A Two-Phase Decomposition Algorithm for Linear Programs with Angular Structure

By

Hideklyo ITAKURA* and Yoshikazu NISHIKAWA*

(Received September 19, 1975)

Abstract

Large scale linear programming problems often have special forms of constraints. An angular structure is a typical instance. The Dantzig-Wolfe decomposition principle is an effective tool for solving the linear programming problem with angular structure.

So far, the decomposition principle has been used only in the second-phase problem of the two-phase simplex procedure. This paper proposes a complete two-phase algorithm, in which the decomposition technique is fully utilized both in the first and the second phases. The present algorithm is then applicable, without any a priori knowledge of an initial feasible solution, to all the classes of linear programs with angular structure, though it may have some computational redundancies.

1. Introduction

Linear programming is a fundamental mathematical technique in various fields of operations research and systems planning. The standard simplex algorithm is a general tool to solve linear programming problems. The algorithm, however, is not almighty in view of numerical computations. If, in particular, a given problem has many thousands of variables and constraints, a storage space requirement and a computation time requirement may become quite large. Then, some idea to reduce them is highly appreciated.

Fortunately, in almost all practical applications, the density of nonzero elements in the constraint coefficient matrix is reduced accordingly as the size of the linear program becomes larger. Furthermore, the arrangement of these elements tends to fall into a special pattern. The most typical pattern is that of so called angular structure.[1] Several effective algorithms have been already proposed for solving a large problem with angular structure. Among them, some typical ones are the

* Department of Electrical Engineering.

decomposition principle of Dantzig-Wolfe[2,3] and the partitioning procedure of Rosen.[4] The decomposition principle is quite simple for those familiar with the mathematics of linear programming. Furthermore, it is readily generalized to a class of nonlinear problems. Particular decomposition algorithms have been applied to practical optimization problems, for example, enormous scheduling problems.[5]

In spite of its usefulness, the conventional decomposition algorithm has a fault, that is, the difficulty in creating an initial feasible solution to start with. In other words, a proper first-phase procedure is lacking. The purpose of this paper is to construct the first phase, also by making use of the decomposition principle,[6] and to complete a two-phase algorithm effective for an angular-structure problem. The new algorithm will be called the two-phase decomposition algorithm.

Section 2 introduces the linear programming problem with angular structure, and outlines the Dantzig-Wolfe decomposition algorithm. Section 3, which is the principal part of the paper, details the construction of the two-phase decomposition algorithm.

2. Linear Programming Problem with Angular Structure and the Decomposition Algorithm

Let us consider the following linear programming problem:

minimize the objective function

$$z=\sum_{i=1}^{p} c_i' x_i \tag{1}$$

subject to the constraints

$$\sum_{i=1}^{p} A_i x_i = b_0 \tag{2}$$

$$B_i x_i = b_i \geq 0, \quad x_i \geq 0 \quad (i=1, 2, \ldots\ldots\ldots, p) \tag{3}$$

where x_i and c_i are n_i-vectors, and b_i a nonnegative m_i-vector $(i=0, 1, \ldots\ldots, p)$. A_i and B_i are $m_0 \times n_i$- and $m_i \times n_i$-matrices $(i=1, 2, \ldots\ldots, p)$, respectively. A prime denotes transposition of a vector or a matrix. Here and throughout the paper, all vectors are in column form.

Equations (2) and (3) are called the coupling constraints and the block constraints, respectively. The constraints in the structure of Eqs. (2) and (3) are called p-block angular. Angular-structure constraints are often encountered in practical problems with many variables and constraints.

Such a kind of linear programming problems can of course be solved with the use of a common simplex or revised simplex procedure. Nonetheless, some other methods are preferable in which the particularity of the constraint forms is taken into consideration. These methods are superior to the conventional ones from the view point of a computer storage and a CPU time. One of them is the decomposition method by Dantzig and Wolfe. In this section, the use of the decomposition method is briefly explained for solving a problem with angular structure. The text[1] is available for the details.

Let S_i be the convex polyhedron composed of x_i satisfying the constraints (3) for each i:

$$S_i \triangleq \{x_i | B_i x_i = b_i,\ x_i \geq 0\} \tag{4}$$

Then, by familiar theories of the set, any element x_i of S_i can be written as a convex combination of x_i^j plus a nonnegative linear combination of $\bar{x}_i^j$:[1]

$$\begin{aligned} &x_i = \sum_{j=1}^{l_i} \mu_i^j x_i^j + \sum_{j=1}^{k_i} \nu_i^j \bar{x}_i^j, \qquad \sum_{j=1}^{l_i} \mu_i^j = 1, \\ &\mu_i^j \geq 0 \quad (j=1, 2, \ldots\ldots, l_i), \quad \nu_i^j \geq 0 \quad (j=1, 2, \ldots\ldots, k_i) \end{aligned} \tag{5}$$

where x_i^j and $\bar{x}_i^j$ are extreme points and extreme rays of the polyhedron S_i; and l_i and k_i are the numbers of these points and rays, respectively.

Substituting Eqs. (5) into (1) and (2), we have a new linear programming problem in the variable λ:

minimize

$$z = c'\lambda \tag{6}$$

subject to

$$A\lambda = b_0, \quad \tilde{e}_i'\lambda = 1 \quad (i=1, 2, \ldots\ldots, p), \quad \lambda \geq 0 \tag{7}$$

In the above, λ is the n-vector with the components μ_i^j and ν_i^j, n being $\sum_{i=1}^{p} (l_i + k_i)$, c is the n-vector with the components $c_i' x_i^j$ and $c_i' \bar{x}_i^j$, and $\tilde{e}_i$ is the $(l_i + k_i)$-vector with the first l_i components unity and the rest zero. A is the $m_0 \times n$-matrix with the columns $A_i x_i^j$ and $A_i \bar{x}_i^j$.

This problem is called the master programming problem generated from the original one. The master program is completely equivalent to the original. That is, provided that the solution $\lambda = \lambda^*$ corresponds to the master, substituting it into Eqs. (5) yields the solution to the original.

The master program has only $m_0 + p$ constraints, compared to the $\sum_{i=0}^{p} m_i$ constraints of the original. The former is much less than the latter, when the value of m_i is large and the value of p is moderate.

In order to construct the complete master program, we must know all the extreme points and the extreme rays of S_i. However, this is very difficult and, if n_i and/or

m_i are large, it may be impossible. Even if all of these were known, the master program would be more cumbersome than the original. Therefore, rather than the problem with all the variables, we consider an alternative form of problem, called the restricted master program. This modified type of problem is with the variables dropped except the current basic variables and those about to enter the basis. To see how such a problem is dealt with is the subject in the remaining part of this section.

Assume that the master program has a feasible solution. Choose m_0+p extreme points and/or rays altogether in $S_1, S_2, \ldots\ldots, S_p$ to find out one basic feasible solution of the master. Let π be the simplex multiplier associated with this basic solution. Partition π as

$$\pi \triangleq (\pi_0', \pi_1, \ldots\ldots, \pi_p)' \tag{8}$$

where π_0, the m_0-vector, corresponds to the constraints $A\lambda=b_0$ in Eqs. (7) and π_i, the scalar, to $\tilde{e}_i'\lambda=1$ for each $i=1, 2, \ldots\ldots, p$.

By using π_0, we make p independent subproblems as follows:

minimize

$$z_i=(c_i'-\pi_0'A_i)x_i \tag{9}$$

subject to the constraints (3)

It is known that,[1] if the minimum objective value z_i^* of these subproblems satisfies

$$z_i^*-\pi_i \geq 0 \tag{10}$$

for all i, then x_i calculated from Eqs. (5), being the optimal solution to the current restricted master program, is optimal to the original. The condition (10) is equivalent to the usual simplex criterion for the current basic feasible solution of the master to be optimal.

If the condition (10) is not satisfied for at least one i, we generate a new restricted master program with m_0+2p variables. The new program is constructed with use of prechosen m_0+p extreme points and/or rays and with p of those corresponding to p subproblem solutions. Each of the latter should be the extreme point x_i minimizing z_i, when z_i^* is bounded, or be the associated extreme ray $\bar{x}_i$, when z_i^* is unbounded. We solve this new program, send the simplex multiplier π_0 to the subproblems (3) and (9), and solve these again. These procedures are iterated, until the optimality test (10) is passed for all i. Just after solving a restricted master program in each iteration, we remove the variables in the program except the optimal basic variables. Then, a restricted master program always has only m_0+2p variables.

There are a number of alternative methods in which the original problem may be decomposed, or a restricted master program be constructed. Among them, the above formulation may be the most standard and advantageous in some sense.

3. Decomposition Algorithm with Two Phases

In the previous section, it has been assumed that the master program is feasible and an associated simplex multiplier is available. When the original constraints are additionally of a special form, it is comparatively easy to find a basic feasible solution, and a computer-programmed code is presented.[7] However, it is generally difficult or, owing to the inconsistency of given constraints, it may originally be impossible. One of the methods for such cases is that a new variable is added to the restricted master program successively until a feasible solution is discovered.[8] The ultimate number of variables to be added, however, is not known initially. Specifically, π may be furnished by the analyst from a previous solution of the problem or a similar one.[9] In any case, these procedures are not very suitable for a computer program.

In this section, a computer-oriented algorithm is developed for obtaining a feasible solution of the master program.[6] The algorithm is based on an application of the decomposition principle to Phase I of the two-phase simplex procedure.

3.1 Problem of Phase I

We consider the following angular-structure problem having various forms of constraints:

minimize the objective function

$$z=\sum_{i=1}^{p} c_i' x_i \tag{11}$$

subject to the constraints

$$\sum_{i=1}^{p} A_i^1 x_i \geq b_0^1, \quad \sum_{i=1}^{p} A_i^2 x_i = b_0^2, \quad \sum_{i=1}^{p} A_i^3 x_i \leq b_0^3 \tag{12}$$

$$B_i^1 x_i \geq b_i^1, \quad B_i^2 x_i = b_i^2, \quad B_i^3 x_i \leq b_i^3, \quad x_i \geq 0 \quad (i=1, 2, \ldots\ldots, p) \tag{13}$$

where x_i and c_i are n_i-vectors and b_i^j (≥ 0) an m_i^j-vector ($j=1, 2, 3$; $i=0, 1, \ldots\ldots, p$); A_i^j and B_i^j are $m_0^j \times n_i$- and $m_i^j \times n_i$-matrices ($j=1, 2, 3$; $i=1, 2, \ldots\ldots, p$), respectively.

Phase-I problem is a linear programming problem with some artificial variables:

minimize the objective function

$$\tilde{z}=\sum_{i=0}^{p} (e_i^{1\prime} w_i^1 + e_i^{2\prime} w_i^2) \tag{14}$$

subject to the constraints

$$\sum_{i=1}^{p} A_i^1 x_i - s_0^1 + w_0^1 = b_0^1, \quad \sum_{i=1}^{p} A_i^2 x_i + w_0^2 = b_0^2, \quad \sum_{i=1}^{p} A_i^3 x_i + s_0^3 = b_0^3 \tag{15}$$

$$B_i^1 x_i - s_i^1 + w_i^1 = b_i^1, \quad B_i^2 x_i + w_i^2 = b_i^2, \quad B_i^3 x_i + s_i^3 = b_i^3,$$
$$\tilde{x}_i \triangleq (x_i', s_i^{1\prime}, s_i^{3\prime}, w_i^{1\prime}, w_i^{2\prime})' \geq 0 \qquad (i=1, 2, \ldots\ldots, p) \tag{16}$$

$$\tilde{x}_0 \triangleq (s_0^{1\prime}, s_0^{3\prime}, w_0^{1\prime}, w_0^{2\prime})' \geq 0 \tag{17}$$

where s_i^j and w_i^j are the m_i^j-slack and artificial variables, respectively, and e_i^j is the m_i^j-vector with all the components unity.

It is natural to put the new variables s_i^j and w_i^j into the block i for each $i=1, 2, \ldots\ldots, p$. As for the variables s_0^j and w_0^j, there are some schemes to deal with them. One is to distribute these variables among the p blocks and to treat them as variables of each block. Then, a question will arise as to how to distribute them. Therefore, we artificially put a $(p+1)$th new block only with s_0^j and w_0^j, for which the block constraint is only the nonnegativity one (17). As a result, Phase-I problem (14) to (17) is of the angular structure with $p+1$ blocks. The decomposition principle is applicable to this problem, if we can readily find a feasible solution of its master program.

3.2 Application of the decomposition algorithm to Phase-I problem

According to Section 2, the master program of the problem (14) to (17) is constructed from extreme points and/or rays of the $p+1$ polyhedra

$$S_i \triangleq \{\tilde{x}_i | \tilde{x}_i \text{ satisfies the constraints (16)}\} \quad (i=1, 2, \ldots\ldots, p),$$
$$S_{p+1} \triangleq \{\tilde{x}_0 | \tilde{x}_0 \geq 0\} \tag{18}$$

It is required to skillfully choose these points or rays to find out a basic feasible solution of the master. With the aid of the introduction of w_i^j, this is possible, if we choose the following m_0+p+1 extreme points and rays, m_0 being $m_0^1+m_0^2+m_0^3$, *i.e.,* one extreme point of S_i for each $i=1, 2, \ldots.., p+1$:

$$\tilde{x}_i = (0', b_i^{3\prime}, b_i^{1\prime}, b_i^{2\prime})' \qquad (i=1, 2, \ldots\ldots, p) \tag{19}$$

$$\tilde{x}_0 = 0 \tag{20}$$

and m_0 extreme rays of S_{p+1}:

$$\tilde{x}_0 = e_{m_0^1+1}, e_{m_0^1+2}, \ldots\ldots, e_{m_0^1+m_0} \tag{21}$$

where e_k is the $(m_0^1+m_0)$-vector with all the components zero except the kth unity. Then we have a set of basic feasible solutions to the master:

$$\begin{aligned} \lambda_i &= 1 \quad (i=1, 2, \ldots\ldots, p+1), \\ \lambda_{p+1+i} &= b_{0i}^3 \quad (i=1, 2, \ldots\ldots, m_0^3), \\ \lambda_{p+m_0^3+1+i} &= b_{0i}^1 \quad (i=1, 2, \ldots\ldots, m_0^1), \\ \lambda_{p+m_0^3+m_0^1+1+i} &= b_{0i}^2 \quad (i=1, 2, \ldots\ldots, m_0^2) \end{aligned} \tag{22}$$

where b_{0i}^j is the ith component of the vector b_0^j and λ_k is a scalar having a subscript indexed in order of extreme points and rays chosen in Eqs. (19) to (21). The objective value by the solutions (22) is

$$\tilde{z}=\sum_{i=1}^{p}(e_i^{1\prime}b_i^1+e_i^{2\prime}b_i^2)\lambda_i+\sum_{i=1}^{m_0^1+m_0^2}\lambda_{p+m_0^3+1+i} \tag{23}$$

and the associated simplex multiplier is

$$\begin{aligned}\pi&\triangleq(\pi_0^{1\prime},\ \pi_0^{2\prime},\ \pi_0^{3\prime},\ \pi_1,\ \ldots\ldots,\ \pi_p,\ \pi_{p+1})'\\&=(e_0^{1\prime},\ e_0^{2\prime},\ 0',\ e_1^{1\prime}b_1^1+e_1^{2\prime}b_1^2,\ \ldots\ldots,\ e_p^{1\prime}b_p^1+e_p^{2\prime}b_p^2,\ 0)'\end{aligned} \tag{24}$$

where π_0^j is the m_0^j-vector and π_i $(i=1, 2, \ldots\ldots, p+1)$ the scalar.

In succession, we make $p+1$ subproblems according to Section 2. With these solutions, we examine the optimality conditions (10) for the solutions (22). If not all of these conditions hold, we add $p+1$ new variables to Eqs. (22) and (23), by using the subproblem solutions. We proceed to obtain the simplex multiplier for an optimal solution to the new master program and again make subproblems. When, after some repetitions of such operations, the optimality conditions have been satisfied, Phase I is completed. At this stage, if the value of $\tilde{z}$ defined by Eq. (14) is positive, the original problem is not feasible.[10] If this is not the case, we can go ahead to Phase II, in which the simplex multiplier should be recalculated for the original objective (11).

3.3 Subproblems and Optimality Conditions

According to Section 2, the subproblems for testing the optimality of Phase-I master-program solution are as follows:

(a) The ith subproblem with $i=1, 2, \ldots\ldots, p$

minimize

$$z_i=-\sum_{j=1}^{3}\pi_0^{j\prime}A_i^jx_i+\sum_{j=1}^{2}e_i^{j\prime}w_i^j \tag{25}$$

subject to the constraints (16)

This problem can be solved by starting with the initial basic feasible solution (19). However, since the variables w_i^j are destined for removal, it is disadvantageous to retain w_i^j all through Phase I. Thus, as a preliminary step, we solve the problem without the first term in Eq. (25):

minimize

$$\tilde{z}=\sum_{j=1}^{2}e_i^{j\prime}w_i^j \tag{26}$$

subject to the constraints (16)

If the minimum value of $\tilde{z}_i$ is positive, the block constraints (12) with the subscript i are infeasible,[10] and consequently so is the total one. If not, we solve, as the next step, the problem without w_i^j:

minimize

$$z_i = -\sum_{j=1}^{3} \pi_0^{j\prime} A_i^j x_i \tag{27}$$

subject to the constraints (13)

What is left is the need to check the condition

$$z_i^* \geq \pi_i \tag{28}$$

for the minimum value z_i^* of z_i.

The subproblems on and after the second iteration are of the form (13) and (27), rather than (16) and (26).

(b) The $(p+1)$th subproblem

minimize

$$z_{p+1} = \pi_0^{1\prime} s_0^1 - \pi_0^{3\prime} s_0^3 + \sum_{j=1}^{2} (e_0^j - \pi_0^j)' w_0^j \tag{29}$$

subject to the constraints (17)

If the variables w_0^j are not our concern (the reason for this is the same as in Case (a)), the optimal objective z_{p+1}^* of z_{p+1} is easily obtained. Only the following two cases are possible:

(i) When $\pi_0^1 \geq 0$ and $\pi_0^3 \leq 0$, we have $z_{p+1}^* = 0$. The extreme point giving z_{p+1}^* is $s_0^1 = s_0^3 = 0$.

(ii) When $\pi_0^1 \ngeq 0$ or $\pi_0^3 \nleq 0$, we have $z_{p+1}^* = -\infty$. Hence, provided that $(\pi_0^{1\prime}, -\pi_0^{3\prime})' \geq 0$ is violated with its kth component, we necessarily choose the extreme ray $(s_0^{1\prime}, s_0^{3\prime})'$ with all the components zero except the kth unity.

Since the polyhedron S_{p+1} has only one extreme point of Eq. (20), λ_{p+1} in Eqs. (22) is always a basic variable of the master program and has the zero coefficient in the objective function. As a consequence, π_{p+1} is always zero and the inequality (10) with $i = p+1$ is reduced to

$$z_{p+1}^* \geq 0 \tag{30}$$

Therefore, we see that in Case (i) the inequality (30) is satisfied; while in (ii) it is not.

Now, we have completed the first phase in our two-phase decomposition algorithm. If a given problem is of a special structure, we need not go through the whole or a part of the above procedure: If $m_i^j = 0$ $(j=1, 2)$ for a particular i, solving the ith subproblem (26) and (16) is skipped, and if $m_i^j = 0$ $(j=1, 2)$ for every i zero through

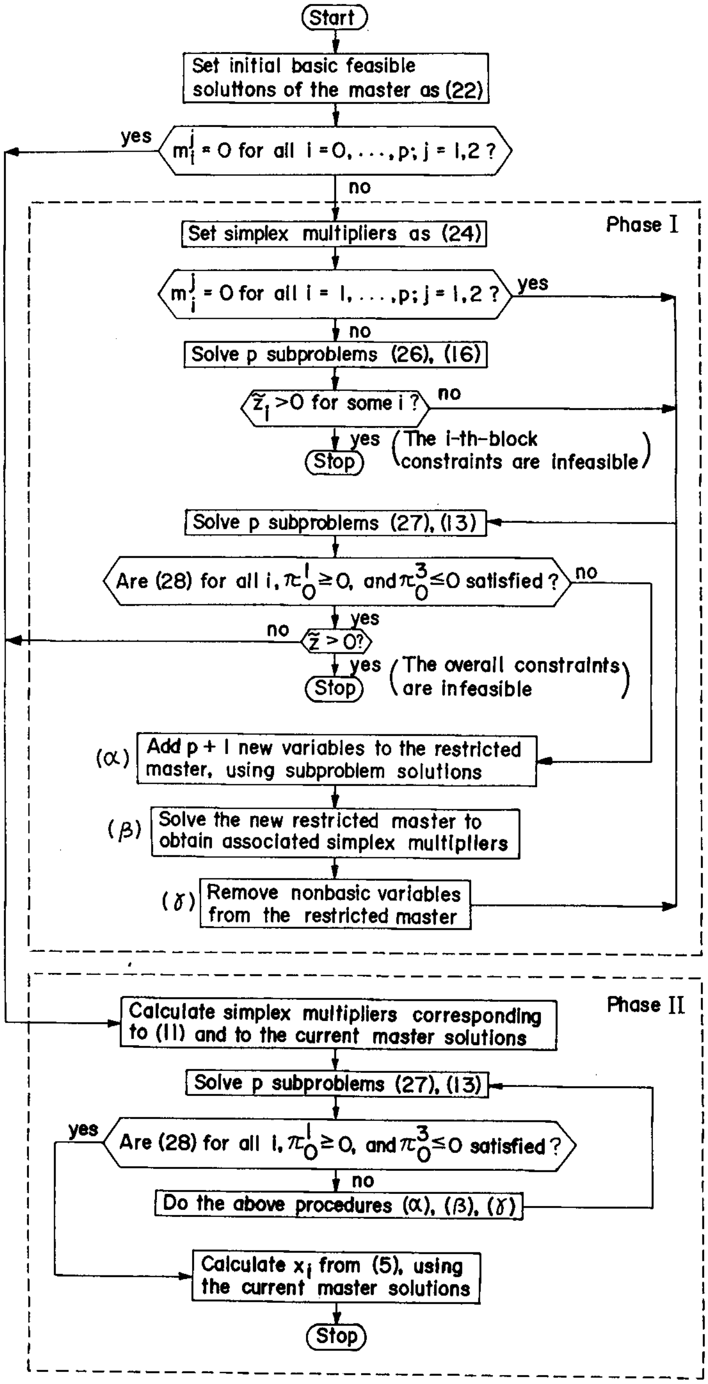

Fig. 1. Flow diagram for the overall procedure.

p, the whole of the above procedure is skipped to begin with the second phase. Together with the second phase, the overall flow of the iteration scheme is summarized in Fig. 1.

4. Conclusion

A difficulty is pointed out in the original decomposition algorithm of Dantzig-Wolfe. Our new algorithm, called the two-phase decomposition algorithm, has succeeded in overcoming the difficulty. This algorithm has a distinct feature that, except for the case of irregularity such as cycling,[10] it always yields a final solution without any a priori knowledge or insight into the problem. However, if some a priori information is available, its use is generally desirable in order to omit redundant steps which otherwise might exist in the computation.

A computer code of the algorithm, named DCOMPS, is completed by the authors, and is currently available at the Scientific Subroutine Library of the Data Processing Center of Kyoto University.

Acknowledgment

The authors would like to acknowledge the helpful discussions of Dr. N. Sannomiya, Lecturer of Kyoto University.

References

1) L. S. Lasdon: "Optimization Theory for Large Systems," Macmillan, New York (1970).
2) G. B. Dantzig and P. Wolfe: "Decomposition Principle for Linear Programs," Operations Research, **8**, 101 (1960).
3) G. B. Dantzig and P. Wolfe: "The Decomposition Algorithm for Linear Programs," Econometrica, **29**, 767 (1961).
4) J. B. Rosen: "Primal Partition Programming for Block Diagonal Matrices," Numerische Mathematik, **6**, 250 (1964).
5) A. C. Williams: "A Treatment of Transportation Problems by Decomposition," J. Soc. Ind. Appl. Math., **10**, 35 (1962).
6) H. Itakura and Y. Nishikawa: "Large-Scale Linear Programming Problems—Algorithms and Applications—," Rep. Prof. Group Autom. Control (Inst. Elect. Engrs. Japan), No. AC-74-8 (1974, in Japanese).
7) H. Künzi: "Numerical Methods of Mathematical Optimization with ALGOL and FORTRAN Programs," Academic Press, New York (1968).
8) Y. Sekine: "Mathematical Programming II," Iwanami, Tokyo (1968, in Japanese).
9) W. Orchard-Hays: "Advanced Linear-Programming Computing Techniques," McGraw-Hill, New York (1968).
10) G. B. Dantzig: "Linear Programming and Extensions," Princeton University Press, Princeton (1963).