Decomposition in Global Optimization

JØRGEN TIND
Department of Operations Research, University of Aarhus, Building 530, Ny Munkegade, DK-8000 Aarhus C, Denmark.

(Received: 6 June 1990; accepted: 4 December 1990)

Abstract. The purpose of this article is to propose a simple framework for the various decomposition schemes in mathematical programming.

Special instances are discussed. Particular attention is devoted to the general mathematical programming problem with two sets of variables. An economic interpretation in the context of hierarchical planning is done for the suggested decomposition procedure.

The framework is based on general duality theory in mathematical programming and thus focussing on approaches leading to global optimality.

Key words. Decomposition, general mathematical programming, general duality, hierarchical systems.

1. Introduction

Decomposition procedures have had a profound influence in mathematical programming since they were introduced by Benders [1] and by Dantzig and Wolfe [2]. They have given rise to a long range of suggestions for improvement in the design of algorithms for the solution of large scale mathematical programming problems. Perhaps even more importantly, they have had a tremendous impact on quantitative modelling of decision making in hierarchical structures. See for example Dirickx and Jennergren [3], Burton and Obel [4] and Obel [5]. Here a hierarchy typically consists of a central unit and one or more subunits with communication lines between the central unit and the subunits. For example in the decomposition scheme given by Dantzig and Wolfe the objective of the central unit is to announce prices for the subunits for their utilization of joint resources. In this way the subunits need not know the actual limits of the joint resources. On the other hand the subunits quote suggestions for the utilization of joint resources together with their contribution to an overall objective. Hence, the central unit needs no information about technical constraints that are local for each subunit. This is the basic structure that is utilized in an iterative process. Originally and in most applications linearity is assumed in the part of the cost and technical structure that interact between the central unit and the subunit. Technically this means that the interaction is modelled by linear programming and the price information for joint resources is obtained from the dual variables in the actual linear model.

The purpose of the present article is to develop a similar but more general decomposition procedure that does not require linearity. The main tool for doing

© 1991 *Kluwer Academic Publishers. Printed in the Netherlands.*

this is to replace linear programming duality by duality in general mathematical programming. The article demonstrates that in this way it is possible to develop a decomposition procedure while still keeping all the main principles unchanged. In particular the usual economic interpretations carry over to this general setting. The main idea for this is to replace the prices, given by the dual variables, by the more detailed cost information provided by dual functions considered in general duality theory.

It is emphasized that since the suggested procedure is based on these more detailed functions the approach does not limit itself to the consideration of only local optima. Hence the procedure should be considered as a global optimization approach for mathematical programming.

Some progress has already been made in this direction. Geoffrion [6] and Holloway [7] have discussed procedures based on convex programming. Wolsey [8] discusses a generalization of Benders' procedure. Burkard, Hamacher and Tind [9] consider the separable case. For an overview and some further results, see Flippo [10].

The remaining part of the paper is divided in sections as follows.

Section 2 restates some basic results from duality in general mathematical programming to be used in the later sections.

Section 3 presents the basic decomposition scheme. This is based on a very simple max-min model. Due to its simplicity and generality it is believed to be an appropriate choice of model. It is important to note that the decomposition procedure provides improving lower and upper bounds for the objective function. They are of significance in convergence considerations, in particular in case of premature interruption of the procedure.

Section 4 deals with decomposition of a general mathematical programming model with two sets of variables. The suggested approach is a special case of the procedure in Section 3. An alternative procedure for this model can be found in Flippo [10], based on an expansion in the dimension of the dual solution space.

In Section 5 we present an economic interpretation of the model and the procedure in the previous section. Section 6 gives a description of the simplifications obtained in case of separability in the objective function and in the constraints. Section 7 discusses how the cross decomposition procedure by Van Roy [11] can be derived. Section 8 concludes the article with some final comments and remarks.

For convenience in notation sup is used at most places instead of max. This is done in order to get always an optimal value of the objective function. A similar remark applies to inf and min.

2. General Dual Programs

The purpose of this section is to present some fundamental definitions and results about duality in general mathematical programming which will be used in the next sections. The results are taken from Tind and Wolsey [12] which also contains the proofs and other details which are omitted here.

Let X and Y be arbitrary nonempty sets and consider a function $K(x, y)$: $X \times Y \to \mathbb{R} \cup \{+\infty\} \cup \{-\infty\}$.

We state the following program as the *primal program*

$$z = \sup_{x \in X} \inf_{y \in Y} K(x, y) . \tag{P}$$

Let $z(x) = \inf_{y \in Y} K(x, y)$. We speak of a feasible solution x_o of (P) with value $z(x_o)$ if $x_o \in X$ and $z(x_o) > -\infty$. Furthermore, x_o is an optimal solution of (P) if it is feasible and if $z(x_o) = \sup_{x \in X} z(x)$.

By transposition of the operators sup and inf we shall also consider the *dual program*

$$w = \inf_{y \in Y} \sup_{x \in X} K(x, y) . \tag{D}$$

Let $w(y) = \sup_{x \in X} K(x, y)$. Similarly y_o is feasible in (D) if $y_o \in Y$ and $w(y_o) < \infty$. Moreover, a feasible solution y_o is optimal in (D) if $w(y_o) = \inf_{y \in Y} w(y)$.

This terminology is in concordance with the conventional terminology applied in mathematical programming. Consider for example a standard mathematical programming problem

$$\begin{aligned} &\sup \ f(x) \\ &\text{s.t.} \ g(x) \leq b \\ &\qquad x \in X \subseteq \mathbb{R}^n , \end{aligned} \tag{2.1}$$

where $f: \mathbb{R}^n \to \mathbb{R}$, $g: \mathbb{R}^n \to \mathbb{R}^m$ and $b \in \mathbb{R}^m$. Let $K(x, y)$ be the corresponding standard *Lagrange function*, i.e., $K(x, y) = f(x) - yg(x) + yb$ where $y \in Y = \mathbb{R}^m_+$. In this setting (P) and (2.1) are equivalent programs. In particular $x \in X$ is feasible in (P) if and only if $x \in X$ is feasible in (2.1) in the conventional sense, i.e., $g(x) \leq b$. The program (D) becomes the standard Lagrangean dual of (2.1).

We always have *weak duality* between (P) and (D), i.e., $z(x_o) \leq w(y_o)$ for all $(x_o, y_o) \in X \times Y$. We speak of *strong duality* when there is no duality gap between (P) and (D), i.e., when $z = w$.

We shall here restate some results from general duality to be used in the following sections.

Consider again the general mathematical programming problem (2.1). Let $\mathcal{F}$ denote a selected class of nondecreasing functions $F: \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\} \cup \{-\infty\}$ and consider with this class a *generalized Lagrange function* of (2.1):

$$K(x, F) = F(b) - F(g(x)) + f(x) . \tag{2.2}$$

Consider also the dual of (2.1) written in the form

$$\begin{aligned} &\inf_{F \in \mathcal{F}} \ F(b) \\ &\text{s.t.} \ F(g(x)) \geq f(x) \quad \forall x \in X . \end{aligned} \tag{2.3}$$

Here we shall summarize some fundamental relationships between the above programs into the following

PROPOSITION 2.1. *Let* $Y = \mathcal{F}$, $X \subseteq \mathbb{R}^n$ *and define* $K(x, F)$ *by* (2.2). *We can then select an appropriate class* $\mathcal{F}$ of functions so that

(i) (2.3) *and* (D) *are equivalent and*

(ii) *strong duality exists between* (P) *and* (D).

An appropriate selection of the class $\mathcal{F}$ depends on the actual program. The aim is of course to select the class as simple and small as possible. This question is fundamentally the same as asking for an efficient algorithm to find an optimal solution of (2.1) and to prove optimality via the construction of an optimal solution F in (2.3). For details see Tind and Wolsey [12].

3. Decomposition

Here we shall attempt to solve (P) by a decomposition approach. The underlying procedure is a direct generalization of the procedures originally developed by Benders [1] and by Dantzig and Wolfe [2].

Let $\bar{Y} \subseteq Y$ and consider the following problem called the *upper problem*:

$$\bar{z} = \sup_{x \in X} \inf_{y \in \bar{Y}} K(x, y) . \tag{U}$$

Since $\bar{z} \geqslant z$ we obtain that (U) produces an upper bound for the optimal value z of (P).

Similarly, let $\bar{X} \subseteq X$ and consider the following *lower problem*:

$$\underline{w} = \inf_{y \in Y} \sup_{x \in \bar{X}} K(x, y) . \tag{L}$$

This problem produces a lower bound for the optimal value w of (D).

Under appropriate conditions $\underline{w}$ is also a lower bound for (P) (or similarly $\bar{z}$ an upper bound for (D)) as demonstrated by the next two remarks.

REMARK 3.1. If $\bar{X}$ contains only one element $x^* \in X$, i.e., $\{x^*\} = \bar{X}$, then $\underline{w} = \inf_{y \in Y} K(x^*, y) \leqslant \sup_{x \in X} \inf_{y \in Y} K(x, y) = z$. Hence, $\underline{w}$ is also a lower bound for (P).

REMARK 3.2. If $z = w$, i.e., strong duality holds between (P) and (D), then $z = w \geqslant \underline{w}$. Hence, $\underline{w}$ is also a lower bound for (P) in this case.

In the following procedure we shall set up specifications for the generation of the subsets $\bar{X}$ and $\bar{Y}$. The procedure is iterative. The iterations are enumerated by an iteration counter k. Now, introduce a map $\tilde{X}(k) : \mathbb{N} \rightarrow \mathcal{D}(X)$, where $\mathcal{D}(X)$ denotes the set of subsets in X. Let $\bar{X} = \tilde{X}(k)$ in the kth iteration. Similarly, introduce a map $\tilde{Y}(k): \mathbb{N} \to \mathcal{D}(Y)$ and let $\bar{Y} = \tilde{Y}(k)$ in the kth iteration. Let u, $l \in \mathbb{R} \cup \{+\infty\} \cup \{-\infty\}$ be the best upper and lower bounds, respectively, to be updated during the procedure.

ASSUMPTION 3.1. In the following procedure we assume that (L) provides a lower bound for z. This assumption is for example fulfilled subject of the conditions stated in Remarks 3.1 or 3.2.

Now the *basic procedure* can be stated as follows.

Start : Let $\bar{X} \subseteq X$ and $\bar{Y} \subseteq Y$ be nonempty.
: Let $u = +\infty$, $l = -\infty$ and $k = 1$.
Step 1 : Solve (U). Update $u := \min\{u, \bar{z}\}$.
Let $\bar{X} = \tilde{X}(k)$.
Step 2 : Solve (L). Update $l := \max\{l, \underline{w}\}$.
Let $\bar{Y} = \tilde{Y}(k)$.
Step 3 : If $u = l$ then stop. Otherwise, let $k := k + 1$ and go to Step 1.

Observe that the entire setup is symmetric in x and y. Hence, an equivalent procedure can be established for (D) as well.

If, in step 1, $\bar{z} = -\infty$ then (U) is infeasible. Hence, (P) is also infeasible, and the procedure will stop in Step 3 as $u = l = -\infty$. This situation will be discovered during the first iteration.

Assumption 3.1 is used in Step 3 to ensure that $z = u = l$ at termination.

We have to find appropriate specifications of the sets $\bar{X}$ and $\bar{Y}$ to ensure convergence. In order to obtain this goal we can state the following

PROPOSITION 3.1. *If*

(i) $\sup_{x \in \bar{X}} \inf_{y \in \bar{Y}} K(x, y) = \sup_{x \in X} \inf_{y \in \bar{Y}} K(x, y)$ *and*
(ii) $\inf_{y \in Y} \sup_{x \in \bar{X}} K(x, y) = \inf_{y \in \bar{Y}} \sup_{x \in \bar{X}} K(x, y)$

then $\bar{z} = \underline{w}$.

Proof. By definition $\bar{z} = \sup_{x \in X} \inf_{y \in \bar{Y}} K(x, y)$ and $\underline{w} = \inf_{y \in Y} \sup_{x \in \bar{X}} K(x, y)$. By Assumption 3.1, $\bar{z} \geqslant z \geqslant \underline{w}$. Since, generally, $\sup_{x \in \bar{X}} \inf_{y \in \bar{Y}} K(x, y) \leqslant \inf_{y \in \bar{Y}} \sup_{x \in \bar{X}} K(x, y)$ we get by (i) and (ii) that $\bar{z} \leqslant \underline{w}$. Hence, $\bar{z} = z = \underline{w}$. □

The conditions (i) and (ii) express that it is sufficient to consider the subsets $\bar{X} \subseteq X$ and $\bar{Y} \subseteq Y$ to ensure convergence. Typically, the sets $\bar{X}$ or $\bar{Y}$ are expanded or otherwise changed during the iterations of the procedure by the maps $\tilde{X}(k)$ or $\tilde{Y}(k)$ until the conditions (i) and (ii) are satisfied. The objective is of course to make any expansion and the number of changes as small as possible to facilitate the computational work and still have convergence.

Before we are going to look at some more general problems let us, for completeness, show that the Dantzig–Wolfe decomposition procedure is a special case of the above procedure.

EXAMPLE (Dantzig–Wolfe decomposition in linear programming). We let $K(x, y) = cx - yA_1x + yb_1$, where $A_1 \in \mathbb{R}^{m\times n}$, $b_1 \in \mathbb{R}^m$. Let also $Y = \mathbb{R}^m_+$ and $X = \{x \in \mathbb{R}^n_+ \mid A_2x \leqslant b_2\}$ where $A_2 \in \mathbb{R}^{q\times n}$ and $b_2 \in \mathbb{R}^q$.

For simplicity we assume here that the problems (L) and (U) always possess an optimal solution. Let x^k denote the optimal solution of (U) during the kth iteration. Let $\tilde{X}(0) = \emptyset$ and update $\tilde{X}(k)$ recursively by $\tilde{X}(k+1) = \tilde{X}(k) \cup x^k$. Further, let $\tilde{Y}(k) = y^k$ where y^k is an optimal solution of (L).

With this terminology (U) gets the form

$$\begin{aligned} \sup_x \ & cx - y^kA_1x \quad (+y^kb_1) \\ \text{s.t. } & A_2x \leqslant b_2 \\ & x \geqslant 0 \end{aligned} \tag{3.1}$$

in the kth iteration.

Let $v \in \mathbb{R}$. Then, in the kth iteration (L) gets the form

$$\begin{aligned} \inf_{y\geqslant 0} \ & v \\ \text{s.t. } & v \geqslant cx^i - yA_1x^i + yb_1 \quad \text{for } i = 1, \ldots, k \end{aligned}$$

or by linear programming duality

$$\begin{aligned} \sup_{\lambda_i} \ & \sum_{i=1}^{k} \lambda_i cx^i \\ \text{s.t. } & \sum_{i=1}^{k} \lambda_i A_1x^i \leqslant b_1 \\ & \sum_{i=1}^{k} \lambda_i = 1 \\ & \lambda_i \geqslant 0 \text{ for } i = 1, \ldots, k\,. \end{aligned} \tag{3.2}$$

The programs (3.1) and (3.2) are here recognized as the subproblem and the master problem, respectively, considered in the classical Dantzig–Wolfe decomposition procedure applied on the problem

$$\begin{aligned} \sup \ & cx \\ \text{s.t. } & A_1x \leqslant b_1 \\ & A_2x \leqslant b_2 \\ & x \geqslant 0\,. \end{aligned}$$

4. Variable Decomposition in General Mathematical Programming

This instance considers the Benders' decomposition principle in a more general context than originally introduced in Benders [1]. A major step in this direction has been taken by Geoffrion [6] and [13]. Further important steps have been done by Wolsey [8] who first presented a general procedure of this type. It is our purpose here to derive such a procedure as a special case of the procedure in Section 3.

For $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$ define $f(x, u): \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$ and $g(x, u): \mathbb{R}^n \times \mathbb{R}^p \to b \in \mathbb{R}^m$. Let $X \subseteq \mathbb{R}^n$ and $U \subseteq \mathbb{R}^p$. We can now consider a general mathematical programming problem with two sets of variables:

$$\begin{aligned} \sup\ & f(x, u) \\ \text{s.t. } & g(x, u) \leq b \\ & x \in X \\ & u \in U\,. \end{aligned} \tag{4.1}$$

We are going to solve this fairly general program by means of the decomposition procedure in Section 3.

Separate (4.1) into

$$\sup_{x \in X} \begin{bmatrix} \sup\limits_{u \in U} & f(x, u) \\ \text{s.t.} & g(x, u) \leq b \end{bmatrix}. \tag{4.2}$$

For fixed x we are going to consider the inner optimization problem in u of (4.2) and on this problem we shall use the duality theory considered in Section 2. Hence, we establish the generalized Lagrangean as created by (2.2). We shall further *assume* that we have selected the class $\mathcal{F}$ big enough to use Proposition 2.1 for any fixed x. In this way (4.2) is rewritten into

$$\sup_{x \in X} \inf_{F \in \mathcal{F}} [F(b) + \sup_{u \in U} (-F(g(x, u)) + f(x, u))]\,. \tag{4.3}$$

We are going to apply the decomposition procedure on (4.3). So in this case $Y = \mathcal{F}$ and $K(x, F) = F(b) + \sup_{u \in U} (-F(g(x, u)) + f(x, u))$. Furthermore, let x^* be an optimal solution of (U) in Step 1 at the kth iteration and let $\bar{X} = \tilde{X}(k) = x^*$, i.e. $\bar{X}$ is equal to the latest optimal solution. Let also F^* denote an optimal solution of (L) in Step 2 and let $\bar{Y} = \tilde{Y}(k) = \bar{\mathcal{F}}$, where $\bar{\mathcal{F}} := \bar{\mathcal{F}} \cup F^*$. That is $\bar{\mathcal{F}}$ is increased by the latest optimal solution.

If an optimal solution in (L) or (U) does not exist, the procedure must be modified along the lines that are usually followed in the case of linear programming. For example, this would replace an optimal solution by an extreme direction in case of unboundedness, etc.

With these specifications program (U) becomes

$$\bar{z} = \sup_{x \in X} \inf_{F \in \bar{\mathcal{F}}} [F(b) + \sup_{u \in U} (-F(g(x, u)) + f(x, u))]\,. \tag{4.4}$$

This is a direct generalization of what in the Benders' decomposition procedure is denoted as the master problem.

The problem (L) takes the form

$$\underline{w} = \inf_{F \in \mathcal{F}} [F(b) + \sup_{u \in U} (-F(g(x^*, u)) + f(x^*, u))] , \tag{4.5}$$

which is a generalization of the standard form of the subproblem in Benders' decomposition.

We note that since $\bar{X} = \{x^*\}$ we get by Remark 3.1 that Assumption 3.1 is fulfilled in the present case.

We also note that $\bar{X} = \{x^*\}$ implies satisfaction of (i) in Proposition 3.1. Hence, by this proposition the procedure stops if (ii) is satisfied, too, i.e. if

$$\inf_{F \in \mathcal{F}} K(x^*, y) = \inf_{F \in \bar{\mathcal{F}}} K(x^*, y) . \tag{4.6}$$

This equation expresses that the current selection $\bar{\mathcal{F}} \subseteq \mathcal{F}$ is sufficient to ensure the optimality of x^* in (4.3) and hence in (4.1). In other words, all the optimal solutions of (4.5) obtained so far and which all have been included in $\bar{\mathcal{F}}$ are enough to prove the optimality of x^*. In particular if $\mathcal{F}$ is finite then (4.6) must be true at some iteration. Hence in this case the procedure terminates after a finite number of steps. This is a frequently used argument for finite convergence and, as shown here, it can be used also in a very general setting.

A further criterion for termination is given in the following proposition.

PROPOSITION 4.1. The procedure terminates if

$$F(g(x, u)) \geq f(x, u), \forall F \in \bar{\mathcal{F}}, \forall x \in X \quad \textit{and} \quad \forall u \in U .$$

Proof. So, let $\sup_{u \in U} (-F(g(x, u)) + f(x, u)) \leq 0$ for all $x \in X$ and $F \in \bar{\mathcal{F}}$. Then

$$F^*(b) + \sup_{u \in U} (-F^*(g(x, u)) + f(x, u)) \leq F^*(b)$$

for all $x \in X$ and $F^* \in \bar{\mathcal{F}}$.

This implies that

$$\inf_{F \in \bar{\mathcal{F}}} [F(b) + \sup_{u \in U} (-F(g(x, u)) + f(x, u))] \leq F^*(b)$$

for all $x \in X$ and $F^* \in \bar{\mathcal{F}}$.

Hence

$$\bar{z} = \sup_{x \in X} \inf_{F \in \bar{\mathcal{F}}} [F(b) + \sup_{u \in U} (-F(g(x, u)) + f(x, u))] \leq F^*(b) \tag{4.7}$$

for all $F^* \in \bar{\mathcal{F}}$.

Now, consider the following program

$$\underline{w} = \inf_{F \in \mathcal{F}} F(b)$$
$$\text{s.t. } F(g(x^*, u)) \geq f(x^*, u) \ \forall u \in U . \tag{4.8}$$

From our continuing assumption about the selection of $\mathcal{F}$ we obtain by Proposition 2.1, part (i), that (4.5) and (4.8) are equivalent programs. The updating procedure of the lower bound l therefore implies that $F^*(b) = \underline{w} \leq l$ for all optimal solutions of (4.5). Similarly, the updating procedure of the upper bound implies that $\bar{z} \geqslant u$. Hence, by (4.7) $F^*(b) \geqslant u$. Since in general $l \leqslant u$ we obtain $l = F^*(b) = u$ and the procedure terminates. □

5. An Economic Interpretation of the Procedure

We shall give an economic interpretation of the procedure in the setting from the previous section. This will be done within the usual activity analysis framework. The problem here consists of two units in a planning environment. These units may be of equal rank of importance. However, in agreement with standard literature one unit shall be called the *central unit* and the other the *subunit*. In most instances it will probably be natural to deal with more subunits, but for ease of exposition only one subunit shall be considered here.

The objective is to solve (4.1) by means of the decomposition procedure where x indicates the activity levels to be determined by the central unit. Similarly, the subunit determines the level u for its activities. The function $f(x, u)$ indicates the overall income. The vector b denotes the capacity of some given production factors and $g(x, u)$ denotes the utilization of those capacities for given activity levels.

Here we assume that only the central unit has complete information about the set X which restricts the set of feasible solutions. Also the subunit has only complete information about the set U for its activities. Additionally, it is assumed that the subunit has all information required to compute the inner part of (4.2) for any given activity level x. However, instead of performing this computation for all potential activity levels at the central unit, the subunit provides cost information for these levels. This cost information is provided by the cost function F^* generated by the subunit as the solution of the lower problem in the form (4.5).

Since F^* in each iteration is going to be included in $\bar{\mathscr{F}}$, the central unit increases its cost information in this way. Based on this information new suggestions x^* for activity levels are proposed via the solution of the upper problem in the form of (4.4). In particular, if termination has not yet occurred then there exists by Proposition 4.1 an activity level $x \in X$ and a cost function $F \in \bar{\mathscr{F}}$ such that

$$\sup_{u \in U} f(x, u) - F(g(x, u)) > 0 \tag{5.1}$$

The function $F(d)$ can here be interpreted as an estimation of the alternate costs for the utilization of a resource vector $d \in \mathbb{R}^m$. In this terminology (5.1) expresses the natural statement that, since termination has not yet occurred, the subunit may contribute to an increase in the overall income via a change in x. So the procedure continues.

In order to keep information decentralized, the subunit subsumes in principle its knowledge into a function $G(x)$ defined on the activity levels belonging to the central unit. $G(x)$: $\mathbb{R}^n \to \mathbb{R} \cup \{+\infty\} \cup \{-\infty\}$ is given by

$$G(x) = F(b) + \sup_{u \in U} (f(x, u) - F(g(x, u))) \quad \forall x . \tag{5.2}$$

So we get a set $\bar{\mathscr{G}}$ of such functions, where

$$\bar{\mathscr{G}} = \{G \mid \exists F \in \bar{\mathscr{F}} \text{ where } G(x) = F(b) + \sup_{u \in U} (f(x, u) - F(g(x, u)))\} .$$

In this terminology the problem of the central unit subsumes to

$$\sup_{x \in X} \inf_{G \in \bar{\mathscr{G}}} G(x) . \tag{5.3}$$

Hence, in the present interpretation the subunit generates the functions $G(x)$ to be used by the central unit. Since the subunit does not know in advance the set X, the subunit must in principle compute $G(x)$ by means of (5.2) for all $x \in \mathbb{R}^n$. This might at first sight look laborious. However, taking the generality of the model (4.1) into account, this should not be too surprising, since the relationships between the activity levels x and u, given by the functions $f(x, u)$ and $g(x, u)$ can be very complicated. In the next section we shall see how the procedure simplifies, when more structure is introduced on these functions.

6. The Separable Case

This instance which has also been considered in Burkard, Hamacher and Tind [9] assumes separability of the objective function and the constraint functions in the following way. Let

$$f(x, u) = c(x) + e(u)$$

and $\quad g(x, u) = h(x) + k(u)$,

where $\quad c(x) : \mathbb{R}^n \to \mathbb{R},\ e(u) : \mathbb{R}^p \to \mathbb{R}$,

$$h(x) : \mathbb{R}^n \to \mathbb{R}^m \quad \text{and} \quad k(u) : \mathbb{R}^p \to \mathbb{R}^m.$$

In this setup (P) takes the form

$$\sup_{x \in X} \inf_{F \in \mathscr{F}} [F(b) + \sup_{u \in U} (-F(h(x) + k(u)) + c(x) + e(u))] . \tag{6.1}$$

We shall here see that a direct computation of $G(x)$ for all x as in the economic interpretation of (5.2) can be avoided.

For each $x \in \mathbb{R}^n$ and $F \in \mathscr{F}$ introduce the function $F_x : \mathbb{R}^n \to \mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$ defined by

$$F_x(d) = F(d + h(x)) \quad \text{for all } d \in \mathbb{R}^m .$$

Then $F(d) = F_x(d - h(x)) \quad$ for all $x \in \mathbb{R}^n$.

We assume that $F_x \in \mathscr{F}$ for all x. This is not a very restrictive assumption, and is satisfied in most cases. In this way (6.1) is converted into

$$\sup_{x \in X} \inf_{F_x \in \mathscr{F}} [F_x(b - h(x)) + \sup_{u \in U} (-F_x(k(u)) + e(u) + c(x))]$$

or by replacing F_x by F into

$$\sup_{x \in X} \inf_{F \in \mathscr{F}} [F(b - h(x)) + \sup_{u \in U} (-F(k(u)) + e(u)) + c(x)] . \tag{6.2}$$

Again, by proposition 2.1 (i), (6.2) is equivalent to

$$\begin{aligned} &\sup_{x \in X} \inf_{F \in \mathscr{F}} \; [F(b - h(x)) + c(x)] \\ &\quad \text{s.t.} \;\; F(k(u)) \geq e(u) \quad \text{for all } u \in U . \end{aligned}$$

Define $\mathscr{G} = \{F \in \mathscr{F} \mid F(k(u)) \geq e(u)\}$. Hence (6.2), which is our present form of (P), can be stated as

$$\sup_{x \in X} \inf_{F \in \mathscr{G}} [F(b - h(x)) + c(x)] .$$

Application of the decomposition procedure then simplifies the upper problem to

$$\sup_{x \in X} \inf_{F \in \bar{\mathscr{G}}} [F(b - h(x)) + c(x)] \tag{6.3}$$

where $\bar{\mathscr{G}} \subseteq \mathscr{G}$, and the lower problem gets the form

$$\inf_{F \in \mathscr{G}} [F(b - h(x^*)) + c(x^*)] .$$

By direct insertion for example in (P) the actual problem has the form

$$\begin{aligned} \sup \;\; & c(x) + e(u) \\ \text{s.t.} \;\; & h(x) + k(u) \leq b \\ & x \in X, u \in U . \end{aligned}$$

In an economic context for this problem along the lines from Section 5 $x \in X$ may typically stand for investment activities to be handled by the central unit and $u \in U$ may stand for production activities to be handled by a subunit. It is observed that this separable case does not require the computation of the function $G(x)$ for all x, as defined by (5.2), in order to solve the upper problem as stated in the form of (5.3). To solve the upper problem which is performed by the central unit, it sufficies here to operate directly with the cost function F. This cost function indicates the value of the resources $b - h(x^*)$ left over to the subunit. Also the subunit needs not to know the currently proposed activity level x^*. Information about the available resources $b - h(x^*)$ suffices together with the contribution $c(x^*)$ in the objective function.

It is the purpose for the central unit via the solution of the upper problem in the form of (6.3) to make an economic balance between the investment and production activities based on all currently known cost functions $\bar{\mathscr{G}}$.

If $e(u)$ and $k(u)$ are linear and $U \subseteq \mathbb{R}^p_+$ then we arrive at the original Benders' decomposition scheme based on linear programming duality. So, in this instance it suffices to let $\mathscr{G}$ contain linear functions.

7. Cross Decomposition

The idea of cross decomposition, as originally proposed by Van Roy [11], is to keep the problems (L) and (U) as simple as possible during the decomposition procedure. Let us, for simplicity, assume that (L) and (U) always have optimal solutions. Let x^k and y^k denote the optimal solutions in the kth iteration of (U) and (L), respectively. Then, the idea is to keep the sets $\bar{X}$ and $\bar{Y}$ small by setting $\bar{X} = \tilde{X}(k) = x^k$ and $\bar{Y} = \tilde{Y}(k) = y^k$. Cross decomposition can therefore be described as a coordinatewise search procedure.

In general, one cannot expect to obtain an optimal solution for (P) with this scheme. So occasionally, one has to modify the map $\tilde{X}(k)$ (or $\tilde{Y}(k)$) for example to the standard updating procedure for Benders' decomposition.

Some special instances are of particular interest because of the attractiveness of (U) and (L) in cross decomposition. Consider for example the classical capacitated plant location problem treated by Van Roy [14] with n plants and m markets. For this let d_j denote the demand of the jth market. Let s_i be the capacity of plant i and let f_i be the fixed cost for opening plant i. Let x_{ij} denote the transportation flow and c_{ij} the unit transportation costs from plant i to market j. Finally, introduce the binary variable y_i so that $y_i = 1$ if plant i is opened, and 0 otherwise.

We can now specify the terms of the general procedure. We shall here use max and min instead of sup and inf, respectively. Let $\mu = (\mu_1, \dots, \mu_n) \in \mathbb{R}^n_+$ and consider

$$K(y, \mu) = \min_{x_{ij}} \sum_{i=1}^{n} \sum_{j=1}^{m} (c_{ij} + d_j \mu_i) x_{ij} - \sum_{i=1}^{n} s_i y_i \mu_i + \sum_{i=1}^{n} f_i y_i$$

$$\text{s.t.} \sum_{i=1}^{n} x_{ij} = 1 \quad \forall j$$

$$0 \leq x_{ij} \leq y_i \quad \forall i, j .$$

Further, let $Y = \{(y_1, \dots, y_n) \mid y_i = 0 \text{ or } 1\}$ and $X = \mathbb{R}^n_+$. We will consider the problem in the form of (D). Hence, we consider

$$\min_{y \in Y} \max_{\mu \geq 0} K(y, \mu) . \tag{7.1}$$

By linear programming duality applied on the inner problem then (7.1) is equivalent to

$$\min_{y \in Y} \min_{x_{ij}} \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} + \sum_{i=1}^{n} f_i y_i$$

$$\text{s.t.} \sum_{i=1}^{n} x_{ij} = 1 \quad \forall j$$

$$\sum_{j=1}^{m} d_j x_{ij} \leq s_i y_i \quad \forall i$$

$$y_i = \begin{cases} 0 \\ 1 \end{cases}$$
$$x_{ij} \geq 0$$

which is the capacitated plant location problem in standard formulation.

In the cross decomposition scheme for (D) then (L) gets the following form with fixed $\mu = \mu^*$:

$$\begin{aligned} \min_{y_i} \min_{x_{ij}} & \sum_{i=1}^{n} \sum_{j=1}^{m} (c_{ij} + d_j \mu_i^*) x_{ij} - \sum_{i=1}^{n} s_i y_i \mu_i^* + \sum_{i=1}^{n} f_i y_i \\ \text{s.t.} & \sum_{i=1}^{n} x_{ij} = 1 \quad \forall j \\ & 0 \leq x_{ij} \leq y_i \quad \forall i, j \\ & y_i = \begin{cases} 0 \\ 1 . \end{cases} \end{aligned}$$

This is the simple plant location problem.

The upper problem (U) gets the form with fixed y_i^*:

$$\begin{aligned} \min_{x_{ij}} & \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} + \sum_{i=1}^{n} f_i y_i^* \\ \text{s.t.} & \sum_{i=1}^{n} x_{ij} = 1 \quad \forall j \\ & \sum_{j=1}^{m} d_j x_{ij} \leq s_i y_i^* \quad \forall i \\ & x_{ij} \geq 0 \end{aligned}$$

which is a transportation problem, providing the values of the dual variable μ_i^* belonging to the constraints, $\Sigma_{j=1}^{m} d_j x_{ij} \leq s_i y_i^*$.

8. Concluding Remarks

The basic decomposition procedure presented in Section 3 includes many special cases, for example the decomposition scheme of Kornai and Lipták [15] for block-angular LP-problems. It also includes the extension to general linear programming by Holmberg [16] and the various proposals in Holmberg [17]. The decomposition scheme for integer programming developed in Holm and Tind [18] is included, too. Finally, it should be mentioned that also approaches for comparison of different decomposition procedures fit well into the suggested framework, for example the comparison performed in Aardal and Ari [19] between the Kornai–Lipták and the cross decomposition procedure.

It all depends on the specific choice of the function $K(x, y)$, the sets X and Y, and the updating formulas for $\bar{X} = \tilde{X}(k)$ and $\bar{Y} = \tilde{Y}(k)$. Due to the simplicity and generality it is believed that all decomposition procedures in mathematical programming can be derived in this way from the present framework.

Hence a common basis has thus been established for categorization, comparison and further development of decomposition procedures.

References

1. Benders, J. F. (1962), Partitioning Procedures for Solving Mixed-Variables Programming Problems, *Numerische Mathematik* **4**, 238–252.
2. Dantzig, G. B. and Wolfe, P. (1961), The Decomposition Algorithm for Linear Programs, *Econometrica* **9**, 767–778.
3. Dirickx, Y. M. I. and Jennergren, L. P. (1979), Systems Analysis by Multi-Level Methods; With Applications to Economics and Management, Wiley.
4. Burton, R. M. and Obel, B. (1984), Designing Efficient Organizations; Modelling and Experimentation, *Advanced Series in Management* **7**, North-Holland.
5. Obel, B. (1981), Issues of Organizational Design; A Mathematical Programming View of Organizations, Pergamon Press.
6. Geoffrion, A. M. (1970), Primal Resource Directive Approaches for Optimizing Nonlinear Decomposable Systems, *Operations Research* **18**, 375–403.
7. Holloway, C. A. (1973), A Generalized Approach to Dantzig–Wolfe Decomposition for Concave Programs, *Operations Research* **21**, 210–220.
8. Wolsey, L. A. (1981), A Resource Decomposition Algorithm for General Mathematical Programs, *Mathematical Programming Study* **14**, 244–257.
9. Burkard, R. E., Hamacher, H. W., and Tind, J. (1985), On General Decomposition Schemes in Mathematical Programming, *Mathematical Programming Study* **24**, 238–252.
10. Flippo, O. E. (1989), Stability, Duality and Decomposition in General Mathematical Programming, PhD Thesis, Econometric Institute, Erasmus University, Rotterdam, The Netherlands.
11. Van Roy, T. J. (1983), Cross Decomposition for Mixed Integer Programming, *Mathematical Programming* **25**, 46–63.
12. Tind, J. and Wolsey, L. A. (1981), An Elementary Survey of General Duality Theory in Mathematical Programming, *Mathematical Programming* **21**, 241–261.
13. Geoffrion, A. M. (1972), Generalized Benders Decomposition, *J. of Optimization Theory and Applications* **10**, 237–260.
14. Van Roy, T. J. (1986), A Cross Decomposition Algorithm for Capacitated Facility Location, *Operations Reseach* **34**, 145–163.
15. Kornai, J. and Lipták, T. (1965), Two-Level Planning, *Econometrica* **33**, 141–169.
16. Holmberg, K. (1990), Linear Mean Value Cross Decomposition: A Generalization of the Kornai–Lipták Method, Publication no: 90/1, Department of Operations Research, University of Aarhus, Aarhus, Denmark, to appear in *European J. of Operational Research*.
17. Holmberg, K. (1985), Decomposition in Large Scale Mathematical Progamming, Dissertation, Department of Mathematics, Linköping University, Linköping, Sweden.
18. Holm, S. and Tind, J. (1988), A Unified Approach for Price Directive Decomposition Procedures in Integer Programming, *Discrete Applied Mathematics* **20**, 205–219.
19. Aardal, K. and Ari, A. (1990), On the Resemblance between the Kornai–Lipták and Cross Decomposition Techniques for Block-Angular Linear Programs, *European J. of Operational Research* **46**, 393–398.