II-APPROXIMATION AND DECOMPOSITION OF LARGE-SCALE PROBLEMS

E.A. Nurminski
International Institute for Applied Systems Analysis, Austria

RR-81-11
June 1981

Reprinted from *Optimization and Optimal Control*
(A. Auslender, W. Oettli, and J. Stoer, editors)

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
Laxenburg, Austria

Reprinted with permission from A. Auslender, W. Oettli, and J. Stoer, editors, *Optimization and Optimal Control*: Proceedings of a Conference held at Oberwolfach, March 16–22, 1980, Springer-Verlag, Berlin, 1981, pages 79–88.

FOREWORD

Systems-analysis problems are frequently large-scale problems, a fact that often forces the calculations supporting their solution to have such a large scale as to cause significant difficulty. One of the possible responses is to decompose the large-scale calculations into parts that offer less calculational difficulty; however, this procedure brings into the analysis the additional process of coordinating the solutions to the simpler subproblems.

This decomposition of large-scale problems and coordinating the solutions of the resulting subproblems is a recurring theme in systems-analysis applications. It has motivated many theoretical and practical studies, both at IIASA and elsewhere.

This paper describes a new approach to this decomposition/coordination problem based on the techniques of nondifferentiable optimization. It is based on an approximation – that the author calls the II-approximation – of functions that characterize the decomposed subproblems, and it offers a computationally efficient algorithm.

ANDRZEJ WIERZBICKI
Chairman
System and Decision Sciences Area

Π-APPROXIMATION AND DECOMPOSITION OF LARGE-SCALE PROBLEMS

E.A. Nurminski
International Institute for
Applied Systems Analysis
A-2361 Laxenburg, Austria

ABSTRACT

Partial or complete dualization of extremum problems often allows the decomposition of initially large-scale problems into smaller ones with some coordinating program of a moderate size. This idea underlies many known schemes of decomposition and the common difficulty often encountered is the problem of restoring the solution of the primal problem. The main idea of this paper is to present an algorithm for providing an easy way of obtaining the solution of the initial primal problem keeping all advantages of the dual one.

The algorithm described here is based on the particular approximation of the aggregated function representing the decomposed way of solving the extremum problem. This approximation looks like a dual problem and its remarkably simple structure makes it possible to solve a corresponding extrememem problem in a few iterations.

1. INTRODUCTION

The effective solution of large-scale problems is possible only if these problems have a specific structure both in theory as well as in application. In many applications the original problem can be reformulated in a two-stage way

$$\min_{x \in X} \min_{z \in Z(x)} f(x,z) \tag{1}$$

where the internal problem of computing

$$\min_{z \in Z(x)} f(x,z) = F(x) \tag{2}$$

is easy to solve for fixed values of x and takes care of the vast majority of the variables leaving unknown only a small number of

the linking variables. If the optimal values for these variables x^* were known in advance then the solution of (1) would be equivalent to solving (2) for $x = x^*$ and would be easy to perform. However the problem of fixing the correct values for linking variables is not a trivial one. The aggregated function $F(x)$ has poor analytical properties so the application of many procedures becomes dubious or unjustified or they fail to reach an optimum.

During the last few years a number of techniques have been proposed for handling extremum problems with relaxed requirements for analytical properties of the objective function and/or constraints. These methods performed quite well in a number of cases and also recent theoretical studies have shown some theoretical advantages of this approach even in classical cases such as linear programming (Khachyan 1979). Here we establish a few facts based on convex duality which provides certain new possibilities.

2. Π-APPROXIMATIONS

In this part we will establish an equivalence under quite general conditions of problems of minimizing convex functions and minimization of their particular approximations which are constructed in a way similar to the standard duality approach.

Let $F(x)$ be a closed convex function bounded from below. Let $F^*(g)$ denote its conjugate

$$F^*(g) = \sup_x \{xg - F(x)\} \quad . \tag{3}$$

Between $F(x)$ and $F^*(g)$ a well-known relationship exists:

$$F(x) = \sup_g \{xg - F^*(g)\} \quad ,$$

(Fenchel 1949).

It is interesting to look at the slightly different formula

$$\hat{F}(x) = \sup_{g \in \Pi} \{xg - F^*(g)\} \quad , \tag{4}$$

which defines a new function $\hat{F}(x)$. The properties of this function

strongly depend on characteristics of set Π. In the case that this set coincides with the whole space $\hat{F}(x) = F(x)$.

In the other extreme, if this set collapses to a single point $\Pi = \{0\}$ then

$$\hat{F}(x) = \sup_{g=0}\{xg - F^*(g)\} = \inf_x F(x) \quad .$$

Definition. Function $\hat{F}(x)$ given by expression

$$\hat{F}(x) = \sup_{\pi\in\Pi}\{\pi x - F^*(\pi)\} \quad ,$$

where

$$F^*(\pi) = \sup_x\{\pi x - F(x)\} \quad ,$$

is called the Π-approximation of $F(x)$.

Here we will give a few simple results concerning $\hat{F}(x)$. These theorems originally appeared in Nurminski (1979).

Theorem 1. If $F(x)$ is bounded from below:

$$\inf F(x) = f$$

and zero belongs to set Π then

$$\inf \hat{F}(x) = f \quad .$$

Proof. For any x

$$\hat{F}(x) = \sup_{\pi\in\Pi}\{\pi x - \sup_z\{\pi z - F(x)\}\} \leq$$

$$\sup_{\pi\in\Pi}\{\pi x - \pi x + F(x)\} = F(x) \quad .$$

On the other hand

$$\hat{F}(x) \geq 0x - \sup_x\{0x - F(x)\} = \inf_x F(x) \quad .$$

These two inequalities prove the theorem.

Theorem 2. If $F(x)$ is a closed convex and bounded from below and Π is an absorbing convex set, then any minimum of $\hat{F}(x)$ is a minimum of $F(x)$.

Proof. Let x^* be the minimum of $F(x)$. According to Theorem 1 $\hat{F}(x^*) = \inf F(x) = f$ and if Theorem 2 is not valid then $F(x^*) > f$. Then in product space $R \times X$ point (f,x^*) and closed set epi $F = \{(\theta,x): \theta \geq F(x)\}$ is strictly separable in the sense that for some $\varepsilon > 0$ vector p exists such that

$$-px^* + f + \varepsilon < -px + F(x) \quad , \tag{5}$$

for any x. Multiplying (5) by $a \in (0,1]$ and adding trivial inequality $f \leq F(x)$ we obtain

$$-\frac{a}{1+a}px^* + f + \frac{a}{1+a}\varepsilon < -\frac{a}{1+a}px + F(x) \quad .$$

Due to the absorbtion property of Π

$$\frac{a}{1+a}p = \bar{\pi} \in \Pi$$

for some $a > 0$ and $\bar{\varepsilon} = \frac{a}{1+a}\varepsilon > 0$. Then

$$\hat{F}(x^*) = \sup_x\{\pi x^* - \sup_z\{\pi z - F(x)\}\} \geq \bar{\pi}x^* - \bar{\pi}x^* + f + \bar{\varepsilon} = f + \bar{\varepsilon} > f$$

which contradicts the original definition.

Theorem 3. If the convex function $F(x)$ attains its minimum at point x^* and set Π is such that

$$\Pi \subset \partial F(x^*) \quad ,$$

then

$$\hat{F}(x) = \inf_x F(x) + \sup_{\pi \in \Pi} \pi(x - x^*) \quad . \tag{6}$$

Proof.

$$\hat{F}(x) = \sup_{\pi \in \Pi} \inf_z\{F(z) + \pi(z-x)\} =$$

$$\sup_{\pi\in\Pi} \inf_{z}\{F(z)-\pi(z-x^*)\} + \pi(x-x^*)\} \quad .$$

Under the conditions of the theorem

$$F(z) - \pi(z-x^*) \geq F(x^*) \quad ,$$

and the left side attains its minimum at $z = x^*$.

Theorems 2 and 3 provide an essential insight into the structure of Π-approximations and conditions under which we may use it to optimize the original function $F(x)$. Theorem 3 states in fact that it is desirable to set Π as small as possible. In this case the Π-approximation $\hat{F}(x)$ will have a very simple structure and the minimization of it will cause no problems. However, if set Π is too small, then according to Theorem 2 only convergency with respect to function value is to be expected because optimal points x^* are not, generally speaking, identifiable from equation (6) if set Π is chosen incorrectly.

Theorem 3 also provides a natural criteria for checking whether set Π is chosen appropriately or not. If the conditions of the theorem are satisfied then the subgradient of function $\hat{F}(x)$ if unique is always an extreme point of set Π. Appearance of another point might be indicative of a wrong choice of set Π.

3. COMPUTATIONAL ASPECTS

It is interesting also to look at computational aspects of dealing with function $\hat{F}(x)$. Due to Theorems 2 and 3 one can substitute the initial difficult problem (1) with the problem of minimizing $\hat{F}(x)$ under appropriate conditions.

$$\min_{x} \hat{F}(x) \quad . \tag{7}$$

The merits of this function is the fact that its calculation and calculation of its subgradient is similar to the solution of a dual problem and hence can be done in a highly decomposed way for problems with block-angular and similar structures.

Let us show how this computation is performed for a fixed point $x = 0$.

$$F(0) = \sup_{\pi\in\Pi}\{-F^{*}(\pi)\} = \sup_{\pi\in\Pi}\inf_{x}\{F(x)-x\pi\} = \sup_{\pi\in\Pi}\Psi(g) \quad , \qquad (8)$$

where $\Psi(\pi)$ is a value of the problem

$$\inf_{x}\{F(x)-x\pi\} = \Psi(\pi) \quad .$$

The potential advantages of this approach make use of the fact that computing $\Psi(\pi)$ might be essentially easier than dealing with the original problem. In doing so we can make use of Lagrangian relaxation of certain binding constraints in (1) simplifying its solution. For problems with block-diagonal structures with a set of linking variables or problems with block-angular structure with common constraints it is possible through this relaxation to decompose them into a set of smaller problems gaining essential economy in memory requirements. Problem (8) might be solved through a process similar to the Dantzig-Wolfe decomposition method, i.e., by coordinating via pricing mechanism solutions of the subproblems. The essential difference with the Dantzig-Wolfe decomposition method is the absence of the last phase, the *execution* phase, as named by Dirickx and Jennergren (1979). During the process of solving (8) in a decomposed way as in the Dantzig-Wolfe decomposition method a pair of "master-slave" problems can be formed and interaction between them goes on as it is organized in the Dantzig-Wolfe decomposition method. However, as a final result of this process we obtain the value of $\hat{F}(0)$ and its subgradient.

The value of the objective function F(0) together with its subgradient, which is equal to g^{*} (the solution of problem (8)) provides us with sufficient information to find an optimum of function $\hat{F}(x)$ and henceforth the minimum of function F(x).

If set Π satisfies the conditions of Theorem 3 and is a polyhedron then it is clear from the structure of function F(x) that one of the simplest algorithms of mathematical programming--the steepest descent method--will solve this problem in a finite number of steps. The second possibility in this case is to use a cutting plane algorithm (Kelley 1960). In this case it would be sufficient to make no more than n+1 iterations where n is a dimensionality of x.

Curiously enough is the fact that if set Π is a sphere with a radius small enough to satisfy the conditions of Theorem 3, then it would be sufficient to make one iteration of the steepest descent method to solve the original problem.

4. TEST PROBLEMS

In an experimental application of this algorithm a limited computational experience was accumulated using the DEC computer PDP-11/70 under the UNIX (Ritchie and Thompson 1978, Nurminski 1980) operating system with artificial random generated problems.

Two randomly generated linear programming problems were solved in these test runs. These problems consist of two blocks with 39 rows and 100 columns each and with a two-dimensional link between these blocks. These subproblems are referred to below as subproblems A and B respectively.

The coefficients of the constraint matrix and the costs associated with variables were generated by the IMSL subroutine *ggub* providing pseudo-random numbers uniformly distributed on [0,1]. A Fortran text of the matrix generator and details of this experiment are given in Nurminski (1980). Here we will discuss only some particular features of the method and its performance for the given test problems.

For solving the equivalent problem (7) the cutting-plane method was used in both cases. In accordance with the theory of this method, function $\hat{F}(x)$ and its subgradient have to be caculated in a few trial points in the space of linking variables which we call *reper* points which may be chosen in a different way. Here we choose this set as follows:

$$r1 = (0.0,\ 0.0) \quad ,$$

$$r2 = (2.0,\ 0.0) \quad ,$$

$$r3 = (2.0,\ 2.0) \quad .$$

It is worth noting that points r2 and r3 are not even feasible. Nevertheless, the method provides a finite value of Π-approximation at these points as well as finite subgradients which show directions of possible changes in linking variables.

Set Π by definition of Π-approximation was a simplex

$$\pi_1 + \pi_2 \leq 0.1 \quad ,$$

which was small enough not to create any problem during computations. Control runs were also made with

$$\Pi = \{\pi_1+\pi_2 \leq 0.01\} \quad ,$$

which showed no difference obtained with the first variant. The following table describes convergency of the coordinating process in each of the three *reper* points. In Table 3 the final results for corresponding *reper* points are given, where g(1) and g(2) are components of the subgradients of the approximating function $\hat{F}(x)$ with respect to linking variables, calculated at correspondent *reper* points.

Table 1. Test problem 1. Convergence of the coordinating process.

iter	r1()			r2()			r3()		
	master	A	B	master	A	B	master	A	B
1	-1.368	-0.933	-1.129	-1.218	-0.933	-1.117	-1.268	-0.933	-1.133
2	-1.964	-0.868	-1.091	-1.754	-0.868	-1.127	-1.875	-0.878	-1.091
3	-1.971	-0.911	-1.133	-1.777	-0.915	-1.127	-1.877	-0.905	-1.133
4	-1.975	-0.926	-1.133	-1.786	-0.933	-1.133	-1.878	-0.919	-1.133
5	-1.976	-0.933	-1.133	-1.792	-0.926	-1.133	-1.879	-0.926	-1.133
6	-1.979	-0.933	-1.133	-1.794	-0.933	-1.133	-1.879	-0.933	-1.133

Table 2. Test problem 2. Convergence of the coordinating process.

iter	r1()			r2()			r3()		
	master	A	B	master	A	B	master	A	B
1	-1.116	-0.454	-1.002	-0.966	-0.454	-1.002	-1.262	-0.454	-0.970
2	-1.386	-0.485	-1.002	-1.296	-0.488	-0.990	-1.289	-0.485	-1.002
3	-1.395	-0.488	-1.002	-1.296	-0.488	-0.999	-1.306	-0.488	-1.002
4	-1.403	-0.488	-0.970	-1.297	10.454	-1.002	-	-	-
5	-1.403	-0.488	-0.970	-1.297	-0.488	-1.002	-	-	-

Table 3. Test problems 1 and 2

reper point	function	g(1)	g(2)	reper point	function	g(1)	g(2)
r1(0,0)	-0.19791d01	0.0d00	-0.1d00	r1(0,0)	-0.14092d01	-0.1d00	0.0d00
r2(2,0)	-0.17929d01	0.1d00	-0.1d00	r2(2,0)	-0.12973d01	0.1d00	-0.1d00
r3(2,2)	-0.18791d01	0.1d00	0.0d00	r3(2,2)	-0.13092d01	0.0d00	0.1d00

The final step of the method consists of solving a linear system of the size defined by the number of linking variables. In the case under consideration these systems have the following forms:

Test 1:

$$-0.19791394d\ 01 - 0.1x_2 = L$$
$$-0.17929368d\ 01 + 0.1(x_1-2) - 0.1x_2 = L$$
$$-0.18791394d\ 01 + 0.1\ (x_1-2) = L$$

Test 2:

$$-0.14092d01 - 0.1x_1 = L$$
$$-0.12973d\ 01 + 0.1(x_1-2) - 0.1x_2 = L$$
$$-0.13092d01 + 0.1(x_2-2) = L$$

and their solutions are

Test 1:

$$x(1) = 0.13x(2) = 0.87$$

Optimal value: -2.065

Test 2:

$$x(1) = 0.63x(2) = 0.37$$

Optimal value: -1.472

CONCLUSIONS

The decomposition approach provides an efficient algorithmic tool for solving large-scale problems. It allows for a separate consideration of submodels and offers a theoretical foundation for linkage procedures. In this approach local variables are treated locally and exchange is restricted to global variables. Numerical experiments have shown that the method requires little information exchange between different subsystems and gives rapid convergency in the coordinating process.

REFERENCES

[1] Fenchel, W., On Conjugate Convex Functions, *Canad. J. Math.*, Vol. 1, 73-77, 1949.

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

[3] Khachyan, L.G., A Polynomial Algorithm in Linear Programming, *Doklady Akademii Nauk SSR*, Vol. 224, 1093-1096, 1979.

[4] Nurminski, E., *Some Theoretical Considerations on Linkage Problems*, WP-79-117, International Institute for Applied Systems Analysis, 1979.

[5] Nurminski, E., *Numerical Experiments with Decomposition of LP on a Small Computer*, WP-80-37, International Institute for Applied Systems Analysis, 1980.

[6] Ritchie, D.M., and K. Thompson, The UNIX Time-Sharing System, *The Bell System Technical Journal*, Vol. 57(6), Part 2, 1905-1930, 1979.