standardize and automate the various complicated data handling problems and transformations which always occur. It is simply impractical for each investigator to start building all his own system gear from scratch. What is needed is an off-the-shelf decomposition "engine" which can be used in a variety of "vehicles".

Step 0 Obtain, generate or guess a master pricing row $\pi_o=(\pi_{o1},\ldots,\pi_{om_o})$, i.e., a set of dual variable values for the A_p. A number of schemes for obtaining π_o have been proposed and several used. Any meaningful approach is worth considering. However, it must be realized that even if the optimal π_o were provided, no algorithm will produce a global optimal solution in one sweep except by sheer chance.

Step 1 Obtain a "good" feasible solution to each subproblem in the following form:

$$\left\{\begin{matrix}\text{maximize}\\ \text{minimize}\end{matrix}\right\} \pi_o A_p \text{ subject to}$$

$$\begin{bmatrix} I_o & A_p \\ 0 & B_p \end{bmatrix} \begin{bmatrix} V_p \\ X_p \end{bmatrix} = \begin{bmatrix} 0 \\ b_p \end{bmatrix}$$

and stated ranges on the X_p^j, where V_p is a column of m_o free variables.

It is probably wasteful to fully optimize each $\pi_o A_p$ on the first sweep but some improvement over the first feasible solution should be obtained. If $\pi_o A_p$ goes unbounded, just stop at that point since presumably π_o is incorrect. [The user may have additional rules to impose here.]

Accumulate $\sum_p V_p$ as the subproblems are solved. If any subproblem is infeasible, the whole model is andthere is no use continuing. Also construct the basic solution column $\beta=\{0,\beta_1,\ldots,\beta_P\}$ where β_p is the basic subcolumn of X_p.

<u>Step 2</u> We have the following (probably infeasible) solution to the whole model.

$B=\Pi_p D_p$, the bases obtained in Step 1, with corresponding $D^{-1}=\Pi_p D_p^{-1}$

Since $V_p + A_p X_p = 0$, with all X_p feasible,

$$\sum_{p=1}^{P} A_p X_p = -\sum_p V_p$$

$E=I_m$ where $m = \sum_{p=0}^{P} m_p$, hence

$E^{-1}=I_m$, $E_{oo}^{-1}=I_o$

Thus,

$$E_{oo}\, U_o + \sum_p A_p\, X_p = b_o$$

$$B_p\, X_p = b_p \qquad\qquad (p=1,\ldots,P)$$

where U_o is the subcolumn of logical variables in X_o and has the vector value $b_o + \sum_p V_p$. Only elements of U_o are primally infeasible. U_p becomes β_o. If the user wishes to use a D-W algorithm, he may alternatively regard the V_p as candidate columns and form the derived problem

$$A_o X_o - \sum_p \sum_j V_{jp}\, \lambda_{jp} = b_o$$

$$\sum_j \lambda_{jp} = 1$$

where only j=1 for each p is presently defined.

The effect is the same in either event: If U_o is not feasible, a Phase 1 π_o is now generated; if it is, a Phase 2 π_o. Then an attempt is made to obtain either feasibility or optimality with $A_o X_o$ holding the V_p constant. If an unbounded feasible solution is found, the whole model is unbounded and nothing more need be done. Otherwise a final π_o for this sweep is obtained, whether Phase 1 or Phase 2. In general, E_{oo} is now of the form initially assumed in Section 4, with corresponding β_o

Step 2A An irrevocable decision must be made as to whether to use factorization or not. If a D-W approach or some other convergence scheme is employed, the rest of the mechanics are essentially repetitions of Step 1, possibly with user's selection and termination rules. Otherwise, proceed to Step 3 for factorization.

Step 3 Establish some tolerance (negative upper limit in the usual scheme) for an acceptable reduced cost or "d_j". This should have a larger magnitude than the standard system tolerance but must progressively approach the latter as the end of the phase nears.

Using the current π_o, form R_o by adding* the f_r to π_o as in Section 5. All $R_p=0$ for $p > 0$. (Alternatively, one could use the dual algorithm with two R-forms but this is less practical, particularly if P is large, requiring dual pricing of all subproblems.)

Compute $\bar{\bar{R}}_o$ and proceed to form the $\bar{\bar{R}}_p$ for $p=1,2,\ldots$ and price the corresponding subproblems until an acceptable d_j is found.

*On the first sweep of Step 3 (the second sweep altogether), all $f_r=0$.

[The user may wish to impose priority rules on selection of p. If these are independent of the current solution, the simplest way is to input the subproblems in priority order in the first place.]

Step 4 A column S (say for variable X_s^j) from some block p=s has been selected to enter the solution (enter the basis or change bound). First form

$$B^{-1} S = D_s^{-1} S = \begin{bmatrix} \bar{S}_o \\ \bar{S}_s \end{bmatrix}$$

as in Section 4. (S_p=0 for $p \neq 0$, s) This column should be saved in case it is needed later.

Now compute $\bar{\bar{S}}_o$ and all $\bar{\bar{S}}_p$ for which $\bar{T}_{pp}$ exist. ($\bar{\bar{S}}_s$ will exist in any event.) As each piece is generated, do pivot selection for

$$\beta_p \text{ vs } \bar{\bar{S}}_p \text{ , } p=0,\ldots,s..$$

retaining the subcolumn $\bar{\bar{S}}_p$ for any winning ratio. (In fact, the entire vector $\bar{\bar{S}}$ should be retained.) At the end, some winning ratio

$$\theta_t^r \text{ , row r in block t}$$

or

$$\theta_s \text{ , change of bound for } X_s^j$$

is at hand.

The new solution vector must now be computed:

$$\hat{\beta} = \beta - \theta \bar{\bar{S}}, \text{ over all nonzero } \bar{\bar{S}}_p \quad .$$

If a change of bound occurred, we may return to Step 3 and continue pricing. Otherwise a change of basis must be made in Step 5.

Step 5 Depending on whether s=0, t=0, and s=t, update the entire basis inverse using the appropriate case from Section 8.

Step 6 Return to block 0 and reoptimize it (whether in Phase 1 or Phase 2), updating the basis as required and obtaining a new π_o. Note that this can possibly eliminate some T-columns. Now return to Step 3.

Terminations:

1. Some subproblem is infeasible in Step 1. No feasible solution to model.
2. An unbounded solution is found in Step 2. Entire model is unbounded.
3. No acceptable d_j found in Step 3, even after tolerance is set to system standard,
   - (a) In Phase 1, no feasible solution to model.
   - (b) In Phase 2, current solution is optimal.

4. No θ-value found in Step 4. (Can only happen in Phase 2 unless digital difficulties occur.) An unbounded solution has been found, viz:

$$\beta - \theta \bar{\bar{S}} \quad \text{for any } \theta \geq 0$$

References

[1] Dantzig, George B., and P. Wolfe: Decomposition Principle for Linear Programs, *Operations Res.*, vol. 8, pp. 101-111, 1960.

[2] Rosen, J.B.: Convex Partition Programming, in Robert L. Graves and Philip Wolfe (eds.), "Recent Advances in Mathematical Programming," pp. 159-176, McGraw-Hill Book Company, New York, 1963 (see list of references on p.176).

[3] Beale, E.M.L.: The Simplex Method Using Pseudobasic Variables for Structural Linear-programming Problems, in Robert L. Graves and Philip Wolfe (eds.), "Recent Advances in Mathematical Programming," pp. 133-148, McGraw-Hill Book Company, New York, 1963.

[4] Orchard-Hays, W.: "Advanced Linear-Programming Computing Techniques," McGraw-Hill Book Company, New York, 1968, particularly Chapters 10-12.

[5] Dantzig, George B., and R.M. Van Slyke: Generalized Upper Bounding Techniques for Linear Programming, pts. 1 and 2, *Operations Res. Ctr. Tech. Repts*. ORC, 64-17 and ORC, 64-18, Univ. of California, Berkeley, August 1964, and February 1965.