DECOMPOSITION OF TWO-BLOCK OPTIMIZATION PROBLEMS

E. Nurminski

June 1981
WP-81-73

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

Decomposition of two-block optimization problems

E.Nurminski

ABSTRACT

This paper is concerned with the problem of balancing an optimization model consisting of two submodels. The submodels are represented by separate linear programming problems and are linked by dependance on common resources, or by the presence of the same variables in both of them.

The method for coordinating the activities of submodels, in order to reach an overall optimum, is based on the direct exchange of proposals between submodels. Computational improvements in comparison with the conventional master-subproblems scheme are shown.

1. Introduction

Here we consider some improvements in solving the problem of balancing an optimization model consisting of two submodels linked by the use of common resources, or by dependance on common variables.

The general framework of the approach follows decomposition ideas of large-scale mathematical programming. In the beginning of the 60's Dantzig and Wolfe proposed the widely known decomposition principle (Dantzig61a). The nature of this conception is to replace the initial large-scale problem by a sequence of smaller problems, each representing different sections of the initial one, with some coordinating master problem balancing the separate solutions of the subproblems.

However, further computational experiments with this principle provided varied results. For linear programming it was shadowed by developments in sparsity techniques. On the other hand, the decomposition principle, as origi-

nally formulated, turned out to be too slow to serve as the theoretical foundation for distributed decision making processes(Dirickx79a).

The main problem with this algorithm is that it often requires many cycles between subproblems and the coordinating master problem.

The frequently observed computational behavior of the Dantzig-Wolfe decomposition principle consists of rather rapid improvement on the initial iterations of the optimization process, with slow convergence on the final stage. The latter takes many cycles between subproblems and master problem, and is the main source of dissapointment for those who unsuccesfully tried to use the Dantzig-Wolfe decomposition principle. It is certainly the main obstacle barring the wide application of decomposition ideas in large-scale mathematical programming. A decrease in the number of cycles between master and subproblems does not necessarily decrease computations, but for large-scale problems it reduces the most critical part-exchange with the secondary memory, or exchange of the coordinating information between subproblems and/or master problem. Improvements in the decomposition algorithm may also bring new insights into mechanisms of distributed decision making.

2. Formulation of the problem

Consider a two-block problem with linking variables:

$$\begin{aligned} &\min\ (c_A z_A + c_B z_B) \\ &A_A z_A + B_A x \le b_A \\ &A_B z_B + B_B x \le b_B \end{aligned} \tag{1}$$

where z_A and z_B can be viewed upon as internal variables of subproblems or submodels A, B and the common variable x links these two subproblems. For a fixed x however the whole problem (1) splits into two independent problems.

$$f_A(x) = \min \{c_A z_A\} \tag{2}$$

$$A_A z_A \leq b_A - B_A x$$

and

$$f_B(x) = \min_{A_B z_B \leq b_B - B_B x} \{ c_B z_B \} \tag{3}$$

each of them requiring a smaller commitment of computer resources.

The methods of direct or resource-directive decomposition tend to consider (1) as a problem of the kind

$$v^* = \min_x \{ f_A(x) + f_B(x) \} \tag{4}$$

where $f_A(x)$ and $f_B(x)$ are given by (2) and (3) respectively.

Indirect or dual decomposition is based on dualization of certain key constraints in a linear programming problem. Partial or complete dualization of extremal problems often allows the decomposition of an initially large-scale problem into smaller ones with some coordinating program of moderate size. This idea also underlies the decomposition principle of Dantzig and Wolfe.

To demonstrate this, notice that for the problem (1), or equivalently (4), the key constraint is the convention that variable x must have the same value in functions (2) and (3).

By explicit formulation of this constraint for the problem (4) and subsequent dualizing of the resulting constrained problem one can obtain the dual problem

$$v^* = - \min_p \{ f_A^*(p) + f_B^*(-p) \} \tag{5}$$

where $f_A^*(p)$ is the conjugate of function $f_A(x)$

$$f_A^*(p) = \sup_x \{ p\ x - f_A(x) \} \tag{6}$$

and $f_B^*(p)$ is the conjugate of function $f_B(x)$

Dual variables p are customarily interpreted as prices for linking variables x. Computation of the values $f_A^*(-p), f_B^*(p)$ can be interpreted as a local optimization in subproblems A, B for given prices p provided by master problem:

$$-f_A^*(-p) = \min_{A_A z_A + B_A x \le b_A} \{ c_A z_A + p\, x \} \tag{7}$$

$$-f_B^*(p) = \min_{A_B z_B + B_B x \le b_B} \{ c_B z_B - p\, x \} \tag{8}$$

It is useful to notice that $f_A^*(-p)$, $f_B^*(p)$ are convex functions with subgradients $-x_A^*$, x_B^* equal to the x-components of the solutions of (7)-(8). In other words subgradients of the functions $f_A^*(-p), f_B^*(p)$ are proposals of the local subproblems in terms of the Dantzig-Wolfe decomposition scheme.

Problem (5) can be solved by a number of methods updating prices p using values of functions $f_A^*(-p)$, $f_B^*(p)$ and their subgradients and the Dantzig-Wolfe decomposition method can be interpreted as a cutting plane algorithm (Kelley60a) applied for optimization of nondifferentiable function $f(p) = f_A^*(p) + f_B^*(-p)$.

Conceptually the cutting plane method consists of maintaining the set $P = \{p^1, p^2, \cdots\}$ of the approximate solutions of the problem (5) and solving on each iteration the linear auxiliary problem

$$\min_{g(p^k)\, p - v \le g(p^k)\, p^k - f(p^k),\ p^k \in P} v \tag{9}$$

where $f(p)$ is a function to be minimized and $g(p^k) \in \partial f(p^k)$ is the subgradient set of the function $f(p)$ at the point p^k.

The solution $(\bar{v}, \bar{p})$ of (9) provides a lower bound $\bar{v}$ for the optimum value of (5) and $\bar{p}$ is added to the set P for performing the next iteration. Some authors

(Topkis70a) considered the variants of the scheme with exclusion of some points from the set P which correspond to nonactive constraints in (9).

The problem (9) forms the master problem of the Dantzig-Wolfe decomposition principle and problems (7)-(8) are subproblems of this scheme reacting to the prices provided by (9).

There is a number of ways to make use of the master problem (9). If (9) is solved for every set of proposals then it is called restricted master problem(Lasdon70a). On the other hand it is possible to make only a few iterations toward optimality in (9) and then turn to subproblems (7)-(8) for the generation of new proposals.

3. Directions for improvements

Convergency properties of the Dantzig-Wolfe decomposition principle and its practical significance have been widely discussed. This scheme, to its advantage, has a nice clear concept of trade-offs between the master problem and subproblems, it appeals to economic interpretations and has inspired many discussions on the mechanisms of optimal decision making.

However, from the computational point of view, this method did not have a good reputation until recently. This may be attributed to the slow convergence of cutting plane method (9) underlying the Dantzig-Wolfe decomposition principle and it is possible to improve the performance of this scheme by replacing master problem (9) with the faster general methods of nondifferentiable optimization (See f.i. Lemarechal78a)

On the other hand, it is also possible to exploit the specific structure of the problem (4) or equivalently (5).

3.1. Direct exchange of proposals

The particular feature of problem (5) is that the objective function there is the sum of two functions each of them referring to the different subproblems. It allows the direct interaction between subproblems A and B to be organized, and it incorporates into the optimization process, not only the information about the primal solutions of (7)-(8), but also dual information associated with these problems.

Mathematically, it consists of introducing two new problems

$$\min_{p} \{ -x_A^* p + f_B^*(p) \} \tag{10}$$

and

$$\min_{p} \{ x_B^* p + f_A^*(-p) \} \tag{11}$$

where x_A^*, x_B^* are some proposals from subproblems A and B respectively, obtained from some previous iteration.

The expressions (10) and (11) provide better entries for the set P then those generated by the rather crude approximation (9) of the original problem (5) and, as a result, the convergence of the balancing procedure is speeded up. As shown by numerical experiments conducted with this approach on the final stages of the optimization process, the solution of (10) or (11) often simply coincides with the overall optimum in (5).

From the technical point of view, the solution of (10) and (11) amounts to solving a slight modification of the problem (7)-(8). Problem (10) for instance, can be looked upon as a dual of the problem

$$\min \{ c_B z_B \}$$

$$A_B z_B + B_B x \leq b_B \tag{12}$$

$$x = x_A^* \tag{13}$$

with the solution $\bar{p}$ of (10) corresponding to optimal dual multipliers for the constraints (13).

Problems (12)-(13) can be interpreted as a result of a direct exchange of proposals between subproblems. In this case the proposal x_A^* generated in one subproblem (A in this case) is used as a constraint in another subproblem (B in this case).

There are several ways of using the solutions to (10)-(11). The simplest one would be to add them to the set P and initiate through (9) the new round of exchanges of proposals between subproblems.

Another way to use prices generated in (10) is to send them directly to subproblem (7) to generate a new proposal in subproblem A, and to use the new proposal x_A^* in (10), and so on.

Symmetrically it can be done with problems (8),(11).

Unfortunately it is not an absolutely safe approach. Computational experience showed that even if it speeds up the convergence on the initial iterations, it is still slow on the final stage--a frequently reported shortcoming of the Dantzig-Wolfe decomposition principle. The reason for this is that the prices generated through (10)-(11) may produce in (7)-(8) in the final stage of the optimization process, proposals which are not active in (9). Then it becomes necessary to invoke (9) to break a deadlock and generate new set of prices. If the master problem (9) is being invoked on every iteration this scheme is becoming equivalent to the conventional Dantzig-Wolfe decomposition principle with the typical computational behavior.

3.2. Master functions

Further improvements may consist of delegating some of the master functions to subproblems.

It may be observed that the proposals generated in one subproblem may be infeasible for another one, that is, the problem (12)-(13) may become infeasible due to constraint (13). If so, the dual problem (10) is unbounded and it is necessary to impose some bounds on prices p to get a solution. It is clear then that this solution would be far away from the solution of (5) anyway, and so bring very little new information about the solution to this problem.

The general idea to overcome this shortcoming is to distribute the constraints of the problem (9) between subproblems (10)-(11) to decrease the variation in prices generated in these subproblems.

Modified problems (10),(11) may have a form

$$\begin{gathered} \min\{ -x_A^* p + f_B^*(p) \} \\ g(p^k)\, p \le \bar{v} + g(p^k)\, p^k - f(p^k) \\ p^k \in P \end{gathered} \tag{14}$$

$$\begin{gathered} \min\{ x_B^* p + f_A^*(-p) \} \\ g(p^k)\, p \le \bar{v} + g(p^k)\, p^k - f(p^k) \\ p^k \in P \end{gathered} \tag{15}$$

where $\bar{v}$ is an upper estimate of the overall optimum in (5) and $g(p^k) = x_B^k - x_A^k$ - is a subgradient of the function $f(p) = f_A^*(-p) + f_B^*(p)$. at the point $p = p^k \in P$. The upper estimate $\bar{v}$ may be updated during the optimization process using the values of the function $f(p)$ which have been already computed.

The solution of (14),(15) may be used in the primal subproblems (7)-(8), generating there new proposals which are then substituted into (14),(15) and so on, as proposed in Section 3.1.

This modification, however, does not annul completely the need to use the master problem (9) and, from time to time, it might be necessary to invoke the

master problem (9). Computational experience discussed below shows that the need to call master problem (9) is, nevertheless, much lower, and in many cases the master problem (9) is not called at all.

4. Numerical example

Here we consider, in a more detailed way, an application of the proposed idea to the mini-scale problem used by E.M.Beale(Beale63a) to illustrate his method of parametric decomposition.

This mini-problem has three linking variables which link together two subproblems shown below each with 6 internal variables and 3 constraints.

Subproblem A.

$$\min \{ z_4 + z_5 + 5.z_6 - 1.5x_1 - x_2 - 0.5x_3 \}$$
$$z_1 + z_4 - z_5 - x_1 + 2.x_3 = 4.$$
$$z_2 + z_4 - z_5 - 2.z_6 + x_2 - x_3 = 0.$$
$$z_3 - z_4 + z_5 - z_6 + x_1 + 3.x_2 = 0.$$

Subproblem B.

$$\min \{ 2.z_1 + z_5 + z_6 - 1.5x_1 - x_2 - 0.5x_3 \}$$
$$z_1 - z_4 - z_5 - z_6 + x_1 + 2.x_2 - 2.x_3 = 2.$$
$$z_2 - z_4 - z_5 + x_1 - x_2 + x_3 = 4.$$
$$z_3 - z_5 - 2.z_6 - x_1 - x_2 + x_3 = 2.$$

In each subproblem the variables $z_1 - z_6$ are internal variables and $x_1 - x_3$ are links.

The Dantzig-Wolfe decomposition method used for the comparison included such features as a restricted master problem and generation of only one new proposal from every subproblem for every new set of prices provided by the master problem. None of the advanced features of the Dantzig-Wolfe decomposition method developed, for instance, in(Loute81a) was implemented.

The performance of this variant of the Dantzig-Wolfe decomposition principle is shown below.

nn	value of master problem	prices		
		p(1)	p(2)	p(3)
1	-0.209170d+03	0.7404d+00	0.1038d+02	0.5969d+01
2	-0.156621d+03	-0.1449d+01	0.1567d+02	0.1268d+01
3	-0.156621d+03	-0.1449d+01	0.1710d+01	0.1268d+01
4	0.925299d+01	0.5888d+00	0.1271d+02	-0.6736d+00
5	0.925299d+01	0.5888d+00	0.7810d+01	-0.6736d+00
6	0.925299d+01	0.5888d+00	-0.1027d+01	-0.6736d+00
7	0.103536d+02	0.5761d+00	0.2104d+01	-0.5390d+00
8	0.124664d+02	0.5517d+00	0.1835d+01	-0.2806d+00
9	0.179431d+02	0.4139d+00	0.1985d+01	-0.5059d+00
10	0.179431d+02	0.4139d+00	0.2048d+01	-0.5059d+00
11	0.184383d+02	0.4139d+00	0.1918d+01	-0.3767d+00
12	0.184982d+02	0.5001d+00	0.2000d+01	-0.4999d+00
13	0.185006d+02	0.5001d+00	0.2000d+01	-0.5000d+00

In contrast the algorithm proposed above took only one round of direct proposal exchange.

The solution was reached through the following sequence of steps:

Step 1

Subproblem A was solved with zero initial prices and produced the following results (only the column section of the correspondent output is shown):

problem name BEAL.A

objective value -1.200000000d+01

section 2 - columns

number	.colum.	at	...activity...	.obj gradient.	.reduced cost.
1	col....1	ll	0. d+00	0. d+00	5.00000d-01
2	col....2	ll	0. d+00	0. d+00	5.00000d-01
3	col....3	ll	0. d+00	0. d+00	2.00000d+00
4	col....4	bs	4.50000d+00	1.00000d+00	0. d+00
5	col....5	ll	0. d+00	1.00000d+00	2.00000d+00
6	col....6	ll	0. d+00	5.00000d+00	2.00000d+00
7	link...1	bs	9.50000d+00	0. d+00	0. d+00
8	link...2	ll	0. d+00	0. d+00	5.50000d+00
9	link...3	bs	4.50000d+00	0. d+00	0. d+00
10	rhs.....	eq	-1.00000d+00	0. d+00	1.20000d+01

Step 2

The proposal from the subproblem A (values of linking variables) was directed to the subproblem B as a constraint. Subproblem B was solved and the correspondent row section of the solution is shown below:

problem name BEAL.B

objective value -6.500000000d+00

section 1 - rows

...row..	at	...activity...	slack activity	.dual activity	..i
cost_row	bs	-6.50000d+00	6.50000d+00	-1.00000d+00	1
row....2	eq	2.00000d+00	0. d+00	0. d+00	2
row....3	eq	4.00000d+00	0. d+00	-1.00000d+00	3
row....4	eq	2.00000d+00	0. d+00	0. d+00	4
n.row..1	eq	9.50000d+00	0. d+00	-5.00000d-01	5
n.row..2	eq	0. d+00	0. d+00	-2.00000d+00	6
n.row..3	eq	4.50000d+00	0. d+00	5.00000d-01	7

Additional rows in this subproblem represent extra constraints which appeared in the problem (12)-(13) and dual variables corresponding to

these constraints are the solution to the problem (10).

Step 3

Dual variables related to additional constraints in subproblem B are used as prices for linking variables in subproblem A.

The column section of the solution shows that subproblem A generated the same value for linking variables:

problem name BEAL.A

objective value -1.450000000d+01

section 2 - columns

number	.column.	at	...activity...	.obj gradient.	.reduced cost.
1	col....1	ll	0. d+00	0. d+00	5.00000d-01
2	col....2	ll	0. d+00	0. d+00	1.00000d-00
3	col....3	ll	0. d+00	0. d+00	2.50000d-00
4	col....4	bs	4.50000d-00	1.00000d+00	0. d+00
5	col....5	ll	0. d+00	1.00000d+00	2.00000d+00
6	col....6	ll	0. d+00	5.00000d+00	5.00000d-01
7	link...1	bs	9.50000d+00	-5.00000d-01	0. d+00
8	link...2	ll	0. d+00	-2.00000d+00	5.50000d+00
9	link...3	bs	4.50000d-00	5.00000d-01	0. d+00
10	rhs.....	eq	-1.00000d-00	0. d+00	1.45000d-01

The fact that the solution of this problem coincides with the previous proposal from subproblem A means that it is optimal.

Stop

Of course a good initial point accounts partly for such rapid convergence but initial prices were rather far from the optimal ones.

Another reason for such good performance is the right order of exchanges: we started with the subproblem A, generated the proposal in this subproblem, sent it to B, generated the prices there, sent them back to A, generated a new proposal from A and stopped. If we started from B the progress would have not been so spectacular.

References

Beale63a. E.M.L. Beale, "The simplex method using pseudo-basic variables for structured linear programmng problems," pp. 133-148 in *Recent advances in mathematical programming*, ed. R.L. Graves and P. Wolfe,McGraw-Hill Book Company, N.Y. (1963).

Dantzig61a. G.B. Dantzig and P. Wolfe, "The decomposition algorithm for linear programming," *Econometrica* **29** pp. 767-778 (1961).

Dirickx79a. Y.M.I. Dirickx and L.P. Jennergren, *System analysis by multilevel methods: with applications to economics and management*, John Wiley & Sons (1979).

Kelley60a. J.E. Kelley, "The Cutting Plane Method for Solving Convex Programs," *Journal of the Society for Industrial and Applied Mathematics* **8**(4) pp. 703-712 (1960).

Lasdon70a. L.S. Lasdon, *Optimization Theory for Large Systems*, Macmillan, New York (1970).

Lemarechal78a. C. Lemarechal and R. Mifflin, *Nonsmooth optimization: Proceedings of a IIASA Workshop, March 28-April 8, 1977*, Pergamon Press (1978).

Loute81a. E. Loute and J.K. Ho, "An advance implementation of the Dantzig-Wolfe decomposition algorithm for linear programming," pp. 425-460 in *Large-Scale Linear programming, Proceedings of a IIASA Workshop,2-6 June 1980, Volume 1*, ed. G.B. Dantzig, M.A.H. Dempster and M.J. Kallio, (1981).

Topkis70a. D.M. Topkis, "Cutting plane method without nested constraint set," *Operation Research* **18**(3) pp. 404-413 (1970).