

A DECOMPOSITION METHOD FOR STRUCTURED  
LINEAR AND NON-LINEAR PROGRAMS

M.D. Grigoriadis & K. Ritter

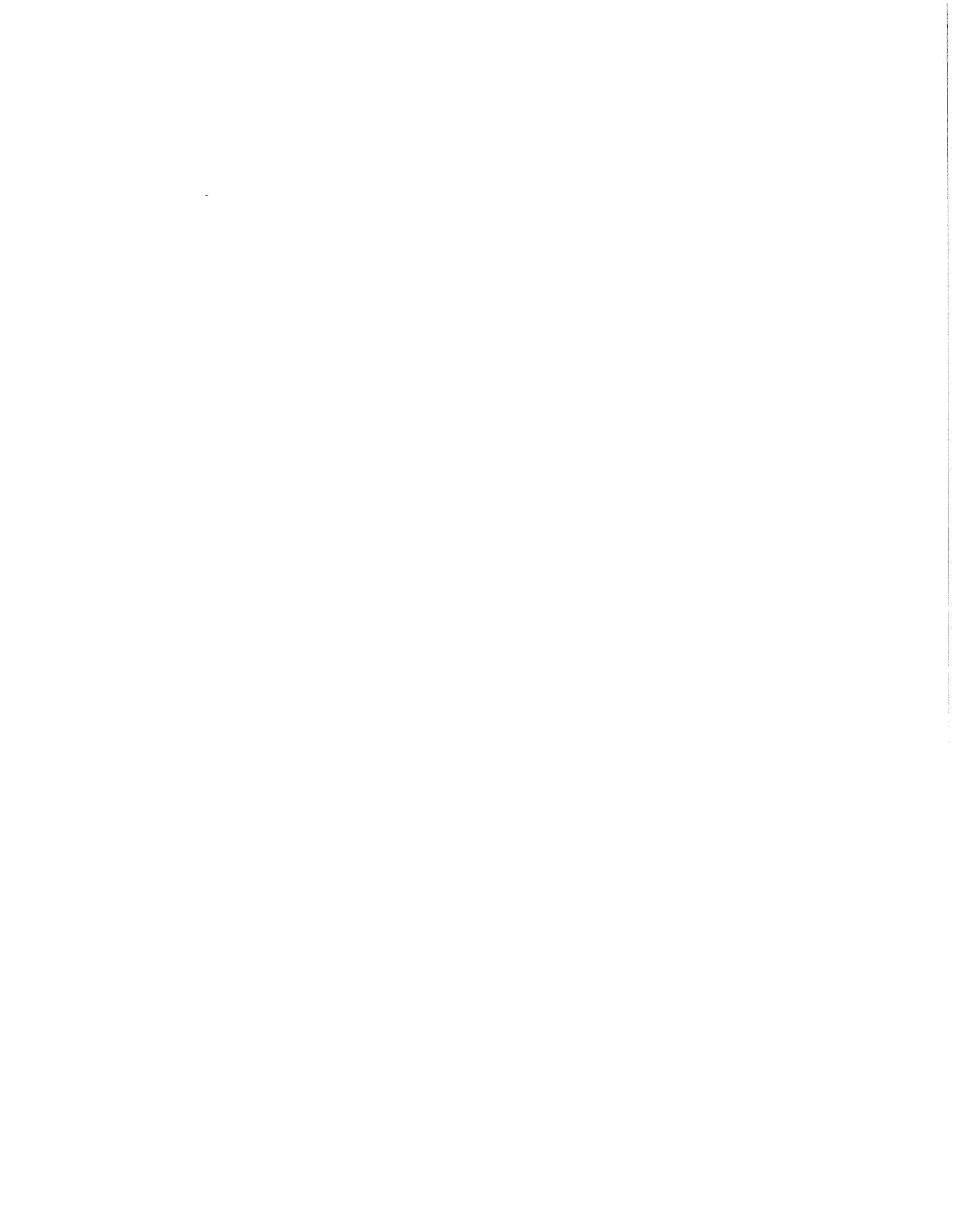
Computer Sciences Technical Report #10

January 1968



## ABSTRACT

A decomposition method for non-linear programming problems with structured linear constraints is described. The structure of the constraint matrix is assumed to be block diagonal with a few coupling constraints and/or variables. The method is further specialized for linear objective functions. An algorithm for performing post optimality analysis - ranging and parametric programming - for such structured linear programs is included. Some computational experience and results for the linear case are presented.



A Decomposition Method for Structured Linear  
and Non-Linear Programs<sup>(†)</sup>

M. D. Grigoriadis<sup>(\*)</sup> and K. Ritter<sup>(\*\*)</sup>

Computer Sciences Department  
University of Wisconsin

§1. Introduction

In practice, large non-linear programming problems with linear constraints, as well as large linear programs, almost always exhibit some structure in their constraint matrix. The most common of these structures is the block diagonal structure with a few coupling constraints and/or variables. To date, various methods for the solution of such large problems with either coupling constraints or coupling variables (both linear and non-linear) and linear, quadratic, separable or general non-linear objective functions have been developed (see e.g. [1, 2, 3, 4, 7, 9]). In [14, 15], Rosen describes partition methods which use the special block diagonal structure of the constraints to reduce the given problem by elimination of variables.

---

<sup>(†)</sup> Sponsored in part by the Mathematics Research Center, United States Army, Madison, Wisconsin, under Contract No.: DA-31-124-ARO-D-462 and in part by the Department of Computer Sciences, University of Wisconsin NSF Research Grant GP6070.

<sup>(\*\*)</sup> and Mathematics Research Center.

<sup>(\*)</sup> and IBM Corporation.

A common assumption in all decomposition or partitioning methods known to the authors is that the constraint matrix represents a "weakly coupled" system: The number of coupling constraints and/or coupling variables is assumed to be much smaller than the corresponding dimension of the problem. Violation of this rather qualitative criterion reportedly has led to poor convergence and other computational irregularities.

The block diagonal structure with a small number of coupling constraints and variables frequently arises in dynamic formulations of multi-plant, multi-commodity production scheduling and distribution models in various industries. This type of a linear model can be converted into the familiar block diagonal structure with only coupling constraints (or only coupling variables) but this conversion results in a drastic increase in the number of such coupling constraints (or variables). Thus, the most desirable property of this inherently weakly coupled system is sacrificed.

This paper describes a decomposition or partitioning method [11] which uses the special structure of the constraints to reduce the given problem through elimination of variables. It may be readily applied to problems having a block diagonal structure with coupling constraints and/or variables. The objective function is assumed to be non-linear, differentiable and concave in all variables. Dual feasibility is maintained throughout the optimization procedure.

The method is further specialized to the case of a linear objective function, first treated by Ritter [10] as a generalization of Rosen's Primal Partition Programming [15]. In addition, an algorithm for performing post-optimality studies for the linear case [12] is offered. This uses the computational tools developed for the linear version of the proposed decomposition algorithm.

In the next section, the non-linear problem is defined and the basic idea of the method, to be detailed in section 3, is summarized. In section 4, the simplifications arising from the linearity of the objective function are discussed. The post-optimality algorithm is given in section 5. The validity of the proposed algorithm is demonstrated in section 6. In the final section some computational aspects and our experience with this algorithm are presented.

## §2. The Non-Linear Problem

We consider the following problem:

Maximize

$$(2.1) \quad F(y, x_1, \dots, x_k)$$

subject to the linear constraints

$$(2.2) \quad \sum_{j=1}^k A_j x_j + D_0 y = b_0$$

$$(2.3) \quad B_j x_j + D_j y = b_j \quad (j=1, \dots, k)$$

$$(2.4) \quad y \geq 0 \quad x_j \geq 0 \quad (j=1, \dots, k),$$

where  $F(y, x_1, \dots, x_k)$  is a differentiable and concave function,

$A_j$  ( $j=1, \dots, k$ ) is an  $(m_0, n_j)$  - matrix,  $B_j$  ( $j=1, \dots, k$ ) is an  $(m_j, n_j)$  - matrix,  $D_j$  ( $j=0, 1, \dots, k$ ) is an  $(m_j, n_0)$  - matrix, while  $x_j$  and  $c_j$  are  $n_j$  - vectors,  $y$  is a  $n_0$ -vector and  $b_j$  ( $j=0, 1, \dots, k$ ) is an  $m_j$  - vector.

The feasible domain defined by (2.2) - (2.4) is assumed to be bounded. This problem will be referred to as the Primal Problem ( $\tilde{P}$ ). The corresponding dual is:

Minimize

$$(2.5) \quad F(y, x_1, \dots, x_k) - u_0' (\sum_{j=1}^k A_j x_j + D_0 y - b_0) \\ - \sum_{j=1}^k u_j' (B_j x_j + D_j y - b_j) + w_0' y + \sum_{j=1}^k w_j' x_j$$

subject to the constraints

$$(2.6) \quad \sum_{j=1}^k D_j' u_j + D_0' u_0 - w_0 - \nabla_y F(y, x_1, \dots, x_k) = 0$$

$$(2.7) \quad B_j' u_j + A_j' u_0 - w_j - \nabla_{x_j} F(y, x_1, \dots, x_k) = 0 \quad (j=1, \dots, k)$$

$$(2.8) \quad w_j \geq 0 \quad (j=0, 1, \dots, k),$$

where the  $u_j$  ( $j=0, 1, \dots, k$ ) and  $w_j$  ( $j=0, 1, \dots, k$ ) represent the dual variables or Lagrange multipliers and are  $m_j$  and  $n_j$  -vectors respectively;  $\nabla_y F$  is a  $n_0$ -vector corresponding to the portion of  $\nabla F$  which consists of the partial derivatives of  $F$  with respect to the components of  $y$  only,



and  $\nabla_{x_j} F$  are  $n_j$  - vectors corresponding to the portions of  $\nabla F$  which consist of partial derivatives of  $F$  with respect to the components of  $x_j$  only.

A more convenient form of the dual problem, which will be referred to as  $\tilde{D}$ , may be obtained by eliminating the variables  $w_j$  ( $j=0, 1, \dots, k$ ) from (2.5) - (2.7) and using (2.8). This is given by:

Minimize

$$(2.9) \quad F(y, x_1, \dots, x_k) + \sum_{j=0}^k u_j' b_j - (y', x_1', \dots, x_k'). \quad \nabla F(y, x_1, \dots, x_k)$$

subject to

$$(2.10) \quad \sum_{j=1}^k D_j' u_j + D_0' u_0 - \nabla_y F(y, x_1, \dots, x_k) \geq 0$$

$$(2.11) \quad B_j' u_j + A_j' u_0 - \nabla_{x_j} F(y, x_1, \dots, x_k) \geq 0 \quad (j=1, \dots, k)$$

The decomposition method described in this paper is mainly based on the following observation. If  $\tilde{P}$  has an optimal solution, then the variables in this solution have non-negative values. Since we have a total of

$$n = \sum_{j=0}^k n_j$$

variables, we would expect to have, at most, as many active constraints in  $\tilde{P}$ . However, the

$$m = \sum_{j=0}^k m_j$$

equality constraints (2.2) - (2.3) are always active. Therefore, provided that (2.2) - (2.3) are linearly independent,<sup>1)</sup> at most  $(n-m)$  of the non-negativity constraints (2.4) are active. The remaining  $m$  non-negativity restrictions, which are inactive, may be canceled with no effect on the optimal solution of  $\tilde{P}$ .

A further simplification may be effected by using the special structure of the constraints (2.3) to eliminate at least  $(m-m_0)$  of the variables which are not restricted in sign. This elimination procedure reduces the maximization problem  $\tilde{P}$  to a concave programming problem with at most

$$s = n - (m - m_0)$$

variables, all of which are restricted to be non-negative, and  $m_0$  linear equality constraints. This problem will be referred to as the Modified Primal Problem ( $\tilde{M}$ ), and may be regarded as analogous to the "master problem" in Dantzig-Wolfe decomposition [1] or the "Problem II" in Rosen's Primal Partition Programming [15].

Clearly, if the set of non-negativity restrictions (2.4) active in the optimal solution to  $\tilde{P}$  were known in advance, then the solution of  $\tilde{M}$  would provide the optimal solution to  $\tilde{P}$ . Generally, however, it is unlikely that one might predict the optimal basic variable set or equivalently the non-negativity restrictions which would be active in the optimal solution to  $\tilde{P}$ . To circumvent this difficulty, we begin by ignoring the non-negativity restrictions for an arbitrary set  $S_1$  of at least  $(m - m_0)$  variables chosen

among the  $x_j$ . In this case, the optimal solution to  $\tilde{M}$  need not be feasible for  $\tilde{P}$  since some of the eliminated variables may take on negative values. If it is feasible, however, then it is also an optimal solution to  $\tilde{P}$  (Theorem 1).

If some variables have negative values, we determine a new set  $S_2$  of at least  $(m - m_0)$  variables and repeat the procedure. It can be shown (Lemma 1 and 2) that corresponding to the sequence of optimal solutions to the modified primal problems  $(\tilde{M})$ , there is a sequence of solutions to  $\tilde{D}$  which give non-increasing values of the dual objective function. From this fact it follows (Theorem 2) that after a finite number of steps, we obtain a modified maximization problem  $(\tilde{M})$  which has the same optimal solution as  $\tilde{P}$ .

### §3. The Algorithm

We assume that each of the matrices  $B_j$  contains a non-singular square matrix of order  $m_j$ . This is no loss of generality since if  $B_j$  does not contain such a matrix, we can add suitable unit vectors and artificial variables having sufficiently large negative entries in the objective function. Then, provided that the original problem  $(P)$  has a feasible solution, the optimal solution to this enlarged problem is identical to that for  $\tilde{P}$ .

Let  $B_{j1}$  be an  $m_j$ -order non-singular square submatrix of  $B_j$ . We denote the matrix formed by the remaining columns by  $B_{j2}$  and partition  $A_j$ ,  $x_j$  and  $c_j$  accordingly into  $A_{j1}$ ,  $A_{j2}$ ,  $x_{j1}$ ,  $x_{j2}$  and  $c_{j1}$ ,  $c_{j2}$  respectively.

Then, the constraints (2.3) can be written as:

$$B_{j1} x_{j1} + B_{j2} x_{j2} = b_j - D_j y$$

or

$$(3.1) \quad x_{j1} = B_{j1}^{-1} b_j - B_{j1}^{-1} B_{j2} x_{j2} - B_{j1}^{-1} D_j y$$

Substituting (3.1) into (2.1) - (2.2) we can eliminate the vectors  $x_{j1}$  ( $j=1, \dots, k$ ) and obtain the "Modified Primal Problem" ( $\tilde{M}$ ) as:

Maximize

$$(3.2) \quad G(y, x_{12}, x_{22}, \dots, x_{k2})$$

subject to the linear constraints

$$(3.3) \quad \sum_{j=1}^k M_j x_{j2} + M_0 y = b$$

$$x_{j2} \geq 0, \quad y \geq 0$$

where the function  $G(y, x_{12}, x_{22}, \dots, x_{k2})$  is concave and differentiable since it is obtained from the function  $F(y, (x_{11}, x_{12}), \dots, (x_{k1}, x_{k2}))$  by the linear transformation (3.1); and

$$(3.4) \quad b = b_0 - \sum_{j=1}^k A_{j1} B_{j1}^{-1} b_j$$

$$M_0 = D_0 - \sum_{j=1}^k A_{j1} B_{j1}^{-1} D_j$$

$$M_j = A_{j2} - A_{j1} B_{j1}^{-1} B_{j2}$$

If  $\tilde{M}$  has no feasible solution, then the original problem  $\tilde{P}$  has no feasible solution since it contains all constraints (3.3) of  $\tilde{M}$ . In the

following, we assume that  $\tilde{P}$  has a feasible solution and  $\tilde{M}$  attains an optimal solution for a finite point  $(y, x_{12}, \dots, x_{k2})$ . (If not, the precautionary procedure outlined in Section 6, Remark 2, may be used).

We note that  $\tilde{M}$  is a concave maximization problem considerably smaller than the original problem  $\tilde{P}$ , with at most  $s$  variables and  $m_0$  linear equality constraints (in addition to the non-negativity restrictions). Efficient and computationally successful methods for the solution of non-linear programming problems subject to linear constraints developed by Rosen [13], Frank and Wolfe [5], and others, may be used. The solution of  $\tilde{M}$ , theoretically, may not be a finite procedure.

Let  $(y^*, x_{j2}^*)$  ( $j = 1, \dots, k$ ) be an optimal solution to  $\tilde{M}$ .

Substituting this solution into (3.1) we obtain

$$(3.5) \quad x_{j1}^* = B_{j1}^{-1} b_j - B_{j1}^{-1} B_{j2} x_{j2}^* - B_{j2}^{-1} D_j y^*$$

Now we apply the following optimality criterion (Theorem 1):

If  $x_{j1}^* \geq 0$  ( $j = 1, \dots, k$ ), then  $(y^*, x_{j1}^*, x_{j2}^*)$  ( $j = 1, \dots, k$ ) is an optimal solution to the original problem  $\tilde{P}$ .

Suppose  $x_{j1}^*$  ( $j = 1, \dots, k_1 \leq k$ ) has at least one negative component. We construct a new problem  $\tilde{M}$ , of the form (3.2) - (3.3), in such a way that in the resulting solution  $(y^{**}, x_{j1}^{**}, x_{j2}^{**})$  one of the components of  $x_j$  ( $j = 1, \dots, k_1$ ) which was negative in  $(y^*, x_{j1}^*, x_{j2}^*)$  is forced to be non-negative. This procedure will now be outlined for a general cycle.

A) Let  $(x_{j1}^*)_1, (x_{j1}^*)_2, \dots, (x_{j1}^*)_l$  be the negative components of  $x_{j1}^*$ . Denote the first  $l$  rows of the matrix  $B_{j1}^{-1} B_{j2}$  by  $g'_{j1}, \dots, g'_{jl}$ . Furthermore, suppose that  $x_{j2}^*$  has  $q$  positive components, say the first  $q$  components. Then, for each  $j \leq k_1$ , consider the following two cases:

I) At least one of the components  $(g_{ji})_v$  ( $i = 1, \dots, l; v = 1, \dots, q$ ) is non-zero.

II)  $(g_{ji})_v = 0$  for  $i=1, \dots, l; v = 1, \dots, q$ .

In case I, let  $(g_{ji})_v \neq 0$ . Denote the  $v^{\text{th}}$  column of  $B_{j2}$  by  $h_{jv}$ . If the  $i^{\text{th}}$  column of  $B_{j1}$  is replaced by  $h_{jv}$ , the new matrix  $B_{j1}^*$  is nonsingular since  $(B_{j1}^{-1} h_{jv})_i = (g_{ji})_v \neq 0$  implies that the columns of  $B_{j1}^*$  are linearly independent. Thus, replace  $B_{j1}$  by  $B_{j1}^*$  for any  $j$  for which  $x_{j1}^*$  has negative components and for which case I holds. Then, the procedure (3.4) which leads to the construction of  $\tilde{M}$  is applied using the new matrices  $B_{j1}^{*-1}$ . It should be noted that those  $M_j$  for which  $x_{j1}^* \geq 0$ , are not altered and need not be recomputed.

In case II, let  $(x_{j1}^*)_v < 0$ . Denote the  $v^{\text{th}}$  component of  $B_{j1}^{-1} B_{j2}$  and  $B_{j1}^{-1} D_j$  by  $g'_{jv}$  and  $e'_{jv}$  respectively, and the  $v^{\text{th}}$  component of  $B_{j1}^{-1} b_j$  by  $\beta_{jv}$ . Then, add the condition:

$$(3.6) \quad -(x_{j1})_v = g'_{jv} x_{j2} + e'_{jv} y - \beta_{jv} \leq 0$$

to the constraints (3.3) of  $\tilde{M}$  after all changes dictated by case I have been implemented.

Finally, the new  $\tilde{M}$  problem ((3.2) - (3.3)) is solved resulting in  $(y^{**}, x_{j2}^{**})$  as its optimal solution. The corresponding  $x_{j1}^{**}$  ( $j = 1, \dots, k$ ) is obtained by inserting this solution into (3.1).

By Theorem 1,  $(y^{**}, x_{j1}^{**}, x_{j2}^{**})$  is an optimal solution to  $\tilde{P}$  if all components of  $x_{j1}^{**} \geq 0$  ( $j = 1, \dots, k$ ).

B) If at least one of the vectors  $x_{j1}^{**}$ , has negative components, the "additional constraints" of the form (3.6) are treated as follows:

Case 1) If  $g'_{jv} x_{j2}^{**} + e'_{jv} y^{**} - \beta_{jv} < 0$  then this constraint is canceled.

Case 2) If  $g'_{jv} x_{j2}^{**} + e'_{jv} y^{**} - \beta_{jv} = 0$  and  $(g_{jv})_{\mu} (x_{j2}^{**})_{\mu} \neq 0$ , then the constraint is canceled and the  $v^{\text{th}}$  column of  $B_{j1}$  is replaced by the  $\mu^{\text{th}}$  column  $h_{j\mu}$  of  $B_{j2}$ . The resulting matrix  $B_{j1}^{**}$  is non-singular since  $(B_{j1}^{-1} h_{j\mu})_v = (g_{jv})_{\mu} \neq 0$  implies that the columns of  $B_{j1}^{**}$  are linearly independent.<sup>3)</sup>

Case 3) If  $g'_{jv} x_{j2}^{**} + e'_{jv} y^{**} - \beta_{jv} = 0$  and  $(g_{jv})_{\mu} (x_{j2}^{**})_{\mu} = 0$  for all  $\mu$ , then this constraint is left unaltered in  $\tilde{M}$ .

Since in this case

$$g'_{jv} x_{j2}^{**} + e'_{jv} y^{**} = e'_{jv} y^{**} = \beta_{jv}$$

$\tilde{M}$  may contain, except for degenerate cases, at most  $n_0$  constraints of the form (3.6) at the conclusion of any cycle.

The presence of linear dependence among the rows of the original constraint matrix (2.2) - (2.3) may cause a slight increase in the number of constraints of  $\tilde{M}$ .

The modification of the "additional constraints" outlined above, completes a decomposition "cycle". We let  $x_{j1}^* = x_{j1}^{**}$  and start the next cycle at A.

Since in each cycle at most  $k$  "additional constraints" are appended to  $\underline{M}$ , it follows from Case 3, that, disregarding degeneracy,  $\underline{M}$  may contain at most  $(n_0 + k)$  "additional constraints" at any cycle.

By Theorem 2, an optimal solution to  $\underline{P}$  is obtained after a finite number of cycles.

Remark:

The above procedure yields the optimal solution to  $\underline{P}$  after a finite number of decomposition cycles, even when only one of the variables negative in the  $t^{\text{th}}$  cycle is forced to be non-negative at the  $(t + 1)^{\text{th}}$  cycle. Consequently, it would suffice to append at most one "additional constraint" of the form (3.6) at each cycle. Then, the number of additional constraints involved in any single cycle would reduce to at most  $(n_0 + 1)$ .



#### §4. The Linear Case -- Simplifications

The linear case is characterized by a linear objective function in (2.1), i.e.

$$(4.1) \quad F(y, x_1, \dots, x_k) = c'_0 y + \sum_{j=1}^k c'_j x_j$$

resulting in the Linear Primal problem (LP) and leading to the following formulation of the dual problem (LD) corresponding to (2.5) - (2.8):

Minimize

$$(4.2) \quad \sum_{j=0}^k b'_j u_j$$

subject to

$$(4.3) \quad \sum_{j=1}^k D'_j u_j + D'_0 u_0 - w_0 = c_0$$

$$(4.4) \quad \sum_{j=1}^k B'_j u_j + A'_j u_0 - w_j = c_j \quad (j = 1, \dots, k)$$

$$(4.5) \quad w_j \geq 0 \quad (j = 1, \dots, k)$$

For the sake of conformity with the previous section we state the linear version of the dual program corresponding to (2.9) - (2.11):

Minimize

$$(4.6) \quad \sum_{j=0}^k b'_j u_j$$

subject to

$$(4.7) \quad \sum_{j=1}^k D'_j u_j + D'_0 u_0 \geq c_0$$

$$(4.8) \quad B'_j u_j + A'_j u_0 \geq c_j \quad (j = 1, \dots, k)$$

The obvious, and most important, simplification resulting from our assumption of a linear objective function is the linearity of  $\underline{M}$ , i.e.:

Maximize

$$(4.9) \quad \alpha + \sum_{j=1}^k d'_j x_{j2} + d'_0 y$$

subject to

$$(4.10) \quad \sum_{j=1}^k M_j x_{j2} + M_0 y = b$$

$$(4.11) \quad y \geq 0, \quad x_{j2} \geq 0 \quad (j = 1, \dots, k)$$

where  $M_0$ ,  $M_j$  ( $j = 1, \dots, k$ ) and  $b$  are given by (3.4) and

$$(4.12) \quad \alpha = \sum_{j=1}^k c'_{j1} B_{j1}^{-1} b_j$$

$$(4.13) \quad d_0 = c_0 - \sum_{j=1}^k (B_{j1}^{-1} D_j)' c_{j1}; \quad d_j = c_{j2} - (B_{j1}^{-1} B_{j2})' c_{j1}.$$

We note that the above linear version of  $\underline{M}$ , which will be referred to as  $\underline{LM}$ , is an ordinary linear programming problem with  $m_0$  equality constraints and  $s$  variables. It may be solved using any of the commercially available linear programming codes. Although the maximum number of constraints in  $\underline{LM}$  may differ from one cycle to the next, the last remark in section 3 suggests that a constant size of at least  $(n_0 + 1)$ , and not more than  $(n_0 + k)$ , rows may be selected in advance and used for all cycles. This will facilitate the use of an existing linear programming code for solving  $\underline{LM}$ .

§5. Post-Optimality Analysis -- The Linear Case

We consider the following parametric form of LP:

Maximize

$$(5.1) \quad \sum_{j=1}^k (c_j + \lambda f_j)' x_j + (c_0 + \lambda f_0)' y$$

subject to the constraints

$$(5.2) \quad \sum_{j=1}^k A_j x_j + D_0 y = b_0 + \lambda e_0$$

$$(5.3) \quad B_j x_j + D_j y = b_j + \lambda e_j \quad (j = 1, \dots, k)$$

$$(5.4) \quad y \geq 0; \quad x_j \geq 0 \quad (j = 1, \dots, k)$$

where  $f_j$  and  $e_j$  ( $j = 0, 1, \dots, k$ ) are given  $n_j$  and  $m_j$  - vectors respectively and  $\lambda$  is a parameter in a specified range  $\lambda_l \leq \lambda \leq \lambda_u$ .

This problem will be referred to as the Parametric Linear Primal problem

(PLP).

Considering the partitioning introduced in section 3, we may write a relation analogous to (3.1) as:

$$(5.5) \quad x_{j1} = B_{j1}^{-1} (b_j + \lambda e_j) - B_{j2}^{-1} x_{j2} - B_{j1}^{-1} D_j y.$$

whence, by substitution into (5.1) and (5.2) we may state the Parametric

Linear Modified primal problem (PLM) as:

Maximize

$$\alpha(\lambda) + \sum_{j=1}^k (d_j^1 + \lambda d_j^2)' x_{j2} + (d_0^1 + \lambda d_0^2)' y$$

subject to the constraints

$$\sum_{j=1}^k M_j x_{j2} + M_0 y = b + \lambda e$$

$$x_{j2} \geq 0, \quad y \geq 0$$

where

$$\alpha(\lambda) = \sum_{j=1}^k (c_{j1} + \lambda f_{j1})' B_{j1}^{-1} (b_j + \lambda e_j)$$

$$b + \lambda e = (b_0 + \lambda e_0) - \sum_{j=1}^k A_{j1} B_{j1}^{-1} (b_j + \lambda e_j)$$

$$d_0^1 + \lambda d_0^2 = (c_0 + \lambda f_0) - \sum_{j=1}^k (B_{j1}^{-1} D_j)' (c_{j1} + \lambda f_{j1})$$

$$d_0^1 + \lambda d_j^2 = (c_{j2} + \lambda f_{j2}) - (B_{j1}^{-1} B_{j2})' (c_{j1} + \lambda f_{j1})$$

Clearly, all properties of LM are shared by PLM and the two problems are equivalent for  $\lambda = 0$ .

We assume that an optimal solution for  $\lambda = \lambda_0$  has already been obtained by the decomposition method outlined in the previous sections.

The questions of post-optimality analysis to be examined here, are:

I) For which values of the parameter  $\lambda$ ,  $\lambda_l \leq \lambda \leq \lambda_u$ ,  $\lambda \neq \lambda_0$  does the current solution remain feasible and optimal? This question is commonly referred to as "ranging information" on the current optimal solution.

II) For a given change in the value of the parameter  $\lambda$ , for which the current basic solution for  $\lambda = \lambda_0$  is no longer feasible and/or optimal, what is the new feasible and optimal basic solution? This is usually

referred to as "parametric programming".

Each of the above questions is coincident to a set of two problems: that of feasibility and that of optimality. The problem of feasibility is associated with the right hand side of the constraints, hence the usual terms "right hand side ranging" and "right hand side parametrization". Similarly, optimality is associated with the cost row, hence the terms "cost ranging" and "cost parametrization". The parametric programming algorithm to be described in this section provides solutions to these questions by using, and expanding on, information available from the current optimal solution for  $\lambda = \lambda_0$ .

The ranging information is obtained by the well known ratios. Thus, we obtain one interval for preserving feasibility in the last PLM and another for the nonnegativity of  $x_{j1}$  ( $j = 1, \dots, k$ ). Clearly, overall feasibility is maintained for those values of the parameter in the intersection of these two intervals. The interval for optimality is obtained by simply applying the optimality criterion of the simplex method to the last PLM.

The parametrization algorithm provides mechanisms for altering the existing optimal solution, so that it remains feasible and optimal when the value of the parameter falls outside one of the computed "ranges". Therefore,

- 1) If feasibility in the last PLM is violated, a basis change is performed using the dual simplex method.
- 2) If the non-negativity of at least one variable  $(x_{j1})_v$  is violated for at least one  $j$ , either the current partitioning is altered by exchanging

$(x_{j1})_v$  by a non-basic variable  $(x_{j2})_u$ , or the violated non-negativity constraint expressed in terms of  $x_{j2}$  and  $y$  is appended to the last PLM.

3) If the optimality condition is violated, a basic change is made in the last PLM using the primal simplex method, provided that the resulting levels of the  $x_{j1}$  ( $j = 1, \dots, k$ ) variables are non-negative. If not, their non-negativity is secured by following (2) above.

#### The Algorithm

Let  $(y(\lambda), x_{j1}(\lambda), x_{j2}(\lambda))$  ( $j = 1, \dots, k$ ) be the optimal solution to PLP for  $\lambda = \lambda_0$  and  $(y(\lambda), x_{j2}(\lambda))$  ( $j = 1, \dots, k$ ) be the solution to the last PLM whose definition and data are also available. The latter is assumed to have  $q$  rows,  $m_0 + p + 1 \leq q \leq m_0 + p + k$ , and  $s$  variables.

We now examine the post-optimality question (I), i.e.

I) For which values of  $\lambda$ ;  $\lambda_l \leq \lambda \leq \lambda_u$ ,  $\lambda \neq \lambda_0$  does  $(y(\lambda), x_{j1}(\lambda), x_{j2}(\lambda))$  satisfy the conditions for:

- 1) feasibility
- 2) optimality
- 3) both feasibility and optimality.

For (1), we wish to determine the largest interval for which overall feasibility is maintained. We distinguish two cases.

a) Feasibility condition on the last PLM:

This is a necessary condition for feasibility in the original problem PLP. The largest interval for which this condition is satisfied,

denoted by  $[\lambda_{\ell 1}^f, \lambda_{ul}^f]$ , is obtained by considering the right hand side vector of the last PLM (i.e. for  $\lambda = \lambda_0$ ) updated by the inverse of its optimal basis  $M_B^{-1}$ , i.e. we consider

$$(g^1 + \lambda g^2) = M_B^{-1} (b + \lambda e)$$

and apply the non-negativity condition

$$(5.6) \quad g^1 + \lambda g^2 \geq 0$$

which gives

i) For  $\lambda > \lambda_0$

$$(5.7) \quad \lambda_{ul}^f = \min \left\{ -\frac{(g^1)_\nu}{(g^2)_\nu} \mid (g^2)_\nu < 0; \nu = 1, \dots, q \right\}$$

or  $\lambda_{ul}^f = +\infty$  if  $(g^2)_\nu \geq 0$  for  $\nu = 1, \dots, q$ .

ii) For  $\lambda < \lambda_0$ :

$$(5.8) \quad \lambda_{\ell 1}^f = \max \left\{ \frac{(g^1)_\nu}{(g^2)_\nu} \mid (g^2)_\nu > 0; \nu = 1, \dots, q \right\}$$

or  $\lambda_{\ell 1}^f = -\infty$  if  $(g^2)_\nu \leq 0$  for  $\nu = 1, \dots, q$ .

b) Feasibility condition on the  $x_{j1}$  variables:

In addition to (a) above, it is necessary to satisfy the non-negativity restrictions on the variables  $x_{j1}(\lambda)$  whose levels, denoted by  $g^1 + \lambda g^2$ , are established by substituting the optimal levels of the basic  $x_{j2}(\lambda)$  and  $y(\lambda)$  into (5.5) which gives the optimal  $x_{j1}(\lambda)$  as linear functions of  $\lambda$ , denoted by  $(g_j^1 + \lambda g_j^2)$  ( $j = 1, \dots, k$ ). The largest interval  $[\lambda_{\ell 2}^f, \lambda_{u 2}^f]$  for which non-negativity of the  $x_{j1}(\lambda)$  is maintained is obtained by:

$$(5.9) \quad \begin{aligned} & \text{(i) For } \lambda > \lambda_0 : \\ & \lambda_{u 2}^f = \min \left\{ - \frac{(g_j^1)_\nu}{(g_j^2)_\nu} \mid (g_j^2)_\nu < 0 ; \nu = 1, \dots, m_j ; j = 1, \dots, k \right\} \end{aligned}$$

or  $\lambda_{u 2}^f = \infty$  if  $(g_j^2)_\nu \geq 0$  for all  $\nu$  and  $j$ .

$$(5.10) \quad \begin{aligned} & \text{(ii) For } \lambda < \lambda_0 \\ & \lambda_{\ell 2}^f = \max \left\{ \frac{(g_j^1)_\nu}{(g_j^2)_\nu} \mid (g_j^2)_\nu > 0 ; \nu = 1, \dots, m_j ; j = 1, \dots, k \right\} \end{aligned}$$

or  $\lambda_{\ell 2}^f = -\infty$  if  $(g_j^2)_\nu \leq 0$  for all  $\nu$  and  $j$ .

For overall feasibility we must therefore have

$$\lambda \in [\lambda_{\ell 1}^f, \lambda_{u 1}^f] \cap [\lambda_{\ell 2}^f, \lambda_{u 2}^f].$$

Now for (2), we wish to determine the largest interval  $[\lambda_{\ell 1}^e, \lambda_{u 1}^e]$  for which the current solution remains optimal. This is easily accomplished by considering the cost row of the last PLM (updated by the inverse of its optimal basis), denoted by  $(h_j^1 + \lambda h_j^2)$  ( $j = 0, 1, \dots, k$ ). Thus, for optimality we must have, for the  $x_{j2}(\lambda)$  variables:



$$(5.11) \quad h_j^1 + \lambda h_j^2 \leq 0 \quad (j = 1, \dots, k)$$

and for the  $y(\lambda)$  variables:

$$(5.12) \quad h_0^1 + \lambda h_0^2 \leq 0.$$

The sought interval is immediately established by:

i) For  $\lambda > \lambda_0$  :

$$(5.13) \quad \lambda_{ul}^e = \min \left\{ \frac{(h_j^1)_{\nu_j}}{(h_j^2)_{\nu_j}} \mid (h_j^2)_{\nu_j} > 0 ; \text{all } ^3) \nu_j ; j = 0, 1, \dots, k \right\}$$

or  $\lambda_{ul}^e = +\infty$  if  $(h_j^2)_{\nu_j} \leq 0$  for all  $^4) \nu_j ; j = 0, 1, \dots, k$ .

ii) For  $\lambda < \lambda_0$

$$(5.14) \quad \lambda_{\ell 1}^e = \max \left\{ \frac{(h_j^1)_{\nu_j}}{(h_j^2)_{\nu_j}} \mid (h_j^2)_{\nu_j} < 0 ; \text{all } ^3) \nu_j ; j = 0, 1, \dots, k \right\}$$

or  $\lambda_{\ell 1}^e = -\infty$  if  $(h_j^2)_{\nu_j} \geq 0$  for all  $^4) \nu_j ; j = 0, 1, \dots, k$ .

Finally, (3) is obtained as an obvious consequence of (1) and (2). That is, the solution will remain both feasible and optimal for  $\lambda \in [\lambda_*, \lambda^*]$  where

$$(5.15) \quad \begin{aligned} \lambda_* &= \max \{ \lambda_{\ell 1}^e, \lambda_{\ell 1}^f, \lambda_{\ell 2}^f, \lambda_{\ell} \} \\ \lambda^* &= \min \{ \lambda_{ul}^e, \lambda_{ul}^f, \lambda_{u2}^f, \lambda_u \} . \end{aligned}$$

If  $\lambda_* = \lambda_{\ell}$  and  $\lambda^* = \lambda_u$  then the post-optimality question I has been answered and question II is clearly not relevant. We must assume, therefore, that either  $\lambda_{\ell} < \lambda_*$  and/or  $\lambda^* < \lambda_u$ . In the ensuing discussion we consider only the case  $\lambda^* < \lambda_u$ . This is no loss of generality since the case  $\lambda < \lambda_*$  leads to

entirely symmetric results.

The post-optimality question II, i. e. "parametric programming", is stated as:

II) Utilizing the information available from the current optimal solution for  $\lambda = \lambda^*$ , obtain the optimal solution to PLP for  $\lambda = \lambda^* + \varepsilon$ ;  $\varepsilon > 0$ .

We consider three cases:

- 1)  $\lambda^* = \lambda_{u1}^f$ , i. e. for  $\lambda = \lambda^* + \varepsilon$ ;  $\varepsilon > 0$  the current optimal solution to the last PLM does not satisfy the feasibility condition of PLM.
- 2)  $\lambda^* = \lambda_{u2}^f$ ; i. e. for  $\lambda = \lambda^* + \varepsilon$ ;  $\varepsilon > 0$  the current optimal solution does not satisfy the non-negativity condition on the  $x_{j1}(\lambda)$ .
- 3)  $\lambda^* = \lambda_{u1}^e$ , i. e. for  $\lambda = \lambda^* + \varepsilon$ ;  $\varepsilon > 0$  the current optimal solution does not satisfy the optimality conditions.

For (1), we would like to effect a basis change in the last PLM optimal basic solution such that the feasibility condition (5.6) will be restored with respect to the new basis. This is easily accomplished by considering the optimal levels of the basic  $y(\lambda)$  and  $x_{j1}(\lambda)$  variables (earlier denoted as  $g^1 + \lambda g^2$ ) for  $\lambda = \lambda_{u1}^f$ . Due to (5.7) we must have  $(g^1 + \lambda g^2)_\nu = 0$  for at least one component  $\nu$ . For  $\lambda = \lambda_{u1}^f + \varepsilon$ , therefore, we have a basic optimal but infeasible solution. The customary rules of the dual simplex method (see e. g. p. 247 in [6]) are applied and the  $\nu^{\text{th}}$  variable is exchanged with one of the non-basic  $x_{j1}$  and  $y$  variables which enters the basis at zero level. This requires exactly one pivot step when  $\nu$  is unique. Several pivot steps may be necessary to obtain feasibility if  $\nu$  is not unique. If no non-basic variable is eligible to enter, i. e. row  $\nu$  of the last PLM simplex tableau has no negative entries, we conclude that there is no feasible solution to PLM, further implying that no such solution to PLP exists.

For (2) we assume that for  $\lambda = \lambda_{u2}^f$  we obtain from (5.5)  $x_{j1} \geq 0$  with  $(x_{j1})_{\mu} = 0$  for at least one  $j$  and  $\mu$ . For  $\lambda = \lambda_{u2}^f + \varepsilon$ ;  $\varepsilon > 0$  we wish to restore the non-negativity of the  $x_{j1}(\lambda)$ . This may be accomplished by an exchange between the  $x_{j1}(\lambda)$  and  $x_{j2}(\lambda)$  variables; that is by updating the current partitioning of the problem, or by appending an additional constraint to PLM. For a fixed value of  $\lambda = \lambda_{u2}^f$  and the corresponding optimal solution to the last PLM, (3.1) gives the form:

$$x_{j1} = p_j - P_j x_{j2} \quad (j = 1, \dots, k)$$

where

$$p_j = B_{j1}^{-1} (b_j + \lambda e_j) - B_{j1}^{-1} D_j y(\lambda)$$

$$P_j = B_{j2}^{-1} B_{j2}$$

At this point we treat two cases:

i) If there exists a column index  $\nu$  such that

$$(P_j)_{\mu\nu} (x_{j2})_{\nu} \neq 0$$

where  $(P_j)_{\mu\nu}$  is the element in position  $(\mu, \nu)$  of  $P_j$ , then the variables  $(x_{j1})_{\mu}$  and  $(x_{j2})_{\nu}$  are exchanged. The current partitioning is updated to reflect this exchange, a revised PLM is defined and solved to optimality. The computational effort required to solve this revised PLM may be drastically reduced by attempting to use as a starting basic set those column indices which were in the optimal basic set of the previous PLM.

ii) If  $(P_j)_{\mu\nu} (x_{j2})_\nu = 0$  for all  $\nu$ , the above exchange is not possible. Nevertheless, we can secure the non-negativity of  $(x_{j1})_\mu$  by generating and appending an "additional constraint" of the form (3.6) expressing this restriction in terms of the  $x_{j2}$  and  $y$ . An optimal solution to this enlarged PLM is then obtained by revising the optimal solution to the current PLM by the well known rules (see e.g. pp. 384-385 in [6]).

The above two cases lead to the consideration of a solution strategy whereby one may keep applying (ii) until either the number of additional constraints becomes excessive, or case (i) is possible. That is (i) may be used at will, whenever possible, to reduce the size of the PLM by eliminating all of the accumulated "additional constraints". The number of "additional constraints" may also be reduced while applying exclusively case (ii), by omitting such constraints as soon as they become inactive. If they become active at later stages the appropriate "additional constraint" will be generated (case (ii)), or alternately the non-negativity of the corresponding  $x_{j1}$  variable will be guaranteed by a revised partitioning (case (i)).

Finally, for (3), we would like to effect a basis change such that the optimality conditions (5.11) - (5.12) are restored. For  $\lambda = \lambda_{u1}^e$ , (3.13) guarantees the existence of at least one non-basic variable  $(z)_\nu$ , among the components of  $x_{j2}(\lambda)$  or  $y(\lambda)$ , with a reduced cost of zero. The rules of the simplex method could be applied to introduce  $(z)_\nu$  into the basis. However, due to the provisional nature of this pivot step (see case b below), we first examine the effect of such a step on the current levels  $x_{j1}^b$  of the variables  $x_{j1}$ .

For restoring feasibility in PLM, we must have, at the conclusion of the pivot step:

$$z_B^a = z_B^b - p_\nu^b (z_\nu^b)_\nu \geq 0$$

where  $z_B^b$  and  $z_B^a$  are the basic optimal solution vectors before and after the pivot step respectively and  $p_\nu^b$  is the column corresponding to the non-basic variable  $(z_\nu^b)_\nu$  in the current simplex tableau of PLM. The effect of the pivot step would thus be to increase the level of  $(z)_\nu$  from zero to:

$$(z^a)_\nu = \min_i \left\{ \frac{(z_B^b)_i}{(p_\nu^b)_i} \mid (p_\nu^b)_i > 0; i = 1, \dots, g \right\}$$

and  $(z_B^a)_\rho = 0$  for at least one component  $\rho$ . If, upon substituting  $z_B^a$  into (5.5), the resulting  $x_{j1}^a$  are strictly positive, then the contemplated pivot step, with  $(\mu, \nu)$  as the pivot position, is carried out.

Alternately, if  $(x_{j1}^a)_\rho = 0$  for at least one  $\rho$  and:

- a)  $x_{j1}^a \geq x_{j1}^b$ , then the contemplated pivot step is performed.
- b)  $x_{j1}^a < x_{j1}^b$ , then the non-negativity of  $x_{j1}(\lambda)$  for  $\lambda = \lambda_{ul}^e + \varepsilon$ ,  $\varepsilon > 0$ , is secured by the procedure outlined in 2(i)-(ii) above.

It should be noted however, that if PLM is solved by the product form of the inverse revised simplex method, it is computationally expedient to carry out the pivot step in advance and subsequently check its validity. If (b) prevails, return to the pre-pivot status of PLM is achieved by simply dropping the last elementary matrix in the produce form of the inverse.

## §6. Verification of the Method

In this section the validity of the algorithm is outlined.

Suppose that in the  $t^{\text{th}}$  cycle the problem  $\underline{M}$  ((3.2) - (3.3)) has  $s_t$  "additional constraints" of the form (3.6). It follows from (3.1) that  $\underline{M}$  is equivalent to the problem (2.1) - (2.3) and the constraints

$$(6.1) \quad \text{and} \quad \begin{aligned} y &\geq 0, \quad x_{j2} \geq 0 \quad (j = 1, \dots, k) \\ (x_{jv1})_{1v} &\geq 0 \quad (v = 1, \dots, s_t) \end{aligned}$$

where  $(x_{jv1})_{1v} \geq 0$  corresponds to the  $v^{\text{th}}$  constraint of the form (3.6).

Since canceling the restriction  $(x_j)_v \geq 0$  ( $j = 1, \dots, k$ ) in the primal problem  $\underline{P}$  results in the removal of the column corresponding to  $(w_j)_v$  from the dual problem (2.6) - (2.8), it follows that in the second formulation of the dual (i.e. in  $\underline{D}$ , (2.9) - (2.11) or for the linear case in  $\underline{LD}$ , (4.6) - (4.8)) the  $v^{\text{th}}$  inequality in the  $j^{\text{th}}$  block is replaced by an equation. Therefore, the dual problem of the problem stated by (2.1) - (2.3) and (6.1) is given by:

Minimize

$$(6.2) \quad \Phi(y, x, u) = F(y, x_1, \dots, x_k) + \sum_{j=0}^k b_j^i u_j - \nabla F(y, x_1, \dots, x_k)^i \cdot (y, x_1, \dots, x_k)$$

subject to:

$$(6.3) \quad \sum_{j=1}^k D_j^i u_j + D_0^i u_0 - \nabla_Y F(y, x_1, \dots, x_k) \geq 0$$

$$(6.4) \quad \begin{aligned} B_{j1}^i u_j + A_{j1}^i u_0 - \nabla_{x_{j1}} F(y, x_1, \dots, x_k) &\stackrel{(\geq)}{=} 0 \\ B_{j2}^i u_j + A_{j1}^i u_0 - \nabla_{x_{j2}} F(y, x_1, \dots, x_k) &\geq 0 \end{aligned} \quad (j = 1, \dots, k)$$

Similarly, the dual of the problem stated by (4.1), (2.2), (2.3) and (6.1), i.e. the linear case, is given by:

$$(6.2a) \quad \begin{array}{l} \text{Minimize} \\ \sum_{j=1}^k b_j^i u_j \end{array}$$

subject to

$$(6.3a) \quad \sum_{j=1}^k D_j^i u_j + D_0^i u_0 \cong c_0$$

$$(6.4a) \quad \begin{array}{l} B_{j1}^i u_j + A_{j1}^i u_0 \stackrel{(\cong)}{=} c_{j1} \\ B_{j2}^i u_j + A_{j2}^i u_0 \cong c_{j2} \end{array} \quad (j = 1, \dots, k)$$

where  $\stackrel{(\cong)}{=}$  means that the constraints corresponding to the variables  $(x_{jv1})_{i_v}$  ( $v = 1, \dots, s_t$ ) are inequalities.

The following theorem states the optimality condition:

#### Theorem 1

Let  $(y^*, x_{j1}^*, x_{j2}^*)$  ( $j = 1, \dots, k$ ) be the vector obtained after  $t$  cycles. It is an optimal solution to  $\tilde{P}$  if and only if  $x_{j1}^* \cong 0$  ( $j = 1, \dots, k$ ).

#### Proof:

The condition is clearly necessary because otherwise (2.4) would not be satisfied. For sufficiency, we note from (3.1) that  $(x_{j1}^*, x_{j2}^*, y^*)$  is an optimal solution to the problem given by (2.1) - (2.3) and (6.1). If  $x_{j1}^* \cong 0$  ( $j = 1, \dots, k$ ) the condition (6.1) can be replaced by (2.4) without changing the optimal solution. Thus,  $(x_{j1}^*, x_{j2}^*, y^*)$  is an optimal solution to  $\tilde{P}$  if

$$x_{j1}^* \cong 0.$$

In order to prove that an optimal solution to  $\tilde{P}$  is obtained after a finite number of cycles, we need the following two statements.

Lemma 1

To each vector  $(y^*, x_{j1}^*, x_{j2}^*)$  ( $j = 1, \dots, k$ ), obtained in the  $t^{\text{th}}$  cycle, there exists a corresponding vector  $(y^*, x_j^*, u_0^*, u_j^*)$  ( $j = 1, \dots, k$ ) which is a feasible point of  $\tilde{D}$  and has the property:

$$(6.5) \quad F(y^*, x_1^*, \dots, x_k^*) = \Phi(y^*, x_1^*, \dots, x_k^*, u_0^*, u_1^*, \dots, u_k^*)$$

where  $x_j^* = (x_{j1}^*, x_{j2}^*)$ .

Proof:

$(y^*, x_{j1}^*, x_{j2}^*)$  ( $j = 1, \dots, k$ ) is an optimal solution to the problem given by (2.1) - (2.3) and (6.1). Since (6.2) - (6.4) define the dual of this problem, it follows from the duality theorem for non-linear programming [19] that there exists a point  $(y^*, x_j^*, u_0^*, u_j^*)$  ( $j = 1, \dots, k$ ) satisfying (6.3) - (6.4) such that the objective functions (2.1) and (6.2) have equal values, immediately establishing the property (6.5). Comparison of (6.3) - (6.4) with (2.10) - (2.11) shows that each feasible point of (6.3) - (6.4) is also a feasible point of (2.10) - (2.11) (but not conversely).



Corrolary 1. For a linear objective function (6.5) implies:

$$(6.6) \quad \sum_{j=1}^k c_j' x_j^* + c_0' y^* = \sum_{j=0}^k b_j' u_j^*$$

Lemma 2

Let  $(y^*, x_{j1}^*, x_{j2}^*)$  and  $(y^{**}, x_{j1}^{**}, x_{j2}^{**})$  ( $j = 1, \dots, k$ ) be the vectors obtained at the  $t^{\text{th}}$  and  $(t+1)^{\text{th}}$  cycle, respectively. Then

$$(6.7) \quad F(y^{**}, x_1^{**}, \dots, x_k^{**}) \leq F(y^*, x_1^*, \dots, x_k^*)$$

Proof:

In the  $t^{\text{th}}$  cycle we have solved a problem given by (2.1) - (2.3) and (6.1). Denote the feasible region of this problem by  $R_1$ . This domain is subsequently altered, according to the procedure described in Section 3, as follows:

1) If  $\tilde{M}$  contains "additional constraints" of the form (3.6) we cancel those which are not active in the optimal solution (case 1). Each remaining "additional constraint" is either left unchanged (case 3) or rewritten (case 2) while one of the constraints  $(x_{j2})_i \geq 0$  (which is not active in the optimal solution since  $(x_{j2}^*)_i > 0$ ) is disregarded.

2) Suppose  $x_{j1}^*$  has at least one negative component, say  $(x_{j1}^*)_v$ . If case I applies, one of the constraints  $(x_{j2}^*)_i \geq 0$  (which is inactive since  $(x_{j2}^*)_i > 0$ ) is canceled and replaced by  $(x_{j1}^*)_v \geq 0$ . If case II applies, an additional constraint of the form (3.6), equivalent to  $(x_{j1})_v \geq 0$ , is added to the problem.

Thus, upon completion of a cycle, say the  $t^{\text{th}}$ , only inactive constraints are canceled while the new "additional constraints" of the form (3.6) which are added to  $\tilde{M}$  are not satisfied by  $(y^*, x_{j1}^*, x_{j2}^*)$ . For a maximization problem this implies (6.7).

Corrolary 2. For a linear objective function (6.7) implies:

$$(6.8) \quad \sum_{j=1}^k c'_j x_j^{**} + c'_0 y^{**} \cong \sum_{j=1}^k c'_j x_j^* + c'_0 y^* .$$

Remark 1:

The above proof shows that the feasible domain  $R_1$  is altered in two steps. First, by canceling some constraints, we obtain a larger domain  $R_2$  in which the objective function remains at its optimal solution value. Then, new constraints (i.e. non-negativity restrictions on the  $x_{j2}$  variables) which are not satisfied by the current optimal solution are added. This results in a smaller feasible domain, say  $R_3$ . It follows, therefore, that strict inequality holds in (6.7) and (6.8), except in the case of an alternate optimal solution in  $R_3$ . In this case, a possibility of cycling exists. Nevertheless, it can easily be prevented by a small perturbation in the coefficients of (2.1) or (4.1). Clearly, cycling will not occur for strictly concave objective functions since in such cases the optimal solution is unique.

Remark 2:

In Section 3 we assumed that if a feasible solution to  $\underline{P}$  exists, then  $\underline{M}$  attains an optimal solution for a finite point  $(y, x_{12}, \dots, x_{k2})$ . Now, suppose that the latter is not true, i.e.  $\underline{M}$  does not attain a finite optimum. In order to prevent such occurrences, we propose the following procedure.

Let  $T$  be a sufficiently large positive number, and  $o_j, q_j$  vectors, conformal to the current partitioning of  $x_j = (x_{j1}, x_{j2})$ , which have as their components all zeroes and ones respectively. Then, the addition of the condition

$$(6.9) \quad \sum_{j=1}^k q_j' x_{j2} + q_o' y \leq T$$

to the existing constraints of  $\underline{M}$ , insures that this enlarged  $\underline{M}$  has an optimal solution provided that  $\underline{M}$  has a feasible solution. Clearly, this is equivalent to the addition of:

$$(6.10) \quad \sum_{j=1}^k (o_j, q_j)' \begin{pmatrix} x_{j1} \\ x_{j2} \end{pmatrix} + q_o' y + \tau = T; \tau \geq 0$$

to the constraints of  $\underline{P}$ . If in the optimal solution to this enlarged  $\underline{P}$  we have  $\tau = 0$  for arbitrary large  $T$ , then the original problem has an unbounded solution.

Since the optimal value of the current  $\underline{M}$  is an upper bound to the objective function values of all subsequent  $\underline{M}$  problems (Lemma 2),

and due to the way in which the feasible domain of  $\underline{M}$  is altered from one cycle to the next, it follows that all subsequent  $\underline{M}$  problems have optimal solutions, provided they have a feasible solution.

### Definition

The constraints of  $\underline{D}$  are said to satisfy a constraint qualification at the point  $(y^0, x_j^0, u_0^0, u_j^0)$  ( $j = 1, \dots, k$ ) if there exists a differentiable mapping

$$(y, x_1, \dots, x_k) = \theta(u_0, u_1, \dots, u_k),$$

such that the point  $(\theta(u_0, u_1, \dots, u_k), u_0, u_1, \dots, u_k)$ , where

the  $u_j \in \{u_j \mid |u_j - u_j^0| < \varepsilon\}$  ( $j = 0, 1, \dots, k$ ), for some  $\varepsilon > 0$ , satisfies the constraints of  $\underline{D}$ .

This constraint qualification is satisfied if  $F(y, x_1, \dots, x_k)$  is a strictly concave function or if the Hessian matrix  $\nabla_{y, x_j}^2 F$  is non-singular at the point  $(y^0, x_1^0, \dots, x_k^0)$ .

### Theorem 2

Suppose that  $\underline{D}$  satisfies the constraint qualification. Then, if  $\underline{P}$  has an optimal solution it is obtained in a finite number of cycles.

### Proof:

Let  $(y^*, x_j^*, u_0^*, u_j^*)$  and  $(y^{**}, x_j^{**}, u_0^{**}, u_j^{**})$  ( $j = 1, \dots, k$ ) be the feasible points of  $\underline{D}$  associated with the vectors  $(y^*, x_{j1}^*, x_{j2}^*)$  and  $(y^{**}, x_{j1}^{**}, x_{j2}^{**})$  ( $j = 1, \dots, k$ ) obtained in the  $t^{\text{th}}$  and  $(t+1)^{\text{th}}$  cycle respectively (Lemma 1). Considering (6.5) in conjunction with Lemma 2 we see that:

$$(6.11) \quad \Phi(y^{**}, x_{j1}^{**}, x_{j2}^{**}, u_0^{**}, u_j^{**}) \leq \Phi(y^*, x_{j1}^*, x_{j2}^*, u_0^*, u_j^*) \quad (j = 1, \dots, k)$$

We observe that  $(y^*, x_j^*, u_0^*, u_j^*)$  ( $j = 1, \dots, k$ ) is an optimal solution to the dual problem (6.2) - (6.4) and that it satisfies as equalities at least those constraints in (6.3) - (6.4) which correspond to cancellation of the non-negativity restrictions on the  $x_{j1}$  in  $\underline{M}$ . Denote the set of equations in (6.3) - (6.4) for the  $t^{\text{th}}$  and  $(t+1)^{\text{th}}$  cycle by  $S^*$  and  $S^{**}$  respectively. If (6.11) is an equality for several consecutive cycles, appropriate methods to prevent cycling can be employed to insure that  $S^*$  reoccurs at most a finite number of times.

If  $\underline{D}$  has an unbounded solution, it follows that a problem (6.2) - (6.4) with an unbounded solution is obtained after a finite number of cycles. Since the corresponding primal problem  $\underline{M}$  is assumed to be bounded and  $G(y, x_{j2})$  is differentiable, by the duality theory for non-linear programming [19], this implies that  $\underline{M}$  has no feasible solution. The latter then implies that  $\underline{P}$  has no feasible solution.

If  $\underline{D}$  has an optimal solution, then it follows from the preceding discussion that it is obtained in a finite number of cycles. At each cycle we obtain a finite point  $(y^t, x_{j1}^t, x_{j2}^t, u_0^t, u_j^t)$  ( $j = 1, \dots, k$ ) and in a finite number of cycles we have a finite number of such points which must be in a bounded set. Since  $\Phi$  is differentiable, it must be bounded. If  $\underline{D}$  satisfies a constraint qualification (which is also satisfied if  $F(y, x_j)$  is strictly concave), the converse duality theorem [8] asserts that the optimal solution of the

corresponding  $\underline{M}$  yields the optimal solution to  $\underline{P}$ . Alternately, for cases where the constraint qualification is not satisfied, the optimal solution to the corresponding  $\underline{M}$  need not be feasible for  $\underline{P}$ .

Remark 3:

For the case of a linear objective function the above proof may be stated in a concise manner as follows:

The relation (6.11) implies:

$$(6.12) \quad \sum_{j=1}^k (c'_{j1} x_{j1}^{**} + c'_{j2} x_{j2}^{**}) + c'_0 y^{**} \leq \sum_{j=1}^k (c'_{j1} x_{j1}^* + c'_{j2} x_{j2}^*) + c'_0 y^* .$$

Therefore, with appropriate methods to prevent cycling, the optimal solution of  $\underline{LD}$ , if it exists, will be reached in a finite number of cycles, say after  $r$  cycles. By the duality theorem for linear programming, the vector  $(x_{j1}^r, x_{j2}^r, y^r)$  ( $j = 1, \dots, k$ ) obtained in the  $r^{\text{th}}$  cycle is an optimal solution to  $\underline{LP}$ . If  $\underline{LD}$  has no optimal solution, it follows again for the duality theorem that  $\underline{LP}$  also has no optimal solution.

## §7. Computational Aspects and Results

The computational efficiency of the algorithm presented in section 3 depends on several factors. First, the distinction between linear and non-linear objective function is an essential one. It is generally known that for most of the available methods, the solution efficiency for nonlinear problems depends almost entirely on the number of variables. This is particularly evident when one uses a method in the dual space such as Gradient Projection [13]. Consequently, the reduction in the total number of variables involved in each  $\underline{M}$  solution should be viewed as a much more important development than the obvious reduction in the number of constraints. This reduction can be impressive for many problems arising in practical applications. Then, little attention is paid to the increase in "additional constraints" of the form (3.6) during the course of the algorithm. Their accumulation is tolerated and their elimination may be deferred until convenient.

The situation may be markedly different for the linear case depending on the method of solving  $\underline{LM}$ . Its solution may be accomplished either by the primal or the dual simplex method. The choice will depend on the size of  $\underline{LM}$  which is related to the size of the original problem  $\underline{LP}$ . If this problem is specified with subproblem matrices  $B_j$  for which  $m_j \ll n_j$ , then the number of variables in  $\underline{LM}$  will still be substantial, thus dictating the use of the primal simplex method for its solution. The accumulation of additional constraints will then be checked by effective pivoting procedures. As mentioned earlier,

however, the existence of non-vanishing pivots cannot be guaranteed for all the variables corresponding to existing "additional constraints". Therefore, even with the emphasis on pivoting, the possibility of a modest accumulation of these constraints remains. On the other hand, if  $m_j \leq n_j$  with  $m_n \approx n_j$ , the number of  $x_{j2}$  variables in LM will be relatively small. In such instances, use of the dual simplex method should prove more efficient. The number of additional constraints may then be allowed to increase more freely, with pivoting assuming a secondary role.

The choice of initial bases  $B_{j1}$  for the subproblem matrices  $B_j$  is an obvious parameter which affects solution efficiency. Clearly, the optimal solution to the complete problem P would be obtained in one cycle if this choice were made to coincide with the optimal basis. In most industrial problems an initial point  $(y^0, x_j^0)$ ,  $j = 1, \dots, k$  (not necessarily feasible) will be known from the physical characteristics of the model or from a previous solution to a slightly modified problem. The columns of  $B_j$  which correspond to the positive components of the  $x_j^0$  specify the partial initial basis which may then be used, whenever linear independence holds, to construct the inverse  $B_{j1}^{-1}$  by appending, if necessary, some linearly independent non-basic columns. Other methods of obtaining an initial subproblem basis may be more advantageous. However, computational evidence will be required to establish their relative merits.

The solution efficiency will also be influenced by the method of variable exchanges, referred to as "pivoting". Such exchanges are required under both steps A and B of the algorithm. Complete lack of non-zero pivots at the



required positions will cause the generation of at least one "additional constraint" for the next cycle. Since generation of an excessive number of such constraints is undesirable, at least when  $\underline{LM}$  is solved by the primal simplex method, intuition suggests that more than one pivot step should be performed for the non-optimal blocks at each cycle. One way of performing this operation is to apply the simplex method to a modified subproblem as follows. Let the current basis for the  $j^{\text{th}}$  subproblem be  $B_{j1}$ , and let the submatrix of  $B_j$  containing the non-basic columns be  $B_{j2}$ . Suppose that the solution of  $\underline{M}$  and the subsequent application of (3.1) gives the following partition of the variables  $x_{j1}$  and  $x_{j2}$ :

$$x_{j1} = \begin{pmatrix} x_{I_1} \\ x_{I_2} \\ x_{I_3} \end{pmatrix} ; \quad x_{j2} = \begin{pmatrix} x_{I_4} \\ x_{I_5} \end{pmatrix}$$

with

$$\begin{aligned} I_1 &= \{i \mid (x_{j1})_i < 0\} \\ I_2 &= \{i \mid (x_{j1})_i = 0; i \in I_a\} \\ I_3 &= \{i \mid (x_{j1})_i \geq 0\} \\ I_4 &= \{i \mid (x_{j2})_i > 0\} \end{aligned}$$

where  $I_a$  represents the set of column indices  $i$  for which "additional constraints" of the form (3.6) were present in  $\underline{M}$ . An effective pivoting strategy would then be to exchange as many of the variables in  $I_1$  and  $I_4$  as possible, and to retain the variables  $I_3$  as basic. This is accomplished by considering the linear program:

Maximize

$$(7.1.1) \quad -T_1 q'_{I_1} x_{I_1} - T_2 q'_{I_2} x_{I_2} + T_4 q'_{I_4} x_{I_4}$$

subject to

$$(7.1.2) \quad B_{I_1} x_{I_2} + B_{I_2} x_{I_2} + B_{I_3} x_{I_3} + B_{I_4} x_{I_4} = 0$$

$$(7.1.3) \quad x_{I_1}, x_{I_2}, x_{I_3}, x_{I_4} \geq 0$$

where  $q_{I_1}, q_{I_2}, q_{I_4}$  are vectors having all ones in their components and the scalars  $T_1, T_2, T_4 > 0$  are specified weighing constants. This problem is solved by the primal simplex method. In order to retain  $x_{I_3}$  in the basis, the usual pivot selection rules of the simplex method are revised to avoid pivoting  $x_{I_3}$  out of the basis. In our program, an initial inverse for the above problem is obtained by reinverting the subproblem basis of the previous decomposition cycle. It would certainly be more efficient to maintain each inverse  $B_{j1}^{-1}$  in the product form which can then be revised, if the block is non-optimal, by the simplex algorithm. If  $T_1 \ll T_2 = T_4$ , then the solution to (7.1.1) - (7.1.3) will obtain the revised subproblem basis  $B_{j1}$  and its inverse by following the best pivoting sequence with preference given to reducing the infeasibility caused by  $x_{I_1}$ . Thus, the exchanging will take place primarily between  $x_{I_1}$  and  $x_{I_4}$  and only to a limited degree between  $x_{I_2}$  and  $x_{I_4}$ . If the weighing constants are chosen so that  $T_2 \ll T_1 = T_4$ , then exchanging will favor the elimination of the existing additional constraints over the reduction in the existing infeasibility. It is reasonable to assume then

that the choice of these constants will also influence the overall efficiency. However, limited computational experience in comparing the two extreme choices stated above, indicated no appreciable differences in the number of cycles.

Finally, the choice of an initial starting point for each  $\underline{M}$  is the key to the overall efficiency. Starting each  $\underline{M}$  from the solution to the previous  $\underline{M}$  seems to be a plausible way. Although such a point will be infeasible for the  $\underline{M}$  of the current decomposition cycle, the method outlined in [13] may be applied to obtain a feasible starting point. It is expected that this choice will be a good one, particularly in the later decomposition cycles. Similarly, for the linear case, the optimal basis to the previous  $\underline{LM}$  will provide a partial basis for starting the solution to the current  $\underline{LM}$ . Our experience has shown that the optimal bases of successive  $\underline{LM}$  problems differ from each other only by a few basic columns. This observation leads us to expect that the use of the previous basis columns, which are still present in the new  $\underline{LM}$ , as a partial starting basis, will result in considerable computational savings.

A small experimental computer program for solving the linear problem  $\underline{LP}$  has been written in FORTRAN for the CDC 3600 and has been tested on a number of randomly generated problems.

The program, which is completely core resident, is divided into a number of subroutines which essentially perform the following functions:

a) Problem data input or generation of input data, b) Construction of initial bases  $B_{j1}; j = 1, \dots, k$  and their inverses, c) Generation of the LM matrices, d) Solution of LM, e) Extraction of solution values, computation of  $x_{j1}$  and optimality tests, f) Variable exchanges for the subproblems and g) Output and solution check.

The input phase reads in the matrices  $A_j, B_j, D_0, D_j$ , the vectors  $b_0, b_j, c_0$ , and  $c_j$  ( $j = 1, \dots, k$ ). Optionally, these matrices and vectors are generated randomly with the necessary precautions insuring feasibility and boundedness. This part of the program also generates the input data for the complete problem LP in a form acceptable by the CDM4 LP system for the 3600. Subsequently this data is used to obtain solutions for each LP as an ordinary linear program. The initial bases, for this early version of the program, are taken as unit matrices representing slack and artificial vectors. The bulk of the computational work is done in (c) above, where the matrices, cost and right hand side vectors of LM are computed and are stored in packed form for later use by the linear programming code CDM4 [20]. CDM4 is then called to solve LM starting from a completely artificial basis. This undesirable manner of starting the solution of LM was chosen in the interest of simplicity since it was found that CDM4 could not effectively handle a given partial starting basis. The solution values, basis, etc. obtained by the CDM4 are extracted by unpacking. Then, the current value of the objective function and the levels of  $x_{j1}; j = 1, \dots, k$  are computed. For each non-optimal block, the necessary variable exchanges are performed by direct pivoting,

or optionally, by solving the modified subproblem (7.1.1) - (7.1.3) by CDM4. The data for these problems are packed and set-up for use by the CDM4 using the subroutines of (c) above. Each problem (7.1.1) - (7.1.3) is solved by first reinverting to the previous (feasible)  $B_{j1}$  and then by carrying Phase II iterations. Each block is handled in succession and requires the original data of  $B_j$  only. The output phase is negligible since the  $(x_{j1}^*, x_{j2}^*, y)$  ( $j = 1, \dots, k$ ) are available from the last LM solution and the computation (3.1) which was necessary for the optimality test. Finally, a solution check is made by substituting the optimal variable levels into (2.2) - (2.4) to obtain a computed right hand side vector and by comparing it to the given  $b_j$  ( $j = 0, 1, \dots, k$ ).

A number of test cases were solved successfully by our experimental program. The data for these problems are randomly generated as follows. The matrices  $B_j$  and  $A_j, D_j$  are 50% and 100% dense respectively with the non-zero elements of  $B_j$  arranged in a checkerboard pattern. Each element is a pseudo-random number in the range  $[0, 10]$ . In addition, unit matrices of appropriate dimensions are appended to  $A_j$  and  $B_j$  ( $j = 1, \dots, k$ ) representing non-negative slack variables. The vectors  $c_j, b_j$  ( $j = 0, 1, \dots, k$ ) are generated by first constructing a known optimal solution to the complete problem. The desired optimal levels of the variables  $x_j$  and  $y$  are chosen so that:

- a) a specified number of the coupling constraints are active
- b) a specified number of the coupling variables  $y$  are at a positive level and
- c) a specified number of the block variables  $x_j$  are at a positive level. Within these

restrictions, randomly selected subsets of the  $x_j$  and  $y$  variables are then declared basic by assigning random levels in the range  $[0, 1]$  and random cost elements which are appropriately magnified to insure that these variables will remain basic in the optimal solution. The right hand sides are then obtained by multiplying the constraint matrix by the generated values of  $x_j$  and  $y$ . The problem thus generated is insured to require a reasonable amount of work for its solution. The computed answers, however, may be slightly different from the generated ones for obvious reasons.

No claims will be made regarding the resemblance of these test cases to actual industrial problems. In fact, our test cases are too small to allow any inference for problems of giant sizes for which this method is primarily intended. Thus, the results of the 26 test cases presented in Figure 7.1 should be regarded only as an indication that our method should not be abandoned. Tests of large problems of practical importance, a sophisticated computer program with the flexibility for introducing the various solution strategies discussed in the previous paragraphs and a large amount of computing time, will be needed before the efficiency of this method, or any other method previously proposed by others, may be accepted on firm ground. Plans for designing such a system and performing extensive large scale testing are reported under consideration [18].

The information given in Figure 7.1 is arranged as follows:

Columns 1 - 6:

Test case identification and problem sizes

Column 7:

The number of decomposition cycles to optimality, which also corresponds to the number of LM's solved

Column 8:

Subproblem pivoting strategy used:

(1): Select one component of  $x_{j1}$  corresponding to one active "additional constraint" for a particular block and exchange it with a positive component of  $x_{j2}$  for the same block. If no non-zero pivot is found, this exchange is abandoned and this "additional constraint" is left in LM. Select one negative component of  $x_{j1}$ , say  $(x_{j1})_v$ , for the same block and exchange it with a positive component of  $x_{j2}$  for the same block. If no non-zero pivot is found, an "additional constraint" for  $(x_{j1})_v$  is generated for inclusion in the next LM.

(2): Multiple pivoting by defining and solving, for each non-optimal block, the subproblem (7.1.1) - (7.1.3) with  $T_1 = 1$ ,  $T_2 = T_4 = 100$ .

Column 9:

Average number of "additional constraints" which are present in LM at each decomposition cycle.

Columns 10, 11:

The average and total number of simplex iterations required to solve LM by the CDM4 LP System. Each LM solution is started "from scratch" i.e. from a full artificial or slack basis.

Columns 12, 13:

The average and total number of variable exchanges performed in the subproblems. For strategy (2) the reported numbers represent the number of simplex iterations after the completion of the reinversion to the current basis  $B_{j1}$ . For each non-optimal block, reinversion generally requires an additional  $m_j$  pivot steps.

Column 14:

The average number of optimal blocks in each cycle. This number is appreciably greater for problems with a large number of blocks. When a block is optimal, the corresponding matrices of LM remain the same for the next cycle. Our program, however, takes little advantage of this and thus some recomputation occurs.

Column 15:

Net computation time required to solve the number of LM problems (given in Column 7), starting each time from a full artificial or slack basis and using the primal simplex method as programmed in the CDM4 LP System.

Column 16:

The total solution time required to solve LP by the decomposition method starting from a full slack basis. This result includes: Preparation of LM matrices and vectors, packing of these for use by CDM4, solution (from scratch) of LM, unpacking of answers, computation of  $x_{j1}$ , optimality checks, subproblem pivoting (in case of strategy (2) preparation of the data for (7.1.1) - (7.1.3) in packed form, reinversion, solution by CDM4, unpacking of answers, etc.), generation of "additional constraints", etc.



Columns 17, 18:

Number of primal simplex iterations and net computation time required to solve the complete problem LP (column 6) by the CDM4 LP System which, for the purpose of this test, was arranged so that both data and program resided in core.

The running times reported for the decomposition method, in some instances, exceed those for the direct LP solution of the same problem. This disconcerting fact may be explained through consideration of several practical factors. First, we note that our experimental program was written with no regard for programming efficiency, for the sole purpose of solving a limited number of small test cases and investigating some of the computational aspects of the method. Consequently, the timing results should not be compared too closely with those obtained by CDM4, which is an efficient production tool. Second, an appreciable part of our program performs operations which allow the use of CDM4 as a subroutine. Most of this work would not be necessary if a more flexible and versatile linear programming code could be used as a subroutine for the decomposition program. Third, a substantial part of the total solution times consists of the optimization of the sequence of LM problems, each one starting from scratch. Comparing columns 15 and 16, we find that for the test cases treated here, an average of 57% of the total computation time has been expended for solving the LM problems. In some cases this percentage exceeds 70%. It is evident, therefore, that considerable savings would result if good starting bases for these LM problems were used. In addition,

the total computation times for solutions by strategy (b) include the time required for reinversion of the non-optimal subproblem bases in every cycle. Such reinversions may, of course, be avoided by maintaining the current subproblem inverses in product form, which will result in further savings. The recomputation of the LM matrices for optimal blocks, is another expensive operation which may be avoided. Finally, the results reported herein are, in a sense, the worst possible, since solutions to these problems were initiated from an all slack basis and the problem data were generated so that only a small number of these slacks would be contained in the optimal basis. Thus a good starting basis for the complete problem should be expected to improve matters considerably.

TEST CASE NO.	NO. OF COUPLING CONSTRAINTS	NO. OF COUPLING VARIABLES	BLOCK SIZE	NO. OF BLOCKS	TOTAL PROBLEM SIZE	SOLUTION BY DECOMPOSITION										DIRECT LP SOLUTION	
						7	8	9	10	11	12	13	14	15	16	17	18
						NO. OF DEC. CYCLES	PIVOTING STRATEGY	AVER. NO. OF ADDL CONSTR. PER CYCLE	AVER. NUMBER OF ITERATIONS	TOTAL SIMPLEX FOR LM PROBLEMS	AVER. NUMBER OF VARIABLE EXCHANGES IN SUBS.	AVER. NO. OF OPTIMAL BLOCKS	TOTAL TIME FOR LM (SECS) *	TOTAL SOLUTION TIME (SECS) *	LP ITTERS.	SOLUTION TIME (SECS) *	
1	5	5	10X20	2	25X45	8	1	5.0	10.7	86	2.3	18	1.5	4.6	10.3	28	11.0
2	10	5	10X20	2	30X45	3	2	4.7	15.6	47	5.7	17	0.0	1.9	4.7	15	6.0
3	5	10	10X20	2	25X50	3	2	2.5	16.4	49	5.3	16	0.7	2.0	4.9	30	8.0
4	10	10	10X20	2	30X50	6	2	7.1	18.7	112	2.3	14	0.2	6.6	12.2	29	9.0
5	5	5	10X30	2	25X65	4	2	2.6	19.3	77	4.5	18	0.2	3.8	8.5	28	10.0
6	10	5	10X30	2	30X65	8	2	5.8	19.4	155	3.4	27	0.3	11.6	20.9	31	12.0
7	5	10	10X30	2	25X70	7	1	10.4	22.3	156	2.6	18	0.1	15.1	25.2	38	13.0
8	10	10	10X30	2	30X70	6	2	9.1	24.0	144	2.8	17	0.2	13.3	20.9	32	13.0
9	5	5	15X30	2	35X65	5	2	5.8	33.5	168	3.4	17	0.4	16.1	20.1	45	13.0
10	10	5	15X30	2	40X65	7	1	5.7	17.4	122	2.6	26	0.2	11.1	25.5	32	12.0

\*CDC - 3600 seconds.

Figure 7.1 - Test results

11	5	10	15X30	2	35X70	8	1	11.1	23.2	186	2.8	22	0.1	15.7	34.5	49	15.0
12	10	10	15X30	2	40X70	6	2	7.6	21.2	127	3.5	21	0.2	9.8	19.0	36	15.0
13	5	5	10X20	5	55X105	13	1	8.6	21.2	127	3.3	20	0.2	10.9	26.3	86	45.0
14	10	5	10X20	5	60X105	4	2	6.2	28.0	112	3.5	14	0.3	8.9	16.8	64	24.0
15	5	10	10X20	5	55X110	7	2	8.4	19.6	255	5.2	68	0.9	24.0	44.3	81	32.0
16	10	10	10X20	5	60X110	9	1	9.3	23.4	164	5.4	38	1.3	14.7	28.2	72	31.0
17	5	5	10X30	5	55X155	6	2	8.9	24.2	268	5.5	49	0.3	21.9	39.7	112	52.0
18	10	5	10X30	5	60X155	9	2	9.3	29.2	175	6.0	36	0.5	17.9	32.0	109	55.0
19	5	10	10X30	5	55X160	10	1	12.0	27.4	274	5.8	58	0.5	31.6	54.8	109	71.0
20	10	10	10X30	5	60X160	6	2	13.0	34.4	204	6.5	39	0.5	23.6	36.4	121	63.0
21	5	5	15X30	5	80X155	9	1	13.0	35.4	329	5.1	46	0.7	43.4	68.2	90	52.0
22	10	5	15X30	5	85X155	9	2	11.8	39.2	352	6.1	55	1.1	45.4	69.0	118	66.0
23	5	10	15X30	5	80X160	18	1	8.6	25.5	459	5.2	93	0.6	70.4	120.0	122	72.0
24	10	10	15X30	5	85X160	11	2	11.8	35.4	390	5.5	60	0.6	63.9	92.4	199	177.0
25	5	5	10X20	10	105X205	15	1	8.6	37.0	555	6.4	96	0.5	103.5	162.0	126	79.0
26	5	5	10X20	15	155X305	12	2	12.2	43.0	516	7.4	89	0.8	100.7	142.8	199	177.0
27	5	5	10X20	15	155X305	16	1	12.3	37.4	597	5.3	85	0.6	108.2	160.9	199	177.0
28	5	5	10X20	15	155X305	14	2	12.8	38.8	544	6.7	93	0.8	96.1	140.6	199	177.0
29	5	5	10X20	15	155X305	11	1	12.0	44.6	491	6.7	74	0.5	104.4	155.5	199	177.0
30	5	5	10X20	15	155X305	10	2	10.3	44.5	445	9.0	90	0.7	89.6	130.7	199	177.0
31	5	5	10X20	15	155X305	25	1	9.5	24.1	578	6.9	172	0.2	74.0	189.2	199	177.0
32	5	5	10X20	15	155X305	17	2	13.3	36.4	617	7.7	131	0.4	83.8	145.8	199	177.0
33	5	5	10X20	15	155X305	21	1	9.0	30.8	646	6.3	132	0.8	97.4	208.9	199	177.0
34	5	5	10X20	15	155X305	14	2	14.1	45.1	632	4.8	67	1.6	108.0	165.2	199	177.0
35	5	5	10X20	15	155X305	20	1	13.4	32.8	656	5.9	117	0.5	98.8	205.4	199	177.0
36	5	5	10X20	15	155X305	19	2	13.9	44.6	850	8.2	156	0.5	133.3	208.0	199	177.0
37	5	5	10X20	15	155X305	17	1	13.4	42.6	723	6.5	111	0.4	129.1	246.1	199	177.0
38	5	5	10X20	15	155X305	14	2	11.6	41.5	582	9.7	136	1.0	97.7	169.7	199	177.0
39	5	5	10X20	15	155X305	33	1	13.0	31.6	1041	6.7	219	3.9	171.5	264.2	199	177.0
40	5	5	10X20	15	155X305	18	2	18.0	45.8	828	5.7	103	3.3	152.0	210.1	199	177.0
41	5	5	10X20	15	155X305	57	1	14.9	34.8	1984	6.0	343	7.5	457.9	659.2	199	177.0

(Maximum row error is less than  $2.4 \times 10^{-7}$  for all solutions)

Figure 7.1 - Test results (continued)

Acknowledgements

The authors wish to acknowledge the helpful suggestions of Professor J. B. Rosen, and the contribution to certain sections of the decomposition program by Mr. Dennis Kuba.

## REFERENCES

- [1] Abadie, J. M., and A. C. Williams: Dual and Parametric methods in Decomposition, Recent Advances in Mathematical Programming (P. Wolfe and R. L. Graves, Eds.) pp. 143-158, New York, McGraw Hill, 1963.
- [2] Beale, E. M. L.: The Simplex Method Using Pseudo-basic Variables for Structured Linear Programming Problems, Recent Advances in Mathematical Programming (P. Wolfe, and R. L. Graves, Eds.) pp. 133-148, New York: McGraw Hill, 1963.
- [3] Benders, J. F.: Partitioning Procedures for Solving Mixed Variables Programming Problems, Numerische Mathematik 4, 238-252 (1962).
- [4] Dantzig, G. B. and Wolfe, P.: Decomposition Principle for Linear Programs. Operations Research 8, 101-111 (1961).
- [5] Frank, M. and Wolfe, P.: An Algorithm for Quadratic Programming, Naval Res. Logistics Quart., 3 (1956).
- [6] Hadley, G.: Linear Programming, Reading, Massachusetts, Addison-Wesley, 1963.
- [7] Huard, P.: Application du Principe de Décomposition aux Programmes Mathématiques Non-Linéaires, Note, Electricité de France HR-547613 (1963).
- [8] Mangasarian. O. L.: Nonlinear Programming, McGraw Hill, New York, 1969, (forthcoming).
- [9] Ritter, K.: A Decomposition Method for Structured Quadratic Programming Problems, Mathematics Research Center Technical Summary Report #718, University of Wisconsin (1967).
- [10] Ritter, K.: A Decomposition Method for Linear Programming Problems with Coupling Constraints and Variables, Mathematics Research Center Technical Summary Report # 739, University of Wisconsin, (1967).
- [11] Ritter, K., and Grigoriadis, M. D.: A Decomposition Method for Large Structured Non-Linear Programs, Mathematics Research Center Technical Summary Report #786, University of Wisconsin, (1967).

- [12] Ritter, K., and Grigoriadis, M. D.: An Algorithm for Post-Optimality Analysis of Structured Linear Programs, Mathematics Research Center Technical Summary Report #815, University of Wisconsin (1967).
- [13] Rosen, J. B.: The Gradient Projection Method for Non-Linear Programming. Part I. Linear Constraints, Journal of the Society for Industrial and Applied Mathematics, Vol. 8, pp. 181-217 (1960).
- [14] Rosen, J. B.: Convex Partition Programming. Recent Advances in Mathematical Programming (P. Wolfe and R. L. Graves, Eds.) pp. 159-176 New York: McGraw-Hill, (1963).
- [15] Rosen, J. B.: Primal Partition Programming for Block Diagonal Matrices, Numerische Mathematik 6, 250-260 (1964).
- [16] Rosen, J. B.: Parametric Requirement Vector in Primal Partition Programming, Private Communication (1965).
- [17] Sanders, J. C.: A Non-Linear Decomposition Principle, Operations Research, 13, pp. 266-271 (1965).
- [18] Webber, D. and White, W. W.: A Partitioning Algorithm for Structured Linear Programming Problems, to be presented at TIMS/OSRA Joint Meeting, San Francisco, May 1968.
- [19] Wolfe, P.: A Duality Theorem in Non-Linear Programming, Quarterly Appl. Math., Vol. XIX, No. 3, pp. 239-244 (1961).
- [20] CDM4: 3400/3600 CDM4 LP System Reference Manual, Control Data Corporation, Palo Alto, California.

## FOOTNOTES

<sup>1</sup>The case of linearly dependent constraints (2.2) - (2.3) may cause a larger number of non-negativity restrictions to be active which may in turn result in a larger number of constraints in  $\underline{M}$  (See Section 3, Case 3).

<sup>2</sup>If  $q = 0$ , proceed as if Case II holds.

<sup>3</sup>Since  $(x_{j1}^{**})_v = 0$ , this procedure changes only the partitioning of  $x_j$  into  $(x_{j1}, x_{j2})$  but not the actual value of  $x_j^{**}$ . In the new partitioning  $(x_{j1}^*)_v$  belongs to the variables which are restricted to non-negative values.

<sup>4</sup>Assuming (for notational purposes) that the  $B_{j1}$  are formed by the first  $m_j$  columns of  $B_j$  for  $j = 1, \dots, k$ , "all  $v_j$ " refers to:

$$v_0 = 1, \dots, p; \quad v_j = m_j + 1, \dots, n_j; \quad j = 1, \dots, k.$$