Business Applications

D. TEICHROEW, Editor

Decomposition Programming: An Analysis of Matrix Substructure

EARL J. BELL
University of Washington, Seattle, Washington*

A petroleum blending problem was analyzed in order to compare the primal and primal-dual decomposition algorithms. In the course of the analysis, a substructure was discovered which has relevance to the relative performance of the two algorithms and to their absolute performance as compared with a standard primal-Simplex solution without decomposition.

In the literature on mathematical programming, examples abound of special algorithms designed to take advantage of particular matrix structures. In this paper, by macro structure we mean a pattern exhibited by a matrix when arranged into submatrices, while micro structure is reserved for the nature of the submatrices themselves. Thus, an identity matrix is a case of micro structure because of the diagonal terms and also because there are only zeros and ones. Similarly, a null vector is an illustration of micro structure. The algorithms which employ the Decomposition Principle illustrate the use of macro structure [1, 2], while the transportation algorithms illustrate the use of micro structure.

The point of this paper is to show that the efficiency of decomposition algorithms can be improved by considering the micro structure in more detail. To some extent this already exists if one allows that the subprograms may be of the network or transportation type, or for that matter even decomposition programs themselves.

* Department of Finance and Statistics, School of Business Administration.

Our vehicle for this exposition is a linear programming formulation of a petroleum blending problem[1] given below as problem (1). Three types of blends are considered whose technical properties are reflected by the lower six constraints, while crude oil availabilities are in the upper four rows.

PROBLEM (1). Maximize

$$.360(x_1 + x_2 + x_3 + x_4) + .889(x_5 + x_6 + x_7 + x_8) + 1.494(x_9 + x_{10} + x_{11} + x_{12})$$

subject to

$$\begin{aligned}
x_1 \quad + x_5 \quad + x_9 \quad &\leqq 3800 \\
x_2 \quad + x_6 \quad + x_{10} \quad &\leqq 2652 \\
x_3 \quad + x_7 \quad + x_{11} \quad &\leqq 4081 \\
x_4 \quad + x_8 \quad + x_{12} \quad &\leqq 1300 \\
-2x_1 + x_2 - 3x_3 + 13.5x_4 \quad &\leqq 0 \\
14x_1 + 3x_2 - 6x_3 + 15x_4 \quad &\geqq 0 \\
-2x_5 + x_6 - 3x_7 + 13.5x_8 \quad &\leqq 0 \\
16.5x_5 + 2x_6 - 4x_7 + 17x_8 \quad &\geqq 0 \\
-2x_9 + x_{10} + 3x_{11} + 13.5x_{12} &\leqq 0 \\
7.5x_9 - 7x_{10} - 13x_{11} + 8x_{12} &\geqq 0 \\
x_1, x_2, x_3, x_4, \quad x_5, x_6, x_7, x_8, \quad x_9, x_{10}, x_{11}, x_{12} &\geqq 0.
\end{aligned}$$

Problem (1) is an example of the familiar decomposition, block-diagonal

PROBLEM (2). Maximize

$$\begin{aligned}
c_1x_1 + c_2x_2 + \cdots + c_Nx_N & \\
A_1x_1 + A_2x_2 + \cdots + A_Nx_N &\leqq b_0 \\
B_1x_1 &\leqq b_1 \\
B_2x_2 &\leqq b_2 \\
\ddots & \quad \vdots \\
B_Nx_N &\leqq b_N \\
x_1, x_2, \cdots, x_N &\geqq 0.
\end{aligned}$$

where the A_j-matrices have m_0 rows and n_j columns, the B_j-matrices have m_j rows and n_j columns. The vectors c_j and x_j have n_j elements and the b_j vectors have m_j elements. Thus, problem (1) has $m_0 = 4$, $m_1 = m_2 = m_3 = 2$, $n_1 = n_2 = n_3 = 4$, and here $N = 3$.

By using the Decomposition Principle, instead of solving directly as a standard linear program, we can reduce any problem of the type (2) to an equivalent problem (3) which has $(m_0 + N)$ rows, new variables, and columns of coefficients which are linear tranformations of non-negative basic or homogeneous solutions of the subsystems $B_jx_j \leqq b_j$. This is accomplished by a change of variable $x_j = \sum_k x_j^k \lambda_j^k$ $(j = 1, \cdots, N)$ due to the fact that any point in a closed, polyhedral, convex set can be written as a convex combination of its extreme points (basic solution) plus non-negative multiples of a finite number of extreme rays (homogeneous solutions).[2]

[1] Reference [3, pp. 62–82]; see also [4, pp. 551–553], and [6].
[2] Reference [5, pp. 448–454].

PROBLEM (3). Maximize

$$\sum_{r=1}^{R} \bar{c}_1{}^r\lambda_1{}^r + \sum_{s=1}^{S} \bar{c}_2{}^s\lambda_2{}^s + \cdots + \sum_{t=1}^{T} \bar{c}_N{}^t\lambda_N{}^t$$

$$\sum_r \bar{A}_1{}^r\lambda_1{}^r + \sum_s \bar{A}_2{}^s\lambda_2{}^s + \cdots + \sum_t \bar{A}_N{}^t\lambda_N{}^t \leqq b_0$$

$$\sum_r \delta_1{}^r\lambda_1{}^r = 1$$

$$\sum_s \delta_2{}^s\lambda_2{}^s = 1$$

$$\vdots$$

$$\sum_t \delta_N{}^t\lambda_N{}^t = 1$$

where $\bar{c}_j{}^k = c_jx_j{}^k$, $\bar{A}_j{}^k = A_jx_j{}^k$, $\sigma_j{}^k = \begin{Bmatrix}1\\0\end{Bmatrix}$ if $x_j{}^k$ is a $\begin{Bmatrix}\text{basic}\\\text{homogeneous}\end{Bmatrix}$ solution of $B_jx_j \leqq b_j$, $x_j \geqq 0$. The summations are taken over all extreme points and extreme rays of the several polyhedra. The $\lambda_j{}^k$ are the system variables and are required to be non-negative; the other symbols are coefficients, though they are generated, not input.

Decomposition algorithms generate coefficient columns for problem (3) as needed by solving subprograms whose objective functions are linear (in the case of the primal algorithm) or quotients of linear functions (in the case of the primal-dual algorithm). Clearly, the efficiency of the algorithms depends largely upon the efficiency in generat- the subprogram solutions. We have further comment upon this later in the paper when we contrast the performance of the two algorithms on this problem.

In terms of the system in problem (2) we notice several features which distinguish our blending problem: (a) the b_j vectors are null vectors, with the exception of b_0, (b) all of the A_j matrices are identity matrices, and (c) the c_j vectors are sum vectors multiplied by constants.

The decomposition algorithms exploit the macro structure; the question now is: can the algorithms be improved by taking explicit account of the micro structure represented by the three features?

(a) When each of the b_j vectors ($j = 1, \cdots, N$) is a null vector, the only basic solution of $B_jx_j \leqq b_j$ is the trivial solution $x_j = 0$. This means that all of the N rows $\sum_k \delta_j{}^k\lambda_j{}^k = 1$ ($j = 1, \cdots, N$) in (3) may be dropped, thus further reducing the number of rows and, hence, the basis size in problem (3).[3] When N is quite large, as if we were considering a large number of blends, this reduction could be helpful. It will still be necessary to solve the subprograms, however. This reduction will work for either the primal or primal-dual decomposition algorithms.

(b) When the A_j matrices are identity matrices the transformations $A_jx_j{}^k = \bar{A}_j{}^k$ are obviously unnecessary and the coefficient columns can be added to the master program just as they are obtained from the subprograms. More importantly, in order to extract the optimal solution in terms of the original x_j vectors, it is generally necessary to maintain the solution $x_j{}^k$ from the subprogram that generated the coefficient column that has just been added to the master program. The reason for this is contained in the relationship of the x_j vectors to the $x_j{}^k$ and $\lambda_j{}^k$, $x_j = \sum x_j{}^k\lambda_j{}^k$ where the only $\lambda_j{}^k$ and $x_j{}^k$ of interest are those in the terminal basic solution of problem (3). Clearly the $x_j{}^k$ are available if they are stored after being generated, as in the revised Simplex method, before they are transformed by pivoting. This feature also holds for both the primal and primal-dual decompsition algorithms.

(c) Ordinarily, by itself, this feature cannot be exploited, but in conjunction with (a) and (b) it can lead to a substantial improvement in the primal-dual algorithm. As in the general primal-dual algorithm, the decomposition version needs a feasible solution for problem (4), the dual of problem (3).

PROBLEM (4). Minimize

$$\sigma \cdot b_0 + \sum_{j=1}^{n} \alpha_j$$

$$\sigma A_jx_j{}^k + \beta_j \geq c_jx_j{}^k \quad j = 1, \ldots, N.$$

$$\sigma \geqq 0.$$

PROBLEM (5). Maximize

$$\sum_{j=1}^{N} c_jx_j$$

$$\sum_{j=1}^{N} A_jx_j \leqq b_0, \qquad x_j \geqq 0 \quad j = 1, \cdots, N.$$

The recommended approach for general decomposition problems is to obtain optimal-dual variables associated with an optimal solution of problem (5) and from them obtain an $(m_0 + N)$ vector satisfying the dual constraints in problem (4).[4]

By virtue of (a), the sum rows of (3) are eliminated. Thus, we need only an m_0-vector σ satisfying:

$$\sigma A_jx_j{}^k \geqq c_j{}^kx_j{}^k \tag{6}$$

and when the A_j are identity matrices[5] this reduces to

$$\sigma x_j{}^k \geqq c_jx_j{}^k. \tag{7}$$

Clearly, to satisfy (7), it is sufficient to find a vector $\sigma \geqq c_j$ for all $j = 1, \cdots, N$.[5] Thus, when (c) prevails, σ may be determined at the time of input by simply determining which c_j vector has the largest constant and setting all the elements of σ at that value initially. Thus, for our blending problem

$$\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = 1.494.$$

This feature has two redeeming characteristics beyond avoiding a lot of computation:

(d) It establishes an upper bound on the maximum of

[3] In such cases all $\delta_j{}^k = 0$ except that $\delta_j{}^k = 1$ for the trivial basic solution which implies that its associated $\lambda_j{}^k = 1$. Hence the constraint may be dropped.

[4] See [1] for a detailed outline of this procedure.

[5] When the A_j are identity matrices, it must be the case that $m_0 = n_j$ for all j.

$\sum_{j=1}^{N} c_j x_j$ at the value of $\sigma \cdot b_0 = 1.494 \cdot 11{,}833 = 17{,}678$ (the optimal value is at 15,425).

(e) It furnishes an immediate indication concerning which subprogram to solve in order to obtain candidates for the initial restricted master program (in the blending problem start with the third subprogram).

Point (e) relates to the relative efficiency of the primal-dual and primal algorithms, at least as far as their performance on this problem is concerned. The primal-dual algorithm for general linear programs, and also for special network structures such as the capacitated transportation problem, are known to have a tighter selection criterion than the primal algorithms. This is because the primal-dual algorithm must move the solution towards primal feasibility while simultaneously causing the dual feasible solutions to move towards optimality and maintaining complementary slackness between primal and dual. This parallelism holds true in the decomposition algorithms as well. To illustrate the lower stringency of the primal decomposition algorithm, note that any candidate vector x_j^k from any of the subprograms which satisfies $(\sigma A_j - c_j) \cdot x_j^k < 0$ (where σ is the current vector of multipliers) can be introduced into the master program and cause an increase (under conditions of nondegeneracy) in the function to be maximized. In the blending problem when starting from a full slack basis $\sigma = (0, 0, 0, 0)$ and because all of the c_j vectors have no negative elements,[6] any $x_j^k \geq 0$ satisfying $B_j x_j \leqq 0$ also satisfies $(\sigma A_j - c_j)x_j^k < 0$. Because of the homogeneous nature of the solutions $(\sigma A_j - c_j)x_j^k$ can be made even more negative by multiplying by a positive scalar. Therefore, there is no way of distinguishing in this case, as in the usual primal decomposition algorithm, which candidate is "best" in the sense of the usual Simplex criterion of minimal $(\sigma A_j - c_j)x_j$.

In the blending problem this means that each of the three subprograms could furnish a possible candidate for the first master pivot operation. In anticipation of this "indeterminate" situation we might suspect that candidates drawn from the subprograms with the higher c_j values—that is, subprograms 2 and 3 with values .889 and 1.494—would tend to supply an optimal solution. Post-optimal analysis of this problem in fact confirms this suspicion. It turns out that x_1, x_2, x_3, x_4 can never appear in any optimal solution. This means that any candidate from subprogram 1 which is introduced into the master program must later be rejected.

Thus, there are three major solution paths, depending upon which subprogram furnishes the first candidate. The number of candidates which had to be generated in each case was 11, 7, and 4, respectively. We remark again that the primal decomposition algorithm, by itself, has no means of determining which of these paths is "best." Only after the fact can we confirm our earlier suspicion. It is now interesting to compare these numbers with the four candidates generated by the primal-dual algorithm, starting with $\sigma = 1.494$ (1, 1, 1, 1), or alternatively the total of five candidates with the initial $\sigma = (0, 0, 0, 0)$.

Hence the primal-dual algorithm was at least as efficient as the primal, two-phase algorithm in spite of the fact that no Phase I was required because a full slack basis was initially available. This illustrates the point that the apparent disadvantage of using a full artificial basis may be deceptive and that methods employing artificial variables, such as primal-dual methods, may be very efficient when few of the available slacks will appear in optimal solutions.

One final point of computational interest: even for this small problem the total computing time of each decomposition algorithm compared favorably with the time to solve as a standard linear program. This was the case in spite of the fact that both decomposition algorithms were "simulated," not automatically programmed. The Dartmouth College Computation Center is to be commended for its excellent time sharing system which provided the flexibility to carry out the many offline calculations without "turnaround" problems, and thus avoid the necessity of large and complex automated computer codes.

It is interesting to note that in making several simplifying assumptions in order to reduce the basis size of this problem, Garvin could make the statement that "the matrix of our problem has *structure* . . . characteristic of blending problems. . . . It is natural to inquire whether it is possible to take advantage of the matrix structure. . . ." Ironically, within six months the Decomposition Principle had been published (in 1960).

We hope, indeed, that we have shown that this problem has "structure," both *macro* and *micro*. Attention to both kinds of structure can pay dividends, particularly when dealing with calculations of routine frequency.

RECEIVED JULY, 1966; REVISED JUNE, 1967

[6] Notationally $x_j \geq 0$, instead of $x_j \geqq 0$, excludes $x_j = 0$.

REFERENCES

1. BELL, E. J. Primal-dual decomposition programming. Rep. ORC 65-23, Operations Research Center, U. of California, Berkeley, Calif., August, 1965; also in Preprints of the Proceedings of the Fourth International Conference on Operational Research, Boston, Mass., Session 1, pp. 1-30.
2. DANTZIG, G. B., AND WOLFE, P. The decomposition algorithm for linear programs. *Econometrica 29*, 4 (Oct. 1964).
3. GARVIN, W. W. *Introduction to Linear Programming*. McGraw-Hill Book Co., Inc., New York, 1960.
4. CHARNES, A. AND COOPER, W. W. *Management Models and Industrial Applications of Linear Programming, Vol. II*. John Wiley, New York, 1961.
5. DANTZIG, G. B. *Linear Programming and Extensions*. Princeton U. Press, Princeton, N. J., 1963.
6. CHARNES, A., COOPER, W. W., AND MELLON, B. Blending aviation gasolines—A study in programming interdependent activities. *Econometrica 20* (April, 1952), 135-159.