OPERATIONS RESEARCH CENTER

working paper

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

Decomposition Methods for Facility Location Problems+

by

Thomas L. Magnanti*

and

Richard T. Wong**

OR 153-86 December 1986

*Thomas L. Magnanti is George Eastman Professor of Management Science, Sloan School of Management, MIT. Research on this paper was supported in part by Grants 79-26225-ECS and 83-16224-ECS from the National Science Foundation's Program on Systems Theory and Operations Research.

**Richard T. Wong is Associate Professor of Management, Krannert Graduate School of Management, Purdue University. Research on this paper was supported in part by Grants 81-17876-ECS and 83-16224-ECS from the National Science Foundation's Program on Systems Theory and Operations Research.

+This paper has been prepared as Chapter 5 for the forthcoming text, Discrete Location Theory, edited by R.L. Francis and P.B. Mirchandani.

5.1 INTRODUCTION

Discrete facility location problems pose many challenging optimization issues. In size alone, they can be difficult to solve: models of facility location applications often require thousands of variables and thousands of constraints. Moreover, the models are complicated: the basic yes-no decisions of whether or not to select candidate sites for facilities endow the models with a complex combinatorial structure. Even with as few as 30 candidate sites, there may be more than one billion potential combinations for the facility locations.

In treating large-scale problems with these complications, mathematical programming, like most disciplines, has relied heavily upon several key concepts. Throughout the years, the closely related notions of bounding techniques, duality, and decomposition have been central to many of the advances in mathematical programming. The optimization problems encountered in facility location have been no exception. Indeed, location problems have served as a fertile testing ground for many ideas from mathematical programming, and advances in location theory have stimulated more general developments in optimization.

In particular, since discrete facility location problems contain two types of inherently different decisions--where to locate facilities and how best to allocate demands to the resulting facilities--the problem class is an attractive candidate for decomposition. Once the discrete-choice facility location decisions have been made, the (continuous) allocation problem typically becomes much simpler to solve. Can we exploit this fact in designing algorithms? If not, decomposition may still be attractive. Even if location problems were not complicated by the discrete-choice site selection decisions and were to be formulated as simpler linear programs (e.g., by relaxing the integrality restrictions on the problem variables), they still would be very large and difficult to solve. Fortunately, though, the problems have a special structure that various decomposition techniques can exploit.

This chapter describes the use of decomposition as a solution procedure for solving facility location problems. It begins by introducing two basic decomposition strategies that are applicable to location problems. It then presents a more formal discussion of decomposition and examines methods for improving the performance of the decomposition algorithms, both in general and in the context of facility location models. This discussion emphasizes recent advances that have led to new insights about decomposition methods and that appear promising for future developments. It also stresses the relationships between bounding techniques, decomposition, and duality. Finally, the chapter discusses the importance of problem formulation and its effect upon the performance of decomposition methods. Most facility location problems can be stated as mixed integer programs in a variety of ways; choosing a "good" formulation from the available alternatives can have a pronounced effect upon the performance of an algorithm.

Most of the chapter, and particularly Sections 5.1, 5.2, 5.6, and 5.7, should be accessible to nonspecialists and requires only a general background in linear and integer programming. Sections 5.3-5.5 discuss more advanced material and contain some new results and interpretations that might be of interest to specialists as well. A good knowledge of linear programming duality is useful for following the proofs in Sections 5.4 and 5.5.

Problem Formulations

Throughout our discussion, we assume that facilities are to be located at the nodes of a given network $G = (V, A)$ with node (vertex) set V and arc set A. We let $[i,j]$ denote the arc connecting nodes v_i and v_j.

We assume that if a facility is located at node $v_j \varepsilon V$, then the node has a demand (output) capacity of K_j units. If no facility is assigned to node $v_j \varepsilon V$, then the node cannot accommodate any demand.

For decision variables, we let

$$x_j = \begin{cases} 0 \text{ if a facility is not assigned to node } v_j \\ 1 \text{ if a facility is assigned to node } v_j \end{cases}$$

and let

$$y_{ij} = \text{ the flow on arc } [i,j] \varepsilon A.$$

Let $\mathbf{x} = (x_1, \ldots, x_n)$ denote the vector of location variables and $\mathbf{y} = (y_{ij})$ denote the vector of flow variables. For notational convenience, we will say that facility v_j is <u>open</u> if $x_j=1$ and that it is <u>closed</u> if $x_j=0$. In general, the set $\{v_1, v_2, \ldots, v_n\}$ of potential facility locations might be a subset of the nodes V. We let A_j denote the subset of arcs directed into the potential facility location v_j.

We assume that the location problem has been formulated as the following mixed integer program:

$$\text{minimize } \mathbf{cx} + \mathbf{dy} \tag{5.1.1}$$

$$\text{subject to } \quad \mathbf{Ny} = \mathbf{w} \tag{5.1.2}$$

$$\mathbf{y} \geq 0 \tag{5.1.3}$$

$$\sum_{[i,j] \varepsilon A_j} y_{ij} \leq K_j x_j \qquad j=1,\ldots,n \tag{5.1.4}$$

$$\mathbf{x}\epsilon X \tag{5.1.5}$$

$$(\mathbf{x},\mathbf{y})\epsilon S. \tag{5.1.6}$$

In this formulation, d_{ij} denotes the per unit cost of routing flow on arc $[i,j]$, c_j denotes the cost for locating a facility at node v_j, N is a node-arc incidence matrix for the network G, and w_i denotes the net demand (weight) at node v_i. Therefore, equation (5.1.2) is the customary mass balance equation from network flows.

The inequalities (5.1.4) state that the total output† from node v_j cannot exceed the node's capacity K_j if a facility is assigned to that node (i.e., $x_j = 1$), and that the node can have no output if a facility is not assigned to it (i.e., $x_j = 0$). The set X contains any restrictions imposed upon the location decisions, including the binary restriction $x_j = 0$ or 1. For example, the set might include multiple choice constraints of the form $x_1 + x_2 + x_3 \leq 2$ that state that at most two facilities can be assigned to nodes v_1, v_2 and v_3. It could also contain precedence constraints of the form $x_1 \leq x_2$, stating that a facility can be assigned to node v_1 (i.e., $x_1 = 1$) only if a facility is assigned to node v_2.

Finally, the set S contains any additional side restrictions imposed upon the allocation variables, or imposed jointly upon the location and allocation variables. For example, it may contain "bundle" constraints of the form $y_{ij} + y_{hk} + y_{rs} \leq u$ that limit the total flow on three separate arcs $[i,j]$, $[h,k]$ and $[r,s]$ or of the form $y_{ij} + y_{hk} + y_{rs} = y_{pq}$ that relate the flow on several arcs. The last equation can be used to model multicommodity flow versions of the problem without the need for any additional notational complexity. In this case, we could view the arcs $[i,j]$, $[h,k]$,

† The flow y_{ij} <u>into</u> node v_j represents the amount of service that node v_i is receiving <u>from</u> node v_j. Therefore, it seems natural to refer to this flow as an <u>output</u> (of service) from node v_j.

[r,s] and [p,q] as having been extracted from four separate copies of the same underlying network (i.e., N is block diagonal with four independent copies of the same node-arc incident matrix). The first three of these networks model different commodities and the fourth models total flow by all commodities. Similar types of specifications for the side constraints or for the topology of the underlying network would permit the formulation to model a wide variety of other potential problem characteristics, such as the distribution of goods through a multi-echelon system of warehouses.

The following special case of this general model has received a great deal of attention in the facility location literature:

$$\text{minimize} \quad \mathbf{cx} + \mathbf{dy} \tag{5.1.7}$$

$$\text{subject to} \quad \sum_{j=1}^{n} y_{ij} = 1 \qquad i=1,\ldots,m \tag{5.1.8}$$

$$y_{ij} \leq x_j \qquad i=1,\ldots,m;\ j=1,\ldots,n \tag{5.1.9}$$

$$\sum_{j=1}^{n} x_j = p \tag{5.1.10}$$

$$y_{ij} \geq 0 \qquad i = 1,\ldots,m;\ j = 1,\ldots,n \tag{5.1.11}$$

$$x_j = 0 \text{ or } 1 \qquad \text{all } j = 1,\ldots,n. \tag{5.1.12}$$

In this model, y_{ij} denotes the fraction of customer demand at node v_i that receive service from a facility at node v_j. The "forcing" constraints (5.1.9), which we could have written as $\sum_i y_{ij} \leq m\, x_j$ to conform with the earlier formulation (5.1.4), restricts the flow to only those nodes v_i that have been chosen as facility sites (i.e., have $x_j = 1$). Finally, constraint (5.1.10) restricts the number of facilities to a prescribed number p. In this formulation, the set of customer locations v_i could be distinct from the set of potential facility locations v_j. Or, both sets of locations might correspond to the same node set V of an underlying graph G = (V, A). In the model (5.1.7)-(5.1.12), which is usually referred to as the p-median problem, d_{ij} denotes the cost of servicing demand from node v_i by a facility at node v_j (see Chapter 2). Throughout our discussion, when referring to the p-median problem, we will assume, as is customary, that each $c_j = 0$. That is, we do not consider the cost of establishing facilities, but merely limit their number.

Although much of our discussion in this chapter applies to the general formulation (5.1.1)-(5.1.6), or can be extended to apply to this model, for ease of presentation, we usually consider the more specialized model (5.1.7)-(5.1.12).

Chapter Summary

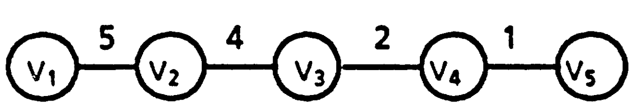
The remainder of the chapter is structured as follows. The next section introduces two forms of decomposition for facility location problems--Benders' (or resource directive) decomposition and Lagrangian relaxation (or price directive decomposition). Section 5.3 describes these decomposition approaches in more detail and casts them in a more general and unifying framework of minimax optimization. The section also describes methods for improving the performance of these algorithms. Section 5.4 specializes one of these improvements to Benders' decomposition as applied to facility location problems. Section 5.5 discusses the important role of model formulation in applying decomposition to facility location problems. This section also focuses on Benders' decomposition (see Chapters 2 and 3 for related discussions of Lagrangian relaxation). Section 5.6 describes computational experience in applying Benders' decomposition to facility location and related transportation problems. Finally, Section 5.7 contains concluding remarks and cites references to the literature.

5.2. INTRODUCTION TO DECOMPOSITION

This section discusses two different decomposition strategies for obtaining lower bounds on the optimal objective function value of location problems. To introduce these concepts, we focus on the p-median location problem introduced in Chapter 2 and reformulated in Section 5.1. The next section derives these bounding techniques more generally for the entire class of location problems (5.1.1)-(5.1.6).

Resource Directive (Benders') Decomposition

Consider the five-node, two-median example of Figure 5.1. In Figure 5.1(a), the arc labels indicate the cost of traversing a particular link; assume that each node has a unit demand. The entries in the transportation cost matrix in Figure 5.1(b) specify costs d_{ij} of servicing the demand at node v_i from a facility located at node v_j. Suppose we have a current



(a)

$$[d_{ij}] = \begin{bmatrix} 0 & 5 & 9 & 11 & 12 \\ 5 & 0 & 4 & 6 & 7 \\ 9 & 4 & 0 & 2 & 3 \\ 11 & 6 & 2 & 0 & 1 \\ 12 & 7 & 3 & 1 & 0 \end{bmatrix}$$

(b)

Figure 5.1 A Two-Median Example:
(a) the underlying transportation network,
(b) the distance matrix

configuration with facilities located at nodes v_2 and v_5 (i.e., $x_2 = x_5 = 1$). The objective function cost for this configuration is 5 + 0 + 3 + 1 + 0 = 9.

Relative to the current solution, let us evaluate the reduction in the objective function cost if facility 1 (i.e., the facility at node v_1) is opened and all other facilities retain their current, open vs. closed, status. This new facility would reduce the cost of servicing the demands at node v_1 from $d_{12} = 5$ to $d_{11} = 0$. Therefore, the savings for opening facility 1 is 5 units. Similarly, by opening facility 3 we would reduce (relative to the current solution) node v_3 cost from $d_{34} + d_{45} = 3$ to 0 so the savings is 3 units. Opening facility 4 would reduce node v_3 cost from 3 to 2 and node v_4 cost from 1 to 0, for a savings of 1 + 1 = 2. Since facilities 2 and 5 are already open in the current solution, the savings for opening any of them is zero.

Note that when these savings are combined, the individual assessments might overestimate possible total savings since the computation might double count the cost reductions for any particular node. For example, our previous computations predict that opening both facilities 3 and 4 would reduce the node v_3 cost and give a total reduction of 3+1 = 4 units even though the maximum possible reduction is clearly 3 units which is the cost of servicing node v_3 in the current solution.

With this savings information, we can bound the cost z of any feasible configuration **x** from below by

$$z \geq B_1(\mathbf{x}) = 9 - 5x_1 - 3x_3 - 2x_4. \qquad (5.2.1)$$

Notice that specifying a <u>different</u> current configuration would change our savings computations and permit us to obtain a different lower bound function. For example, the configuration $x_1 = x_3 = 1$ would produce a lower bound inequality

$$z \geq B_2(\mathbf{x}) = 9 - 4x_2 - 4x_4 - 4x_5 \tag{5.2.2}$$

on the objective function value z of any feasible solution, and thus the optimal solution of the problem.

Since each of these two bounding functions is always valid, by combining them we obtain an improved lower bound for the optimal two-median cost. Solving the following mixed integer program would determine the best location of the facilities that uses the combined lower bounding information:

$$\begin{aligned} &\text{minimize} && z \\ &\text{subject to} && z \geq B_1(\mathbf{x}) = 9 - 5x_1 - 3x_3 - 2x_4 \\ &&& z \geq B_2(\mathbf{x}) = 9 - 4x_2 - 4x_4 - 4x_5 \\ &&& x_1 + x_2 + x_3 + x_4 + x_5 = 2 \\ &&& x_j = 0 \text{ or } 1 \qquad j = 1, \ldots, 5 \end{aligned} \tag{5.2.3}$$

which yields a lower bound of $z^* = 5$ obtained by setting $x_1^* = x_2^* = 1$ and $x_3^* = x_4^* = x_5^* = 0$ (or by setting $x_1^* = x_4^* = 1$, or $x_1^* = x_5^* = 1$, or $x_3^* = x_4^* = 1$ and all other $x_j^* = 0$ in each case).

This bounding procedure is the essential ingredient of Benders' decomposition. In this context, (5.2.3) is referred to as a Benders' master problem and (5.2.1) and (5.2.2) are called Benders' cuts or inequalities. When applied to mixed integer programs with integer variables $\mathbf{x}$ and continuous variables $\mathbf{y}$, Benders' decomposition repeatedly solves a master problem like (5.2.3) in the integer variables $\mathbf{x}$; at each step, the algorithm uses the simple savings computation to refine the lower bound information by adding a new Benders' cut to the master problem. Each solution $(z^*, \mathbf{x}^*)$ to the master problem yields a new lower bound z^* and a new configuration $\mathbf{x}^*$. For p-median problems, with the facility locations fixed at $\mathbf{x}=\mathbf{x}^*$, the resulting allocation problem becomes a trivial linear program (assign all demand at node v_i to the closest open facility - i.e., minimize d_{ij} over all j with $x_j^* = 1$). The optimal solution $\mathbf{y}^*$ to this linear program generates a new feasible median solution $(\mathbf{x}^*, \mathbf{y}^*)$. The cost $\mathbf{d}\mathbf{y}^*$ of this solution is an upper bound on the optimal objective function value of the p-median problem. As we will see in Section 5.4, the savings from any current configuration $\mathbf{x}^*$ can be viewed as dual variables for this linear program. Therefore, in general, the solution of a linear program (and its dual) would replace the simple savings computation.

The method terminates when the current lower bound z^* equals the cost of the best (least cost) configuration $\hat{\mathbf{x}}$ found so far. This equality implies that the best upper bound equals the best lower bound and so $\hat{\mathbf{x}}$ must be an optimal configuration.

Since Benders' decomposition generates a series of feasible solutions to the original median problem, it may be viewed as a primal method that utilizes dual information. We next discuss a dual method.

Price-Directive (Lagrangian) Decomposition

Lagrangian relaxation offers another type of decomposition technique that produces lower bounds. Consider a mixed integer programming formulation of the example of Figure 5.1:

$$\text{minimize} \quad \sum_{i=1}^{5} \sum_{j=1}^{5} d_{ij}\, y_{ij} \tag{5.2.4}$$

$$\text{subject to} \quad \sum_{j=1}^{5} y_{ij} = 1 \qquad i=1,\ldots, 5 \tag{5.2.5}$$

$$y_{ij} \le x_j \qquad i=1,\ldots,5;\ j=1,\ldots,5 \tag{5.2.6}$$

$$\sum_{j=1}^{5} x_j = 2 \tag{5.2.7}$$

$$y_{ij} \geq 0,\ x_j = 0 \text{ or } 1 \quad i=1,\ldots,5;\ j=1,\ldots,5. \tag{5.2.8}$$

Larger versions of problem (5.2.4)-(5.2.8) are too complicated to solve directly, since with 100 instead of 5 nodes the problem would contain 10,000 variables y_{ij} and 10,000 constraints of the form (5.2.6).

As an algorithmic strategy for simplifying the problem, suppose that we remove the constraints (5.2.5), weighting them by Lagrange multipliers (dual variables) λ_i, and placing them in the objective function to obtain the Lagrangian subproblem:

$$L(\lambda) = \min \quad \sum_{i=1}^{5} \sum_{j=1}^{5} d_{ij}\, y_{ij} + \sum_{i=1}^{5} \lambda_i (1 - \sum_{j=1}^{5} y_{ij}) \tag{5.2.9}$$

subject to (5.2.6)-(5.2.8).

Each "penalty" term $\lambda_i\ (1 - \sum_{j=1}^{5} y_{ij})$ will be positive if λ_i has the appropriate sign and the i^{th} constraint of (5.2.5) is violated. Therefore, by adjusting the penalty values λ_i, we can "discourage" the subproblem (5.2.9) from having an optimal solution that violates (5.2.5).

Note that since the penalty term is always zero for all λ whenever y satisfies (5.2.5), the optimal subproblem cost $L(\lambda)$ is always a valid lower bound for the optimal p-median cost.

The primary motivation for adopting this algorithmic strategy is that problem (5.2.9) is very easy to solve. Set $y_{ij} = 1$ only when $x_j = 1$ and the modified cost coefficient $(d_{ij} - \lambda_i)$ of y_{ij} is nonpositive. Thus, summed over all nodes v_i, the optimal benefit of setting $x_j = 1$ is

$$r_j = \sum_{i=1}^{5} \min\,(0,\ d_{ij} - \lambda_i)$$

and we can rewrite (5.2.9) as

$$L(\lambda) = \min \sum_{j=1}^{5} r_j x_j + \sum_{i=1}^{5} \lambda_i \qquad (5.2.10)$$

subject to (5.2.7) and (5.2.8).

This problem is solved simply by finding the two smallest r_j values and setting the corresponding variables $x_j = 1$.

For example, let $\bar{\lambda} = (3, 3, 3, 3, 3)$. Then (5.2.10) becomes

$$L(\bar{\lambda}) = \min -3x_1 - 3x_2 - 4x_3 - 6x_4 - 5x_5 + 15$$

subject to (5.2.7) and (5.2.8).

The corresponding optimal solution for (5.2.9) has a Lagrangian objective value $L(\bar{g}) = 15 - 6 - 5 = 4$; the solution has $x_4 = x_5 = 1$, $y_{34} = y_{44} = y_{45} = y_{54} = y_{55} = 1$, and all other variables set to zero. Notice that this solution for the Lagrangian subproblem is not feasible for the p-median problem since with $i = 1$ or 2 it does not satisfy the demand constraint (5.2.5).

For another dual variable vector $\lambda^* = (5, 5, 3, 2, 3)$, (5.2.10) becomes

$$L(\lambda^*) = \min -5x_1 - 5x_2 - 4x_3 - 5x_4 - 4x_5 + 18$$

subject to (5.2.7) and (5.2.8).

Its optimal objective function value $L(\lambda^*) = 8$ is a tight lower bound since the optimal p-median cost is also 8.

This example illustrates the importance of using "good" values for the dual variables λ_i in order to obtain strong lower bounds from the

Lagrangian subproblem. In fact, to find the sharpest possible Lagrangian lower bound, we need to solve the optimization problem

$$\max_{\lambda} L(\lambda).$$

This optimization problem in the variables λ has become known as a Lagrangian dual problem to the original facility location model. (See Chapter 2 for further discussion of the use of the Lagrangian dual for the p-median problem, and Chapter 3 for applications to the uncapacitated facility location problem.)

5.3 DECOMPOSITION METHODS AND MINIMAX OPTIMIZATION

The previous section introduced two of the most widely used strategies for solving large-scale optimization problems. Lagrangian relaxation, or price directive decomposition, simplifies problems by relaxing a set of complicating constraints. Resource directed decomposition, which includes Benders' method as a special case, decomposes problems by projecting (temporarily holding constant) a set of strategic resource variables.

We noted how the techniques can be applied directly to location problems. In addition, they can be combined with other solution methods; for example, Lagrangian relaxation, rather than a linear programming relaxation, can be embedded within the framework of a branch-and-bound approach for solving location and other discrete optimization problems.

In this section, we study these two basic decomposition techniques by considering a broader, but somewhat more abstract, minimax setting that captures the essence of both the resource directive and Lagrangian relaxation approaches. That is, we consider the optimization problem

$$v = \min_{u \in U} \max_{s \in S} \{f(s) + ug(s)\} \tag{5.3.1}$$

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

To relate this minimax setting to Benders' decomposition applied to the facility location problem (5.1.7)-(5.1.12), we can argue as follows. Let $X = \{x: \sum_{j=1}^{n} x_j = p \text{ and } x_j = 0 \text{ or } 1 \text{ for } j=1,\ldots,n\}$. An equivalent form of formulation (5.1.7)-(5.1.12) is

$$\underset{\mathbf{x}\in X}{\text{minimize}} \;\; \underset{\mathbf{y}\geq 0}{\text{minimize}} \{\mathbf{cx} + \mathbf{dy}: (5.1.8),\ (5.1.9) \text{ and } (5.1.11) \text{ are satisfied}\}. \quad (5.3.2)$$

For any fixed value of the configuration vector $\mathbf{x}$, the inner minimization is a simple network flow linear program. Assume the network flow problem is feasible and has an optimal solution for all $\mathbf{x}\in X$[†]; then dualizing the inner minimization problem over $\mathbf{y}$ gives the equivalent formulation

$$\underset{\mathbf{x}\in X}{\text{minimize}} \;\; \underset{(\lambda,\pi)\in\Lambda\Pi}{\text{maximize}} \; \{\sum_{i=1}^{m} \lambda_i - \sum_{j=1}^{n} (\sum_{i=1}^{m} \pi_{ij}) x_j + \mathbf{cx}\} \quad (5.3.3)$$

where $\Lambda\Pi = \{(\lambda,\pi): \lambda\in R^m,\ \pi\in R^{mn},\ \lambda_i - \pi_{ij} \leq d_{ij} \text{ for all } i,j \text{ and } \pi \geq 0\}$.

Observe that this problem is a special case of (5.3.1) with (λ,π) and $\mathbf{x}$ identified with $\mathbf{s}$ and $\mathbf{u}$, respectively. This reformulation is typical of the resource directive philosophy of solving the problem parametrically--in terms of complicating variables like the configuration variables $\mathbf{x}$.

Dualizing (5.1.8) in the location model (5.1.7)-(5.1.12) gives a maximin form of the problem. The resulting Lagrangian dual problem is

$$\underset{\lambda}{\text{maximize}} \;\; \underset{(\mathbf{x},\mathbf{y})\in XY}{\text{minimize}} \; \{\mathbf{cx} + \mathbf{dy} + \sum_{i=1}^{m} \lambda_i (1 - \sum_{j=1}^{n} y_{ij})\} \quad (5.3.4)$$

or, equivalently,

$$\underset{\lambda}{\text{maximize}} \;\; \underset{(\mathbf{x},\mathbf{y})\in XY}{\text{minimize}} \; \{\mathbf{cx} + \sum_{i=1}^{m} \lambda_i + \sum_{i=1}^{m} \sum_{j=1}^{n} (d_{ij} - \lambda_i)\, y_{ij}\} \quad (5.3.5)$$

where $XY = \{(\mathbf{x},\mathbf{y}): \mathbf{x}\in X,\ \mathbf{y} \geq 0 \text{ and } (5.1.9)\text{-}(5.1.12) \text{ are satisfied}\}$ and λ and $(\mathbf{x},\mathbf{y})$ correspond to $\mathbf{u}$ and $\mathbf{s}$ in (5.3.1).

Note that duality plays an important role in both the minimax formulation (5.3.3) and Lagrangian maximin formulation (5.3.5). Benders' decomposition uses duality to convert the inner minimization in (5.3.2) into a maximization

---

† These assumptions can be relaxed quite easily, but with added complications that cloud our main development.

problem and (5.3.5) is just a slightly altered restatement of the Lagrangian dual problem (5.3.4).

5.3.1 Solving Minimax Problems by Relaxation

For any $u \in U$, let $v(u)$ denote the value of the maximization problem in (5.3.1); that is

$$v(u) = \max_{s \in S} \{f(s) + ug(s)\}. \tag{5.3.6}$$

We refer to this problem, for a fixed value of u, as a <u>subproblem</u>. Note that

$$v = \min_{u \in U} v(u).$$

To introduce a "relaxation" strategy for solving this problem, let us rewrite (5.3.1) as follows:

$$\begin{array}{lll} \text{minimize} & z & \\ \text{subject to} & z \geq f(s) + ug(s) & \text{for all } s \in S \\ & u \in U,\ z \in R. & \end{array} \tag{5.3.7}$$

Observe that this problem has a constraint for each point $s \in S$. Since S may be very large, and possibly even infinite, the problem (5.3.7) often has too many constraints to solve directly. Therefore, let us form the following relaxation of this problem:

$$\begin{array}{lll} \text{minimize} & z & \\ \text{subject to} & z \geq f(s^k) + ug(s^k) & k = 1,2,\ldots,K \\ & u \in U,\ z \in R & \end{array} \tag{5.3.8}$$

which is obtained by restricting the inequalities on z to a finite subset $\{s^1, s^2, \ldots, s^K\}$ of elements s^k from the set S. The solution (u^K, z^K) of this <u>master problem</u> (5.3.8) is optimal for (5.3.7) if it satisfies all of the constraints of that problem, that is, if $v(u^K) \leq z^K$. If, on the other hand, $v(u^K) > z^K$ and s^{K+1} solves† the subproblem (5.3.6) when $u = u^K$, then we add

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

$$z \geq f(s^{K+1}) + ug(s^{K+1})$$

as a new constraint or, as it is usually referred to, a new "cut" to the master problem (5.3.8). The algorithm continues in this way, alternately solving the master problem and subproblem.

In Section 5.3.2, we give a numerical example that illustrates both the algorithm and the conversion of mixed integer programs into the minimax form (5.3.7).

Maximin problems like the Lagrangian dual problem (5.3.5) can be treated quite similarly. Restating the relaxation algorithm for these problems requires only minor, and rather obvious, modifications.

When applied to problem (5.3.3) this relaxation algorithm is known as Benders' decomposition and when applied to (5.3.5), it is known as generalized programming or Dantzig-Wolfe decomposition. For Benders' decomposition, the master problem is an integer program with one continuous variable z, and the subproblem (5.3.6) is a linear program whose solution s^* can be chosen as an extreme point of S. Since S has a finite number of extreme points, Benders' algorithm will terminate after a finite number of iterations. (In the worst possible case, eventually $\{s^1, s^2, \ldots, s^K\}$ equals all extreme points of S and (5.3.8) becomes identical to (5.3.7)). For Dantzig-Wolfe decomposition applied to (5.3.5), the master problem is a linear program and, consequently, we can replace the set S=XY by the set of its extreme points (since the inner minimization problem always solves at an extreme point). Consequently, the algorithm again will terminate in a finite number of iterations and the sequence of solutions $\{\lambda^K\}_{K \geq 1}$ (i.e., the u variable for problem 5.3.1) will converge to an optimum for (5.3.5) (see Section 5.7 for comments on more general convergence properties of the algorithm).

5.3.2 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. There are several ways to improve the algorithm's performance:

(i) make a good selection of initial cuts (i.e., initial values of the s^k for the master problem);

(ii) modify the master problem to alter the choice of u^K at each step, or to exploit the information available from master problems solved in previous iterations;

(iii) reduce the number of master problems to be solved by using alternative mechanisms to generate cuts (i.e., values of the s^k),

(iv) formulate the problem "properly"; or

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

Let us briefly comment on each of these enhancements. Sections 5.5.6 and 5.5.7 cite computational studies that support many of the observations in this discussion.

(i) Initial Cuts

Various computational studies have demonstrated that the initial selection of cuts can have a profound effect upon the performance of Benders' algorithms applied to facility location and other discrete optimization problems. The initial cuts can be generated from institutional knowledge about the problem setting being studied or from heuristic methods that provide "good" choices u for the integer variables. Solving the subproblem (5.3.6) for these choices of u generates points s from S that define the initial cuts. Unfortunately, little theory is available to guide analysts in the choice of initial cuts.

(ii) Modifying the Master Problem

In the context of Dantzig-Wolfe decomposition, several researchers have investigated approaches for implementing the relaxation method more efficiently by altering the master problem. Scaling the constraints of the master problem to find the "geometrically centered" value of u^K at each step can be beneficial. Another approach is to restrict the solution to the master problem at each step to lie within a box centered about the previous solution. This procedure prevents the solution from oscillating too wildly between iterations. When there are choices, selecting judiciously among multiple optima of the master problem can also result in better convergence.

We can also modify Benders' decomposition to exploit the inherent nesting of constraints in the sequence of master problems and thus avoid solving a complete integer program at each iteration. Let $\bar{z}^k$ be the value of the best solution from the points $u^1, u^2, \ldots, u^K$ generated after K iterations, i.e., $\bar{z}^K = \min \{v(u^k): k=1,2,\ldots K\}$. Instead of solving the usual master problem (5.3.8), consider the integer programming feasibility version of that problem:

Find $u \epsilon U$ satisfying

$$(\bar{z}^K - \varepsilon) \geq f(s^k) + ug(s^k) \qquad \text{for } k = 1,2,\ldots,K \qquad (5.3.9)$$

where ε is a prespecified target reduction in objective value. If this system has a feasible solution u^{K+1}, then solve the subproblem (5.3.6) with $u = u^{K+1}$ to generate a new point s^{K+1} from S and add the associated cut to (5.3.9). Notice that the $(K+1)^{st}$ system has a feasible region that is a proper subset of the K^{th} system (since $\bar{z}^{K+1} \leq \bar{z}^{K}$ and the $(K+1)^{st}$ system has one more constraint). This property allows us to solve a sequence of these systems by incorporating information from previous iterations.

If $z = v$ and $u = u^*$ is optimal for (5.3.7), then u^* is feasible in (5.3.9) for any $\bar{z}^K \geq v + \varepsilon$. So if (5.3.9) is infeasible, then $\bar{z}^K < v + \varepsilon$ which implies that we have found a solution within ε-units of the optimal objective value. For this reason, this technique is referred to as the <u>ε-optimal method</u> for solving the Benders' master problem.

An implementation of the ε-optimal method has been very effective in solving facility location problems with a number of side constraints, that is, when the set U has a very complicated structure (see Section 5.6).

(iii) <u>Avoiding the Master Problem</u> (Cross Decomposition)

An alternative to modifying the master problem is to reduce the number of master problems that must be solved. Although our discussion applies to any mixed integer program with continuous variables $\mathbf{y}$ and integer variables $\mathbf{x}$, for concreteness, consider the location model (5.1.7)-(5.1.12). Suppose we apply the relaxation algorithm to the Lagrangian dual problem (5.3.5) and obtain a master problem solution $u^K = \lambda^K$ and a solution $s^K = (\mathbf{x}^K, \mathbf{y}^K)$ to the Lagrangian subproblem (5.3.6). Instead of solving another master problem (in this case a linear program) to generate λ^{K+1}, we could fix the network configuration at $\mathbf{x} = \mathbf{x}^K$ and solve the inner maximization of the Benders' formulation (5.3.3) to obtain $(\lambda^{K+1}, \pi^{K+1})$. Solving the subproblem (5.3.6) with $\lambda = \lambda^{K+1}$, we obtain $(\mathbf{x}^{K+1}, \mathbf{y}^{K+1})$.

These computations can be carried out very efficiently. The inner maximization of (5.3.3) is the dual of a special transportation problem (i.e., the inner minimization problem in (5.3.2)) that can, as we saw in Section 5.2, be solved by inspection. This special linear program is the subproblem that arises when Benders' decomposition is applied to the location problem (5.1.7) - (5.1.12). Since this technique combines the advantages of an easily solved Lagrangian dual subproblem (5.3.6) and an easily solved Benders' subproblem, it is sometimes referred to as cross decomposition.

By exploiting two special structures of the facility location problem, cross decomposition can compute a new dual solution λ^{K+1} and a new cut corresponding to (x^{K+1}, y^{K+1}) much more quickly than the usual Dantzig-Wolfe decomposition algorithm, which needs to solve a linear programming master problem (5.3.8) to find a new dual solution $u^{K+1} = \lambda^{K+1}$. Cross decomposition iteratively continues this process of solving a Lagrangian subproblem and a Benders' subproblem. Periodically, the method solves a linear programming master problem corresponding to the Lagrangian dual (5.3.5) in order to guarantee convergence to a dual solution.

An implementation of cross decomposition has provided the most effective method available for solving certain capacitated plant location problems (see Section 5.6).

(iv) Improving Model Formulation

Current research in integer programming has emphasized the importance of problem formulation for improving the performance of decomposition approaches and other algorithms. Two different formulations of the same problem might have identical feasible solutions, but might have different computational characteristics. 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. Since the issue seems to be so essential for ensuring computational success, in Section 5.5 we discuss in some detail the role of model formulation in the context of Benders' decomposition applied to facility location problems.

(v) <u>Choosing Good Cuts</u>

In many instances, the selection of good cuts at each iteration can significantly improve the performance of the relaxation algorithm as applied to minimax problems. For facility location models, the Benders' subproblem (5.3.6) often has multiple optimal solutions; equivalently, the dual problem is a transportation linear program which is renowned for its degeneracy.

In the remainder of this section, we introduce general methods and algorithms for choosing from the alternate optima to (5.3.6) at each iteration, a solution that defines a cut that is in some sense "best". Section 5.4 specializes this methodology to facility location models.

To illustrate the selection of good cuts to add to the master problem, consider the following example of a simple mixed integer program:

$$\begin{aligned} z = \text{minimize} \quad & y_3 + u \\ \text{subject to} \quad & -y_1 + y_3 + 2u = 4 \\ & -y_2 + y_3 + 5u = 4 \\ & y_1 \geq 0,\ y_2 \geq 0,\ y_3 \geq 0 \\ & u \geq 0 \text{ and integer.} \end{aligned}$$

The equivalent formulation (5.3.7) written as the linear programming dual of this problem for any fixed value of u is

$$\begin{array}{lll} \text{minimize} & z & \\ \text{subject to} & z \geq u & \\ & z \geq 4-u & \qquad (5.3.10) \\ & z \geq 4-4u & \\ & u \geq 0 \text{ and integer.} & \end{array}$$

The constraints correspond to the linear programming weak duality inequality $z \geq (4-2u)s_1 + (4-5u)s_2 + u$ written for the three extreme points $s^1 = (0,0)$, $s^2 = (1,0)$ and $s^3 = (0,1)$ of the dual feasible region S.

Suppose that we initiate the relaxation algorithm with the single cut $z \geq u$ in the master problem. The optimal solution is $z^1 = u^1 = 0$. At $u = u^1 = 0$, both the extreme points s^2 and s^3 (and every convex combination of them) solves the subproblem

$$\begin{array}{ll} \text{maximize} & (4-2u)\, s_1 + (4-5u)\, s_2 + u \\ \text{subject to} & (s_1, s_2) \in S\ . \end{array}$$

Stated in another way, both the second and third constraints of (5.3.10) are most violated at $z = u = 0$. Thus, the corresponding extreme points s^2 and s^3 must solve this subproblem.

Adding the second constraint gives the optimal solution $z^2 = u^2 = 2$ to the original problem as the next solution to the master problem. Adding the third constraint gives the nonoptimal solution $z^2 = u^2 = 1$ and requires another iteration that adds the remaining constraint of (5.3.10).

In this instance, the second constraint of (5.3.10) dominates the third in the sense that

$$4-u \geq 4-4u$$

whenever $u \geq 0$ with strict inequality if $u > 0$. That is, the second constraint provides a sharper lower bound on z.

To identify the dominant cut in this case, we check to see which of the second or third constraints of (5.3.10) has the largest right-hand side value for <u>any</u> $u^o > 0$. In terms of the subproblem, this criterion becomes: from among the alternate optimal solutions to the subproblem at $\hat{u} = 0$, choose a solution that maximizes the subproblem's objective function when $u = u^o > 0$.

Figure 5.2 illustrates this procedure and serves as motivation for our subsequent analysis. In the figure, we have plotted the three constraints from the dual version (5.3.10) of the subproblem. Note that as a function of u, the minimum objective function value v(u) to the problem is the upper envelope of the three lines in the figure. At $\hat{u} = 0$, the lines $z = 4-u$ and $z = 4-4u$ both pass through this lower envelope. Equivalently, the extreme points $s^2 = (1,0)$ and $s^3 = (0,1)$ of the dual feasible region S both solve the dual problem. Note, however, that as we increase u from $\hat{u} = 0$, the line $z = 4-u$ lies above the line $z = 4-4u$. It, therefore, provides a better approximation to the dual objective function v(u). To identify this preferred line, we can conceptually pivot about the solution point $(\hat{z},\hat{u}) = (4,0)$, choosing the line through this point that lies most to the "northeast". Note

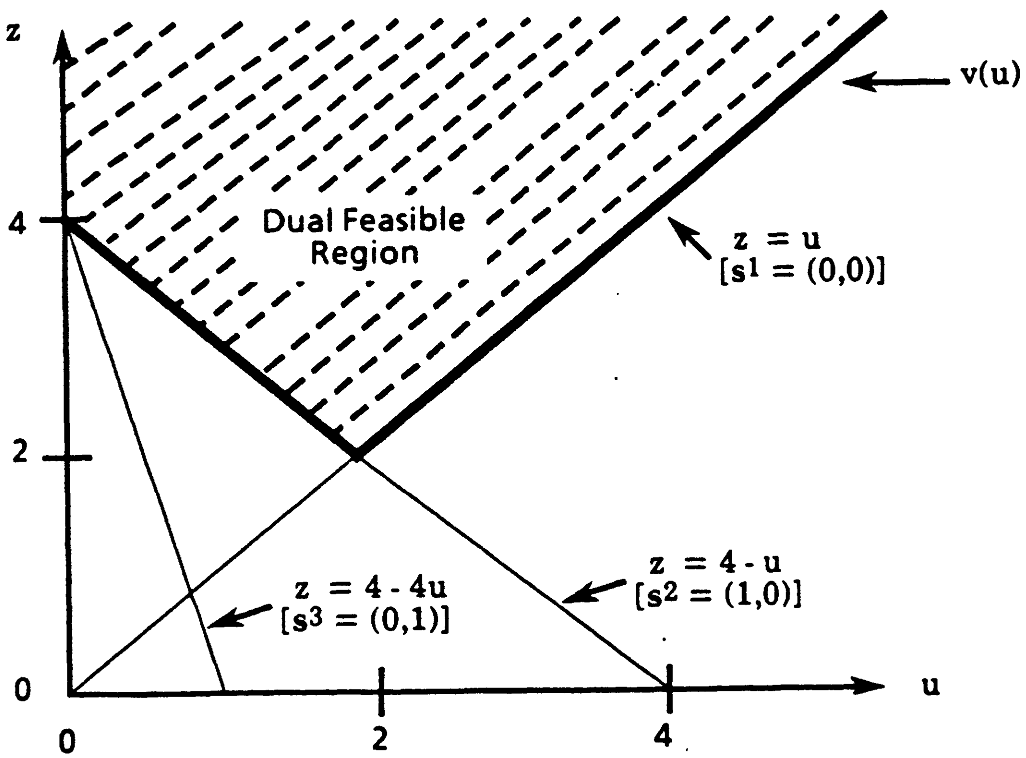


Figure 5.2 Subproblem (Dual) Feasible Region

that to discover this line, we need to find which of the two lines through the pivot point $(\hat{z},\hat{u})$ is higher for <u>any</u> value of $u > 0$.

Before extending this observation to arbitrary minimax problems, we formalize some definitions.

We say that the cut (or constraint)

$$z \geq f(s^1) + ug(s^1)$$

in the minimax problem (5.3.7) <u>dominates</u> or is <u>stronger</u> than the cut

$$z \geq f(s) + ug(s)$$

if

$$f(s^1) + ug(s^1) \geq f(s) + ug(s)$$

for all $u \in U$ with a strict inequality for at least one point $u \in U$. We call a cut <u>pareto-optimal</u> if no cut dominates it. Since a cut is determined by the vector $s \in S$, we shall also say that s^1 dominates (is stronger) than s if the associated cut is stronger, and we say that s is pareto-optimal if the corresponding cut is pareto-optimal.

In the previous example, we showed how to generate a pareto-optimal cut by solving an auxiliary problem defined by any point $u^o > 0$. Note that any such point is an interior point of the set $\{u: u \geq 0\}$. This set, in turn, is the convex hull of the set $U = \{u: u \geq 0 \text{ and integer}\}$.

The following theorem shows that this observation generalizes to any minimax problem of the form (5.3.7). Again, we consider the convex hull of U, denoted U^c, but now we will be more delicate and consider the relative interior (or core)† of U^c, denoted $ri(U^c)$, instead of its interior.

---

† The relative interior of a set is the interior relative to the smallest affine space that contains it. For example, the relative interior of a disk in 3-space (which has no interior) is the interior of the disk when viewed as a circle in 2-space.

The result will always be applicable since the relative interior of the convex set U^c is always nonempty. For notation, let us call any point u^o contained in the relative interior of U^c, a <u>core</u> point of U.

<u>**Theorem 5.1**</u>: Let u^o be a core point of U, that is, $u^o \in ri(U^o)$, and let $S(u)$ denote the set of optimal solutions to the optimization problem

$$\underset{s \in S}{\text{maximize}} \{f(s) + ug(s)\}. \tag{5.3.11}$$

Also, let $\hat{u}$ be any point in U and suppose that s^o solves the problem

$$\underset{s \in S(\hat{u})}{\text{maximize}} \{f(s) + u^o g(s)\}. \tag{5.3.12}$$

Then s^o is pareto-optimal.

<u>Proof</u>: Suppose to the contrary that s^o is not pareto-optimal; that is, there is a $\bar{s} \in S$ that dominates s^o. We first note that the inequalities

$$f(\bar{s}) + ug(\bar{s}) \geq f(s^o) + ug(s^o) \qquad \text{for all } u \in U, \tag{5.3.13}$$

imply that

$$f(\bar{s}) + ug(\bar{s}) \geq f(s^o) + ug(s^o) \qquad \text{for all } u \in U^c. \tag{5.3.14}$$

To establish the last inequality, recall that any point $\bar{u} \in U^c$ can be expressed as a convex combination of a finite number of points in U, that is,

$$\bar{u} = \sum_{u \in U} \{\lambda_u u\colon u \in U\} = 1$$

where $\lambda_u \geq 0$ for all $u \in U$, at most a finite number of the λ_u are positive, and $\sum_{u \in U} \lambda_u = 1$. Therefore, (5.3.14) with $u = \bar{u}$ can be obtained from (5.3.13) by multiplying the u^{th} inequality by λ_u and adding these weighted inequalities.

Also, note from the inequality (5.3.13) with $u = \hat{u}$ that $\bar{s}$ must be an optimal solution to the optimization problem (5.3.11) when $u = \hat{u}$, that is, $\bar{s} \in S(\hat{u})$. But then, (5.3.12) and (5.3.13) imply that

$$f(\bar{s}) + u^{o}g(\bar{s}) = f(s^{o}) + u^{o}g(s^{o}). \tag{5.3.15}$$

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

$$f(\bar{s}) + \bar{u}g(\bar{s}) > f(s^{o}) + \bar{u}g(s^{o}) \tag{5.3.16}$$

for at least one point $\bar{u}\epsilon U$. Also, since $u^{o}\epsilon ri(U^{c})$ for some scalar $\theta > 1$

$$u \equiv \theta\, u^{o} + (1 - \theta)\, \bar{u}$$

belongs to U^{c}. Multiplying equation (5.3.15) by θ, multiplying inequality (5.3.16) by $(1 - \theta)$, which is negative and reverses the inequality, and adding gives

$$f(\bar{s}) + ug(\bar{s}) < f(s^{o}) + ug(s^{o}).$$

But this inequality contradicts (5.3.14), showing that our supposition that s^{o} is not pareto-optimal is untenable. □

We should note that varying the core point u^{o} might conceivably generate different pareto-optimal cuts. Also, any implementation 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 (5.3.12) as compared to the number of iterations that it saves.

In many instances, it is easy to specify a core point u^{o} for implementing the pareto-optimal cut algorithm. If, for example,

$$U = \{u\epsilon R^{k}: u \geq 0 \text{ and integer}\}$$

then any point $u^{o} > 0$ will suffice; if

$$U = \{u\epsilon R^{k}: u_{j} = 0 \text{ or } 1 \text{ for } j = 1,2,\ldots,k\}$$

then any vector u^{o} with $0 > u > 1$ for $j = 1,2,\ldots,k$ suffices; and if

$$U = \{u\epsilon R^{k} : \sum_{j=1}^{k} u_{j} \leq p,\ u \geq 0 \text{ and integer}\}$$

as in the inequality version of the p-median problem (here $X = U$), then any u^{o}

with $u^o > 0$ and $\sum_{j=1}^{n} u_j^0 < p$ suffices. In particular, if $p > n/2$ then $u^o = (1/2, 1/2, \ldots, 1/2)$ is a core point.

When Benders' decomposition is applied to the location model (5.1.7) - (5.1.12), problem (5.3.11) is a specially structured linear program that can be solved efficiently. The next section specifies details of this solution procedure.

5.4 ACCELERATING BENDERS METHOD FOR FACILITY LOCATION

5.4.1 Introduction

Section 5.3 described both Benders' decomposition and Dantzig-Wolfe decomposition within a unifying framework of minimax optimization. This section, which is problem-specific and somewhat more detailed, considers Benders' decomposition as applied to facility location problems. The discussion will serve two purposes. First, it shows how decomposition can be streamlined to exploit the special structure of facility location problems. Second, it introduces several different types of Benders' cuts for facility location models that can be used to design new algorithms. In particular, they can be used as bounding procedures within branch-and-bound algorithms or can be used to design heuristic algorithms. For example, when interpreted properly, a number of successful heuristics for several classes of integer programming problems can be viewed as applying a (heuristic) version of Benders' decomposition that retains only the most recently generated Benders' cut(s) in the master problem. Therefore, the cuts themselves are of interest, independent of their use within decomposition.

Although solving the linear program (5.3.12) generates pareto-optimal cuts for Benders' method applied to general mixed integer programs, the special structure of facility location problems makes it possible to generate strong cuts more efficiently with specialized network algorithms. In this section, we discuss several different cut-generating procedures, ranging from those that produce cuts that dominate the standard Benders' cuts, to more elaborate algorithms that actually produce pareto-optimal cuts. In this discussion, we return to the notation used in Sections 5.1 and 5.2.

Suppose we fix $x = \bar{x} \in X$; then (5.1.7) - (5.1.12), and more generally (5.1.1)-(5.1.6), reduce to a pure linear programming subproblem in the variables y_{ij} which we will call the associated linear program:

$$\begin{aligned} v(\bar{x}) = \min \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} y_{ij} + \sum_{j=1}^{n} c_j \bar{x}_j \\ \text{subject to} \quad & \sum_{j=1}^{n} y_{ij} = 1 \qquad i=1,\dots,m \\ & y_{ij} \leq \bar{x}_j \\ & y_{ij} \geq 0 \qquad i=1,\dots,m;\ j=1,\dots,n. \end{aligned} \tag{5.4.1}$$

For future reference, let us adopt the following notation:

$O = \{j | \bar{x}_j = 1\}$, the index set of open facilities, and

$C = \{j | \bar{x}_j = 0\}$, the index set of closed facilities

corresponding to the configuration $x = \bar{x}$.

The dual of the associated linear program is

$$\begin{aligned} v(\bar{x}) = \max \quad & \sum_{i=1}^{m} [\lambda_i - \sum_{j=1}^{n} \bar{x}_j \pi_{ij}] + \sum_{j=1}^{n} c_j \bar{x}_j \\ \text{subject to} \quad & \lambda_i - \pi_{ij} \leq d_{ij} \qquad i=1,\dots,m;\ j=1,\dots,n \\ & \pi_{ij} \geq 0 \qquad i=1,\dots,m;\ j=1,\dots,n. \end{aligned} \tag{5.4.2}$$

Note that $\Sigma c_j \bar{x}_j$ in the objective function is a constant since the $\bar{x}_j$ are fixed. Any solution λ_i, π_{ij} to this problem determines a cut of the form

$$z \geq \sum_{i=1}^{m} (\lambda_i - \sum_{j=1}^{n} \pi_{ij} x_j) + \sum_{j=1}^{n} c_j x_j \,. \tag{5.4.3}$$

Let us define

$$d_{ij(i)} = \min \{d_{iq}: q \in O\}.$$

Therefore, facility $v_{j(i)}$ is a closest opened facility to node v_i. Then, with $x_j = \bar{x}_j$ for $j = 1,\ldots,n$, the associated linear program (5.4.1) has the following optimal solution:

$$y_{ij} = \begin{cases} 1 & \text{if } j = j(i) \\ 0 & \text{otherwise} \end{cases} \qquad \begin{array}{l} i = 1,\ldots,m \\ j = 1,\ldots,n. \end{array}$$

Also, the dual program (5.4.2) has the following "natural" solution:

For each $i=1, 2,\ldots,m$,

$$\begin{aligned} \bar{\lambda}_i &= d_{ij(i)} & & \\ \bar{\pi}_{ij} &= 0 & &\text{if } j \in O,\ j=1,\ldots,n \\ \bar{\pi}_{ij} &= \max\ (0, \bar{\lambda}_i - d_{ij}) & &\text{if } j \in C,\ j=1,\ldots,n. \end{aligned} \qquad (5.4.4)$$

The optimal dual variables have a convenient interpretation: $\bar{\lambda}_i$ is the cost of servicing node v_i when $x = \bar{x}$; $\bar{\pi}_{ij}$ is the reduction in the cost of servicing node v_i when facility v_j is opened and $x_i = \bar{x}_i$ for all $i \neq j$. So from the dual subproblem solution, we can construct the following cut

$$z \geq \bar{\omega} - \sum_{j=1}^{n} \mu_j x_j + \sum_{j=1}^{n} c_j x_j \qquad (5.4.5)$$

whose coefficients ω and μ_j are defined by

$$\bar{\omega} = \sum_{i=1}^{m} \bar{\lambda}_i \quad \text{and} \quad \mu_j = \sum_{i=1}^{m} \bar{\pi}_{ij}\ .$$

Note that $\bar{\omega}$ is the total servicing costs when $x = \bar{x}$ and that μ_j is the total reduction in servicing costs if facility v_j is opened and all other facilities retain their current, open vs. closed, status.

For reference purposes, we shall refer to the cut in (5.4.5) as a usual cut. We discussed this cut in our introduction to decomposition in Section 5.2.

For $x = \bar{x}$, the associated linear program can be viewed as a transportation problem with demand constraints (5.1.8) for each destination v_i and a set of unconstrained origins v_j (i.e., each has an unlimited supply). Typically, transportation problems have a degenerate optimal basis which implies that the dual problem (5.4.2) has multiple optimal solutions. Because of this property, it is usually possible to derive more than one Benders' cut. We next describe procedures for generating alternative cuts that will usually be superior to the usual cut (5.4.5).

An Improved Cut

In deriving the Benders' cut (5.4.5), we considered only the savings from opening a new facility, i.e., increasing some x_j from 0 to 1. We did not, however, consider the added servicing costs produced by closing a facility. If facility $v_{j(i)}$ is closed, then node v_i must be serviced from a different facility and the service cost for node v_i must be at least the cost $d_{ik(i)}$ of servicing node v_i from the best alternative node $V_{k(i)}$; that is,

$$d_{ik(i)} = \min \{d_{iq} : 1 \leq q \leq n \text{ and } q \neq j(i)\}.$$

Note that since $v_{j(i)}$ must be an open facility and $v_{k(i)}$ need not be open, $d_{ik(i)}$ might be less than $d_{ij(i)}$.

Let

$$\sigma_i = \max \{d_{ik(i)} - d_{ij(i)}, 0\}.$$

Whenever $\sigma_i > 0$, node v_i suffers an increase in service cost of at least σ_i if facility $v_{j(i)}$ is closed, i.e., if $x_{j(i)}$ is decreased from 1 to 0. Therefore,

$$\nu_j = \Sigma\{\sigma_i : 1 \leq i \leq n \text{ and } j = j(i)\} \qquad (5.4.6)$$

is the minimum total service cost incurred from all customers by closing facility v_j. So we can write a new cut, which we will refer to as a closing facility cut, as follows:

$$z \geq \bar{\omega} + \sum_{j\varepsilon O} (1-x_j)\nu_j - \sum_{j\varepsilon C} \mu_j x_j + \sum_{j=1}^{m} c_j x_j \; . \qquad (5.4.7)$$

Notice that whenever some $\nu_j \neq 0$ and $\hat{x}_j \neq 0$ for some $\hat{x}\varepsilon X$, the closing facility cut will dominate the usual cut.

Example 5.4.1

To illustrate the concepts introduced in this subsection, we consider once again the two-median problem given in Figure 5.1. Let the current configuration $\bar{x} = (1,0,1,0,0)$. Then from (5.4.4) we have the "natural" dual solution

$$\bar{\lambda}_1 = 0,\; \bar{\lambda}_2 = 4,\; \bar{\lambda}_3 = 0,\; \bar{\lambda}_4 = 2,\; \bar{\lambda}_5 = 3$$

and $\bar{\pi}_{ij} = \max\,(0, \bar{\lambda}_i - d_{ij})$ $\qquad i=1,2,3,4,5;\; j=1,2,3,4,5.$

Substituting into (5.4.5), we obtain the usual cut

$$z \geq 9 \quad - 4x_2 \quad - 4x_4 \quad - 4x_5 \qquad (5.4.8)$$

which we specified earlier as (5.2.2).

To compute the closing facility cut for the current configuration $\bar{x}$, note that $d_{ik(i)} = 0$ for $i=2,4,5$, $d_{1k(1)} = 5$, and $d_{3k(3)} = 2$. So,

$$\nu_1 = \sigma_1 = 5$$

$$\nu_3 = \sigma_2 + \sigma_3 + \sigma_4 + \sigma_5 = 0 + 2 + 0 + 0 = 2$$

and substituting into (5.4.7) yields the closing facility cut

$$z \geq 9 + 5(1-x_1) - 4x_2 + 2(1-x_3) - 4x_4 - 4x_5. \qquad (5.4.9)$$

Notice that in this example, since every node is a candidate for a facility, it is possible to open a facility at each closed node and consequently $d_{ik(i)} = 0$ for each closed node $v_i \in C$. In general, these costs could be positive.

5.4.2 Pareto-Optimal Cuts

In this section, we derive an efficient special purpose algorithm for solving the linear program for generating pareto-optimal cuts for the facility location model. The algorithm uses a parametric solution technique to solve each of the subproblems.

First, we note that for any choice of $\bar{x} \in X$, the linear programs (5.4.2) decompose into separate subproblems, one for each index $i=1,2,\ldots,m$. Also, the "natural solution" (5.4.4) to the linear programming dual problem (5.4.2) has the property that the optimal value of the ith subproblem is $v_i(\bar{x}) = \bar{\lambda}_i$ and $(\lambda_i - \sum_{j=1}^{n} \bar{x}_j \pi_{ij}) \le v_i(\bar{x})$ for any (λ, π) that is feasible for (5.4.2).

Consequently,

$$\sum_{i=1}^{m} (\lambda_i - \sum_{j=1}^{n} \bar{x}_j \pi_{ij}) = v(\bar{x}) = \sum_{i=1}^{m} v_i(\bar{x}) - \sum_{j=1}^{n} c_j \bar{x}_j$$

if and only if $(\lambda_i - \sum_{j=1}^{n} \bar{x}_j \pi_{ij}) = v_i(\bar{x})$. This observation implies that we can also decompose (5.3.11) into a series of subproblems. For each i, solving the subproblem

$$\begin{aligned} \text{maximize} \quad & \lambda_i - \sum_{j=1}^{n} x_j^o \pi_{ij} & & \\ \text{subject to} \quad & \lambda_i - \sum_{j=1}^{n} \bar{x}_j \pi_{ij} = \bar{\lambda}_i & & \\ & \lambda_i - \pi_{ij} \le d_{ij} & & j = 1,\ldots,m \\ & \pi_{ij} \ge 0 & & j = 1,\ldots,m \\ & \lambda_i \ge 0 & & \end{aligned} \tag{5.4.10}$$

provides a pareto-optimal vector with components λ_i and π_{ij} for $i=1,\ldots,m$ and $j=1,\ldots,n$. In this formulation, as before, $\bar{x}$ denotes the current value of the integer variables and x^o belongs to the core of X (which was S in Section 5.3); that is, $x^o \varepsilon ri(X^c)$.

Our first objective is to show that, for each i, the optimal value of the subproblem (5.4.10) is piecewise linear as a function of λ_i. Since the equality constraint of this problem reads

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

and since $\lambda_i - \pi_{ij(i)} \leq d_{ij(i)}$ and $\pi_{ij} \geq 0$ for all j, we have $\pi_{ij} = 0$ for all $j \neq j(i)$, $j \varepsilon O$ and $\pi_{ij(i)} = \lambda_i - d_{ij(i)} = \lambda_i - \bar{\lambda}_i$. Also, if we substitute for λ_i in the objective function of (5.4.10) from the equality constraint, the objective becomes

$$\text{maximize } \bar{\lambda}_i + \sum_{j=1}^{n} (\bar{x}_j - x_j^o)\, \pi_{ij}.$$

For any index $j \varepsilon C$, $\bar{x}_j = 0$ and the coefficient $\varepsilon_j \equiv (\bar{x}_j - x_j^o)$ of π_{ij} is nonpositive. Thus, an optimal choice of π_{ij} in (5.4.10) is $\pi_{ij} = \max(0, \lambda_i - d_{ij})$.

Collecting these results, we see that the optimal value of problem (5.4.10) as a function of the variable λ_i is

$$\bar{\lambda}_i + \varepsilon_{j(i)}(\lambda_i - \bar{\lambda}_i) + \sum_{j\varepsilon C} \varepsilon_j \max(0, \lambda_i - d_{ij}). \qquad (5.4.11)$$

To aid us in optimizing (5.4.10), we note upper and lower bounds $\bar{\lambda}_i \leq \lambda_i \leq L_i$ on λ_i where, by definition, $L_i = \min\{d_{ij} : j \varepsilon O \text{ and } j \neq j(i)\}$. The lower bound is a simple consequence of the equality constraint of problem (5.4.10), because each $\bar{x}_j \geq 0$ and each $\pi_{ij} \geq 0$. The upper bound follows from our previous observation that $\pi_{ij} = 0$ whenever $j \neq j(i)$ and $j \in O$, and, therefore, for these j the constraint $\lambda_i - \pi_{ij} \leq d_{ij}$ becomes $\lambda_i \leq d_{ij}$.

Now, since the function (5.4.11) is piecewise linear and concave in λ_i, we can minimize it by considering its linear segments 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. More formally, for a current configuration $\bar{x}$ and a core point x^o, we have the following procedure:

Pareto-Optimal Cut Generation Algorithm

(0) For all $i=1,\ldots,m$ and $j=1,\ldots,n$, compute

$$\varepsilon_j = (\bar{x}_j - x_j^o)$$

$$d_{ij(i)} = \min \{d_{ij}: j \in O\}$$

$$L_i = \min \{d_{ij}: j \in O \text{ and } j \neq j(i)\}.$$

For every $i=1,2,\ldots,m$ perform the following steps:

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

(2) Let $T = \{j \in C : d_{ij} \leq \lambda_i\}$ and let $s = \varepsilon_{j(i)} + \Sigma\{\varepsilon_j : j \in T\}$. s is the slope of the function (5.4.11) to the right of λ_i.

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

(4) Let $d_{ik} = \min \{d_{ij} : j \in C \text{ and } j \notin T\}$. If $d_{ik} > L_i$, set $\lambda_i = L_i$ and stop. Otherwise, increase λ_i to d_{ik}.

Repeat steps (2) - (4).

Once the optimal value of λ_i for each i is found using this algorithm, the remaining variables π_{ij} can be set using the rule $\pi_{ij} = \max (0, \lambda_i - d_{ij})$. Then, the cut obtained by substituting these values in (5.4.3) is pareto-optimal.

This algorithm should be very efficient. For each node v_i, at most n (the number of possible facilities) steps must be executed, so the total number of steps required by this procedure is bounded by (number of demand nodes)(number of possible facilities).

We might emphasize that this algorithm determines a pareto-optimal cut for any given point x° in the core of X. Also, the algorithm applies to any of the possible modeling variations that we might capture in X, such as the contingency and configuration constraints mentioned in Section 5.1.

Example 5.4.2

Once again, consider the two-median example depicted in Figure 5.1. To apply the pareto-optimal cut generation algorithm, we must first choose a core point $x^o \in ri(\bar{X}^c)$ of the set

$$\bar{X} = \{x: \sum_{i=1}^{5} x_i = 2 \text{ and } 0 \le x_i \le 1 \text{ for } 1 \le i \le 5\} .$$

One possible value for the core point is $x^o = (1/2,1/4,1/2,1/4,1/2)$. The conditions of a core point are satisfied since

$$\sum_{i=1}^{5} x_i^o = \frac{1}{2} + \frac{1}{4} + \frac{1}{2} + \frac{1}{4} + \frac{1}{2} = 2$$

and $0 < x_i^o < 1$ for $1 \le i \le 5$.

As in Section 5.2, assume that the current configuration $\bar{x}$ is (1,0,1,0,0). From computations in example 5.4.1, we know $\bar{\lambda} = (0,4,0,2,3)$.

Now we apply the steps of the pareto-optimal cut generation algorithm. For i=1, we have $\bar{\lambda}_1 = 0 \le \lambda_1 \le L_1 = 9$. Step 1 initializes $\lambda_1 = 0$. Step 2 computes $T = \emptyset$ and $s = \varepsilon_1 = 1 - \frac{1}{2} = \frac{1}{2}$. Since $s > 0$, step 4 increases λ_1 to $d_{12} = 5$. Next we return to step 2 and find $T = \{2\}$ and $s = \varepsilon_1 + \varepsilon_2 = (1- \frac{1}{2}) +$

$(0 - \frac{1}{4}) = \frac{1}{4}$. Since $s > 0$, step 4 increases λ_1 to $d_{13} = 9$. But since $\lambda_1 = L_1 = 9$, we can stop with the optimal value $\lambda_1 = 9$.

Similar computations show that $\lambda_2 = 5$, $\lambda_3 = 3$, $\lambda_4 = 2$, and $\lambda_5 = 3$.

Using the values for λ permits us to compute the corresponding values for the π_{ij} from the relation

$$\pi_{ij} = \max(0, \lambda_i - d_{ij}).$$

Substituting these values of the λ_i and π_{ij} into (5.4.3) produces the following pareto-optimal Benders' cut for the two-median problem:

$$z \geq 22 - 9x_1 - 9x_2 - 4x_3 - 5x_4 - 4x_5. \qquad (5.4.12)$$

5.4.2 Neighborhood Interpretation and a New Cut

As we noted in Section 5.4.1, the standard Benders' cut considers savings in servicing costs obtained by opening new facilities. The improved closing facility cut introduces additional servicing costs that must be incurred whenever an open facility is closed. In this section, we show that any Benders' cut generated from an optimal solution to the dual problem (5.4.2) has a similar interpretation. We also present a new type of cut for the p-median problem and discuss its interpretation.

An Interpretation

First, we introduce some new notation. The δ-neighborhood, denoted $N_i(\delta)$, of demand node v_i with respect to the dual variable $\lambda_i = \bar{\lambda}_i + \delta$, with value $\bar{\lambda}_i$ defined in (5.4.4), is the set of facility locations v_j satisfying $d_{ij} \leq \bar{\lambda}_i + \delta$. We define the interior of the δ-neighborhood as

$N_i^o(\delta) \equiv \{v_j: d_{ij} < \bar{\lambda}_i + \delta\}$.

Recall from the last section that $\lambda_i \geq \bar{\lambda}_i$ in any optimal solution to the dual of subproblem (5.4.1); therefore, $\lambda_i = \bar{\lambda}_i + \delta_i$ for some $\delta_i \geq 0$ and varying δ_i (and hence the size of the δ-neighborhood) is equivalent to varying λ_i. The operation in the pareto-optimal cut generation algorithm of increasing λ_i until $s \leq 0$ has the following interpretation: increase the δ-neighborhood about node v_i until $\Sigma\{\varepsilon_j: v_j \varepsilon N_i(\delta_i)\} \leq 0$.

Figure 5.3 gives a small example of a δ-neighborhood for the p-median example shown in Figure 5.1. Assume distances in the figure are drawn to scale (they aren't) and that the neighborhood is constructed around demand node v_5 with the current configuration $\bar{x} = (1,0,0,1,0)$. As indicated in the figure, the current δ-neighborhood contains nodes v_5, v_4, v_3, and v_2. If $\varepsilon_5 + \varepsilon_4 + \varepsilon_3 + \varepsilon_2 = s > 0$, then the pareto-optimal cut algorithm would expand the neighborhood to the next nearest facility, which is node v_1.

The cut determined by the neighborhood pictured in Figure 5.3 has the

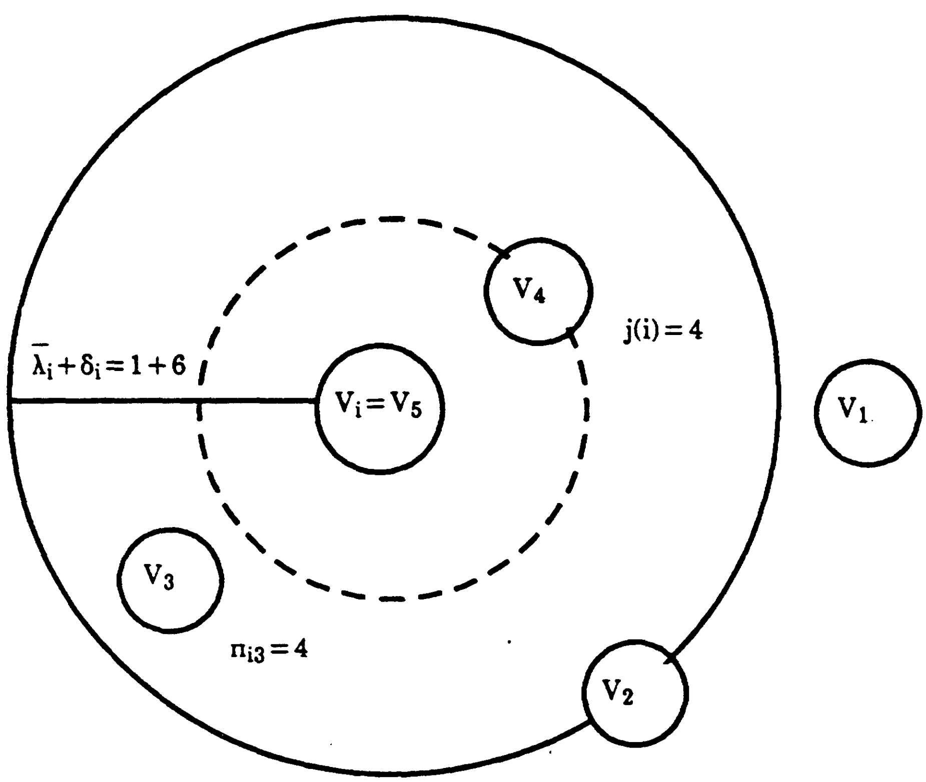


Figure 5.3. Neighborhood about a demand node $V_i = V_5$

following interpretation. As we noted in deriving (5.4.11) from (5.4.10), every optimal solution to (5.4.2) can be written as

$$\lambda_i = \bar{\lambda}_i + \delta_i \quad \text{for some } \delta_i \geq 0$$

$$\bar{\lambda}_i = d_{ij(i)} \equiv \min \{d_{ij}: j\varepsilon O\}$$

$$\pi_{ij} = \begin{cases} \lambda_i - \bar{\lambda}_i = \delta_i & \text{if } j = j(i) \\ \lambda_i - d_{ij} & \text{if } j\varepsilon C \cap N_i(\delta_i) \\ 0 & \text{otherwise.} \end{cases}$$

For notational convenience, let us assume at this point that node v_5 is the only demand node. Substituting the dual variable values corresponding to Figure 5.3 into (5.4.3) yields (after some rearrangement) the cut

$$z \geq \bar{\lambda}_5 + \delta_5(1-x_4) - (\bar{\lambda}_5 + \delta_5 - d_{53})x_3 - (\bar{\lambda}_5 + \delta_5 - d_{55})x_5$$

or

$$z \geq 1 + 6(1 - x_4) - 4x_3 - 7x_5 \; .$$

If we set $x_4 = 0$ to close facility 4, then node v_5 must be serviced from outside the neighborhood (or on its boundary) at a cost z of at least $\bar{\lambda}_5 + \delta_5 = 7$ unless a facility at node v_3 or node v_5 is opened. If facility 5 is opened, then the service cost for node v_5 becomes $d_{55} = 0$. The coefficient of x_5 compensates for this reduction in service cost when $x_5 = 1$. The coefficient for x_3 has a similar interpretation.

The general situation is much the same. Given any neighborhoods for the nodes, let $\bar{\omega} = \sum_{i=1}^{m} \bar{\lambda}_i$ be the current allocation cost, let

$\nu_j = \Sigma\{\delta_i: 1 \leq i \leq n \text{ and } j=j(i)\} = \Sigma\{\pi_{ij}: 1 \leq i \leq j \text{ and } j=j(i)\}$, and let $\mu_j = \sum_{i=1}^{n} \pi_{ij}$.

Substituting these values in the form of the cut expressed in (5.4.3) gives

$$z \geq \bar{\omega} + \sum_{j\varepsilon O} \nu_j(1-x_j) - \sum_{j\varepsilon C} \mu_j x_j + \sum_{j=1}^{n} c_j x_j. \qquad (5.4.13)$$

The coefficient ν_j accounts for the fact that the open facility v_j might lie interior to several neighborhoods. Closing this facility increases allocation costs to the boundary of each of these neighborhoods unless some closed facility within any neighborhood is opened. The coefficient μ_j for $j \in C$ records the savings for opening facility v_j considering all the neighborhoods that contain it.

Suppose, as before, that $d_{ik(i)}$ denotes the cost to the closest alternate facility $k(i) \neq j(i)$ to node v_i. Setting $\lambda_i = \bar{\lambda}_i$, and $\delta_i = \max\{0, d_{ik(i)} - d_{ij(i)}\}$, reduces expression (5.4.13) to the closing facility cut introduced in Section 5.4.1.

Note that the cut (5.4.13) is only an <u>alternative</u> way of describing the Benders' cut (5.4.3) using the neighborhood interpretation. This alternative description not only provides a new interpretation, but will permit us to introduce a new type of Benders' cut that is easily described using the alternative cut (5.4.13). We first use an example to illustrate the equivalence of the two cut descriptions (5.4.3) and (5.4.13).

<u>Example 5.4.3</u>

We continue example 5.4.2 and discuss the two-median problem given in Figure 5.1 with $\bar{x} = (1,0,1,0,0)$. Recall from the example that $\bar{\lambda} = (0,4,0,2,3)$ and the final computed value for λ was (9,5,3,2,3). We will translate this solution to our neighborhood interpretation and use it to show the equivalence of the two cut descriptions (5.4.3) and (5.4.13).

Using $\bar{\lambda}$ and the final value for λ permits us to compute $\delta_1 = \lambda_1 - \bar{\lambda}_1 = 9$, $\delta_2 = 5-4=1$, $\delta_3 = 3-0=3$, $\delta_4 = 2-2=0$, and $\delta_5 = 3-3=0$.

We can now determine the corresponding neighborhoods. For example, $N_1(9) = \{v_1,v_2,v_3\}$; $N_1^o(9) = \{v_2,v_3\}$; $N_2(1) = \{v_1,v_2\}$; and $N_2^o(1) = \{v_2\}$.

$$\bar{\omega} = \sum_i \bar{\lambda}_i = 0 + 4 + 0 + 2 + 3 = 9.$$

Also, since $\bar{x}_1$ and $\bar{x}_3$ are 1, we compute

$$\nu_1 = \delta_1 = 9$$

$$\nu_3 = \delta_2 + \delta_3 + \delta_4 + \delta_5 = 1 + 3 + 0 + 0 = 4.$$

In this case, $j(i) = 1$ for $i=1$, and $j(i) = 3$ for $i=2,3,4,5$. Now $\bar{x}_2$, $\bar{x}_4$, and $\bar{x}_5$ are zero, so we compute

$$\mu_2 = \pi_{12} + \pi_{22} + \pi_{32} + \pi_{42} + \pi_{52} = 4 + 5 + 0 + 0 + 0 = 9$$

$$\mu_4 = \pi_{14} + \pi_{24} + \pi_{34} + \pi_{44} + \pi_{54} = 0 + 0 + 1 + 2 + 2 = 5$$

$$\mu_5 = \pi_{15} + \pi_{25} + \pi_{35} + \pi_{45} + \pi_{55} = 0 + 0 + 0 + 1 + 3 = 4.$$

Substituting these values into (5.4.13), we obtain the cut

$$z \geq 9 + 9(1-x_1) + 4(1-x_3) - 9x_2 - 5x_4 - 4x_5$$

which is the same as (5.4.12), the cut computed from (5.4.3) in example 5.4.2.

A New Cut for the p-Median Problem:

When specialized, the cut-generating technique described in the last section provides a new type of Benders' cut for the p-median problem, one that dominates the closing facility cut. To simplify our development, we temporarily assume that all servicing costs d_{ij} are nonnegative and that $d_{ii} = 0$ for all i.

As we have seen, the closing facility cut introduces penalties for customers forced to travel to nodes other than the ones to which they are currently assigned. For the p-median problem, these penalties separate into two groups:

(a) a demand node and a facility are located at the same node v_i. Then the servicing cost for that demand node is $d_{ii} = 0$ and the penalty in servicing cost for this node is

$$d_{ik(i)} \equiv \min \{d_{ij}: j \neq i\}$$

if the facility at node v_i is closed.

(b) a demand node, but no facility, is located at node v_i. Then the closing of any open facility need not incur any servicing penalty, since the demand node might conceivably be serviced by a facility at node v_i at cost $d_{ii} = 0$.

Stated in terms of the neighborhood interpretation, these observations imply that the δ_i-neighborhood about demand node v_i is of minimal size, $\delta_i = 0$, if $x_i = 0$ in the current solution $\mathbf{x}$; if $x_i = 1$, then $d_{ij(i)} \equiv \min \{d_{ij}: j \varepsilon O\}$ and $\delta_i = d_{ij(i)}$ is the size of the neighborhood.

Since closing a facility at node v_j contributes only to the penalty in the closing facility cut of the demand at that node, the term $\nu_j \equiv \Sigma\{\delta_i: 1 \leq i \leq n \text{ and } j = j(i)\}$ equals $d_{jk(j)}$, the distance to node v_j's second nearest neighbor and the closing facility cut is written in the form of expression (5.4.13) as

$$z \geq \bar{\omega} + \sum_{j \varepsilon O} d_{jk(j)} (1-x_j) - \sum_{j \varepsilon C} \mu_j x_j \qquad (5.4.14)$$

whose terms $\bar{\omega}$ and μ_j are defined as before in (5.4.5).

The algorithm presented in Section 5.4.1 shows how to expand the neighborhoods about every demand node from the values associated with the usual cut in order to obtain pareto-optimality. Although the new cut must be pareto-optimal, there is no guarantee that it dominates the usual cut or the closing facility cut.

To develop a cut that dominates the closing facility cut, we proceed as follows. We maintain the neighborhood about nodes whose facilities are closed at their minimal size $\delta_j = 0$, and we increase the neighborhoods about the other nodes all by the same amount $\bar{\delta}$. That is, we set $\delta_j = d_{jk(j)} + \bar{\delta}$ for every node v_j with $\bar{x}_j = 1$. This procedure avoids the formal slope checking mechanism of the algorithm for generating pareto-optimal cuts. Although other options are certainly possible, choosing to expand every neighborhood equally leads to a very simple implementation.

The choice of $\bar{\delta}$ for δ is governed by two restrictions. First, the resulting value of δ cannot be too large, since otherwise $\lambda_i = (\bar{\lambda}_i + \delta_i) = \bar{\lambda}_i + d_{ik(i)} + \delta$ for $i \in O$ will violate

$$\lambda_i \leq L_i = \min\{d_{ij}: j \in O \text{ and } j \neq j(i)\}.$$

Recall that we identified this bound in Section 5.4.1 by considering the subproblem (5.4.10).

This bound on the λ_i is equivalent to the restriction that if $j \neq j(i)$ and $j \in O$, then $\lambda_i = (\bar{\lambda}_i + \delta_i) \leq d_{ij}$ or j cannot be in the interior of the neighborhood about node v_i. Since node $v_{j(i)}$ is in the interior of the neighborhood about node v_i whenever $\delta_i > 0$, this bound on λ_i is equivalent to the restriction that the interior of every neighborhood may contain at most one open facility. The second restriction is that every closed facility lie interior to at most one neighborhood about an open facility. Although this restriction is not imposed by the linear programs (5.4.10), later we will show by an example that the new cut need not dominate the closing facility cut if this condition is not fulfilled. Our choice of δ is made as large as possible, consonant with these two restrictions.

We will call the result of this procedure

$$v \geq \bar{\omega} + \sum_{j\epsilon O} (d_{jk(j)} + \bar{\delta})(1-x_j) - \sum_{j\epsilon C} (\mu_j + \bar{\Delta}_j)x_j \qquad (5.4.15)$$

an expanding neighborhood cut. Note that the coefficient of the closed facilities v_j must be altered from the values μ_j in the closing facility cut (5.4.14). Since our restrictions on $\bar{\delta}$ ensure that every closed facility v_j lies interior to only one neighborhood $N_q(\delta)$, if any, about an open facility, as we increase δ only the term π_{qj} in the saving expression $\mu_j = \sum_{i=1}^{m} \pi_{ij}$ changes. Each compensation factor $\bar{\Delta}_j$ to the savings expression equals the difference between $\pi_{qj} = \max (\bar{\lambda}_q - d_{qj}, 0) = \max (d_{qk(q)} + \delta - d_{qj}, 0)$, at $\delta = 0$ and at $\delta = \bar{\delta}$ (see Section 5.4.1). Note that this observation implies that $\bar{\Delta}_j \leq \bar{\delta}$ for all $j\epsilon C$.

In comparing cuts, we noted that the closing facility cut dominates the usual cut whenever at least one $\nu_j \neq 0$. The following result summarizes the relationship between closing facility and expanding neighborhood cuts.

Proposition 5.1: For a given iteration of Benders' decomposition applied to the p-median problem, an expanding neighborhood cut will either dominate or be equivalent to a closing facility cut.

Proof. Let $\mathbf{x} = \bar{\mathbf{x}}$ be any values for the configuration variables satisfying the p-median constraint $x_1 + x_2 + \dots + x_n = p$. Let $R_E(\mathbf{x})$ and $R_C(\mathbf{x})$ denote the right-hand sides of the expanding neighborhood cut (5.4.15) and the closing facility cut (5.4.14). Then

$$R_E(x) - R_C(x) = \sum_{j\epsilon O} \bar{\delta}(1-x_j) - \sum_{j\epsilon C} \bar{\Delta}_j x_j \ .$$

By the p-median constraint, if K of the facilities v_j for $j\varepsilon C$ are opened, then K of the facilities v_j for $j\varepsilon O$ must be closed. As we have noted just prior to the proposition, though, $\bar{\Delta}_j \leq \bar{\delta}$ for all $j\varepsilon C$. These two facts imply that $R_E(\mathbf{x}) - R_C(\mathbf{x}) \geq 0$, so the expanding neighborhood cut is always at least as strong as the closing facility cut. □

Reviewing the definition of the expanding neighborhood cut and the proof of this proposition shows that our assumptions that service costs are nonnegative and that $d_{ii} = 0$ for all i are dispensable. These assumptions merely lead to more attractive interpretations and motivation.

Example 5.4.4

We continue discussing the two-median example of Figure 5.1 with the current configuration $\bar{\mathbf{x}} = (1,0,1,0,0)$. Recall from the example 5.4.1 that $\bar{\lambda} = (0,4,0,2,3)$, $d_{1k(1)} = 5$ and $d_{3k(3)} = 2$. Also, $\mu_2 = 4$, $\mu_4 = 4$, and $\mu_5 = 4$. Using the δ-neighborhood concept, we find that $\bar{\delta}$ equals 2 (for $\bar{\delta} > 2$, node v_2 lies in the interior of the neighborhoods about node v_1 and node v_3), and we obtain the following expanding neighborhood cut

$$z \geq 9 + (5+2)(1-x_1) - (4+2)x_2 + (2+2)(1-x_3) - (4+2)x_4 - (4+1)x_5. \tag{5.4.16}$$

Note that the expanding neighborhood cut dominates the usual cut (5.4.8) and the closing facility cut (5.4.9).

If we ignored the restriction prohibiting node v_2 from being in the interior of the neighborhoods about both nodes v_1 and v_3, we could expand the neighborhoods until $\delta = 4$ and the cut would become

$$z \geq 9 + 9(1-x_1) - 10x_2 + 6(1-x_3) - 8x_4 - 7x_5.$$

Observe that this cut does not dominate the closing facility cut (5.4.9) take $x_1 = x_2 = 1$, $x_3 = x_4 = x_5 = 0$). The difficulty is that $\bar{\Delta}_2 = 6$ exceeds $\bar{\delta} = 4$.

Finally, notice the expanding neighborhood cut (5.4.16) does not dominate the pareto-optimal cut (5.4.12) (take $x_4 = x_5 = 1$, $x_1 = x_2 = x_3 = 0$) nor does the pareto-optimal cut dominate it (take $x_1 = x_2 = 1$, $x_3 = x_4 = x_5 = 0$).

5.5. A MODEL SELECTION CRITERION FOR BENDERS' DECOMPOSITION

Selecting the "proper" model formulation is another important factor that affects the computational performance of Benders' decomposition applied to facility location 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 "natural" mixed integer programming formulations. For example, as we noted in Section 5.1, different variations of the facility location problem can be stated in several possible ways as mixed integer programs. In this section, we demonstrate why some formulations lead to such pronounced improvements over others in the performance of Benders' decomposition.

To illustrate the role of model selection, we consider an example of Benders' decomposition applied to the p-median facility location problem. In Section 5.1, (5.1.7) - (5.1.10) gives a formulation, say P, of the p-median problem when

$$X = \{x: \sum_{j=1}^{n} x_j = p\}.$$

Replacing (5.1.9), i.e., $y_{ij} \leq x_j$, with

$$\sum_{i=1}^{m} y_{ij} \leq m\, x_j \qquad j=1,\ldots,n \tag{5.5.1}$$

gives an equivalent formulation, say Q. Note that (5.5.1) represents an aggregation of (5.1.9). Consequently, although P and Q are equivalent mathematical descriptions, if we relax the integrality constraints on the x_i, the feasible region for P is a proper subset of the feasible region for Q.

Let us examine the p-median problem represented in Figure 5.4:

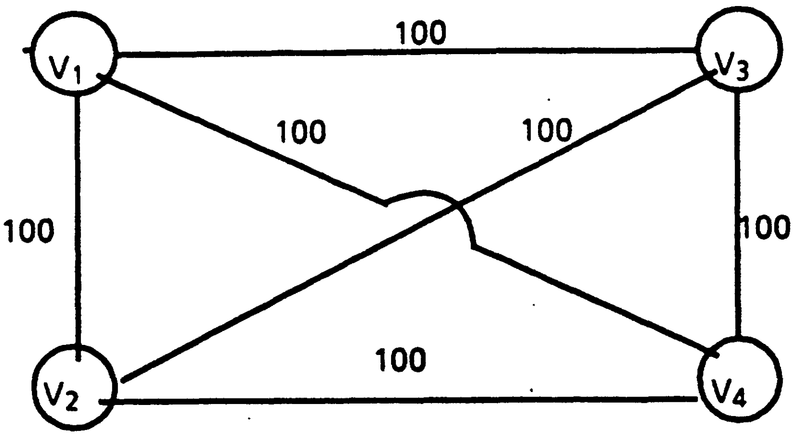


Figure 5.4 Example of a p-median problem with n = 4 and p = 2

In this example, n=4, p=2, and all d_{ij} are 100 except for $d_{ii} = 0$.

The application of Benders' decomposition to this example with formulation Q yields the following set of Benders' cuts:

$$z \geq 200 - 400x_1 - 400x_2 + 0x_3 + 0x_4$$
$$z \geq 200 - 400x_1 + 0x_2 - 400x_3 + 0x_4$$
$$z \geq 200 - 400x_1 - 0x_2 + 0x_3 - 400x_4$$
$$z \geq 200 + 0x_1 - 400x_2 - 400x_3 + 0x_4$$
$$z \geq 200 + 0x_1 - 400x_2 + 0x_3 - 400x_4$$
$$z \geq 200 + 0x_1 + 0x_2 - 400x_3 - 400x_4.$$

It turns out (see exercise 5.8) that every single one of these cuts must be generated in order for Benders' algorithm to converge.

Recall from Section 5.4 that applying Benders' decomposition to our example with formulation P yields several different sets of cuts. The first set, consisting of the usual cuts, is identical to the previous set except that all coefficients of value -400 become -100. So all six cuts are again necessary for convergence. In contrast, generating a set of closing facility cuts requires the single cut

$$z \geq 400 - 100x_1 - 100x_2 - 100x_3 - 100x_4$$

for convergence.

We can generalize this example in the following way: let $p = n/2$ and let $d_{ij} = 100$ for all $i \neq j$ and $d_{ii} = 0$ for all i. For this class of examples, the Q formulation requires an exponential number $\binom{n}{n/2}$ of cuts 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. Our results apply not only to facility location problems (5.1.1) - (5.1.6), but also to general mixed integer programming problems. Since it will facilitate our notation, we cast our development in its most general form. Later in this section, we discuss the application of this theory to specific facility location models.

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

$$\text{(P)} \quad \underset{\mathbf{x} \in X}{\text{Minimize}} \, [v^P(\mathbf{x})] \text{ where } [v^P(\mathbf{x})] = \underset{\mathbf{y} \geq 0}{\text{minimum}} \, \{\mathbf{cx}+\mathbf{dy} : H\mathbf{x}+G\mathbf{y} = \mathbf{h}\} \qquad (5.5.2)$$

and

$$\text{(Q)} \quad \underset{\mathbf{x} \in X}{\text{Minimize}} \, [v^Q(\mathbf{x})] \text{ where } [v^Q(\mathbf{x})] = \underset{\mathbf{y} \geq 0}{\text{minimum}} \, \{\mathbf{cx}+\mathbf{ty} : R\mathbf{x}+T\mathbf{y} = \mathbf{r}\}. \qquad (5.5.3)$$

$\mathbf{x}$ and $\mathbf{y}$ are column vectors of problem variables; $\mathbf{h}$ and $\mathbf{r}$ are column vectors; $\mathbf{c}$,$\mathbf{d}$, and $\mathbf{t}$ are row vectors; H,G,R, and T are matrices with appropriate dimensions. The set X is a set of integer-valued vectors and captures the integer constraints of the problem. We assume that the set X is finite.

We will say that P and Q are equivalent mixed integer programming representations of the same problem if $v^P(\mathbf{x}) = v^Q(\mathbf{x})$ for all $\mathbf{x} \in X$. That is, the two models have the same integer variables, but may have different constraints and continuous variables; nevertheless, they 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 $v^P(\mathbf{x}) = v^Q(\mathbf{x})$ for all $\mathbf{x}$ belonging to the convex hull of X.

Note that in the context of Benders' decomposition, $v^P(\mathbf{x})$ and $v^Q(\mathbf{x})$ represent the linear programming subproblems when Benders' decomposition is applied to P and Q. Consequently, the two models are equivalent if their respective Benders' subproblems always have the same optimal value.

We evaluate the two models (5.5.2) and (5.5.3) by comparing the cuts generated from the application of Benders' decomposition. Following the derivation of Benders' decomposition given in Section 5.3, we can rewrite P and Q, respectively, as:*

$$(P) \quad \underset{\mathbf{x} \in X,\ z \in R}{\text{minimize}} \{z : z \geq \pi(\mathbf{h} - H\mathbf{x}) + \mathbf{cx} \ \text{ for all } \pi \in \Pi\}$$

where Π is the set of points in the polyhedron $\pi G \leq \mathbf{d}$; and

$$(Q) \quad \underset{\mathbf{x} \in X,\, z \in R}{\text{minimize}} \{z : z \geq \gamma(\mathbf{r} - R\mathbf{x}) + \mathbf{cx} \ \text{ for all } \gamma \in \Gamma\}$$

where Γ is the set of points in the polyhedron $\gamma T \leq \mathbf{t}$.

The inequalities $z \geq \pi(\mathbf{h} - H\mathbf{x}) + \mathbf{cx}$ and $z \geq \gamma(\mathbf{r} - R\mathbf{x}) + \mathbf{cx}$ 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

---

* As in earlier sections, we assume that the linear programming subproblems $v^P(\mathbf{x})$ and $v^Q(\mathbf{x})$ are feasible and have optimal solutions for all $\mathbf{x} \in X$. These restrictions can be relaxed, but with added complications that do not enrich the development in an essential way.

possible cuts than the usual definition which restricts the cuts to those corresponding to the extreme points of the subproblem dual feasible region. The results of this section are not always valid for the usual definition of Benders' cuts (see exercise 5.10). To compare equivalent model formulations, we adapt the concept of a pareto-optimal cut introduced in Section 5.3 by saying that a Benders' cut (or constraint) $z \geq \pi(\mathbf{h}-\mathbf{Hx}) + \mathbf{cx}$ for P dominates a Benders' cut $z \geq \gamma(\mathbf{r}-\mathbf{Rx}) + \mathbf{cx}$ for Q if $\pi(\mathbf{h}-\mathbf{Hx}) + \mathbf{cx} \geq \gamma(\mathbf{r}-\mathbf{Rx}) + \mathbf{cx}$ for all $\mathbf{x} \in X$ with a strict inequality for at least one point $\mathbf{x} \in X$.

A cut $z \geq \gamma(\mathbf{r}-\mathbf{Rx}) + \mathbf{cx}$ for Q will be called unmatched with respect to the formulation P if no cut for P is equal to it (in the sense that two cuts are equal if their righthand sides are equal for all $\mathbf{x} \in X$) or dominates it.

A formulation Q is "cut richer" than P if they are equivalent formulations and the set of Benders' cuts for P is a "proper subset" of the Benders' cuts for Q (i.e., some cut in Q equals or dominates each cut in P and Q has a cut that is unmatched in P).

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

Lemma 5.1 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 an $\mathbf{x}°$ belonging to the convex hull X^c of X that satisfies $v^Q(\mathbf{x}°) > v^P(\mathbf{x}°)$.

Proof: To establish the necessity of the inequality condition, let $z \geq \gamma^*(\mathbf{r}-\mathbf{Rx}) + \mathbf{cx}$ be a Benders' cut that is unmatched with respect to P. That is, for every cut $z \geq \pi(\mathbf{h}-\mathbf{Hx}) + \mathbf{cx}$ in P, there is an $\mathbf{x} \in X$ with $\pi(\mathbf{h}-\mathbf{Hx}) + \mathbf{cx} < \gamma^*(\mathbf{r}-\mathbf{Rx}) + \mathbf{cx}$. Since we are assuming that the set X is finite, this inequality implies that

$$\max_{\pi G \leq d} [\min_{x \in X} \pi(h-Hx) + cx - \gamma^*(r-Rx) - cx] < 0.$$

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

$$\min_{x \in X^c} [\max_{\pi G \leq d} \pi(h-Hx) + cx - \gamma^*(r-Rx) - cx] < 0.$$

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

$$\min_{x \in X^c, y \geq 0} \{cx+dy - [\gamma^*(r-Rx) + cx] : Hx + Gy = h\} < 0.$$

Now let $x^\circ \epsilon X^c$ be an optimal value for x in this problem. Then

$$\min_{y \geq 0} \{cx^o + dy : Gy = h-Hx^o\} = v^P(x^o) < \gamma^*(r-Rx^o) + cx^o.$$

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

$$v^P(x^o) < \gamma^*(r-Rx^o) + cx^o \leq \min_{y \geq 0} \{ cx^o + ty : Ty = r-Rx^o\}$$

or

$$v^P(x^o) < v^Q(x^o).$$

The sufficiency of this inequality condition has essentially the same proof with all the steps reversed. Explicit details are left as an exercise (see exercise 5.9). □

This lemma leads to the following theorem concerning preferred formulations:

Theorem 5.2: Let P and Q be equivalent formulations of a mixed integer programming problem. Q is cut richer than P if, and only if, $v^Q(x) \geq v^P(x)$ for all $x \epsilon X^c$ with a strict inequality for at least one $x \epsilon X^c$.

Proof: If $v^Q(\mathbf{x}) \geq v^P(\mathbf{x})$ for all $\mathbf{x} \varepsilon X^c$, Lemma 5.1 says that P does not have any Benders' cuts that are unmatched with respect to Q. But because there is a $\mathbf{x}^o \varepsilon X^c$ satisfying $v^Q(\mathbf{x}^o) > v^P(\mathbf{x}^o)$, Lemma 5.1 implies that Q has a cut that is unmatched in P. So Q satisfies the definition of being cut richer than P.

If Q is cut richer than P, then P, by definition of cut richer, does not have any cuts that are unmatched with respect to Q. Lemma 5.1 then tells us that $v^Q(\mathbf{x}) \geq v^P(\mathbf{x})$ for all $\mathbf{x} \varepsilon X^c$. The definition of cut richer also states that Q has a cut that is unmatched with respect to P and using Lemma 5.1 we know that some $\mathbf{x}^o \ \varepsilon X^c$ satisfies $v^Q(\mathbf{x}^o) > v^P(\mathbf{x}^o)$. □

The implications of Theorem 5.2 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 X by its convex hull X^c. Theorem 5.2 states that the preferred formulation of a mixed integer program for generating strong Benders' cuts is the one with the smallest possible feasible region (or the "tightest" possible constraint set) for its relaxed primal problem. For any formulation P, a smaller feasible region for its relaxed primal problem will result in larger values of the function $v^P(\mathbf{x})$; this property is desirable because of Lemma 5.1 and Theorem 5.2.

As an example, consider the p-median problem of Figure 5.4. Formulations P and Q differ only in that P has constraints of the form $y_{ij} \leq x_j$ for all (i,j), whereas Q has constraints of the form

$$\sum_{i=1}^{4} y_{ij} \leq 4x_j \quad \text{for all } j.$$

Since the latter set of constraints is an aggregation of the former set of constraints, the feasible region for the relaxed primal problem of P is no larger than that for Q. So $v^P(\mathbf{x}) \geq v^Q(\mathbf{x})$ for all $\mathbf{x} \varepsilon X^c$. A straightforward computation shows that $v^P(\mathbf{x}^o) = 200 > v^Q(\mathbf{x}^o) = 0$ for $\mathbf{x}^o = (1/2,1/2,1/2,1/2)$. So the formulation P is cut richer than Q for this example. The comparison of the cuts given at the beginning of this section dramatically illustrates the superiority of the Benders' cuts for formulation P.

As a general consequence of Theorem 5.2, for any mixed integer programming formulation, the convex hull of its feasible region will be a model formulation that is "optimal" for 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 (5.5.2), let C(P) denote the mixed integer program whose feasible region is the convex hull of the feasible region for P.

Theorem 5.3: Given any formulation P of a mixed integer program, $v^{C(P)}(\mathbf{x}) \geq v^Q(\mathbf{x})$ for all $\mathbf{x} \epsilon X^c$ and for all equivalent formulations Q of this problem.

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

Combining this theorem with Theorem 5.2 establishes the following result.

Corollary 5.1: Given any two equivalent formulations P and Q of a mixed integer program, the convex hull formulation C(P) of P is either cut richer than or identical to Q.

Another distinguishing feature 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:

$$v^{C(P)} = \min_{x \in X, y \geq 0} \{cx + dy : H_1x + G_1y = h_1\}.$$

Then we have the following result.

Theorem 5.4: For any formulation of the mixed integer program (5.5.2), the convex hull formulation requires only one Benders' cut for convergence.

Proof: $v^{C(P)} = \min_{x \in X} \min_{y \geq 0} \{cx + dy : H_1x + G_1y = h_1\}.$

Since C(P) is the convex hull formulation, we can substitute X^c for X without affecting the optimal solution value. Then, applying linear programming duality theory (and again assuming that $v^{C(P)}(x)$ is feasible for all $x \in X$), we have

$$v^{C(P)} = \min_{x \in X^c} \max_{uG_1 \leq d} u(h_1 - H_1x) + cx.$$

Another application of linear programming duality theory yields

$$v^{C(P)} = \min_{x \in X^c} \{u^*(h_1 - H_1 x) + cx\}$$

for some u^* satisfying $u^* G_1 \leq d$. Since the last objective function is a linear function of x, we can substitute X for its convex hull and write

$$v^{C(P)} = \min_{x \in X, z \in R} \{z : z \geq u^* (h_1 - H_1 x) + cx\}.$$

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

For facility location problems, our results show that a formulation with a reduced feasible region for the relaxed primal problem is desirable. However, there are other issues that must be considered in selecting a model for use with Benders' decomposition.

First, constructing alternative models for mixed integer programming problems can be a difficult task. 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 the constraints representing the convex hull of a set of points.

Recently, researchers have been successful in partially characterizing such constraints for the plant location problem (see Chapter 3). Section 5.7 cites a number of related studies.

computational experiments verified the utility of tight model formulations discussed earlier in Section 5.5. For several problems, using an aggregated (looser) formulation increased the number of Benders' iterations required by a factor of more than 10.

Benders' decomposition has also been used successfully in two other transportation applications: (i) constructing airline routes for long-haul passenger markets, and (ii) in railway planning, selecting the mix of engine types and scheduling the available engines to trains. The model for the first application chooses, for each given origin-destination city pair, a route plan consisting of intermediate stops and the number of passengers transported between pairs of cities on the route. The objective is to maximize revenue while honoring operational constraints such as the capacity of the aircraft. The mixed integer programming model for this problem contains a number of integer side constraints (i.e., constraints imposed upon the integer variables). The implementation of Benders' decomposition for this problem used the ε-optimal method for solving the master problem and an initial selection of cuts to accelerate convergence (see Section 5.3.2). For test problems with 40 to 160 0-1 variables (corresponding to routing problems over a 12 to 17-city network), the solution procedure required from 10 to 16 Benders' iterations and about 80 seconds per problem on a DEC-10 system. Comparative tests indicate that the initial selection of cuts substantially reduced the number of Benders' iterations.

In the railway planning application, the integer variables represent engine assignments and determine bounds on train movements that are modeled by a minimum cost flow subproblem. The model also contains a number of integer side constraints that represent engine requirements for each particular train. For a problem with 432 0-1 variables, 986 continuous flow variables and 1972 integer side constraints, Benders' decomposition required 9 iterations and 1500 seconds on a CYBER 74 computer to determine upper and lower bounds that differed by about 6.6%. For another version (with different costs) of the same model, the decomposition procedure was able to find upper and lower bounds only within 19.3% after 20 iterations and 3168 seconds of computation time. This study concluded that the procedure was suitable for solving moderate-sized, but not large-scale problems.

In the context of facility location, computational experiments performed with p-median problems have tested the effectiveness of the usual, closing facility, and expanding neighborhood cuts that we discussed in Section 5.4.3. When applied to 10 and 33 node problems, the expanding neighborhood cut performed slightly better than the closing facility cut and both cuts clearly outperformed the usual cut technique. Both cut implementations required at least two or three times fewer cuts than the usual cut implementation to achieve comparable levels of accuracy. These results indicate the relative utility of the improved Benders' cuts. However, the performance of all three cut types indicates that Benders' decomposition is not competitive for solving large-scale p-median problems (see Chapter 2). For example, a four-median problem on a 33-node network required ten Benders' iterations to compute bounds that were within 9% of each other. Also, for larger problems, all three cut types exhibited pronounced tailing effects, that is, the convergence of the bounds slowed considerably as the number of iterations increased.

The improved Benders' cut methodology has also been tested on a close relative of discrete location problems, namely, the uncapacitated fixed charge network design problem. (Splitting of nodes permits facility location problems to be converted into network design problems, and adding nodes to arcs permits network design problems to be converted to faclity location models.) In the network design setting, multiple commodities must be routed over a network with linear flow costs and with fixed charge costs on the arcs. The solution procedure compared three types of Benders' cuts including a usual cut, a "strong" cut that dominates the usual cut, and a pareto-optimal cut computed with the methodology described in Section 5.3. For ten moderate-sized test problems with 35 to 45 0-1 variables and 1800 to 3600 continuous variables, the pareto-optimal cuts required from 8 to 30 Benders' iterations to solve a problem to optimality. The computation time on a VAX 11/780 ranged from 7 to 700 seconds. The strong cuts performed almost as well, but the usual cuts were much less effective and exhibited severe tailing effects. On the average, the usual cut implementation required fifty times more computation time and four times as many iterations as the pareto-optimal implementation to solve a problem to optimality. For most of the test problems, the pareto-optimal implementation required fewer cuts than the strong implementation, though at the expense of larger computation time for generating each cut. Generating a pareto-optimal cut required solving the auxiliary problem (5.3.11) which, for the network design application, reduces to a series of minimum cost-flow problems. The strong cut required additional work equivalent to solving only a shortest-path problem. The overall computational results seemed to justify the additional complexity of using the pareto-optimal cuts.

These computational experiments also used a second set of 24 larger problems with 90 0-1 variables and from 10,000 to 15,000 continuous variables. For these problems, with all three cut types, Benders' decomposition encountered severe convergence problems and required excessive numbers of iterations and computation times. However, a preprocessing technique based upon a dual ascent procedure (see Chapter 2) was able to significantly improve the computational performance for these problems. This procedure used dual information to eliminate variables and to provide an initial feasible solution and an initial Benders' cut. For seven of the test problems, the preprocessing routine was usually able to eliminate 75 to 90% of the 0-1 variables. For these seventeen problems, the pareto-optimal and strong cuts both performed significantly better than the usual cuts. The strong cuts were generally more effective than the pareto-optimal cuts except that neither implementation converged after 7000 seconds for five difficult problems. For these problems, the pareto-optimal implementation produced slightly stronger lower bounds. In general, the pareto-optimal cuts strategy required more time to generate a cut than the strong-cut strategy. On the other hand, the pareto-optimal cuts seem to generate somewhat better cuts and this improvement seems to be more pronounced for the more difficult problems.

Finally, an implementation of cross decomposition (see Section 5.3.2) has been very successful in solving capacitated plant location problems. In one recent study, the procedure performed only one full cross-decomposition iteration and did not solve any Benders' master problems. It alternated between solving Benders' subproblems and Lagrangian subproblems while occasionally solving the master problem resulting from applying the relaxation algorithm to the Lagrangian dual problem (5.3.5). The method generated "strong" Benders' cuts that dominate the usual cuts derived from the transportation subproblems. Computational results indicate that the procedure is about ten times faster than other proposed techniques for the capacitated plant location. The cross-decomposition implementation has solved (computed upper and lower bounds within 0.12% of each other) a series of problems with 100 potential plant sites and 200 demand nodes in 6 to 400 seconds on an MV8000 computer.

The computation results discussed in this section indicate that Benders' decomposition and related techniques can be successful in solving location problems and other combinatorial optimization models. The acceleration techniques discussed in this chapter appear to be quite effective. The cut selection techniques (for strong and pareto-optimal cuts) can improve the performance of Benders' decomposition. Also, the selection of an appropriate mixed integer programming model is vital to the success of Benders' method. Other techniques such as the ε-optimal method for solving the master problem and making a good selection of initial cuts have also been quite useful in various applications.

However, these methods should be used with caution. For the multicommodity (Hunt-Wesson) location, airline route construction, and railway planning applications for which Benders' decomposition was fairly successful, the models were complicated by a large number of integer side constraints. The Benders' master problems inherited these side constraints and so in each application a specialized technique was used to exploit the unique structure of the master problem. We suspect that these tailored master problem algorithms contributed significantly to the successful use of Benders' decomposition. For the p-median problem, network design problem, and other optimization problems that are basically combinatorial, however, and do not have many complicating integer side constraints, Benders' decomposition by itself has been much less successful. The p-median problem, for example, has been solved more effectively with other decomposition techniques, most notably Lagrangian decomposition and dual ascent. Other problems have required that Benders' decomposition be combined with problem preprocessing or with Lagrangian decomposition and other decomposition schemes.

In summary, computational results support the effectiveness of Benders' decomposition and related techniques in solving discrete location problems. The techniques presented in this chapter can definitely enhance the performance of decomposition methods. However, for many problems it is also useful to exploit the problem structure of integer side constraints or combine Benders' decomposition with other solution methods.

5.7. NOTES AND REFERENCES

Section 5.2

Over twenty years ago, Benders (1962) first introduced his resource directive decomposition scheme which he cast in a setting of general mixed integer programming. Later, Geoffrion (1972) generalized the approach to solve nonlinear programming problems. Balinski and Wolfe (1963), Balinski (1965), Geoffrion and Graves (1974), and Magnanti and Wong (1981) discuss the application of Benders' decomposition to facility location problems. In particular, Geoffrion and Graves have been able to solve complicated large-scale location problems that arise in distribution system design.

Lagrangian relaxation techniques have a long and illustrious history and have, over the last two hundred years, become a staple of nonlinear programming. The use of this solution strategy for discrete optimization is much more recent, though, and dates to the seminal work of Held and Karp (1970, 1971) on the travelling salesman problem. Subsequently, the method has been applied to a vast range of combinatorial optimization problems (see Geoffrion, 1974; Shapiro, 1979; and Fisher, 1981).

Numerous researchers have used Lagrangian techniques to analyze location models including Cornuejols, Fisher, and Nemhauser (1977) for uncapacitated facility location; Geoffrion and McBride (1978), Nauss (1978), and Christofides and Beasley (1983) for capacitated facility location; Narula, Ogbu, and Samuelson (1977), Christofides and Beasley (1982), Weaver and Church (1983), and Mirchandani, Oudjit and Wong (1985) for the p-median problem and its variants; and Karkazis and Boffey (1981) for a multi-commodity location problem.

Guinard and Spielberg (1979) and Van Roy and Erlenkotter (1982) proposed other dual-based methods. Bitran, Chandra, Sempolinski and Shapiro (1981) proposed an inverse optimization approach using both Lagrangian and group theoretic techniques.

In an annotated bibliography, Wong (1985) discusses a number of other references in the facility location location literature. See also the survey by Tansel, Francis and Lowe (1983) and books by Francis and White (1984) and Handler and Mirchandani (1979).

Section 5.3

The minimax view of decomposition that we have adopted in this section is drawn from Magnanti (1976) and Magnanti and Wong (1981).

Because the general minimax problem (3.1) is typically nondifferentiable, a number of authors including Dem'yanov and Malozemov (1974), Lemarechal (1975), Mifflin (1977), and Wolfe (1975) have modified and extended differentiable optimization algorithms to solve these problems.

Dantzig (1963) and Magnanti, Shapiro and Wagner (1976) discuss the convergence properties of Dantzig-Wolfe decomposition. The second of these references stresses the relationship between Dantzig-Wolfe decomposition and Lagrangian duality in the context of general mathematical programs (including discrete optimization problems).

Mervert (1979), and Richardson (1976) have studied the effect of the initial selection of cuts on the performance of Benders' algorithm.

Nemhauser and Widhelm (1971), O'Neill and Widhelm (1972), Marsten, Hogan and Blankenship (1975), Marsten (1975), and Holloway (1973) have proposed alternatives for modifying the Dantzig-Wolfe decomposition master problem.

Van Roy (1983) introduced the cross-decomposition method. He has successfully applied the technique to solve large-scale capacitated plant location problems (see Van Roy, 1981 and 1986).

Geoffrion and Graves (1974) introduced the ε-optimality method when applying Benders' decomposition to distribution system facility location.

Historically, the study of equivalent formulations for a discrete optimization model seems to have originated with the location problems (see Davis and Ray, 1969). Beale and Tomlin (1972), Williams (1974), Jeroslow and Lowe (1984, 1985, Eppen and Martin (1985), and Martin and Schrage (1985) have studied equivalent formulations for various discrete optimization models. Cornuejols, Fisher and Nemhauser (1977) studied equivalent formulations for an uncapacitated location model.

Section 2 of Magnanti and Wong (1981) discusses Theorem 5.1 and related material. Rockefeller (1970) discusses the relative interior of convex sets in great detail.

<u>Section 5.4</u>

The pareto-optimal cut generation algorithm is taken from Section 3 of Magnanti and Wong (1981). The algorithm is similar to dual ascent procedures for the plant location problem proposed by Bilde and Krarup (1977) and Erlenkotter (1978). Balinski (1965) originally introduced the idea of the closing facility cut. In an unpublished report, Magnanti and Wong (1977) proposed the expanding neighborhood cut and gave computational results comparing the usual, closing, and expanding neighborhood cuts. Magnanti, Mireault and Wong (1986) discuss some computational results with the pareto-optimal Benders' cuts applied to a class of network design problems. Magnanti and Wong (1984) show that, when interpreted properly, several branch-and-bound and heuristic algorithms for network design problems use Benders' cuts for bounding the optimal objective function value. The cuts described in this section could be used in similar ways.

Section 5.5

Our discussion of the Benders' decomposition model selection criteria is taken from Section 4 of Magnanti and Wong (1981).

In recent years, the mathematical programming research community has intensively studied facets of the convex hull of discrete optimization problems (see Pulleyblank, 1983). Although facet inequalities have been used chiefly as cutting planes (see, for example, Groeschel and Padberg, 1971a and 1979b; Crowder and Padberg, 1980; Crowder Padberg and Johnson, 1983; Barany, Van Roy and Wolsey, 1984a and 1984b; Johnson, Kostreva and Suhl, 1985; and Groetschel, Junger and Reinalt, 1984 and 1985), they could also be used to derive alternative models for mixed integer programming problems. Several authors including Cornuejols, Fisher and Nemhauser (1977), Guignard (1980), Cornuejols and Thizy (1982a), Cho, Johnson, Padberg and Rao (1983), Cho, Padberg and Rao (1983), Van Roy and Wolsey (1985), Leung (1985) and Leung and Magnanti (1986) have studied valid inequalities and facets for plant location problems. Lemke (1986) and Lemke and Wong (1985) have studied facets for the p-median problem.

Section 5.6

The studies of the Hunt-Wesson Foods distribution system, the airline route selection problem, and the railway engine scheduling application by Benders' decomposition were conducted, respectively, by Geoffrion and Graves (1974), Richardson (1976), and Florian, Guerin and Bushel (1976).

Magnanti and Wong (1977) performed the computational results on using several types of cuts for the p-median problem. Magnanti, Mireault and Wong (1986) discuss the acceleration of Benders' decomposition for fixed charge network design problems. Van Roy (1986) used cross-decomposition to solve capacitated plant location problems.

Acknowledgment

We are grateful to Pitu Mirchandani and to Rita Vachani for many constructive comments on an earlier version of this paper.

EXERCISES

Exercise 5.1

Consider the following capacitated facility location problem:

$$\begin{aligned}
&\text{minimize} \quad \mathbf{cx} + \mathbf{dy} \\
&\text{subject to} \quad \sum_{j=1}^{n} y_{ij} = w_i \qquad i=1,\ldots,m \\
&\qquad \sum_{i=1}^{m} y_{ij} \le K_j x_j \qquad j=1,\ldots,n \\
&\qquad y_{ij} \ge 0 \qquad i=1,\ldots,m; \quad j=1,\ldots,n \\
&\qquad x_j = 0 \text{ or } 1 \quad j=1,\ldots,n.
\end{aligned}$$

(a) Suppose that we apply Lagrangian relaxation to this problem by associating Lagrange multipliers $\alpha_j \ge 0$ with the constraints $\sum_{i=1}^{m} y_{ij} \le K_j x_j$ for $j=1,\ldots,n$. Let $L(\alpha)$ denote the optimal value of the Lagrangian subproblem as a function of the Lagrange multipliers $\alpha = (\alpha_1,\ldots,\alpha_n)$ and let $d = \max_{\sigma \ge 0} L(\alpha)$. Show that d equals the value v_{LP} of the linear programming relaxation of the original problem (obtained by replacing $x_j = 0$ or 1 by $0 \le x_j \le 1$ for all j).

(b) Suppose that we define another Lagrangian relaxation by associating Lagrange multipliers λ_i with the constraints $\sum_{j=1}^{m} x_{ij} = w_i$ for $i=1,\ldots,m$. Let $\bar{L}(\lambda)$ denote the optimal value of the Lagrangian subproblem as a function of $\lambda = (\lambda_1,\ldots,\lambda_m)$. Also let $\bar{d} = \max_{\lambda} \bar{L}(\lambda)$ and let d be defined as in part a. Show that $\bar{d} \ge d$.

(c) Specify a numerical example to show that $\bar{d} > d$ is possible.

This exercise shows that some Lagrangian relaxations for solving locations problems might provide sharper lower bounds than other Lagrangian relaxations.

Exercise 5.2

Suppose that we add the valid inequalities

$$y_{ij} \leq \min\{w_i, K_j\} x_j \qquad i=1,\ldots,m; \; j=1,\ldots,n$$

to the formulation in Exercise 5.1 and define the two Lagrangian relaxations specified in parts a and b of that exercise.

(a) How do the Lagrangian relaxations defined after the addition of the new valid inequalities compare with those defined before the addition of those inequalities?

(b) Show how to solve the new Lagrangian relaxations.

Exercise 5.3

Outline the steps required to apply Benders' decomposition to the capacitated facility location problem formulated in Exercise 5.1. Show how to generate pareto-optimal cuts for this problem.

Exercise 5.4

Suppose that we compare the formulation given in Exercise 5.1 with the formulation that adds the constraints specified in Exercise 5.2 Is one formulation necessarily cut richer than the other?

Exercise 5.5

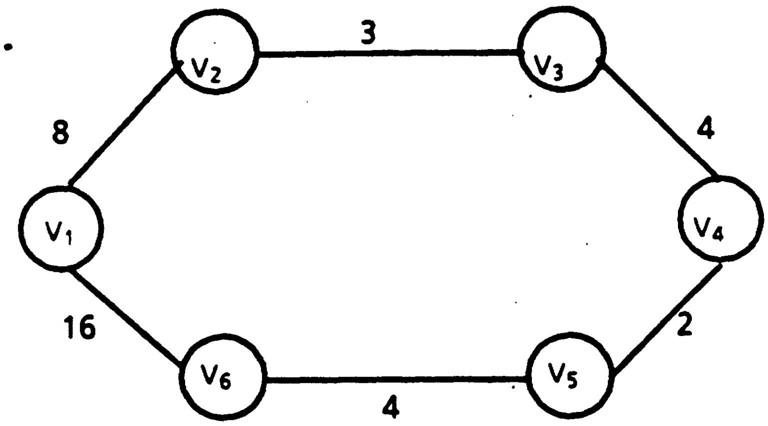


Figure 5.5 Three-Median Example

Consider a three-median problem defined on the network in Figure 5.5. The arc labels indicate the cost of traversing a particular link and the demands are assumed to be unity. The current configuration $\bar{x} = (0,1,0,0,1,1)$.

a) Compute the usual cut for the current configuration.

b) Compute the closing cut for the current configuration.

c) Compute a pareto-optimal cut for the current configuration using (1/8, 5/8, 1/8, 4/8, 6/8, 7/8) as the core point.

d) Compute the expanding neighborhood cut for the current configuration.

e) For every cut generated, determine which of the other cuts dominates it.

Exercise 5.6

Re-do Exercise 5.1 assuming that the problem is an uncapacitated facility location model. Let the current configuration be defined by $\bar{x} = (1,0,1,0,0,0)$ and assume that $c_1=c_2=c_3=5$ and $c_4=c_5=c_6=3$.

Exercise 5.7

Recall the two-median example described in Figure 5.1. Assume the current configuration is defined by $\bar{x} = (1,0,1,0,0)$.

a) Formulate the linear subproblem that results from (5.1.7)-(5.1.12) when we fix $x = \bar{x}$. Formulate the dual subproblem (the dual of the linear subproblem).

b) Compute the dual subproblem solution associated with the usual cut (5.4.5).

c) Compute the dual subproblem solution associated with the closing facility cut (5.4.7).

d) Compute the dual subproblem solution associated with the pareto-optimal cut generated by the core point (1/2, 1/4, 1/2, 1/4, 1/4).

e) Show that the dual solutions computed in parts b, c, and d are all optimal solutions for the dual subproblem.

f) Formulate the optimization problem (5.3.12) specialized to the two-median example of Figure 5.1. Use $x^o = (1/2, 1/4, 1/2, 1/4, 1/2)$.

g) Show that the dual solutions computed in parts b, c, and d are all feasible solutions for the optimization problem of part f. Compute the objective function values for the three solutions.

Notice that the objective function value of the dual solution of part d is the largest of the three objective function values. This result is expected since the dual solution of part d is an optimal solution for the optimization problem of part e.

Exercise 5.8

Show that each of the cuts given in Section 5.5 are required for Benders' decomposition to converge for the two-median problem shown in Figure 5.4.

Exercise 5.9

Prove the reverse implication of Lemma 5.1. That is, show that if there is an x^o belonging to the convex hull X^c of X that satisfies $v^Q(x^o) > v^P(x^o)$, then Q has a Benders' cut that is unmatched with respect to P.

Exercise 5.10

Consider the following pair of mixed integer programming formulations.

$$(P) \quad \underset{x \in \bar{X} = \{0,1,2\}}{\text{minimize}} \quad v^P(x)$$

$$\text{where } v^P(x) = \min \; y \quad \text{subject to} \quad y \geq 0 + x, \quad y \geq 2 + x, \quad y \geq 0$$

and

$$(Q) \quad \underset{x \in \bar{X}}{\text{minimize}} \; v^Q(x)$$

$$\text{where} \quad v^Q(x) = \min \quad y_1 + y_2$$

$$\text{subject to} \quad -y_1 + y_3 = \frac{1}{2} - x$$

$$y_2 - y_3 = \frac{3}{2} - x$$

$$y_1 \geq 0, \; y_2 \geq 0, \; y_3 \geq 0 \; .$$

a) Show that the two-integer programming formulations P and Q are equivalent.

b) Show that the two-integer programming formulations are not identical.

c) Assume that any point in the subproblem dual feasible region generates a Benders' cut. Show that Q cannot have a Benders' cut that is unmatched with respect to P. Also show that the formulation P is cut richer than the formulation Q.

d) Now assume that only the extreme points of the subproblem dual feasible region can generate a Benders' cut. Show that Q has a Benders' cut that is unmatched. Also, show that P is not cut richer than Q. (This exercise demonstrates that without the extended definition of a Benders' cut, given in part c, some of the results in Secton 5.5 might not be valid.)

Exercise 5.11

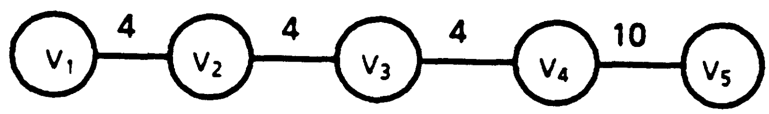


Figure 5.6 Two-Median Example

The network in Figure 5.6 describes a two-median problem: the arc labels indicate the cost of traversing a particular link and the demands are assumed to be unity. Let the current configuration be defined by $\bar{x} = (1,0,1,0,0)$.

a) Compute the expanding neighborhood cut (5.4.15) for the current configuration.

b) Show that the cut generated in part a is pareto-optimal. (Hint: compute the optimal dual objective function value $z_1(\mathbf{x})$ associated with expanding neighborhood cut. Let

$$\bar{X} = \{\mathbf{x}: \sum_{i=1}^{5} x_i = 2, \quad 0 \le x_i \le 1, \quad 1 \le i \le 5\}.$$

Prove that for any cut corresponding to another optimal dual solution with objective function value $z_2(\mathbf{x})$, there is a $\hat{\mathbf{x}} \epsilon \bar{X}$ satisfying $z_1(\hat{\mathbf{x}}) > z_2(\hat{\mathbf{x}})$.)

c) Prove that the expanding neighborhood cut generated in part a cannot be generated as a solution to the optimization exercise 5.3.11. That is, show that for any core point $\mathbf{x}^o \in ri(\bar{X}^c)$, the dual solution corresponding to the expanding neighborhood cut is not a solution for (5.3.11).

(This exercise demonstrates that some pareto-optimal cuts cannot be generated by solving (5.3.12). So although we are guaranteed that the solution to (5.3.12) generates a pareto-optimal cut, we cannot necessarily generate all pareto-optimal cuts with this technique.)

REFERENCES

Balinski, M.L. (1965). "Integer Programming: Methods, Uses, Computation," Management Science, 12, 253-313.

Barany, I., T.J. Van Roy, and L.A. Wolsey (1984a). "Uncapacitated Lot-Sizing: The Convex Hull of Solutions," Mathematical Programming Study 22, 32-43.

Barany, I., T.J. Van Roy, and L.A. Wolsey (1984b). "Multi-Item Capacitated Lot Sizing Problems. Management Science 30, 1255-1261.

Beale, E.M.L. and J.A. Tomlin (1972. "An Integer Programming Approach to a Class of Combinatorial Problems," Mathematical Programming, 3, 339-344.

Benders, J.F. 1962. "Partitioning Procedures for Solving Mixed Variables Programming Problems," Numerische Mathematik, 4, 238-252.

Bilde, O. and J. Krarup (1977). "Sharp Lower Bounds and Efficient Algorithms for the Simple Plant Location Problem," Annals of Discrete Mathematics, 1, 79-97.

Bitran, G. V. Chandru, D. Sempolinski, and J. Shapiro (1981). "Inverse Optimization: An Application to the Capacitated Plant Location Problem," Management Science, 27, 1120-1141.

Christofides, N. and J.E. Beasley (1982). "A Tree Search Algorithm for the P-Median Problem," European Journal of Operational Research, 10, 196-204.

Christofides, N. and J.E. Beasley (1982). "Extensions to a Lagrangian Relaxation Approach for the Capacitated Warehouse Location Problem," European Journal of Operational Research, 12, 19-28.

Cho, D,C., E.L. Johnson, M.W. Padberg, and M.R. Rao (1983). "On the Uncapacitated Plant Location Problem I: Valid Inequalities and Facets," Mathematics of Operations Research, 8, 579-589.

Cho, D.C., M.W. Padberg, and M.R. Rao (1983). "On the Uncapacitated Plant Location Problem II: Facets and Lifting Theorems," Mathematics of Operations Research, 8, 590-612.

Cornuejols, G., M.L, Fisher, and G.L. Nemhauser (1977a). "On the Uncapacitated Location Problem," Annals of Discrete Mathematics, 1, 163-177.

Cornuejols, G., M.L, Fisher, and G.L. Nemhauser (1977b). "Location of Bank Accounts to Optimize Float," Management Science, 23, 789-810.

Cornuejols, G., and J.M. Thizy (1982a). "Some Facets of the Simple Plant Location Polytope," Mathematical Programming, 23, 50-74.

Cornuejols, G., and J.M. Thizy (1982b). "A Primal Approach to the Simple Plant Location Problem," SIAM Journal of Algebraic Discrete Methods, 3, 504-510.

Crowder, H. and M.W. Padberg (1980). "Large Scale Symmetric Travelling Salesman Problems," Management Science, 26, 495-509.

Crowder, H., E.L. Johnson, and M.W. Padberg (1983). "Solving Large-Scale Zero-One Linear Programming Problems," Operations Research, 31, 803-834.

Dantzig, G.B. (1963). Linear Programming and Extensions, Princeton University Press, Princeton, New Jersey.

Davis, P.S. and T.L. Ray (1969). "A Branch-Bound Algorithm for Capacitated Facilities Location Problem," Naval Research Logistics Quarterly, 16, 331-344.

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

Eppen, G.D. and R.K. Martin (1985). "Solving Multi-Item Capacitated Lot Sizing Problems Using Variable Redefinition," Graduate School of Business, University of Chicago, Chicago.

Erlenkotter, D. (1978). "A Dual-Based Procedure for Uncapacitated Facility Location," Operations Research, 26, 992-1009.

Fisher, M. (1981). "Lagrangian Relaxation Method for Solving Integer Programming Problems," Management Science, 27, 1-18.

Florian, M., G. Guerin and G. Bushel (1976). "The Engine Scheduling Problem in a Railway Network," INFOR, 14, 121-138.

Francis, R. and J. White, (1974). Facility Layout and Location: An Analytic Approach, Prentice-Hall, Englewood Cliffs, NJ.

Geoffrion, A.M. (1974). "Lagrangian Relaxation and Its Uses in Integer Programming," Mathematical Programming Study 2, 82-114.

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

Geoffrion, A.M. and R. McBride (1978). "Lagrangian Relaxation Applied to Facility Location Problems," AIIE Transactions, 10, 40-47.

Groeschel, M. and M.W. Padberg (1979a). "On the Symmetric Travelling Salesman Problem I: Inequalities," Mathematical Programming, 16, 265-280.

Groeschel, M. and M.W. Padberg (1979b). "On the Symmetric Travelling Salesman Problem II: Lifting Theorems and Facets," Mathematical Programming, 16, 281-302.

Groeschel, M., M. Junger, and G. Reinelt (1984). "A Cutting Plane Algorithm for the Linear Ordering Problem," Operations Research, 31, 1195-1220.

Groeschel, M., M. Junger, and G. Reinelt (1985). "Facets for the Linear Ordering Problem," Mathematical Programming, 33, 43-60.

Guignard, M. (1980). "Fractional Vertices, Cuts and Facets of the Simple Plant Location Problem," Mathematical Programming Study 12, 150-162.

Guignard, M. and K. Spielberg (1979). "A Direct Dual Method for the Mixed Plant Location Problem with Some Side Constraints," Mathematical Programming, 17, 198-228.

Handler, G.Y. and P.B. Mirchandani (1979). Location on Networks: Theory and Algorithms, MIT Press, Cambridge, MA.

Held, M. and R.M. Karp (1970). "The Travelling Salesman Problem and Minimum Spanning Trees," Operations Research, 18, 1138-1162.

Held, M. and R.M. Karp (1971). "The Travelling Salesman Problem and Minimum Spanning Trees: Part II," Mathematical Programming, 1, 6-25.

Holloway, C. (1973). "A Generalized Approach to Dantzig-Wolfe Decomposition for Concave Programs," Operations Research, 21, 210-220.

Jeroslow, R.G. and J.K. Lowe (1985). "Experimental Results on the New Techniques for Integer Programming Formulations," Journal of the Operational Research Society, 36, 393-403.

Jeroslow, R.G. and J.K. Lowe (1981). "Modelling with Integer Variables," Mathematical Programming Study 22, 167-184.

Johnson, E.L., M.M. Kostreva, and U.H. Suhl (1985). "Solving 0-1 Integer Programming Problems Arising From Large Scale Planning Models," Operations Research, 33, 803-820.

Lemarechal, C. (1975). "An Extension of Davidon's Methods of Non-Differentiable Problems," Mathematical Programming Study 3, 95-103.

Lemke, P. (1986). "Characteristics of the K-Median Polyhedron," Doctoral Dissertation, Rensselaer Polytechnic Institute, Troy, NY (in preparation).

Lemke, P. and R.T. Wong (1985). "Valid Inequalities and Facets for the K-Median Location Problem," TIMS/ORSA Bulletin, 19, 170.

Leung, J.M.Y. (1985). "Polyhedral Structure of Capacitated Fixed Charge Problems and a Problem in Delivery Route Planning," Ph.D. Thesis, Department of Electrical Engineering and Computer Science, M.I.T.

Leung, J.M.Y. and T.L. Magnanti (1986). "Valid Inequalities and Facets for Capacitated Facility Location Problems," Working Paper OR 149-86, Operations Research Center, M.I.T., Cambridge, MA.

Magnanti, T.L. (1976). "Optimization for Sparse Systems," Sparse Matrix Computations, (J.R. Bunch and J.F. Rose, eds.), Academic Press, New York, 147-176.

Magnanti, T.L. and R.T. Wong (1977). "Network Design via Benders' Decomposition," unpublished manuscript, Massachusetts Institute of Technology.

Magnanti, T.L. and R.T. Wong (1981). "Accelerating Benders' Decomposition: Algorithmic Enhancement and Model Selection Criteria," Operations Research, 29, 464-484.

Magnanti, T.L. and R.T. Wong (1984). "Network Design and Transportation Planning: Models and Algorithms," Transportation Science, 18, 1-55.

Magnanti, T.L., P. Mireault and R.T. Wong (1986). "Tailoring Benders' Decomposition for Network Design," Mathematical Programming Study 26, 112-154.

Magnanti, T.L., J.F. Shapiro and M.H. Wagner (1976). "Generalized Linear Programming Solves the Dual," Management Science, 22, 1195-1203.

Marsten, R.E. (1975). "The Use of the BOXSTEP Method in Discrete Optimization," Mathematical Programming Study 3, 127-144.

Marsten, R.E., W.W. Hogan, and J.W. Blankenship (1975). "The BOXSTEP Method for Large-Scale Optimization," Operations Research, 23, 389-405.

Martin, K. and L. Schrage (1985). "Subset Coefficient Reduction Cuts for 0/1 Mixed Integer Programming," Operations Research, 33, 505-526.

Mevert, P. (1977). "Fixed Charge Network Flow Problems: Applications and Methods of Solution," presented at Large Scale and Hierarchical Systems Workship, Brussels, Belgium, May.

Mifflin, R. (1977). "An Algorithm for Constrained Optimization with Semismooth Functions," Mathematics of Operations Research, 2, 191-207.

Mirchandani, P.B., A. Oudjit, and R.T. Wong (1985). "Multidimensional Extensions and a Nested Dual Approach for the M-Median Problem," European Journal of Operational Research, 21, 121-137.

Narula, S.C., U.I. Ogbu, and H.M. Samuelsson (1977). "An Algorithm for the P-Median Problem," Operations Research, 25, 709-712.

Nauss, R.M. (1978). "An Improved Algorithm for the Capacitated Facility Locaton Problem," Journal of the Operational Research Society, 29, 1195-1201.

Nemhauser, G.L. and W.B. Widhelm (1971). "A Modified Linear Program for Columnar Methods in Mathematical Programming," Operations Research, 19, 1051-1060.

O'Neill, R.P., and W.B. Widhelm (1975). "Computational Experience with Normed and Non-normed Column-Generation Procedures in Nonlinear Programming," Operations Research, 23, 372-382.

Pulleyblank, W.R. (1983). "Polyhedral Combinatorics," Mathematical Programming: The State-of-the-Art, (A. Bachem, M. Groeschel, and B. Korte, eds.), Springer-Verlag.

Richardson, R. (1976). "An Optimization Approach to Routing Aircraft," Transportation Science, 10, 52-71.

Rockefellar, R.T. (1970). Convex Analysis, Princeton University Press, Princeton, NJ.

Shapiro, J.F. (1979). "A Survey of Lagrangian Techniques for Discrete Optimization," Annals of Discrete Mathematics, 5, 113-138.

Tansel, B.C., R.L. Francis, and T.J. Lowe (1983). "Location on Networks: A Survey, Parts I and II," Management Science, 29, 482-511.

Van Roy, T.J. (1986). "A Cross Decomposition Algorithm for Capacitated Facility Location," Operations Research, 34, 145-163.

Van Roy, T.J. (1981). "Cross Decomposition for Mixed Integer Programming with Applications to Facility Location," (J.P. Brans, ed.), Operational Research, North-Holland, Amsterdam, 579-587.

Van Roy, T.J. (1983). "Cross Decomposition for Mixed Integer Programming," Mathematical Programming, 25, 46-63.

Van Roy, T.J. and D. Erlenkotter (1982). "A Dual-Based Procedure for Dynamic Facility Location," Management Science, 28, 1091-1105.

Van Roy, T.J. and L.A. Wolsey (1985). "Valid Inequalities and Separation for Uncapacitated Fixed Charge Networks," Operations Research Letters, 4, 105-112.

Weaver, J.R. and R.L. Church (1983). "Computational Procedures for Location Problems on Stochastic Networks," Transportation Science, 17, 168-180.

Williams, H.P. (1974). "Experiments in the Formulation of Integer Programming Problems," Mathematical Programming Study 2, 180-197.

Wolfe, P. (1975). "A Method of Conjugate Subgradients for Minimizing Nondifferentiable Functions," Mathematical Programming Study, 3, 145-173.

Wong, R.T. (1985). "Location and Network Design," in Combinatorial Optimization: Annotated Bibliographies (M. O'hEigeartaigh, J.K. Lenstra and A.H.G. Rinnooy Kan eds.), New York, Wiley-Interscience, 127-147.