# OPERATIONS RESEARCH CENTER

working paper

# MASSACHUSETTS INSTITUTE OF TECHNOLOGY

ACCELERATING BENDERS DECOMPOSITION:
ALGORITHMIC ENHANCEMENTS AND
MODEL SELECTION CRITERIA

by

Thomas L. Magnanti

and

Richard T. Wong

OR 093-79 October 1979

This research was supported by the U.S. Department of Transportation under Contract DOT-TSC-1058, Transportation Advanced Research Program (TARP).

Benders decomposition is an algorithm for mixed integer programming that has been applied successively to a variety of applications. Florian, Guerin, and Bushel [11] have used the algorithm to schedule the movement of railway engines, Richardson [33] has applied the algorithm to airline routing, and Geoffrion and Graves [16] have had great success applying the algorithm to design industrial distribution systems. Fisher and Jaikumar [9] have recently discussed the advantages of using the algorithm for vehicle routing problems. These contributions demonstrate the potential for using Benders decomposition to solve specially structured mixed integer programs.

Unfortunately, Benders procedure has not been uniformly successful in all applications. Geoffrion and Graves [16], among others, have noted that reformulating a mixed integer program can have a profound effect upon the efficiency of the algorithm. In earlier computational experience with a class of network design problems [38], we have observed that straightforward implementation of Benders algorithm often converged very slowly, requiring the solution of an exorbitant number of integer programming problems. This experience raises a number of questions. How does problem formulation effect the performance of Benders algorithm? Is there a "best" formulation for a given problem? For those applications where the algorithm does not work well, is there a mechanism for improving its convergence properties? The resolution of these questions would enhance prospects for applying Benders decomposition and has prompted the study reported in this paper.

In the first section, we review the essential concepts of Benders decomposition. In section 2, we describe an acceleration technique for reducing the number of iterations of Benders procedure. We accomplish this by choosing judiciously from the possible cuts that could be generated at any iteration to obtain "strong" or "pareto-optimal" cuts. This selection process involves a choice, made by solving a linear program, from the multiple optimal solutions of another linear program.

The results given in sections 1 and 2 apply to a broader minimax setting that also includes Dantzig-Wolfe decomposition for linear and nonlinear programs, and related "cutting plane" type algorithms that arise in resource directive and price directive decomposition. For the sake of generality, we cast our development of these two sections in the broader setting of cutting plane, or relaxation, algorithms which includes, as a special case, Benders procedure applied to general mixed integer programs.

In section 3, we show how this general methodology might be specialized for particular classes of applications. We consider Benders decomposition applied to facility location problems on networks, developing a very efficient algorithm for generating pareto-optimal cuts that exploits the underlying structure of these models. Since the linear program for generating cuts in these applications is the dual of a network optimization problem, multiple optimal solutions and, hence multiple choices for Benders cuts, will be commonplace, thus providing an excellent opportunity for applying our proposed methodology for selecting "good" cuts.

Section 4 describes a related approach for accelerating Benders decomposition via the "proper" formulation of mixed integer programs. Improved formulations can also allow the generation of stronger cuts for Benders decomposition. We discuss criteria for comparing various mixed integer formulations of a problem.

The final section summarizes our results and suggests possibilities for further investigation.

## 1. BENDERS DECOMPOSITION AND MINIMAX OPTIMIZATION

### Minimax Problems

Two of the most widely-used strategies for solving large scale optimization problems are resource directive decomposition and Lagrangian relaxation. Several papers in the mathematical programming literature (see, for example, Geoffrion [13] and [14], and Magnanti [22]) point out the central importance and unifying nature of these solution techniques. The techniques are not only applied directly; their use is, at times, combined with other approaches as when Lagrangian relaxation is embedded within the framework of branch and bound for solving integer programming problems (Fisher and Shapiro [10], Geoffrion [15]).

Since Benders algorithm, the focus of our analysis, is but one manifestation of resource directive decomposition, we shall consider a broader, but somewhat more abstract, minimax setting that captures the essence of both the resource directive and Lagrangian relaxation approaches. We study the optimization problem

$$v = \underset{y \in Y}{\text{Min}} \ \underset{u \in U}{\text{Max}} \{f(u) + yg(u)\} \tag{1}$$

where Y and U are given subsets of $R^k$ and $R^m$, f is a real valued function defined on U and g(u) is an m-dimensional vector for any $u \in U$. Note that we are restricting the objective function $f(u) + yg(u)$ to be linear-affine in the outer minimizing variable y for each choice of the inner maximizing variable u.

Benders decomposition leads to this minimax setting by considering the following mixed integer program:

$$\begin{aligned} \text{Minimize} \quad & cx + dy \\ \text{subject to:} \quad & Ax + Dy = b \qquad\qquad (2) \\ & x \geq 0,\ y \in Y . \end{aligned}$$

In this formulation, x is an n-vector of continuous variables, y is a k-vector of discrete variables, and Y is a subset of the integer points in k-dimensions. The matrices A and D and vectors c, d, and b have dimensions compatible with those of x and y.

We can formulate the mixed integer program in the equivalent form:

$$\underset{y \in Y}{\text{Minimize}} \quad \underset{\substack{x \geq 0 \\ Ax = b - Dy}}{\text{Minimize}} \{cx + dy\} \qquad\qquad (3)$$

For any fixed value of y, the inner minimization is a linear program. If it is feasible and has an optimal solution for all $y \in Y$[1], then dualizing gives the equivalent formulation

$$\underset{y \in Y}{\text{Minimize}} \quad \underset{u \in U}{\text{Maximize}} \{ub - uDy + dy\}$$

which is a special case of (1) in which $U = \{u \in R^m : uA \leq c\}$, $f(u) = ub$,

---

[1] These assumptions can be relaxed quite easily, but with added complications that cloud our main development. See Garfinkel and Nemhauser [12] or Lasdon [20] for a review of the algorithm in full generality.

and $g(u) = d - uD$. This reformulation is typical of the resource directive philosophy of solving parametrically in terms of complicating variables, like the integer variables y of a mixed integer program.

The minimax problem (1) also arises when dualizing the constraints $g(u) \geq 0$ of the optimization problem

$$\begin{array}{ll} \text{Maximize} & f(u) \\ \text{subject to:} & g(u) \geq 0 \\ & u \in U\ . \end{array} \tag{4}$$

The resulting optimization problem is the Lagrangian dual, a form of the minimax problem in which Y is the nonnegative orthant, or, more generally, the convex subset of the nonnegative orthant for which the maximization problem over U is finite valued.

<u>Solving minimax problems by relaxation</u>

For any given $y \in Y$, let $v(y)$ denote the value of the maximization problem in (1); that is,

$$v = \min_{y \in Y} v(y)$$

where

$$v(y) = \max_{u \in U} \{f(u) + yg(u)\}\ . \tag{5}$$

Since $v(y)$ is defined as the pointwise maximum of linear-affine functions, it is convex, though generally nondifferentiable. Consequently, whenever the set Y is convex, the minimax problem can be viewed as a convex program. There has been a great flourish of activity recently in modifying and extending algorithms of differentiable optimization to solve this class of problems (see Dem'yanov and Malazemov [ 7 ], Lemarechal [21], Mifflin [27], Wolfe [37], and the references that they cite). An alternative solution strategy that applies even when Y is not convex is a relaxation approach. Rewrite (4) as

$$\begin{aligned} &\text{Minimize} && z \\ &\text{subject to:} && z \geq f(u) + yg(u) \quad \text{for all } u \in U \\ & && y \in Y, \quad z \in R \end{aligned} \tag{6}$$

and form a relaxation

$$\begin{aligned} &\text{Minimize} && z \\ &\text{subject to:} && z \geq f(u^j) + yg(u^j) \quad (j = 1,2,\ldots,K) \\ & && y \in Y, \quad z \in R \end{aligned} \tag{7}$$

where each $u^j$ is an element of U. The solution $y^K$, $z^K$ of this "master problem" (7) is optimal in (6) if it satisfies all of the constraints of that problem; that is, if $v(y^K) \leq z^K$. If, on the other hand, $v(y^K) > z^K$, and $u^{K+1}$ solves[2] the "subproblem" (5) when $y = y^K$, then we add

$$z \geq f(u^{K+1}) + yg(u^{K+1})$$

as a new constraint, or cut as it is usually called, to the master problem (7). The algorithm continues in this way, alternately solving the master problem and subproblem.

When applied to problems (3) and (4), this algorithm is known, respectively, as Benders Decomposition and Dantzig-Wolfe Decomposition or

---

[2] As before, to simplify our discussion we assume that this problem always has at least one optimal solution.

generalized programming. The master problem is an integer program with one continuous variable when Benders algorithm is applied to mixed integer programs; it is a linear program when Dantzig-Wolfe decomposition is applied to nonlinear programs. The convergence properties of the relaxation algorithm are well-known, although usually stated in the context of particular instances of the algorithm, (see, for example, Benders [ 3 ], Dantzig [5, CH.24 ], and Magnanti et al. [23]). If the subproblem is a linear program, then the point $u^j$ in (7) can be chosen as extreme points of U and the algorithm terminates after a finite number of iterations. If the set U is compact and the functions f and g are continuous, then any limit point $y^* \varepsilon Y$, if one exists, to the sequence $\{y^K\}_{K\geq 1}$ is optimal in (1). Neither of these convergence properties depends upon structural properties of Y. Nevertheless, the structure of Y does determine whether or not the master problem (7) can be solved efficiently.

## 2. ACCELERATING THE RELAXATION ALGORITHM

A major computational bottleneck in applying Benders Decomposition is that the master problem, which must be solved repeatedly, is an integer program. Even when the master problem is a linear program as in the application of Dantzig-Wolfe Decomposition, the relaxation algorithm has not generally performed well due to its poor convergence properties (Orchard-Hays [31], Wolfe [36]). There are several possibilities for improvement:

(i) making a good selection of initial cuts, i.e., values of the $u^j$, for the master problem;

(ii) modifying the master problem to alter the choice of $y^K$ at each step;

(iii) formulating the problem "properly"; and

(iv) if there are choices, selecting good cuts to add to the master problem at each step.

In a number of studies of mixed integer programs, Mevert [26] found that the initial selection of cuts can have a profound effect upon the performance of Benders algorithm. Geoffrion and Graves [16] have reported similar experience with facility location problems.

There have been several proposals to alter the master problem for Dantzig-Wolfe Decomposition. Nemhauser and Widhelm [28] (see also O'Neill and Widhelm [30]) show that scaling the constraints of the master problem to find the "geometrically centered" value of $y^K$ at each step, can be beneficial. Marsten, Hogan, and Blankenship [25], (see also Marsten [24]), have had success in restricting the solution to the master problem at each step to lie within a box centered about the previous solution. Hollaway [19] shows how to select among multiple optima of the master problem to obtain better convergence.

Several researchers have illustrated the importance of problem formulation. Two different formulations of the same problem might be identical in terms of feasible solutions, but might be distinguishable in other ways. For example, they might have different linear programming or Lagrangian relaxations, one being preferred to the other when used in conjunction with algorithms like branch and bound or Benders decomposition. Recent studies by Cornuejols, Fisher, and Nemhauser [4] and Geoffrion and McBride [17] provide theoretical insight and computational experience concerning the role of model formulation in Lagrangian relaxation. Davis and Ray [6], Beale and Tomlin [2], and Williams [35], in the context of linear programming relaxation for branch and bound, and Geoffrion and Graves [16], in the context of Benders decomposition applied to facility location models, show that proper model formulation can generally improve the computational efficiency of these procedures. Section 4 explores further the question of proper model formulation for mixed integer programs in the context of Benders decomposition.

In many instances, as when Benders decomposition is applied to network optimization problems, the selection of good cuts at each iteration becomes an issue. Recall from section 1 that *any* solution to the subproblem (5) at each iteration of a relaxation algorithm such as Benders decomposition defines a cut. In network applications where the matrix A of the mixed integer program (2) models network flow structure, multiple optimal solutions to the subproblem are the norm; equivalently, degenerate solutions to the inner maximization problem in (3)

$$\text{Minimize} \quad \{dy + cx : Ax = b - Dy,\ x \geq 0\}$$

are to be expected because the shortest route, transshipment, and other network optimization problems are reknowned for their degeneracy. In this section, and the following one, we introduce methods and algorithms for choosing from the alternative optima to (5) at each iteration, a solution that generates a cut that is in some sense "best".

First, we must formalize some definitions. We say that the cut (or constraint), $z \geq f(u^1) + yg(u^1)$ in the minimax problem (1) *dominates* or is *stronger* than the cut, $z \geq f(u) + yg(u)$, if $f(u^1) + yg(u^1) \geq f(u) + yg(u)$ for all $y \in Y$ with a strict inequality for at least one point $y \in Y$. We call a cut *pareto optimal* if no cut dominates it. Since a cut is determined by the vector $u \in U$, we shall also say that $u^1$ dominates (is stronger) than u if the associated cut is stronger, and we say that u is pareto optimal if the corresponding cut is pareto optimal. Let us call any point $y^o$ contained in the relative interior of $Y^c$, a *core point* of Y.

The following theorem provides a method for choosing from among the alternate optimal solutions to the subproblem (5) to generate pareto-optimal cuts.

Theorem 1: *Let* $y^o$ *be a core point of* Y, *i.e.,* $y^o \in ri(Y^c)$, *let* $U(\hat{y})$ *denote the set of optimal solutions to the optimization problem*

$$\max_{u \in U} \{f(u) + \hat{y}g(u)\} \tag{8}$$

*and let* $u^o$ *solve the problem:*

$$\max_{u \in U(\hat{y})} \{f(u) + y^o g(u)\} \ . \tag{9}$$

*Then* $u^o$ *is pareto optimal.*

Proof: Suppose to the contrary that $u^o$ is not pareto optimal; that is, there is a $\bar{u} \in U$ that dominates $u^o$. We first note that since

$$f(\bar{u}) + yg(\bar{u}) \geq f(u^o) + yg(u^o) \qquad \text{for all } y \in Y, \tag{10}$$

it is true that

$$f(\bar{u}) + wg(\bar{u}) \geq f(u^o) + wg(u^o) \qquad \text{for all } w \in Y^c \ . \tag{11}$$

To establish the last inequality, recall that any point $w \in Y^c$ can be expressed as a convex combination of a finite number of points in Y, i.e., $w = \Sigma\{\lambda_y y : y \in Y\}$, where $\lambda_y \geq 0$ for all $y \in Y$, at most a finite number of the $\lambda_y$ are positive, and $\Sigma\{\lambda_y : y \in Y\} = 1$.

Also, note from the inequality (10) with $y = \hat{y}$, that $\bar{u}$ must be an optimal solution to the optimization problem (8), that is, $\bar{u} \in U(\hat{y})$. But then (10) and (9) imply that

$$f(\bar{u}) + y^o g(\bar{u}) = f(u^o) + y^o g(u^o) \ . \tag{12}$$

Since $\bar{u}$ dominates $u^o$,

$$f(u^o) + \bar{y}g(u^o) < f(\bar{u}) + \bar{y}g(\bar{u}) \tag{13}$$

for at least one point $\bar{y} \in Y$. Also, since $y^o \in ri(Y^c)$, there exists (see [34, Theorem 6.4]) a scalar $\theta > 1$ such that $w \equiv \theta y^o + (1 - \theta)\bar{y}$ belongs to $Y^c$. Multiplying equation (12) by $\theta$ and multiplying inequality (13) by $(1 - \theta)$, which is negative and reverses the inequality, and adding gives:

$$f(u^o) + wg(u^o) > f(\bar{u}) + wg(\bar{u}) \ .$$

But this inequality contradicts (11), showing that our supposition that $u^o$ is not pareto optimal is untenable. This completes the proof. □

When $f(u) = ub$, $g(u) = (d - uD)$, and $U = \{u \in R^k : uA \leq c\}$, as in Benders Decomposition for mixed integer programs, problem (8) is a linear program. In this case, $U(\hat{y})$ is the set of points in U satisfying the linear equation, $u(b - D\hat{y}) = v(\hat{y}) - d\hat{y}$, where $v(\hat{y})$ is the optimal value of the subproblem (5). Therefore, to find a pareto optimal point among all the alternate optimal solutions to problem (8), we solve problem (9), which is the linear program:

$$\begin{aligned} \text{Maximize} \quad & \{dy^o + u(b - Dy^o)\} \\ \text{subject to:} \quad & u(b - D\hat{y}) = v(\hat{y}) - d\hat{y} \\ \text{and} \quad & uA \leq c \ . \end{aligned} \tag{14}$$

We should note that varying the core point $y^o$ might conceivably generate different pareto optimal cuts. Also, any implementation of a strong cut version of Benders algorithm has the option of generating pareto optimal cuts at every iteration, or possibly, of generating these cuts only periodically. The tradeoff will depend upon the computational burden of solving problem (9) as compared to the number of iterations that it saves.

In many instances, it is easy to specify a core point $y^o$ for implementing the pareto optimal cut algorithm. If, for example, $Y = \{y \in R^k : y \geq 0 \text{ and integer}\}$, then any point $y^o > 0$ will suffice; if $Y = \{y \in R^k : y_j = 0 \text{ or } 1 \text{ for } j = 1,2,...,k\}$, then any vector $y^o$ with $0 < y_j^o < 1$ for $j = 1,2,...,k$ suffices; and if

$$Y = \{y \in R^k : \sum_{j=1}^{k} y_j \leq p,\ y \geq 0 \text{ and integer}\},$$

as in the inequality version of the p-median problem, then any point $y^o$ with $y^o > 0$ and $\sum_{j=1}^{k} y_j^o < p$ suffices. In particular, if $p > k/2$, then $y^o = (1/2,1/2,...,1/2)$ is a core point.

One particular version of the preceding theorem merits special mention. Suppose that U is a product of sets $U = U^1 \times U^2 \times ... \times U^J$ and that f and g are additively separable over the sets $U^j$; that is,

$$f(u) = \sum_{j=1}^{J} f_j(u_{(j)})$$

and

$$g(u) = \sum_{j=1}^{J} g_j(u_{(j)}),$$

where $u = (u_{(1)},u_{(2)},...,u_{(J)})$ is a partition of u with $u_{(j)} \in U^j$. The notation $u_{(j)}$ distinguishes this vector from the component $u_j$ of $u \in U$. Then, for any $y \in Y$, the subproblem (5) separates as:

$$v(y) = \sum_{j=1}^{J} v_j(y)$$

where, for each j,

$$v_j(y) = \max_{u_{(j)} \varepsilon U^j} \{f_j(u_{(j)}) + yg(u_{(j)}) \ . \qquad (15)$$

Since, for any $u_{(j)} \varepsilon U^j$,

$$f_j(u_{(j)}) + \hat{y}g_j(u_{(j)}) \leq v_j(\hat{y})$$

the vector u belongs to $U(\hat{y})$, meaning that the sum over j of the lefthand sides of these expressions equals the sum of the righthand sides if, and only if,

$$f_j(u_{(j)}) + \hat{y}g_j(u_{(j)}) = v_j(\hat{y})$$

for all j. That is, choosing u to be one of the alternate optimal solutions to (8) is equivalent to $u_{(j)}$ being an alternate optimal solution to (15) when $y = \hat{y}$. Consequently, finding a pareto optimal cut decomposes into independent subproblems, as recorded formally in the following corollary stated in terms of the notation just introduced.

Corollary 1: *Let* $y^o$ *be a core point of* Y, *and, for each* $j = 1,2,\ldots,J$, *let* $U^j(\hat{y})$ *denote the set of optimal solutions to the optimization problem*

$$\max_{u_{(j)} \varepsilon U^j} \{f_j(u_{(j)}) + \hat{y}g_j(u_{(j)})\}$$

*and let* $u^o_{(j)}$ *solve the problem:*

$$\max_{u_{(j)} \varepsilon U^j(\hat{y})} \{f_j(u_{(j)}) + y^o g_j(u_{(j)})\} .$$

*Then* $u^o = (u^o_{(1)}, u^o_{(2)}, \ldots, u^o_{(J)})$ *is pareto optimal for (1).*

The separability of f and g in this discussion has historically been a major motivation for considering resource directive decomposition and Lagrangian relaxation. Whenever the constraint matrix A of the variables x in problem (2) is block diagonal, this separability property applies. In this case, problem (14) decomposes into several linear programs, one for each subvector $u_{(j)}$ of u. In Lagrangian relaxation, dualizing induces separability in the subproblem, from the "complicating constraints"

$$g(x) \equiv \sum_{j=1}^{J} g_j(u_{(j)}) \geq 0$$

of the original problem formulation (4) if $U = U^1 \times U^2 \times \ldots \times U^J$ is separable.

## 3. ACCELERATING BENDERS METHOD FOR NETWORK OPTIMIZATION

Although solving the linear program (14) always generates pareto optimal cuts whenever Benders method is applied to mixed integer programs, it might be possible to generate strong cuts more efficiently in certain situations. In particular, when the Benders subproblem involves network optimization, special purpose network algorithms might be preferred to the general purpose methodology.

In this section, we describe special network algorithms for generating strong cuts for facility location problems. We begin by considering a facility location problem formulated as the following mixed integer program:

$$v = \text{Min} \quad \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij}x_{ij} + \sum_{j=1}^{m} d_j y_j$$

$$\begin{aligned}
\text{subject to:} \quad & \sum_{j=1}^{m} x_{ij} \geq 1 && (i = 1,2,\ldots,n) && (16)\\
& x_{ij} \leq y_j && (1 \leq i \leq n) \\
& x_{ij} \geq 0 && (1 \leq j \leq m) \\
& y_j = 0 \text{ or } 1 \\
& y \varepsilon Y
\end{aligned}$$

where m = number of potential facilities

n = number of customers

and Y = set of feasible values for $y \subseteq (0,1)^n$ .

If $y_j = 1$, we construct facility j and incur a fixed cost of $d_j$. If $x_{ij} = 1$, customer i receives service at facility j. The first constraint requires that each customer be serviced by some facility. The second constraint states that no customer can be serviced at a facility unless that facility is constructed. The references cited at the outset of section 2 and our discussion in section 4 suggest reasons for choosing this particular form of the problem formulation instead of an equivalent formulation with constraints $\sum_i x_{ij} \leq ny_j$ for all j in place of the constraints $x_{ij} \leq y_j$ for all i and j.

If $Y = \{y | \sum_{j=1}^{m} y_j = p\}$, n = m, and $c_{jj} = 0$ for all j, then (16) becomes the well-known uncapacitated plant location problem. The set Y might incorporate a number of additional conditions imposed upon the configuration of open (i.e., $y_j = 1$) facilities. Among these might be contingency constraints such as "location i is opened only if location j is opened," multiple choice constraints such as "open at most two of the locations i, j, and k," and other conditions of this nature.

Suppose we fix $y = \bar{y} \in Y$; then (16) reduces to the following pure linear programming subproblem:

$$
\begin{aligned}
v(\bar{y}) = \text{Min} \quad & \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} \\
\text{subject to:} \quad & \sum_{j=1}^{m} x_{ij} \geq 1 \qquad (i = 1,2,\ldots,n) \\
& 0 \leq x_{ij} \leq 1 \qquad j \in \mathcal{O} \\
& 0 \leq x_{ij} \leq 0 \qquad j \in \mathcal{C} \\
& \qquad\qquad\qquad (1 \leq i \leq n)
\end{aligned}
\tag{17}
$$

where

$\mathcal{O} = \{j \mid \bar{y}_j = 1\}$, the set of *open* facilities, and

$\mathcal{C} = \{j \mid \bar{y}_j = 0\}$, the set of *closed* facilities.

The linear program dual of this problem is:

$$
\begin{aligned}
v(\bar{y}) = \text{Max} \quad & \sum_{i=1}^{n} [\lambda_i - \sum_{j=1}^{m} \bar{y}_j \pi_{ij}] \\
\text{subject to:} \quad & \lambda_i - \pi_{ij} \leq c_{ij} \qquad (1 \leq i \leq n) \\
& \lambda_i \geq 0 \qquad (1 \leq j \leq m) \\
& \pi_{ij} \geq 0 \,.
\end{aligned}
\tag{18}
$$

Any solution to this problem determines a cut of the form:

$$v \geq \sum_{i=1}^{n} (\lambda_i - \sum_{j=1}^{m} \pi_{ij} y_j) + \sum_{j=1}^{m} d_j y_j \ . \tag{19}$$

(Note that we have appended the term $\Sigma d_j y_j$ to the right hand side of the cut. This term was omitted from the objective function of the subproblem (17) because it is a constant for any given choice of the configuration variables $y_j$.)

Careful inspection of the linear program (17) reveals that, for most problems, it will have a degenerate optimal basis. This implies that it usually will be possible to derive more than one Benders cut. We next describe a procedure for generating pareto-optimal cuts.

Note that for any choice of $\bar{y} \in Y$, the linear programs (17) and (19) decompose into separate subproblems, one for each index $i = 1,2,\ldots,n$. Also, the "natural solution" (see Balinski [ 1 ])[3]

$$\bar{\lambda}_i = c_{ij(i)} \equiv \min \{c_{ij};\ j \in \mathcal{O}\}$$

$$\bar{\pi}_{ij} = 0 \qquad\qquad \text{if } j \in \mathcal{O} \tag{20}$$

and

$$\bar{\pi}_{ij} = \max\ (0, \bar{\lambda}_i - c_{ij}) \qquad \text{if } j \in \mathcal{C}\ ,$$

to the linear programming dual problem (18) has the property that the optimal value of the $i^{th}$ subproblem is $v_i(y) = \lambda_i$. Consequently, corollary 1 with $u_{(i)} = (\lambda_i, \pi_{i1}, \pi_{i2}, \ldots \pi_{in})$ implies that solving for each i the

---

[3] The optimal dual variables have a convenient interpretation in terms of the facility location problem. $\bar{\lambda}_i$ is the cost of servicing customer i when $y = \bar{y}$. $\bar{\pi}_{ij}$ is the reduction in the cost of servicing customer i when facility j is opened and $y_i = \bar{y}_i$ for all $i \neq j$.

the subproblem

$$\text{Max} \quad [\lambda_i - \sum_{j=1}^{m} y_j^o \pi_{ij}]$$

$$\text{subject to:} \quad \lambda_i - \sum_{j=1}^{m} \bar{y}_j \pi_{ij} = \bar{\lambda}_i \qquad (21)$$

$$\lambda_i - \pi_{ij} \leq c_{ij} \quad (j = 1,2,\ldots,m)$$

$$\pi_{ij} \geq 0$$

$$\lambda_i \geq 0$$

provides a pareto-optimal vector with components $\lambda_i$ and $\pi_{ij}$ for $i = 1,2,\ldots,n$ and $j = 1,2,\ldots,m$. Here, as before, $\bar{y}$ denotes the current value of the integer variables and $y^o$ belongs to the core of y, i.e., $y^o \varepsilon \, ri(Y)^c$.

Our first objective is to show that, for each i, the subproblem (21) is piecewise linear as a function of $\lambda_i$. Note that, since the equality constraint of this problem reads,

$$\lambda_i - \sum_{j\varepsilon \mathcal{O}} \pi_{ij} = \bar{\lambda}_i = c_{ij(i)}$$

and since $$\lambda_i - \pi_{ij(i)} \leq c_{ij(i)}$$

and $$\pi_{ij} \geq 0 \quad \text{for all } j,$$

it must be true that $$\pi_{ij} = 0 \quad \text{for all } j \neq j(i),\ j \ \varepsilon \ \mathcal{O} \quad \text{and}$$

$$\pi_{ij(i)} = \lambda_i - c_{ij(i)} = \lambda_i - \bar{\lambda}_i \ .$$

Also, if we substitute for $\lambda_i$ in the objective function of (21) from the equality constraint, the objective becomes:

$$\text{Max} \quad \bar{\lambda}_i + \sum_{j=1}^{m} (\bar{y}_j - y_j^o)\pi_{ij} \ .$$

Consider any index $j \in \mathcal{C}$. Since $\bar{y}_j = 0$, the coefficient

$$\varepsilon_j \equiv \bar{y}_j - y_j^o$$

of $\pi_{ij}$ is nonpositive. Thus, an optimal choice of $\pi_{ij}$ satisfying the two constraints

$$\lambda_i - \pi_{ij} \leq c_{ij} \text{ and } \pi_{ij} \geq 0$$

is
$$\pi_{ij} = \max \ \{0, \lambda_i - c_{ij}\} \ .$$

Collecting these results, we see that the optimal value of problem (21) as a function of the variable $\lambda_i$ is:

$$\bar{\lambda}_i + \varepsilon_{j(i)}(\lambda_i - \bar{\lambda}_i) + \sum_{j \in \mathcal{C}} \varepsilon_j \ \max \ \{0, \lambda_i - c_{ij}\} \ .$$

As an aid to optimizing (21), we note the following upper and lower bounds on $\lambda_i$:

$$\bar{\lambda}_i \leq \lambda_i \leq L_i$$

where, by definition, $L_i = \min \ \{c_{ij} : j \in \mathcal{O} \text{ and } j \neq j(i)\}$. The lower bound is simply a consequence of the equality constraint of problem (21), because each $\bar{y}_j \geq 0$ and each $\pi_{ij} \geq 0$. The upper bound is a consequence of our previous observation that, for all $j \neq j(i)$ and $j \in \mathcal{O}$, $\pi_{ij} = 0$ and, therefore, the constraint $\lambda_i - \pi_{ij} \leq c_{ij}$ becomes $\lambda_i \leq c_{ij}$.

Now, since the function (27) is piecewise linear and concave in $\lambda_i$, we can minimize it by considering the linear segments of the curve in the interval $\bar{\lambda}_i \leq \lambda_i \leq L_i$ in order from left to right until the slope of any segment becomes nonpositive. Formally,

(1) Start with $\lambda_i = \bar{\lambda}_i$ .

(2) Let $T = \{j \in C : c_{ij} \leq \lambda_i\}$ and let $s = \varepsilon_{ij(i)} + \Sigma\{\varepsilon_j : j \in T\}$ . $s$ is the slope of the function (25) to the right of $\lambda_i$.

(3) If $s \leq 0$, then stop; $\lambda_i$ is optimal. If $s > 0$ and $T = C$, then stop, $\lambda_i = L_i$ is optimal.

(4) Let $c_{ik} = \min \{c_{ij} : j \in C$ and $j \notin T\}$. If $L_i \leq c_{ik}$, set $\lambda_i = L_i$ and stop. Otherwise, increase $\lambda_i$ to $c_{ik}$. Repeat steps (2)-(4).

Once the optimal value of $\lambda_i$ is found using this algorithm for each i, the remaining variables $\pi_{ij}$ can be set using the rules given above. Then, by virtue of corollary 1, the cut obtained by substituting these values in (19) is pareto optimal.

The above algorithm should be very efficient. For each customer i, at most m (m = # of possible facilities) steps must be executed. So in the worst case, the number of steps required by this procedure is bounded by (# of customers) (# of possible facilities).

We might emphasize that this algorithm determines a pareto optimal cut for any given point $y^o$ in the core of Y. Also, the algorithm applies to any of the possible modeling variations that we might capture in Y, such as the contingency and configuration constraints mentioned at the beginning of this section.

This algorithm can also be extended to more complex models like the capacitated facility location problem [17,18]. The basic concepts of such a procedure are similar to the algorithm that we have just described and will not be given here.

It is interesting to note that the pareto-optimal algorithm specified above is similar to a dual ascent procedure proposed by Erlenkotter [ 8 , pp. 997-998]. Both procedures essentially give a set of rules for increasing the $\lambda_i$. One distinguishing feature of our algorithm is that it solves problem (21) exactly, whereas Erlenkotter's procedure gives an approximate solution to the dual of the linear programming relaxation of problem (16).

To conclude this section, we note that it is possible to generate cuts stronger than the natural cuts defined by setting $\lambda_i = \bar{\lambda}_i$ and $\pi_{ij} = \bar{\pi}_{ij}$ in (19), but without assurance of pareto-optimality. For details, see Balinski [1 ], Nemhauser and Wolsey [29], and Wong [38].

## 4. A MODEL SELECTION CRITERION FOR BENDERS DECOMPOSITION

Selecting the "proper" model formulation is another important factor that effects the computational performance of Benders decomposition applied to network design and other mixed integer programming models. This section discusses a criterion for distinguishing between different but "equivalent" formulations of the same mixed integer programming problem to identify which formulation is preferred in the context of Benders decomposition.

Many network optimization problems have several "natural" mixed integer formulations. For example, as we noted in section 3, various variations of the facility location problem can be stated in several possible ways as mixed integer programs. We demonstrate in this section why some formulations lead to such pronounced improvements over others in the performance of Benders decomposition (see also references [ 4 ], [ 6 ], and [15], cited earlier).

To illustrate the role of model selection, we consider an example of Benders decomposition applied to the p-median facility location problem [ 4 ]. The p-median problem can be formulated as:

$$\text{(P)} \qquad \text{Minimize} \quad \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} x_{ij}$$

$$\text{subject to:} \quad \sum_{i=1}^{N} x_{ij} = 1 \qquad \forall_j \tag{23}$$

$$x_{ij} \le y_i \qquad \forall (i,j) \tag{24}$$

$$\sum_{i=1}^{N} y_i = p \tag{25}$$

$$x_{ij} \ge 0 \ \text{ and } y_i \text{ integer } \forall (i,j) \ . \tag{26}$$

N is the number of nodes in the problem and p is the number of facilities to be located. $y_i$ indicates whether a facility is located at node i and $x_{ij}$ indicates whether customer j is serviced at node i.

As we noted in section 3, an equivalent formulation is:

$$\text{(Q)} \qquad \text{Minimize} \quad \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} x_{ij}$$

$$\text{subject to:} \quad (23), (25), (26), \text{ and}$$

$$\sum_{j=1}^{N} x_{ij} \le N y_i \ . \tag{27}$$

Note that (27) represents an aggregation of the constraints in (24). Consequently, although P and Q are equivalent mathematical descriptions, if we relax the integrality constraint on the $y_i$, the feasible region for

$Q$ is a proper subset of the feasible region for $P$ .

Let us examine the following p-median problem represented in figure 5.1:

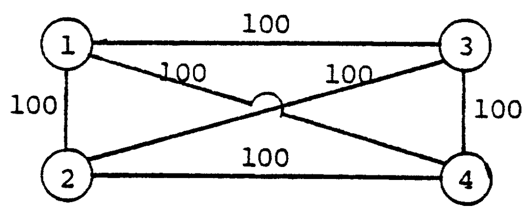


Figure 5.1 p-median Example

$N$ is equal to 4, $p$ is equal to 2, and all $d_{ij}$ are 100.

The application of Benders decomposition to this example formulated as $Q$ yields the following set of Benders cuts:

$$z \geq 200 - 400y_1 - 400y_2 + 0y_3 + 0y_4$$
$$z \geq 200 - 400y_1 + 0y_2 - 400y_3 + 0y_4$$
$$z \geq 200 - 400y_1 + 0y_2 + 0y_3 - 400y_4$$
$$z \geq 200 + 0y_1 - 400y_2 - 400y_3 + 0y_4$$
$$z \geq 200 + 0y_1 - 400y_2 + 0y_3 - 400y_4$$
$$z \geq 200 + 0y_1 + 0y_2 - 400y_3 - 400y_4$$

This set of cuts has the property that every single one must be generated in order for Benders algorithm to converge.

Applying Benders decomposition to our example formulated as $P$ requires the single cut:

$$z \geq 400 - 100y_1 - 100y_2 - 100y_3 - 100y_4 .$$

We can generalize this example in the following way: let $p = N/2$ and let $d_{ij} = 100$ for all $i \neq j$ and $d_{ij} = 0$ for all $i = j$. With this class of examples, we have problems where the Q formulation requires $\binom{N}{N/2}$ cuts, an exponential number of cuts with respect to N, for Benders algorithm to converge. For these same problems, the P formulation, in every case, requires only one Benders cut for convergence! This example dramatically illustrates the importance of intelligent model formulation for Benders decomposition.

Now we present a formal framework for comparing model formulations for Benders decomposition. This framework is then utilized to prove our main results.

Suppose we have two mixed integer programs P and Q that are represented as:

$$\text{(P)} \quad \underset{y \in Y}{\text{Minimize}}\ [v^P(y)] \qquad \text{where } [v^P(y)] = \text{Minimum} \quad cx + dy$$
$$\text{subject to:} \quad Ax + By = b$$
$$x \geq 0$$

and

$$\text{(Q)} \quad \underset{y \in Y}{\text{Minimize}}\ [v^Q(y)] \qquad \text{where } [v^Q(y)] = \text{Minimum} \quad hw + dy$$
$$\text{subject to:} \quad Dw + Gy = g$$
$$w \geq 0$$

x,w, and y are column vectors of problem variables; b and g are column vectors; c, d, and h are row vectors; A, B, D, and G are appropriately dimensioned matrices. The set Y is a set of *integer* valued vectors that captures the integer constraints of the problem. We assume that the set Y is finite.

We will say that P and Q are *equivalent* mixed integer programming representations of the same problem if $v^P(y) = v^Q(y)$ for all $y \in Y$ .

That is, the two models have the same integer variables and may have different continuous variables and constraints, but always give the same objective function value for any feasible assignment of the integer variables. We will say that the two formulations are *identical* if $v^P(y) = v^Q(y)$ for all y belonging to the convex hull of Y.

In the context of Benders decomposition, another possible interpretation of equivalence is that $v^P(y)$ and $v^Q(y)$ represent the linear programming subproblems when Benders decomposition is applied to P and Q. So the two models are equivalent if their respective Benders subproblems always have the same optimal value.

We evaluate these two models by comparing the cuts generated from the application of Benders decomposition to these models. Following the derivation of Benders decomposition given in section 1, we can rewrite P and Q as[4]

$$\begin{array}{lll} \text{Minimize} & z & \\ \text{subject to:} & z \geq \pi(b - By) + dy & \pi \in \Pi \\ & y \in Y & \end{array}$$

where $\Pi$ is the set of points in the polyhedron $\pi A \leq c$; and

$$\begin{array}{lll} \text{Minimize} & z & \\ \text{subject to:} & z \geq \gamma(g - Gy) + dy & \gamma \in \Gamma \\ & y \in Y & \end{array}$$

---

[4]As in earlier sections, we assume that the linear programming subproblems $v^P(y)$ and $v^Q(y)$ are feasible and have optimal solutions for all $y \in Y$. These constraints can be relaxed, but with added complications that do not enrich the development in an essential way.

where $\Gamma$ is the set of points of the polyhedron $\gamma D \leq h$.

The inequalities $z \geq \pi(b - By) + dy$ and $z \geq \gamma(g - Ey) + dy$ will be referred to as the Benders cuts for P and Q, respectively. We remark that our definition of Benders cuts, in which a cut can be generated from any point in the subproblem dual feasible region, produces a larger set of possible cuts than the usual definition which restricts the cuts to those corresponding to the extreme points of the subproblem dual feasible region. With this limited definition of Benders cuts, the results of this section need not always be valid. To compare equivalent model formulations, we adapt the concept of a pareto optimal cut, introduced in section 2, by saying that a Benders cut (or constraint)

$$z \geq \pi(b - By) + dy$$

for P *dominates* a Benders cut

$$z \geq \gamma(g - Gy) + dy$$

for Q if

$$\pi(b - By) + dy \geq \gamma(g - Gy) + dy$$

for all $y \in Y$ with a strict inequality for at least one point $y \in Y$.

A cut $z \geq \gamma(g - Gy) + dy$ for Q will be called *unmatched* with respect to the formulation P if there is no cut for P that is equal to it (in the sense that two cuts are equal if their right-hand sides are equal for all $y \in Y$) or dominates it.

A formulation Q is *superior* to an equivalent formulation P if Q has at least one Benders cut that is unmatched with respect to P, but P does not have any cuts that are unmatched with respect to Q.

In a very loose sense, Q is superior to P if they are equivalent formulations and the set of Benders cuts for P is a proper subset of the Benders cuts for Q.

With these definitions we can now prove several properties concerning model formulation and the strength of Benders cuts.

Lemma 1 *Let P and Q be equivalent formulations of a mixed integer programming problem. Q has a Benders cut that is unmatched with respect to P if, and only if, there is a* $y^o$ *belonging to the convex hull* $Y^c$ *of Y that satisfies* $v^Q(y^o) > v^P(y^o)$.

Proof: ($\Longrightarrow$) Let $z \geq \gamma^*(g - Gy) + dy$ be a Benders cut that is unmatched with respect to P. Since we are assuming that the set Y is finite, the definition of an unmatched cut implies:

$$\underset{\pi A \leq c}{\text{Max}} \; [\underset{y \in Y}{\min} \; \pi(b - By) + dy - \gamma^*(g - Gy) - dy] < 0 \; .$$

Now observe that the above inequality still holds if we replace the set Y by $Y^c$. Using linear programming duality theory, we can reverse the order of the max and min operation to obtain

$$\underset{y \in Y^c}{\text{Min}} \; [\underset{\pi A \leq c}{\max} \; \pi(b - By) + dy - \gamma^*(g - Gy) - dy] < 0 \; .$$

Linear programming duality theory, when applied to the inner maximization, allows us to rewrite the above expression as

$$\begin{array}{ll} \text{Min} & cx + dy - [\gamma^*(g - Gy) + dy] < 0 \\ \text{subject to:} & Ax + By = b \\ & x \geq 0, \quad y \in Y^c \; . \end{array}$$

This implies that there is a $y^o \in Y^c$ satisfying

$$\begin{array}{ll} \text{Min} & cx + dy^o = v^P(y^o) < \gamma^*(g - Gy) + dy^o \; . \\ \text{subject to:} & Ax = b - By^o \\ & x \geq 0 \; . \end{array}$$

Another application of linear programming duality theory, in this case to Q, gives us:

$$v^P(y^o) < \gamma^*(g - Gy^o) + dy^o \leq \begin{array}{ll} \text{Min} & hw + dy^o \\ \text{subject to:} & Dw = g - Gy^o \\ & w \geq 0 \end{array}$$

or

$$v^P(y^o) < v^Q(y^o) \ .$$

($\Longleftarrow$) The reverse implication has essentially the same proof with all the steps reversed. Explicit details will not be given here. □

This lemma leads to the following theorem about preferred formulations:

Theorem 2: *Let* P *and* Q *be equivalent formulations of a mixed integer programming problem.* Q *is superior to* P *if, and only if,* $v^Q(y) \geq v^P(y)$ *for all* $y \in Y^c$ *with a strict inequality for at least one* $y \in Y^c$ .

Proof: ($\Longleftarrow$) If $v^Q(y) \geq v^P(y)$ for all $y \in Y^c$, Lemma 1 says that P does not have any Benders cuts that are unmatched with respect to Q. But because there is a $y^o \in Y^c$ such that $v^Q(y^o) > v^P(y^o)$ Lemma 1 implies that Q has a cut that is unmatched in P. So Q satisfies the definition of being superior to P.

($\Longrightarrow$) If Q is superior to P, then P, by definition of superior, does not have any cuts that are unmatched with respect to Q. Lemma 1 then tells us that $v^Q(y) \geq v^P(y)$ for all $y \in Y^c$. The definition of superior also states that Q has a cut that is unmatched with respect to P and using lemma 1 we can say that there exists a $y^o \in Y^c$ such that $v^Q(y^o) > v^P(y^o)$. □

The implications of Theorem 2 may become more apparent when interpreted in another way. Let the *relaxed primal problem* for any formulation of a mixed integer program be defined by replacing Y by its convex hull $Y^c$. Theorem 2 states that for a formulation of a mixed integer programming problem, the smallest possible feasible region (or the "tightest" possible constraint set) for its relaxed primal problem is preferred for generating strong Benders cuts. For any formulation P, a smaller feasible region for its relaxed primal problem will result in larger values of the function $v^P(y)$ which Lemma 1 and Theorem 2 indicate is desirable.

As a concrete example, the p-median problem discussed at the beginning of this section has two formulations P and Q. They differ only in that P has constraints of the form $x_{ij} \leq y_i$ for all $(i,j)$, whereas Q has constraints of the form $\sum_{j=1}^{4} x_{ij} \leq 4y_i$ for all i.

Since the later set of constraints is an aggregation of the former constraints, the feasible region for the relaxed primal problem of P is no larger than that for Q. So $v^P(y) \geq v^Q(y)$ for all $y \in Y^c$. A straightforward computation shows that $v^P(y^o) = 200 > v^Q(y^o) = 0$ for $y^o = (\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2})$. So the formulation P is superior to Q for this example.

A general consequence of Theorem 2 is that for any mixed integer programming formulation, the convex hull of its feasible region will be a model formulation that is "optimal" in terms of generating Benders cuts since it has a relaxed primal problem whose feasible region is the smallest. In order to formalize this observation for any formulation P of a mixed integer program as in (2), let C(P) denote the mixed integer program whose feasible region is the convex hull of the feasible region for P.

Theorem 3: *Given any formulation* P *of a mixed integer program,* $v^{C(P)}(y) \geq v^Q(y)$ *for all* $y \in Y^c$ *and for all equivalent formulations* Q *of this problem.*

Proof: Let $y^* \in Y^c$ be arbitrary and let $x^*$ be an optimal solution to C(P) when $y = y^*$; that is, $v^{C(P)}(y^*) = cx^* + dy^*$. By definition of convex hull $(y^*,x^*)$ is a convex combination with weights $\lambda_i$ of a finite number of points $(y^i,x^i)$ that are feasible in P. Linearity of the objective function $cx + dy$ implies that $cx^* + dy^* = \Sigma\, \lambda_i(cx^i + dy^i)$. Since $(y^i,x^i)$ is feasible in P, $v^P(y^i) \leq cx^i + dy^i$. Therefore, $v^{C(P)}(y^*) \geq \Sigma\, \lambda_i v^P(y^i)$. But since P and Q are equivalent formulations $v^{C(P)}(y^*) \geq \Sigma\, \lambda_i v^Q(y^i)$ and by convexity of $v^Q(y)$, the right-hand side of this last expression is no smaller than $v^Q(y^*)$. Consequently, $v^{C(P)}(y^*) \geq v^Q(y^*)$ for all $y^* \in Y^c$. □

Combining this theorem with Theorem 2 establishes

Corollary 2: *Given any two equivalent formulations* P *and* Q *of a mixed integer program, the convex hull formulation* C(P) *of* P *is either superior or identical to* Q.

Another interesting property of the convex hull formulation of a problem is that when Benders algorithm is applied to it, only one cut is necessary for it to converge. More formally, let us suppose that the constraints of the following problem define the convex hull of the mixed integer program P in (2):

$$\begin{aligned} v^{C(P)} = \min \quad & cx + dy \\ \text{subject to:} \quad & Rx + Qy = q \\ & x \geq 0, y \in Y\,. \end{aligned}$$

Then we have

Theorem 4: *For any formulation of a mixed integer program, the convex hull formulation* C(P) *requires only one Benders cut for convergence.*

Proof:
$$v^{C(P)} = \min_{y \in Y} \min_{x \geq 0} \; cx + dy$$
$$\text{subject to:} \quad Rx + Qy = q\,.$$

Since C(P) is the convex hull formulation, we can substitute $Y^c$ for Y without affecting the optimal solution value. Then applying linear programming duality theory (and again assuming that $v^{C(P)}(y)$ is feasible for all $y \in Y$) we have $v^{C(P)} = \min_{y \in Y^c} \max_{uR \leq c} u(q - Qy) + dy$. Another application of linear programming duality theory yields $v^{C(P)} = \min_{y \in Y^c} u^*(q - Qy) + dy$ for some $u^*$ satisfying $uR \leq c$. Since the last objective function is a linear function of y, we can substitute Y for its convex hull and write:

$$v^{C(P)} = \min \quad z$$
$$\text{subject to:} \quad z \geq u^*(q - Qy) + dy$$
$$y \in Y .$$

Let $y^*$ be a solution of this problem. Then $v^{C(P)} = v^{C(P)}(y^*)$. So the single Benders cut generated by $u^*$ is sufficient to solve the convex hull formulation C(P). □

Although we have shown that a reduced feasible region for the relaxed primal problem of a formulation is desirable, there are other issues that must be considered in selecting a model for use with Benders decomposition.

First, there remains the difficulty of constructing alternative models for mixed integer programming problems. Although, in principle, the convex hull formulation of a problem requires only a single Benders cut for convergence, in general, it will be very difficult to determine this cut by building such a model. There is no efficient procedure known for generating the constraints representing the convex hull of a set of points. Padberg and Hong [32] have recently had success generating such constraints iteratively for traveling salesman problems. Finding efficient methods for generating alternative models appears to be an area for future research.

The formulation of network optimization problems is potentially one rich application area for the results of this section. Network problems usually have several evident "natural" formulations. The facility location problem discussed in section 3, the multicommodity distribution system problem solved by Geoffrion and Graves [16], and the capacitated plant location problem described by Guignard and Spielberg [18], are all network examples that have several easily derived formulations. For these problems, since the alternative formulations usually have the same problem variables, we can compare them by inspecting the size of the feasible region for their respective relaxed primal problems. Due to the comparatively simple constraint sets of network problem formulations, it may also be possible to derive additional constraints from the current ones. In such a situation, these new constraints could be evaluated by testing if they reduce the size of the feasible region for the modified primal problem.

Another issue that should be considered is the difficulty of solving the Benders (linear programming) subproblems. Adding constraints to a formulation strengthens the Benders cuts that can be derived, but also complicates the solution of the linear subproblems. So there is a trade-off between the quality of Benders cuts available and the time needed to solve the Benders subproblems.

Finally, a related issue is that adding constraints to a formulation can cause the linear programming subproblems to become degenerate since we are adding constraints to a linear program while keeping the number of variables constant. Thus, there may be a choice as to which cut to generate at each iteration of Benders algorithm.

So by "tightening" the formulation of a problem we can get stronger Benders cuts, but these stronger cuts may have to be distinguished from other weaker cuts. The methodology described in section 2 should be useful in such a situation.

## 5. CONCLUSIONS AND FUTURE WORK

Recent computational successes have underscored the promise of Benders decomposition. However, the straightforward application of this solution strategy frequently leads to computational excesses. This paper has described new results and methodology for accelerating the convergence of Benders decomposition. This approach is also applicable to a broader class of relaxation algorithms for minimax problems such as Dantzig-Wolfe decomposition for the Lagrangian dual of a nonlinear problem. The adaptation of this technique in section 3 to Benders decomposition applied to facility location models yielded an efficient special purpose algorithm. Computational work applying these techniques to facility location and network design problems is currently in progress. We hope to report these results in a future paper.

Section 4 discussed the relationship between the proper mathematical formulation of mixed integer programming models such as the facility location problem and the computational performance of Benders decomposition. We presented a criterion for selecting among alternate model formulations for use with Benders decomposition. Suggestions were also made for modifying model formulations in order to improve the performance of Benders procedure. A potentially fruitful avenue for future research would be to construct new formulations for mixed integer programs based upon our results and to perform computational tests evaluating our criteria for selecting among alternative model formulations.

REFERENCES

1. Balinski, M. L., "Integer Programming: Methods, Uses, Computation," *Man. Sci.*, Vol. 12, pp. 253-313, 1965.

2. Beale, E. M. L. and Tomlin, J. A., "An Integer Programming Approach to a Class of Combinatorial Problems," *Math. Prog.*, Vol. 3, pp. 339-344, 1972.

3. Benders, J. F., "Partitioning Procedures for Solving Mixed Variables Programming Problems," *Num. Math.*, Vol. 4, pp. 238-252, 1962.

4. Cornuejols, G., Fisher, M. L., and Nemhauser, G. L., "Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms," *Man. Sci.*, Vol. 23, pp. 789-810, 1977.

5. Dantzig, G. B., *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 1973.

6. Davis, P. S. and Ray, T. L., "A Branch-Bound Algorithm for Capacitated Facilities Location Problems," *Nav. Res. Log. Q.*, Vol. 16, pp. 331-344, 1969.

7. Dem'yanov, V. F. and Malozemov, V. N., *Introduction to Minimax*, (Translated from Russian by D. Louvish), John Wiley & Sons, New York, 1974.

8. Erlenkotter, D., "A Dual Based Procedure for Uncapacitated Facility Location," *Opns. Res.*, Vol. 26, pp. 992-1009, 1978.

9. Fisher, M. L. and Jaikumar, R., "A Decomposition Algorithm for Large Scale Vehicle Routing," Report No. 78-11-05, Department of Decision Sciences, University of Pennsylvania, 1978.

10. Fisher, M. L. and Shapiro, J. F., "Constructive Duality in Integer Programming," *SIAM Journal on Applied Math*, Vol. 27, pp. 31-52, 1974.

11. Florian, M. G., Guerin, G., and Bushel, G., "The Engine Scheduling Problem in a Railway Network," *INFOR Journal*, Vol. 14, pp. 121-138, 1976.

12. Garfinkel, R. and Nemhauser, G., *Integer Programming*, Wiley, New York, 1972.

13. Geoffrion, A. M., "Elements of Large Scale Mathematical Programming, Parts I and II," *Man. Sci.*, Vol. 16, pp. 652-691, 1970.

14. Geoffrion, A. M., "Primal Resource-Directive Approaches for Optimizing Nonlinear Decomposable Systems," *Opns. Res.* Vol. 18, pp. 375-403, 1970.

15. Geoffrion, A. M., "Lagrangian Relaxation and Its Uses in Integer Programming," *Math. Prog. Study 2*, pp. 82-114, 1974.

16. Geoffrion, A. M. and Graves, G., "Multicommodity Distribution System Design by Benders Decomposition," *Man. Sci.*, Vol. 5, pp. 822-844, 1974.

17. Geoffrion, A. M. and McBride, R., "Lagrangian Relaxation Applied to Facility Location Problems," *AIIE Transactions*, Vol. 10, pp. 40-47, 1979.

18. Guignard, M. and Spielberg, K., "Search Techniques With Adaptive Features for Certain Mixed Integer Programming Problems," *Proceedings IFIPS Congress*, Edinburgh, 1968.

19. Holloway, C., "A Generalized Approach to Dantzig-Wolfe Decomposition for Concave Programs," *Opns. Res.*, Vol. 21, pp. 210-220, 1973.

20. Lasdon, L. S., *Optimization Theory for Large Systems*, The Macmillan Company, New York, 1970.

21. Lemarechal, C., "An Extension of Davidon's Methods to Non-Differentiable Problems," *Math. Prog. Study 3*, pp. 95-109, 1975.

22. Magnanti, T. L., "Optimization for Sparse Systems," in *Sparse Matrix Computations*, (J. R. Bunch and D. J. Rose, eds.), pp. 147-179, Academic Press, New York, 1976.

23. Magnanti, T. L., Shapiro, J. F., and Wagner, M. W., "Generalized Linear Programming Solves the Dual," *Man. Sci.*, Vol. 22, pp. 1194-1203, 1976.

24. Marsten, R. E., "The Use of the BOXSTEP Method in Discrete Optimization," *Math. Prog. Study 3*, pp. 127-144, 1975.

25. Marsten, R. E., Hogan, W. W., and Blankenship, J. W., "The BOXSTEP Method for Large-Scale Optimization," *Opns. Res.*, Vol. 23, pp. 389-405, 1975.

26. Mevert, P., "Fixed Charge Network Flow Problems: Applications and Methods of Solution." Presented at Large Scale and Hierarchical Systems Workshop, Brussels, May, 1977.

27. Mifflin, R., "An Algorithm for Constrained Optimization with Semi-smooth Functions," International Inst. for Appl. Sys. Analysis, Laxenberg, Austria, 1977.

28. Nemhauser, G. L. and Widhelm, W. B., "A Modified Linear Program for Columnar Methods in Mathematical Programming," *Opns. Res.*, Vol. 19, pp. 1051-1060, 1971.

29. Nemhauser, G. L. and Wolsey, L. A., "Maximizing Submodular Set Functions: Formulations and Analysis of Algorithms," Technical Report No. 398, School of Operations Research and Industrial Engineering, Cornell University, 1978.

30. O'Neill, R. P., and Widhelm, W. B., "Computational Experience with Normed and Nonnormed Column-Generation Procedures in Nonlinear Programming," *Opns. Res.*, Vol. 23, pp. 372-382, 1972.

31. Orchard-Hays, W., *Advanced Linear Programming Computing Techniques*, McGraw-Hill, New York, 1968.

32. Padberg, M. and Hong, S., *On the Symmetric Traveling Salesman Problem: A Computational Study*, Report #77-89, New York University, 1977.

33. Richardson, R., "An Optimization Approach to Routing Aircraft," *Trans. Sci.*, Vol. 10, pp. 52-71, 1976.

34. Rockafellar, R. T., *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

35. Williams, H. P., "Experiments in the Formulation of Integer Programming Problems," *Math. Prog. Study 2*, pp. 180-197, 1974.

36. Wolfe, P., "Convergence Theory in Nonlinear Programming," in (J. Abadie, ed.), *Integer and Nonlinear Programming*, North Holland, Amsterdam, 1970.

37. Wolfe, P., "A method of Conjugate Subgradients for Minimizing Non-differentiable Functions," *Math Prog. Study 3*, pp. 145-173, 1975.

38. Wong, R., "Accelerating Benders Decomposition For Network Design," Ph.D. Thesis, Department of Elec. Eng. and Comp. Sci., Mass. Inst. of Tech., January, 1978.