ON A DECOMPOSITION OF STRUCTURED PROBLEMS

E.A. Nurminski

March 1981
WP-81-31

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

On a decomposition of structured problems

E.A. Nurminski

IIASA

ABSTRACT

The construction and the analysis of complicated comprehensive models of complex social-economical, technical and/or environmental systems is greatly facilitated by modular design and implementation. However it creates specific difficulties in coordinating activities of separate modules. These questions are considered within the framework of the theory of decomposition of large-scale optimization problems. Theoretical foundations of the newly developed technique and its computational aspects and experience are discussed.

1. Introduction

The construction and the analysis of complicated comprehensive models of complex social-economical, technical and/or environmental systems is greatly facilitated by the equivalence of the modular principle in systems programming: split the whole job into pieces and supply for every piece a module which is responsible for a particular function or which gives an adequate description of a particular aspect of the system's behavior.

However it creates specific difficulties in coordinating activities of modules (subsystems, submodels, blocks,etc ...) in a way which at least allows the system as a whole to function. It is worth noticing that in practice the difficulties with this problem are the same as trying to coordinate those activities in an *optimal* way in order to attain an extreme value of one of the characteristics of the system. Such an optimal solution often provides additional insight into the system's inner mechanics.

In connection with optimal decisions these questions have always attracted a lot of attention from theoreticians and practicians of systems analysis. In applied mathematics these questions were studied within the framework of the theory of decomposition of large-scale problems, distributed and parallel processing, hierarhical and decentralized decision making, etc... Discussion on the practical impotance of this approach can be found for instance in(Dirickx79a).

These studies produced a number of important theoretical results but from the practical point of view, until recently, they did not produce computationally superior methods in comparison with other techniques. However, recently there were a few improvements and implementations in this field which may change this opinion. Especially successful were results obtained by E.Loute,J.Ho,(Loute80a, Loute78a). T.v.J.Roy,D.Erlenkotter,(Erlenkotte80a). T.v.J.Roy(Roy80a). A.Geoffrion, G.Graves(Geoffrion74a). A.Geoffrion,(Geoffrion70a). T.Aonuma,(Aonuma78a). and others. Successful applications were reported by J.Polito and others,(Polito80a). K. Jornsten(Jornsten79a).

Here we consider the theoretical foundations of a newly developed technique for decomposition of optimization problems as well as a numerical experiment with the number of test problems. The aim of the latter exercise is to demonstrate some directions for the potential improvment of the performance of the algorithm.

Theoretical background for our approach stems from one simple observation from the standard duality theory. This idea allows us to unite in a certain way primal (direct) and dual (indirect) approaches to decomposition of large-scale problems.

Let us first discuss briefly the advantages and shortcomings of these two approaches.

2. Direct decomposition

As a typical example we may consider a LP problem consisting of the two blocks

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

where z_A and z_B can be viewed 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

$$f_A(x) = \begin{cases} \min (c_A z_A) \\ A_A z_A + B_A x \leq b_A \end{cases} \tag{2}$$

and

$$f_B(x) = \begin{cases} \min (c_B z_B) \\ A_B z_B + B_B x \leq b_B \end{cases} \tag{3}$$

each of them requiring a smaller commitment of computer and human resources. Moreover computational efforts for solving both (2) and (3) are less then those needed to solve (1) even without taking into account the economy of core requirements.

Inspired by this argument 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.

The success of this approach depends on the degree of connectedness of subsystems A and B. It is worthwhile for a relatively weakly connected A and B with comparatively few x variables to the number of internal variables $z_{A,B}$.

The direct way of solving (4) would be through numerical methods updating

the linking variable x, using computed values of functions $f_A(x)$ and $f_B(x)$ and their differential characteristics (subgradients in this case so long as these functions are convex and nondifferentiable in a nontrivial case). Well-known Bender's decomposition scheme(Benders62a), is an example of such a development. It may be looked upon as a cutting plane algorithm (Kelley60a), applied to the problem (4). There is a significant number of partitioning methods which are applicable to the structured problems of the form (1) development of which started with(Rosen63a). Review of advances in this field and a bibliography on the subject is published in(Molina79a).

Another approach widely used in theory and in practice is subgradient optimization which is based on computing function values $f_A(x)$ and $f_B(x)$ togerther with subgradients of these functions. An application of subgradient optimization procedure for decomposition of linear programming problems was considered in.(Ermolev73a).

For the function (2) the computation of its subgradients is based on a solution of a dual problem

$$\max\ p(B_A x - b) = \left\{ p^*(B_A x - b) \mid p^* \in P^* \right\} \qquad (5)$$
$$pA_A + c_A \le 0$$

where P^* is a solution set of the problem (5).

The subdifferential $\partial f_A(x)$ consists of the vectors

$$\partial f_A(x) = \left\{ p^* B_A \mid p^* \in P^* \right\}$$

and it gives a constructive base for developing subgradient-based optimization routines. Of course for $f_B(x)$ there are similar relations.

This approach has a long list of striking achievements starting with pioneering works in the 60's (Shor62a, Ermolev66a, Polyak67a), not to mention recent

advances connected with the polynomially bounded method for LP. See (Wolfe80a) for details on the latter and(Lemarechal78a) for a bibliography on subgradient optimization.

The major difficulty with this approach is the complicated nature of the aggregated functions $f_A(x)$ and $f_B(x)$ and it creates a number of specific difficulties in solving the problem (4). Generally speaking, they are convex piecewise linear functions possibly undefined for some values of x for which either (2) or (3) is infeasible.

Let us first discuss problems connected with the domain of definition of the functions (2),(3). We assume for a moment that it is (2) which is infeasible for some x. For such x, the dual problem (5) becomes unbounded and provides no information about the direction of desirable changes in x, first - to restore feasibility, second - to reach optimality. As a remedy it is nessessary either introduce artificial variables, or take into account directly the domain of definition of the functions $f_A(x), f_B(x)$ in optimization routines.

Let us consider these two ways.

The domain of definition for functions $f_A(x), f_B(x)$ can, theoretically, be easily represented through Farkas' lemma:

$$X = X_A \cap X_B$$

where

$$X_{A,B} = \left\{ \textit{set of all } x \textit{ such that } f_{A,B}(x) < \infty \right\}$$

Using Farkas' lemma it can be reduced to the constraint

$$b_A - B_A x \in K(A_A) \tag{6}$$

where $K(A_A)$ is a cone

$$K(A_A) = \left\{ u \mid uA_A \geq 0 \right\}$$

Constraint (6) adds to problem (4) a number of linear constraints equal to the number of internal variables in subproblems A and B and significantly complicates it. Indirect usage of (6) through row generation technique complicates the logic of an algorithm and slows down the rate of convergence.

The second possibility which we have mentioned is to add artificial variables. Artificial variables can be added to (2) to ensure its feasibility in a simple way:

$$f_A(x) = \begin{cases} \min\,(c_A z_A + C_A y_A) \\ A_A z_A - y_A \le b_A - B_A x \end{cases} \qquad (7)$$

where $C_A \ge 0$ is a penalty cost associated with violation of the constraints in the subproblem A. Problem (7) is always feasible but difficulties may arise with finding C_A such that at an optimal point x artificial variables y_A are equal to zero. Big values of C_A may cause numerical instability. Also (7) has an enlarged number of variables and it creates additional computational overhead.

Even without problems connected with the domain of definition of the functions (2),(3), the rather complicated nature of these piecewise linear functions appearing after such decomposition of large scale problems makes it difficult to develop fast computational methods for their minimization. Sometimes simple subgradient minimization procedures are efficient enough for solving these problems but in other cases their convergence is reported to be slow.

3. Indirect decomposition

Indirect or dual decomposition is based on dualization of certain key constraints in an LP 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 underlies

many known schemes of decomposition the most widely known being the decomposition principle due to Dantzig and Wolfe.(Dantzig61a)

For problem (1) or equivalently (4) this key constraint may be the convention that variable x must have the same value in the functions (2) and (3).

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

$$v = -\min_{p} f^*_A(p) + f^*_B(-p) \tag{8}$$

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

$$f^*_A(p) = \sup_{x} \{ px - f_A(x) \} \tag{9}$$

and $f^*_B(p)$ is the conjugate of function $f_B(x)$ respectevely.

Dual variables p are customarily interpreted as prices for linking variables x. Computation of the values f^*_A, f^*_B can be interpreted as a local optimization in subproblems A, B for a set of a given prices p provided by master problem.

Problem (8) can then be solved by a number of methods updating prices p using values of functions $f^*_{A,B}$ and their subgradients.

It is useful to notice that $f^*_{A,B}$ are convex functions with subgradients $x^*_{A,B}$ equal to the solutions of (9). In other words subgradients of the functions $f^*_{A,B}$ are proposals of the local subproblems in terms of the Dantzig-Wolfe decomposition scheme. It then becomes clear that the Dantzig-Wolfe decomposition method can be interpreted from the point of view of nondifferentiable optimization as a cutting plane algorithm applied to the dual problem (8).

Convergency properties of this scheme 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.

This approach has an advantage that in due process only the objective functions of subproblems A,B are going to be changed so it is possible to use a previous local optimal solution as a starting point for the new iteration. Also the problem of local infeasibility of (2),(3) does not appear in this case, all links are under the control of local subproblems and *they* determine the most profitable values of the linking variables.

However, as a rule, it violates the balance of the system as a whole and consequently it requires special means to restore the balanced solution. Also computationally it does not have a good reputation, mainly due to the slow convergency on the final stages of solution.

Another problem with indirect decomposition is the problem of restoring a primal solution x^* from the solution p^* of the dual problem (8). Straightforward use of the relation

$$f_A(x_A^*) - p_A^* x_A^* = \sup\,(f_A(x) - p_A^* x\,) \tag{10}$$

$$f_B(x_B^*) - p_B^* x_B^* = \sup\,(f_B(x) - p_B^* x\,) \tag{11}$$

may and will produce quite different local solutions x^*_A, x^*_B even for optimal p^* and for bringing them together one needs to know *all* the solutions of (10),(11) which is practically impossible. What is currently being done is to keep track of all the solutions of (9) for prices generated in the course of the optimization process. The final solution is then generated as a convex combination of these intermediate solutions.(Lasdon70a).

4. Theoretical background

A possible way to overcome the difficulties associated with the primal approach (4) is to simplify functions $f_A(x), f_B(x)$ leaving unchanged those properties which are essential from the optimization point of view. On the other hand, it would also be desirable to preserve the advantages mentioned above of the dual problem (8).

For this purpose we will use a particular type of approximation for the functions $f_A(x), f_B(x)$, the nature of which is exemplified below. The most interesting feature of this approximation is that it has the same optimal solution as the original problem. The potential interest of this approximation from an optimization point of view is that it can also be computed in a way which resembles the dual problem (8). The remarkably simple structure of this approximation makes it possible to solve the correspondent extremal problem in a few iterations.

First we consider some general results concerning this approximation.

Let $f(x)$ be a closed convex function bounded from below. Let $f^*(\pi)$ denotes its conjugate

$$f^*(\pi) = \sup_x \{ \pi x - f(x) \}$$

We have the well-known relationship between $f(x)$ and $f^*(\pi)$

$$f(x) = \sup_\pi \{ \pi x - f^*(\pi) \}$$

which is valid under rather broad assumptions.(Fenchel49a).

For a given convex function $f(x)$ we define a new function $f_\Pi(x)$:

$$f_\Pi(x) = \sup_{\pi \in \Pi} \{ \pi x - f^*(\pi) \} \qquad (12)$$

where Π is a subset of the space X^* of dual variables.

Definition. Function $f_{\Pi}(x)$ given by expression (12) is called the Π-approximation of f(x).

The properties of this function which are essential from an optimization point of view depend on characteristics of the set Π and the behavior of the function $f(x)$ in the neighborhood of extremal points.

Here are a few extreme cases: if this set coincides with the whole dual space X^* then under the rather broad assumptions $f_{\Pi}(x) = f(x)$.(Fenchel49a). On the other hand if set Π collapses to a single point $\Pi = \{ 0 \}$ then

$$f_{\Pi}(x) = x \cdot 0 - f^*(0) = \inf_x f(x) = f^*$$

where f^* is the inf-value of the function $f(x)$.

In a nontrivial case if $\{ 0 \} \subset \Pi \subset X^*$, when both inclusions are strict , the function $f_{\Pi}(x)$ is something of an intermediate between these two extremes.

Here we give a few simple results concerning $f_{\Pi}(x)$ which originally appeared in(Nurminski79a). Proofs are essentialy simplified.

Theorem 1. If $0 \in \Pi$ then

$$\inf f_{\Pi}(x) = \inf f(x)$$

Proof. First:

$$f_{\Pi}(x) = \sup_{\pi \in \Pi} \{ \pi x - f^*(\pi) \} =$$

$$\sup_{\pi \in \Pi} \{ \pi x - \sup_z \{ \pi z - f(z) \} \} \leq$$

$$\sup_{\pi \in \Pi} \{ \pi x - \{ \pi x - f(x) \} \} = f(x)$$

On the other hand

$$f_{\Pi}(x) \geq 0 \cdot x - f^*(0) = f^*$$

which proves the statement.

Theorem 2. If f(x) is a closed convex function and Π is an absorbing set, then

any minimum of $f_{\Pi}(x)$ if it exists, is a minimum of f(x).

Proof. Let us assume that $f_{\Pi}(x)$ attains its minimum at some point x^0. Then if the theorem is not true $f_{\Pi}(x^0) < f(x^0)$ and due to separation arguments there is a vector p and scalar $\varepsilon > 0$ such that for any x

$$f(x) - f_{\Pi}(x^0) \geq p(x - x^0) + \varepsilon$$

Also for any x due to the theorem 1

$$f(x) - f_{\Pi}(x^0) \geq 0$$

Multiplying the first inequality by $\alpha > 0$ and the second by $1-\alpha \geq 0$, and summing them, we obtain:

$$f(x) - f_{\Pi}(x^0) \geq \alpha p(x - x^0) + \varepsilon';$$

We can always take α small enough to insure $\alpha p = \bar{\pi} \in \Pi$ but also $\varepsilon' = \varepsilon\alpha > 0$. Then

$$f(x) - f_{\Pi}(x^0) \geq \bar{\pi}(x - x^0) + \varepsilon';$$

$$\bar{\pi}x^0 - f_{\Pi}(x^0) \geq \bar{\pi}x - f(x) + \varepsilon';$$

$$\bar{\pi}x^0 - f_{\Pi}(x^0) \geq f^*(\pi) + \varepsilon';$$

And finally

$$f_{\Pi}(x^0) = \sup_{\pi \in \Pi} \{ \pi x^0 - f^*(\pi) \} \geq \bar{\pi}x^0 - f^*(\bar{\pi}) \geq f_{\Pi}(x^0) + \varepsilon';$$

This contradiction proves the statement.

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

$$\Pi \subset \partial f(x^*)$$

then

$$f_{\Pi}(x) = f^* + \sup_{\pi \in \Pi} \pi(x - x^*) \qquad (13)$$

Proof. Rather straightforward:

If $\Pi \subset \partial f(x^*)$ and $\pi \in \Pi$ then

$$f^*(\pi) = \sup_x \{ \pi x - f(x) \} = \pi x^* - f^* \qquad (14)$$

To demonstrate it notice that at least

$$f^*(\pi) \geq \pi x^* - f^*$$

On the other hand

$$f(x) \geq f^* + \pi(x - x^*)$$

So

$$f^*(\pi) \leq \sup_x \{ \pi x - f^* - \pi x + \pi x^* \} = \pi x^* - f^*$$

Substituting (14) into the definition of $f_\Pi(x)$ yields (13).

5. Computational aspects.

Consider now some computational problems of how to deal with the function $f_\Pi(x)$. Results from the previous chapter show that under rather mild conditions

$$\inf_x f(x) = \inf_x f_\Pi(x) \qquad (15)$$

and $f_\Pi(x)$ inherently may have a very simple structure (13)

Curiously enough , for instance , if the function $f(x)$ has strictly positive directional derivatives at the optimum, and set Π is a sphere small enough to be contained in $\partial f(x^*)$ then

$$f_\Pi(x) = f(x^*) + \gamma |x - x^*|$$

where γ is a radius of a sphere and *one* iteration of the steepest descent method applied to function $f_\Pi(x)$ provides us with a solution of the problem (15).

Results of the previous chapter show that for the purposes of simplifying of the function $f_\Pi(x)$ it is desirable to have the set Π as small as possible. Once the conditions of the theorem 3 are satisfied the Π-approximation $f_\Pi(x)$ will have a very simple structure and its minimization will create no problems.

However if the set Π is too small, optimal point x^* and optimal value f^* are not generally speaking identifiable from the equation (13). First , if Π doesn't satisfy the conditions of the theorem 1, then $f_\Pi(x)$ may have no minima at all no matter how well-defined the function $f(x)$ is. Moreover , if conditions of the theorem 1 are satisfied but not those of the theorem 2, then $f_\Pi(x)$ may have extra minimas which are not solutions of the original problem. One example of such a situation was considered above with the set Π being a singlton $\{ 0 \}$.

In this connection one important property of nondifferentiable problems is relevant. Formally it may be expressed as

$$0 \in int\, \partial f(x^*) \tag{16}$$

and in slightly different forms it was studied on different occasions by many authors. One example of the related property is the Haar condition which is very important for minmax optimization.(Hald79a). It is easy to show that for descrete minmax problems the Haar condition implies (16). Property (16) was formulated in (Nurminski74a). and was called a condition of essential nondifferentiability.

The particular features of the essentially nondifferentiable problems are the uniqueness of the optimal point and strict positiveness of all directional derivatives at this point. It was also observed that such problems have additional stability properties and it has been used for gaining computational advantages.

Condition (16) leaves enough room to satisfy conditions of the theorem 3 and also have a set Π rich enough to identify x^* and inf $f(x)$ from (13). In this case, due to the simple structure of the function $f_\Pi(x)$ its minimization can be effectively performed by a great variety of simple methods.

As an typical example we may consider a minimization of this function by

the cutting plane method.(Kelley60a). This method maintains some set of points at which correspondent values of the function and its gradient (subgradient to be exact) are calculated. Calculated function values and subgradients are used to form a system of linear equations which defines the next approximation of an optimal solution to enter this set or to be a terminating point.

So long as an epigraph of $f_{\Pi}(x)$ is a cone this method is particularly well suited for solving (15). Once a nonsingular linear system of the cutting plane method applied to the function $f_{\Pi}(x)$ is formed it gives an *exact* solution of the problem (15).

Taking this into account it becomes clear that the only precaution in using the cutting plane method in this case , is to choose the set of trial points in such a way that this set is representative enough so that the final linear system is nonsingular.

For a number of technical reasons it is preferable to choose these points as the vertices of a large enough simplex-like body. This problem will be discussed latter on.

The simplicity of the function $f_{\Pi}(x)$ is however deceiving because computation of a single value $f_{\Pi}(x)$ involves a solution of two nested optimization problems and it is not quite clear why it brings about any computational advantage at all.

In fact it is difficult to expect any advantages for the optimization problem (16) of the general kind. However for the structured problems of the type (1) it may make sense and result in a decomposition algorithm which combines some advantages of the primal and dual approaches.

In application of the cutting plane algorithm to the problem (15) it is nessassary to compute the value and subgradients of the function $f_{\Pi}(x)$ at some

trial points. In what follows we consider these calculations for the function $f(x) = f_A(x) + f_B(x)$ generated by the problem (1).

Without loss of generality consider calculation of $f_\Pi(x)$ and its subgradient at $x = 0$. By definition

$$f_\Pi(0) = \sup_{\pi \in \Pi} \{ - \sup_x \{ \pi x - f_A(x) - f_B(x) \} \} \tag{17}$$

Notice that a subgradient of $f_\Pi(0)$ is a solution π^0 of the external "sup" in the problem (17).

By introducing two distinct variables x_A, x_B and dualizing the constraint

$$x_A = x_B$$

one can obtain the following expression

$$f_\Pi(0) = \sup_{\pi \in \Pi} \inf_{x_A, x_B} \sup_p \{ f_A(x_A) + p x_A + f_B(x_B) - p x_B + \frac{1}{2}\pi x_A + \frac{1}{2}\pi x_B \} =$$

$$\sup_{p, \pi \in \Pi} \{ \inf_{x_A} \{ f_A(x_A) + p x_A + \frac{1}{2}\pi x_A \} + \inf_{x_B} \{ f_B(x_B) - p x_B + \frac{1}{2}\pi x_B \} =$$

$$\sup_{p, \pi \in \Pi} \{ -f_A^*(-p - \frac{1}{2}\pi) - f_B^*(p - \frac{1}{2}\pi) =$$

$$\sup_{\pi_A + \pi_B \in \Pi} \{ -f_A^*(-\pi_A) - f_B^*(-\pi_B) =$$

$$\inf_{\pi_A + \pi_B \in -\Pi} \{ f_A^*(\pi_A) + f_B^*(\pi_B) \}$$

If the set Π is centrally symmetric: $\Pi = -\Pi$ then

$$f_\Pi(0) = - \inf_{\pi_A + \pi_B \in \Pi} \{ f_A^*(\pi_A) + f_B^*(\pi_B) \} \tag{18}$$

This value may be represented as a solution of the problem

$$\begin{gathered} \max (-v) \\ v \le f_A^*(\pi_A) + f_B^*(\pi_B) \\ \pi_A + \pi_B \in \Pi \end{gathered} \tag{19}$$

and it generalizes a master problem appearing in resourse-directive decomposition schemes. Comparing (18) and (8) it becomes clear that complete dualization or the Dantzig-Wolfe decomposition scheme corresponds to the choice $\Pi = \{ 0 \}$.

Computation of the functions $f_A^*(\pi_A), f_B^*(\pi_B)$ can be interpreted as a local optimization in separate subproblems A, B Taking subproblem A as an example one can see

$$f_A^*(\pi_A) = \sup_{x_A} \{ \pi_A x_A - f_A(x_A) \} = \tag{20}$$

$$= - \min (c_A z_A - \pi_A x_A)$$

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

so computation of the function $f_A^*(\pi_A)$ is equivalent to the solution of the local subproblem A (8) with additional cost accociated with priced linking variables. In this sence the approach proposed above has all the advantages of the dual decomposition of the Dantzig-Wolfe type.

Applying the cutting plane procedure for (19) one can think of it as being organized in the following way:

Phase 1.

For given prices $\bar{\pi}_A, \bar{\pi}_B$ solve subproblems (20) and obtain subgradients-proposals $\bar{x}_A, \bar{x}_B$ together with the optimal values in subproblems v_A, v_B:

$$v_A = f_A^*(\pi_A)$$
$$v_B = f_B^*(\pi_B)$$

Phase 2.

Modify the master problem (20) of the cutting plane method by including a new constraint:

$$v \leq \bar{v}_A + \bar{v}_B + \bar{x}_A \pi_A + \bar{x}_B \pi_B$$

where $\bar{v}_A, \bar{v}_B$ are some constant terms:

$$\bar{v}_A = v_A - \bar{x}_A \bar{\pi}_A$$
$$\bar{v}_B = v_B - \bar{x}_B \bar{\pi}_B$$

Solve the new master problem,obtain new prices $\bar{\pi}_A, \bar{\pi}_B$ and if the stopping criteria is not satisfied go to Phase 1.

It is known that after a finite number of such steps a solution (π_A^*, π_B^*) will be obtained. Then the sum $\pi_A^* + \pi_B^*$ is a subgradient of the function $f_\Pi(x)$ at the point $x = 0$ and the optimal value $-v^*$ is equal to the value of the function $f_\Pi(x)$ at zero. Combining these values calculated at different points $x^i, i=1,\ldots,I$ one can form a system of linear equations:

$$f^* + \pi_i^*(x^i - x^*) = f_i \qquad (21)$$
$$i=1,\ldots,I$$

where f^*, x^* are unknown optimal value and solution,for every i π_i^* is a subgradient of $f_\Pi(x)$ at $x = x^i$ and $f_i = f_\Pi^*(x^i)$.

The computational process can be controlled first by the choice of points x^i in (18) which can be done either in an adaptive or predetermined way. An important feature of this approach is that so far as the geometry of level sets of the function $f_\Pi(x)$ is determined by the set Π, these points can be chosen in advance making computations of π_i^* and f_i corresponding to different x^i independent of each other. It allows wide use of parallel computing in solving (13) and of the sharing of the computational efforts between independent but similar processes.

Another degree of freedom in this approach is the choice of a set Π in a definition of the Π-approximation. It is possible to obtain extremely simple results in the case when this set is equal to the difference of two simplices with some scaling:

$$\frac{1}{\varepsilon}\Pi = \{\, x - y : x, y \geq 0; \sum x_i \leq 1; \sum y_i \leq 1 \,\} \qquad (22)$$

where $\varepsilon > 0$ is small enough to ensure satisfaction of the conditions of the theorem 3.

6. Numerical example

Preliminary numerical experience with this algorithm was given in(Nurminski80a). with the randomly generated LP problems. 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 problem has three linking variables which link together two subproblems each with 6 internal variables and 3 equations. The constraint matrix, coefficients of the objective function and right-hand side are shown in the tables 1 and 2.

Table 1. Subproblem A.

row	z1	z2	z3	z4	z5	z6	x1	x2	x3	rhs
cost				2.	1.	1.	-1.5	-1.	-0.5	
eq	1.			-1.	-1.	-1.	1.	2.	-2.	2.
eq		1.		-1.	-1.		1.	-1.	1.	4.
eq			1.		-1.	-2.	-1.	-1.	1.	2.

Table 2. Subproblem B.

row	z1	z2	z3	z4	z5	z6	x1	x2	x3	rhs
cost				1.	1.	5.	-1.5	-1.	-0.5	
eq	1.			1.	-1.		-1.		2.	4.
eq		1.		1.	-1.	-2.		1.	-1.	
eq			1.	-1.	1.	-1.	1.	3.		5.

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

In accordance with the theory , one has to choose a set of points in which correspondent values of $f_\Pi(x)$ and subgradients are to be computed. These points were chosen in the following way : 3 of them were taken as nonzero ver-

tices of the simplex

$$x_1 + x_2 + x_3 \leq R \tag{23}$$
$$x_1 \, , \, x_2 \, , \, x_3 \geqslant 0$$

and the fourth point was taken as

$$x_1 = x_2 = x_3 = -R \tag{24}$$

where R is some constant large enough.

Not only this is the simplest way of choosing these points but it also simplifies theoretical considerations.

An advantage of (22) and the way how trial points are chosen is that it is possible to show that for nondegenerate problems and for R big enough, subgradients of the function $f_{\Pi}(x)$ computed at these points automaticaly form a nonsingular linear system of the cutting plane method. It also allows a simple way of representing the final results and avoiding some numerical problems during the concluding phase.

The pattern of the subgradients of the function $f_{\Pi}(x)$ calculated at different trial points can be done visibly with the the help of some graphs. So far as the subgradients of the $f_{\Pi}(x)$ are almost everywhere extreme points of the set Π, they can be naturally represented as directed arcs of a graph with $n+1$ vertices, where n is the number of linking variables. In this way the (i,j) arc may represent a subgradient equal to the difference of $i-th$ and $j-th$ extreme points of the simplex with some scheme for enumeration of the vertices. For brevity we will call this graph a subgradient graph.

Notice that there are no (i,i) arcs unless the optimal solution coincides with one of the points x^i.

The natural enumeration of the simplex vertices with 0 as an origin and the $i-th$ node having nonzero $i-th$ coordinate is very convenient and will be used in

what follows.

The arcs in the graph may be given values corresponding to the right-hand side of the system (21) Then if the constant R is big enough there is a cycle in a correspondent subgradient graph and the optimal value of the problem (13) is the average cost associated with the arcs in the cycle.

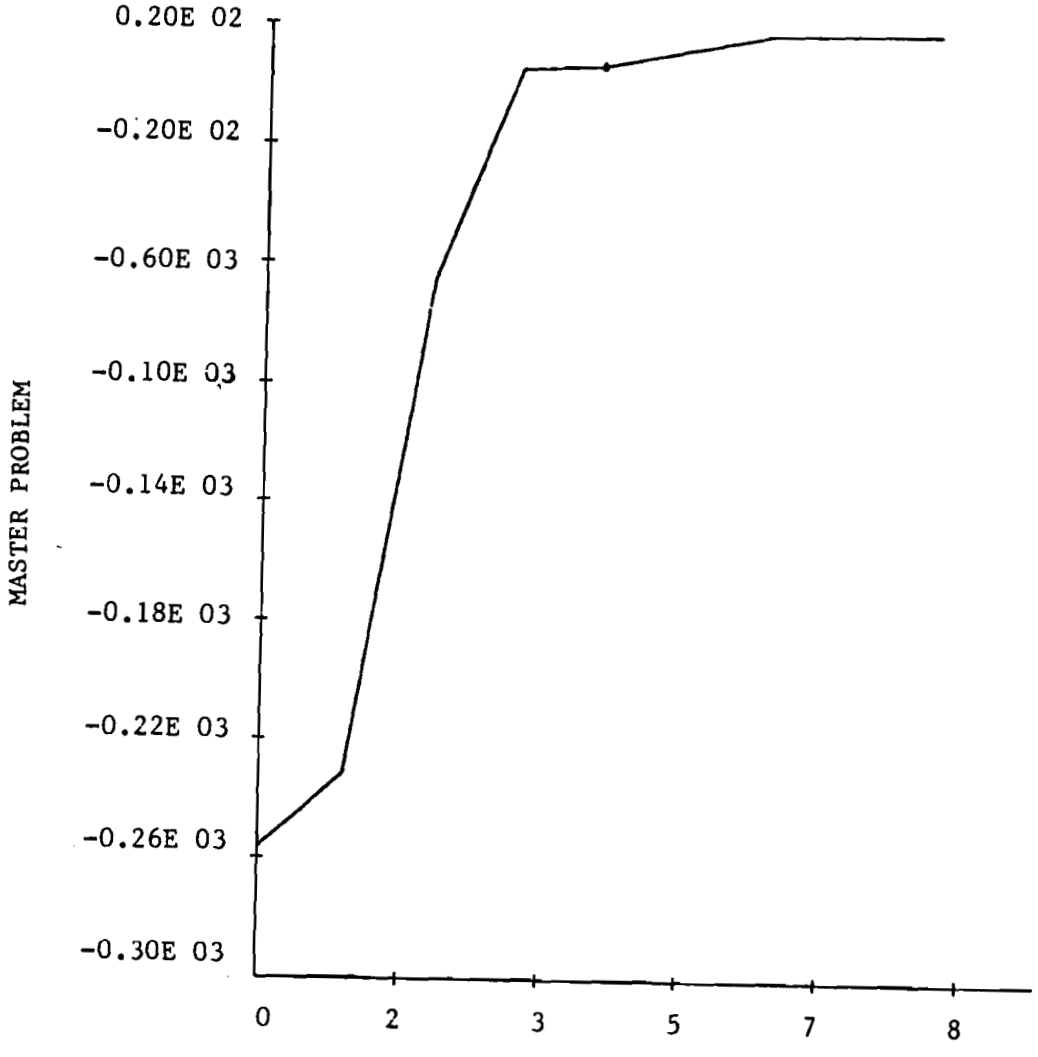
The algorithm described above was applied to the given problem with different values for ε and R. It occurred that its performance did not depend significantly on the particular numerical values for these parameters so far as they were realistic. Results discussed below were obtained for ε=0.01, R = 10000.

On fig.1 the convergence in the value of the master problem for one of the trail points is shown.

It took 9 cycles for all points to reach a solution of the problem (15), which is not a very good result for such a modest problem. It can be improved ,however, by using another faster method for solving (15) rather than the cutting plane algorithm, which is known for its slow convergence.

The justification for such a belief lies in the fact that during the computation very early (actually even on the first cycle between master and subproblems) the correct values for subgradients of function $f_{\Pi}(x)$ were found and further progress was done only in getting more and more accurate value of $f_{\Pi}(x)$ at correspondent points.

Fig. 1. Convergence of the master problem.

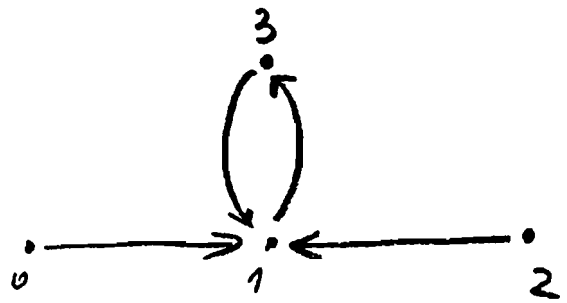


This phenomenon gives rise to the possibility of forming an accurate matrix of the linear system of the cutting plane method already on the first cycles of the coordinating process. During succeeding iterations only the right-hand side of this system changes allowing early estimates of the values of linking variables and structure of bases to be chosen in the subproblems.

On the other hand it allows one to get rid of nonactive constraints in (15), reduce this problem to an unconditional problem of nondifferentiable optimization and use for the resulting problem the fast numerical methods developed for instance by Shor(Shor79c), Lemarechal(Lemarechal80a), and others.

The pattern of the subgradients of the function $f_{\Pi}(x)$ is shown in fig.2

Fig.2. The subgradient graph.



Convergence in linking variables and estimates of the optimal value of the objective function are illustrated in table 3.

Table 3. Convergence of the linking variables.

Beal's problem				
cycle	optimum	x1	x2	x3
1	0.255269e+03	0.4928e+02	0.1363e+02	0.1286e+02
2	0.230243e+03	0.4979e+02	0.1220e+02	0.1258e+02
3	0.652849e+02	0.4320e+02	0.1319e+01	0.1197e+02
4	-0.612910e+01	0.3557e+02	0.8429e+00	0.7896e+01
5	-0.713570e+01	0.3506e+02	0.8385e+00	0.7857e+01
6	-0.128964e+02	0.2898e+02	0. e+00	0.7636e+01
7	-0.179449e+02	0.8945e+01	0. e+00	0.4500e+01
8	-0.184409e+02	0.9441e+01	0. e+00	0.4500e+01
9	-0.185000e+02	0.9500e+01	0. e+00	0.4500e+01

It is also interesting to analyse the changes in the structure of the bases in the subproblems for the approximate solutions shown in the table 3.

Table 4. Structure of the basis in the subproblems

	basis variables					
cycle	subproblem A			subproblem B		
1(a)	2	4	6	2	4	5
2	2	4	6	2	3	5
3	2	4	6	1	3	5
4	2	4	6	1	3	5
5	2	4	6	1	3	5
6	2	4	6	1	3	5
7	1(b)	2	4	1	3	5
8	1(b)	2	4	1	3	5
9	1(b)	2	4	1	3	5

These results show that there is some stability in the bases for the subproblems generated during the optimization process. It obviously can be used for speeding up the whole process.

Another interesting peculiarity of this method is the possibility of revealing a hidden decomposition in linking variables. In fact from fig.2 one may notice that for calculating the optimal value of the problem one does not need to know all the arcs of the subgradient graph and the values associated with them because (1,3) and (3,1) arcs already form a cycle. Adding to them the (0.1) arc one has the possibility of determining the optimal values of the linking variables x_1 and x_3 without any knowledge of x_2. Then x_1 and x_3 can be fixed at their optimal levels and a reduced problem of the same type be formed with constraints matrices, cost rows and right-hand sides given by the following tables derived from tables 1 and 2 :

(a) During this cycle subproblem B had also variable z3 at upper boundary which was additionaly set to avoid unboundness in subproblems.
(b) Degenerate basis variable.

Table 5. Subproblem A'.

row	z1	z2	z3	z4	z5	z6	x	rhs
cost				2.	1.	1.	-1.	
eq	1.			-1.	-1.	-1.	2.	1.5
eq		1.		-1.	-1.		-1.	-10.
eq			1.		-1.	-2.	-1.	7.

Table 6 Subproblem B'.

row	z1	z2	z3	z4	z5	z6	x	rhs
cost				1.	1.	5.	-1.	
eq	1.			1.	-1.	-1.		4.5
eq		1.		1.	-1.	-2.	1	4.5
eq			1.	-1.	1.	-1.	3.	-4.5

Variables $z_1 - z_6$ are internal variables for the new subproblems A',B', x is the only remaining link corresponding to the variable x_2 in the original formulation.

Computational experience shows that the number of cycles between master and subproblems strongly depends on the number of linking variables so the reduction of this number at an early stage of the solution may bring significant savings in computational efforts. In this particular example decomposed solution of the problem given by tables 5,6 required only 2 cycles between master and subproblems.

Summary

It is too early to make any definite conclusions about the merits and the shortcomings of the proposed method. It is at the eary stage of its development and is not so mature that it can compete with well established techniques.

The notion of Π-approximation is based on the general convex duality and can be applied to nonlinear problems as well.

In applications to structured linear programming problems it allows to combine price-directed decomposition approach with resource-directed decomposition.

The particular implementation of the computational process makes it possible to reveal some subsets of linking variables which can be determined well before completing the whole process, reducing in this way the total amount of computations.

References

Aonuma78a. T. Aonuma, "A two-level algorithm for two-stage linear programs," *Journal of the Operational Research Society of Japan* **21**(2), pp.171-187 (June 1978).

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

Benders62a. J.F. Benders, "Partitioning procedures for solving mixed variables programming problems," *Numerische mathematik* **4**, pp.238-252 (1962).

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).

Erlenkotte80a. D. Erlenkotter and Tony J. Van Roy, "A dual-based procedure for dynamic faculty location," WP-80-31, IIASA and Afdeling Industrieeel Beleid Katholieke Universiteit Leuven (March 1980).

Ermolev66a. Yu.M. Ermolev, "Methods of solution of nonlinear extremal problems," *Kibernetika* **2**, pp.1-17, 4 (1966).

Ermolev73a. Yu.M. Ermolev and L.G. Ermoleva, "The Method of Parametric Decomposition," *Cybernetics* **9**(2), pp.262-266 (1973).

Fenchel49a. W. Fenchel, "On conjugate convex functions," *Canad. J. Math.* **1**, pp.73-77 (1949).

Geoffrion70a. A.M. Geoffrion, "Elements of large-scale mathematical programming, Parts I and II, ," *Management Science*(16), pp.652-691 (1970).

Geoffrion74a. A.M. Geoffrion and G. Graves, "Multicommodity Distribution System Design by Benders' Decomposition," *Management Science* **20**(5), pp.822-844 (January 1974).

Hald79a. J. Hald and K. Madsen, "Combined LP and quasi-Newton methods for minimax optimization," NI-79-05, Numerisk Institut Danmarks Tekniske Hojskole, 2800 Lyngby Danmark (June 1979).

Jornsten79a. K.O. Jornsten, "A Benders decomposition method for the network design problem in a combined distribution and assignment environment," LiTH-MAT-R-79-30, University of Linkoping (September 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).

Lemarechal75a. C. Lemarechal and R. Mifflin, *Nonsmooth optimization: Proceedings of a IIASA Workshop, March 28- April 8, 1977*, Pergamon Press (1978).

Lemarechal80a. C. Lemarechal, "Nonsmooth optimization. Use of the code DYNEPS," WP-80-36, IIASA (1980).

Loute78a. E. Loute and J.K. Ho, and B. Culot, "Decomposition of multi-regional energy flow optimization models," Manuscript, Center for Operation Research and Econometric Universite Catholique de Louvain (September 1978).

Loute80a. E. Loute and J.K. Ho, "A comperative study of two methods for staircase linear programs," *ACM Transaction on Mathematical Software* **6**(1), pp.17-30 (March 1980).

Molina79a. F.W. Molina, "A survey of resource directive decomposition in mathematical programming," *Computing Survey* 11(2), pp.95-104 (June 1979).

Nurminski79a. E. Nurminski, "Some theoretical considerations on linkage problems," WP-79-117, IIASA (December 1979).

Nurminski80a. E. Nurminski, "Numerical experiments with decomposition of LP on a small computer," WP-80-37, IIASA (March 1980).

Nurminski74a. E.A. Nurminski, "Minimization of nondifferentiable function under noise," *Kibernetika*(4), pp.59-61 (1974).

Polito80a. J. Polito, B.A. McCarl, and T.L. Morin, "Solution of spartial equilibrium problems with Benders decomposition.," *Management Science* **26**(6), pp.593-605 (June 1980).

Polyak67a. B.T. Polyak, "A general method of solving extremal problems," *Doklady Academii Nauk SSSR (Soviet Math. Doklady)* **174 (8)**(1), pp.33-36 (593-597) (1967).

Rosen63a. J.B. Rosen, "Convex partition programming," pp. 159-176 in *Recent Advances in Mathematical Programming*, ed. R.L. Graves, McGraw-Hill,Inc, New Jork (1963).

Roy80a. Tony J. Van Roy, "A cross decomposition for integer programming," Working Paper 80-7, Afdeling Industrieeel Beleid Katholieke Universiteit Leuven (March 1980).

Shor62a. N.Z. Shor, "Application of the gradient method for the solution of network transportation problems," in *Notes of the Seminar on Theory and Application of Cybernetics and Operation research.*, Institute of Cybernetics, Kiew (1962).

Shor79a. N.Z. Shor, *Methods of Minimization of Nondifferentiable Functions and their Applications*, Naukova Dumka, Kiew (1979).

Wolfe80a. P. Wolfe, *A bibliography for the ellipsoid algorithm*, July 3,1980.