Large-Scale Systems

12

As mathematical-programming techniques and computer capabilities evolve, the spectrum of potential applications also broadens. Problems that previously were considered intractable, from a computational point of view, now become amenable to practical mathematical-programming solutions. Today, commercial linear-programming codes can solve general linear programs of about 4000 to 6000 constraints. Although this is an impressive accomplishment, many applied problems lead to formulations that greatly exceed this existing computational limit. Two approaches are available to deal with these types of problems.

One alternative, that we have discussed in Chapter 5, leads to the partitioning of the overall problem into manageable subproblems, which are linked by means of a hierarchical integrative system. An application of this approach was presented in Chapter 6, where two interactivelinear-programming models were designed to support strategic and tactical decisions in the aluminum industry, including resource acquisition, swapping contracts, inventory plans, transportation routes, production schedules and market-penetration strategies. This hierarchical method of attacking large problems is particularly effective when the underlying managerial process involves various decision makers, whose areas of concern can be represented by a specific part of the overall problem and whose decisions have to be coordinated within the framework of a hierarchical organization.

Some large-scale problems are not easily partitioned in this way. They present a monolithic structure that makes the interaction among the decision variables very hard to separate, and lead to situations wherein there is a single decision maker responsible for the actions to be taken, and where the optimal solution isvery sensitive to the overall variable interactions. Fortunately, these large-scale problems invariably contain special structure. The large-scale system approach is to treat the problem as a unit, devising specialized algorithms to exploit the structure of the problem. This alternative will be explored in this chapter, where two of the most important large-scale programming procedures—decomposition and column generation—will be examined.

The idea of taking computational advantage of the special structure of a specific problem to develop an efficient algorithm is not new. The upper-bounding technique introduced in Chapter 2, the revised simplex method presented in Appendix B, and the network-solution procedures discussed in Chapter 8 all illustrate this point. This chapter further extends these ideas.

12.1 LARGE-SCALE PROBLEMS

Certain structural forms of large-scale problems reappear frequently in applications, and large-scale systems theory concentrates on the analysis of these problems. In this context, structure means the pattern of zero and nonzero coefficients in the constraints; the most important such patterns are depicted in Fig. 12.8. The first illustration represents a problem composed of independent subsystems. It can be written as:

$$\text{Minimize } \sum_{j=1}^{r} c_j x_j + \sum_{j=r+1}^{s} c_j x_j + \sum_{j=s+1}^{n} c_j x_j,$$

subject to:

$$
\begin{aligned}
&\sum_{j=1}^{r} a_{ij}x_j &&= b_i \quad (i = 1, 2, \ldots, t),\\
&\qquad\sum_{j=r+1}^{s} a_{ij}x_j &&= b_i \quad (i = t+1, t+2, \ldots, u),\\
&\qquad\qquad\sum_{j=s+1}^{n} a_{ij}x_j &&= b_i \quad (i = u+1, u+2, \ldots, m),\\
&&x_j &\geq 0 \quad (j = 1, 2, \ldots, n).
\end{aligned}
$$

Observe that the variables $x_1, x_2, \ldots, x_r$, the variables $x_{r+1}, x_{r+2}, \ldots, x_s$, and the variables $x_{s+1}, x_{s+2}, \ldots, x_n$ do not appear in common constraints. Consequently, these variables are independent, and the problem can be approached by solving one problem in the variables $x_1, x_2, \ldots, x_r$, another in the variables $x_{r+1}, x_{r+2}, \ldots, x_s$ and a third in the variables $x_{s+1}, x_{s+2}, \ldots, x_n$. This separation into smaller and independent subproblems has several important implications.

First, it provides significant computational savings, since the computations for linear programs are quite sensitive to m, the number of constraints, in practice growing proportionally to m^3. If each subproblem above contains $\frac{1}{3}$ of the constraints, then the solution to each subproblem requires on the order of $(m/3)^3 = m^3/27$ computations. All three subproblems then require about $3(m^3/27) = m^3/9$ computations, or approximately $\frac{1}{9}$ the amount for an m-constraint problem without structure. If the number of subsystems were k, the calculations would be only $1/k^2$ times those required for an unstructured problem of comparable size.

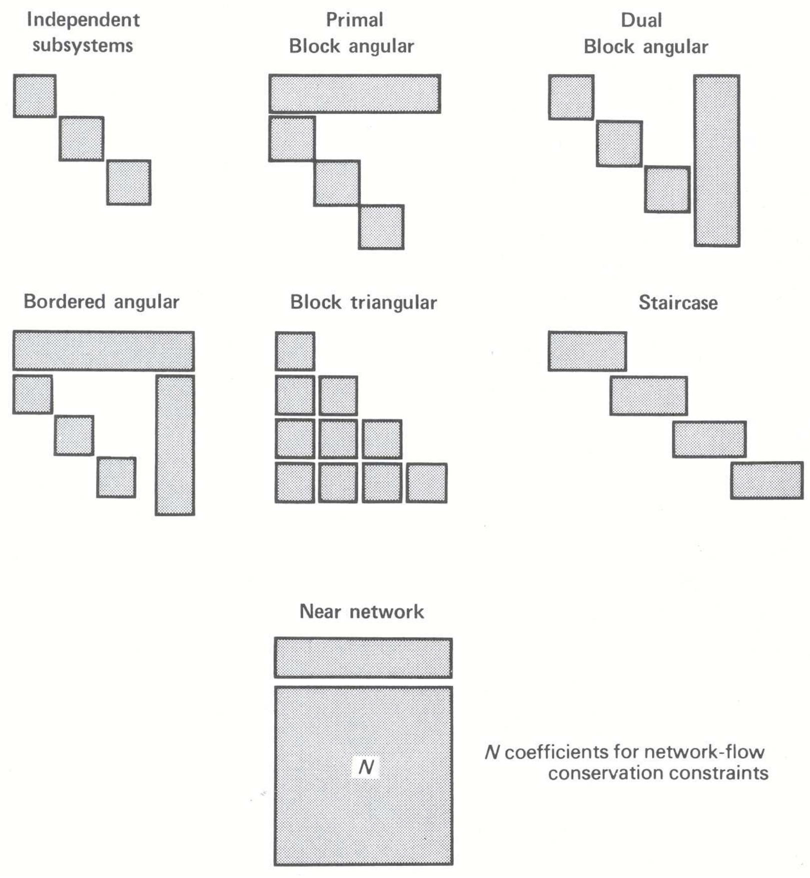

Figure 12.1

Second, each of the independent subproblems can be treated separately. Data can be gathered, analyzed, and stored separately. The problem also can be solved separately and, in fact, simultaneously. Each of these features suggests problems composed solely with independent subsystems as the most appealing of the structured problems.

The most natural extensions of this model are to problems with *nearly* independent subsystems, as illustrated by the next three structures in Fig. 12.8. In the primal block angular structure, the subsystem variables appear together, sharing common resources in the uppermost ‘‘coupling’’ constraints. For example, the subsystems might interact via a corporate budgetary constraint specifying that *total* capital expenditures of all subsystems cannot exceed available corporate resources.

The dual block angular structure introduces complicating ‘‘coupling’’ variables. In this case, the subsystems interact only by engaging in some common activities. For example, a number of otherwise independent subsidiaries of a company might join together in pollution-abatement activities that utilize some resources from each of the subsidiaries.

The bordered angular system generalizes these models by including complications from both coupling variables and coupling constraints. To solve any of these problems, we would like to decompose the system, removing the complicating variables or constraints, to reduce the problem to one with independent subsystems. Several of the techniques in large-scale system theory can be given this interpretation.

Dynamic models, in the sense of multistage optimization, provide another major source of large-scale problems. In dynamic settings, decisions must be made at several points in time, e.g., weekly ormonthly. Usually decisions made in any time period have an impact upon other time periods, so that, even when every instantaneous problem is small, timing effects compound the decision process and produce large, frequently *extremely* large, optimization problems. The staircase and block triangular structures of Fig. 12.8 are common forms for these problems. In the staircase system, some activities, such as holding of inventory, couple succeeding time periods. In the block triangular case, decisions in each time period can directly affect resource allocation in any future time period.

The last structure in Fig. 12.8 concerns problems with large network subsystems. In these situations, we would like to exploit the special characteristics of network problems.

It should be emphasized that the special structures introduced here do not exhaust all possibilities. Other special structures, like Leontief systems arising in economic planning, could be added. Rather, the examples given are simply types of problems that arise frequently in applications. To develop a feeling for potential applications, let us consider a few examples.

Multi-Item Production Scheduling

Many industries must schedule production and inventory for a large number of products over several periods of time, subject to constraints imposed by limited resources. These problems can be cast as large-scale programs as follows. Let

$$\theta_{jk} = \begin{cases} 1 & \text{if the } k\text{th production schedule is used for item } j, \\ 0 & \text{otherwise;} \end{cases}$$

$$K_j = \text{Number of possible schedules for item } j.$$

Each production schedule specifies how item j is to be produced in each time period $t = 1, 2, \ldots, T$; for example, ten items in period 1 on machine 2, fifteen items in period 2 on machine 4, and so forth. The schedules must be designed so that production plus available inventory in each period is sufficient to satisfy the (known) demand for the items in that period. Usually it is not mandatory to consider every potential production schedule; under common hypotheses, it is known that at most 2^{T-1} schedules must be considered for each item in a T-period problem. Of course, this number can lead to enormous problems; for example, with $J = 100$ items and $T = 12$ time periods, the total number of schedules (θ_{jk} variables) will be $100 \times 2^{11} = 204,800$.

Next, we let

c_{jk} = Cost of the kth schedule for item j (inventory plus production cost, including machine setup costs for production),

b_i = Availability of resource i $(i = 1, 2, \ldots, m)$,

a^i_{jk} = Consumption of resource i in the kth production plan for item j.

The resources might include available machine hours or labor skills, as well as cash-flow availability. We also can distinguish among resource availabilities in each time period; e.g., b_1 and b_2 might be the supply of a certain labor skill in the first and second time periods, respectively.

The formulation becomes:

$$\text{Minimize } \sum_{k=1}^{K_1} c_{1k}\theta_{1k} + \sum_{k=1}^{K_2} c_{2k}\theta_{2k} + \cdots + \sum_{k=1}^{K_J} c_{Jk}\theta_{Jk},$$

subject to:

$$\begin{aligned}
&\sum_{k=1}^{K_1} a^i_{1k}\theta_{1k} + \sum_{k=1}^{K_2} a^i_{2k}\theta_{2k} + \cdots + \sum_{k=1}^{K_J} a^i_{Jk}\theta_{Jk} \le b_i \qquad (i = 1, 2, \ldots, m),\\
&\left.\begin{array}{llll}
\sum_{k=1}^{K_1}\theta_{1k} & & & = 1\\
& \sum_{k=1}^{K_2}\theta_{2k} & & = 1\\
& & \ddots & \vdots\\
& & \sum_{k=1}^{K_J}\theta_{Jk} & = 1
\end{array}\right\} J \text{ constraints}\\
&\theta_{jk} \ge 0 \quad \text{and integer}, \qquad \text{all } j \text{ and } k.
\end{aligned}$$

The J equality restrictions, which usually comprise most of the constraints, state that exactly one schedule must be selected for each item. Note that any basis for this problem contains $(m + J)$ variables, and at least one basic variable must appear in each of the last J constraints. The remaining m basic variables can appear in no more than m of these constraints. When m is much smaller than J, this implies that most (at least $J - m$) of the last constraints contain one basic variable whose value must be 1. Therefore, most variables will be integral in any linear-programming basis. In practice, then, the integrality restrictions on the variables will be dropped to obtain an approximate solution by linear programming. Observe that the problem has a block angular structure. It is approached conveniently by either the decomposition procedure discussed in this chapter or a technique referred to as generalized upper bounding (GUB), which is available on many commercial mathematical-programming systems.

Exercises 11–13 at the end of this chapter discuss this multi-item production scheduling model in more detail.

Multicommodity Flow

Communication systems such as telephone systems or national computer networks must schedule message transmission over communication links with limited capacity. Let us assume that there are K types of messages to be transmitted and that each type is to be transmitted from its source to a certain destination. For example, a particular computer program might have to be sent to a computer with certain running or storage capabilities.

The communication network includes sending stations, receiving stations, and relay stations. For notation, let:

x^k_{ij} = Number of messages of type k transmitted along the communication link from station i to station j,

u_{ij} = Message capacity for link i, j,

c^k_{ij} = Per-unit cost for sending a type-k message along link i, j,

b^k_i = Net messages of type k generated at station i.

In this context, $b^k_i < 0$ indicates that i is a receiving station for type-k messages; $(-b^k_i) > 0$ then is the number of type-k messages that this station will process; $b^k_i = 0$ for relay stations. The formulation is:

$$\text{Minimize } \sum_{i,j} c^1_{ij} x^1_{ij} + \sum_{i,j} c^2_{ij} x^2_{ij} + \cdots + \sum_{i,j} c^K_{ij} \; x^K_{ij},$$

subject to:

$$\begin{array}{lllll}
x^1_{ij} + x^2_{ij} & + \cdots + & & x^K_{ij} & \leq u_{ij}, \text{ all } i, j \\
\left(\sum_j x^1_{ij} - \sum_r x^1_{ri}\right) & & & & = b^1_i, \text{ all } i \\
& \left(\sum_j x^2_{ij} - \sum_r x^2_{ri}\right) & & & = b^2_i, \text{ all } i \\
& & \ddots & \vdots & \vdots \\
& & & \left(\sum_j x^K_{ij} - \sum_r x^K_{ri}\right) & = b^K_i, \text{ all } i \\
& & & x^k_{ij} & \geq 0, \quad \text{all } i, j, k.
\end{array}$$

The summations in each term are made only over indices that correspond to arcs in the underlying network. The first constraints specify the transmission capacities u_{ij} for the links. The remaining constraints give flow balances at the communication stations i. For each fixed k, they state that the total messages of type k sent from station i must equal the number received at that station plus the number generated there. Since these are network-flow constraints, the model combines both the block angular and the near-network structures.

There are a number of other applications for this multicommodity-flow model. For example, the messages can be replaced by goods in an import–export model. Stations then correspond to cities and, in particular, include airport facilities and ship ports. In a traffic-assignment model, as another example, vehicles replace messages and roadways replace communication links. A numerical example of the multicommodity-flow problem is solved in Section 12.5.

Economic Development

Economic systems convert resources in the form of goods and services into output resources, which are other goods and services. Assume that we wish to plan the economy to consume b^t_i units of resource i at time t ($i = 1, 2, \ldots, m$; $t = 1, 2, \ldots, T$). The b^t_i's specify a desired consumptionschedule. We also assume that there are n production (service) activities for resource conversion, which are to be produced to meet the consumption schedule. Let

x_j = Level of activity j,
a^t_{ij} = Number of units of resource i that activity j ''produces'' at time t per unit of the activity level.

By convention, $a^t_{ij} < 0$ means that activity j consumes resource i at time t in its production of another resource; this consumption is internal to the production process and does not count toward the b^t_i desired by the ultimate consumers. For example, if $a^1_{1j} = -2$, $a^1_{2j} = -3$, and $a^1_{3j} = 1$, it takes 2 and 3 units of goods one and two, respectively, to produce 1 unit of good three in the first time period.

It is common to assume that activities are defined so that each produces exactly one output; that is, for each j, $a^t_{ij} > 0$ forone combination of i and t. An activity that produces output in period t is assumed to utilize input resources only from the current or previous periods (for example, to produce at time t we may have to train workers at time $t - 1$, ''consuming'' a particular skill from the labor market during the previous period). If J_t are the activities that produce an output in period t and j is an activity from J_t, then the last assumption states that $a^\tau_{ij} = 0$ whenever $\tau > t$. The feasible region is specified by the following linear constraints:

$$
\begin{array}{rcl}
\displaystyle\sum_{j \text{ in } J_1} a^1_{ij}x_j + \sum_{j \text{ in } J_2} a^1_{ij}x_j + \sum_{j \text{ in } J_3} a^1_{ij}x_j + \cdots + \sum_{j \text{ in } J_T} a^1_{ij}x_j &=& b^1_i \quad (i = 1, 2, \ldots, m),\\
\displaystyle\sum_{j \text{ in } J_2} a^2_{ij}x_j + \sum_{j \text{ in } J_3} a^2_{ij}x_j + \cdots + \sum_{j \text{ in } J_T} a^2_{ij}x_j &=& b^2_i \quad (i = 1, 2, \ldots, m),\\
\displaystyle\sum_{j \text{ in } J_3} a^3_{ij}x_j + \cdots + \sum_{j \text{ in } J_T} a^3_{ij}x_j &=& b^3_i \quad (i = 1, 2, \ldots, m),\\
\ddots \qquad & \vdots & \qquad \vdots\\
\displaystyle\sum_{j \text{ in } J_T} a^T_{ij}x_j &=& b^T_i \quad (i = 1, 2, \ldots, m),\\
x_j &\geq& 0 \quad (j = 1, 2, \ldots, n).
\end{array}
$$

One problem in this context is to see if a feasible plan exists and to find it by linear programming. Another possibility is to specify one important resource, such as labor, and to

$$\text{Minimize } \sum_{j=1}^{n} c_j x_j,$$

where c_j is the per-unit consumption of labor for activity j.

In either case, the problem is a large-scale linear program with triangular structure. The additional feature that each variable x_j has a po sitive coefficient in exactly one constraint (i.e., it produces exactly one output) can be used to devise a special algorithm that solves the problem as several small linear programs, one at each point in time $t = 1, 2, \ldots, T$.

12.2 DECOMPOSITION METHOD—A PREVIEW

Several large-scale problems including any with block angular or near-network structure become much easier to solve when some of their constraints are removed. The decomposition method is one way to approach these problems. It essentially considers the problem in two parts, one with the ''easy'' constraints and one with the ''complicating'' constraints. It uses the shadow prices of the second problem to specify resource prices to be used in the first problem. This leads to interesting economic interpretations, and the method has had an important influence upon mathematical economics. It also has provided a theoretical basis for discussing the coordination of decentralized organization units, and for addressing the issue of transfer prices among such units.

This section will introduce the algorithm and motivate its use by solving a small problem. Following sections will discuss the algorithm formally, introduce both geometric and economic interpretations, and develop the underlying theory.

Consider a problem with bounded variables and a single resource constraint:

$$\text{Maximize } z = 4x_1 + x_2 + 6x_3,$$

subject to:

$$\begin{array}{rcrcrcl}
3x_1 & + & 2x_2 & + & 4x_3 & \le & 17 \quad (\text{Resource constraint}), \\
\hline
x_1 & & & & & \le & 2, \\
 & & x_2 & & & \le & 2, \\
 & & & & x_3 & \le & 2, \\
x_1 & & & & & \ge & 1, \\
 & & x_2 & & & \ge & 1, \\
 & & & & x_3 & \ge & 1.
\end{array}$$

We will use the problem in this section to illustrate the decomposition procedure, though in practice it would be solved by bounded-variable techniques.

First, note that the resource constraint complicates the problem. Without it, the problem is solved trivially as the objective function is maximized by choosing x_1, x_2, and x_3 as large as possible, so that the solution $x_1 = 2$, $x_2 = 2$, and $x_3 = 2$ is optimal.

In general, given any objective function, the problem

$$\text{Maximize } c_1x_1 + c_2x_2 + c_3x_3,$$

subject to:

$$\begin{array}{ccccl}
x_1 & & & \le & 2, \\
 & x_2 & & \le & 2, \\
 & & x_3 & \le & 2, \\
x_1 & & & \ge & 1, \\
 & x_2 & & \ge & 1, \\
 & & x_3 & \ge & 1,
\end{array} \tag{1}$$

is also trivial to solve: One solution is to set x_1, x_2, or x_3 to 2 if its objective coefficient is positive, or to 1 if its objective coefficient is nonpositive.

Problem (1) contains some but not all of the original constraints and is referred to as a *subproblem* of the original problem. Any feasible solution to the subproblem potentially can be a solution to the original problem, and accordingly may be called a subproblem *proposal*. Suppose that we are given two subproblem proposals and that we combine them with weights as in Table 12.1.

Observe that if the weights are nonnegative and sum to 1, then the weighted proposal also satisfies the subproblem constraints and is also a proposal. We can ask for those weights that make this composite proposal best for the overall problem, by solving the optimization problem:

$$\text{Maximize} z = 22\lambda_1 + 17\lambda_2, \qquad \begin{array}{l}\textit{Optimal} \\ \textit{shadow} \\ \textit{prices} \\ \hline\end{array}$$

subject to:

$$\begin{aligned} 18\lambda_1 + 13\lambda_2 &\le 17, \qquad 1\\ \lambda_1 + \lambda_2 &= 1, \qquad 4 \end{aligned}$$

$$\lambda_1 \ge 0, \qquad \lambda_2 \ge 0.$$

The first constraint states that the composite proposal should satisfy the resource limitation. The remaining constraints define λ_1 and λ_2 as weights. The linear-programming solution to this problem has $\lambda_1 = \frac{4}{5}$, $\lambda_2 = \frac{1}{5}$, and $z = 21$.

We next consider the effect of introducing any new proposal to be weighted with the two above. Assuming that each unit of this proposal contributes p_1 to the objective function and uses r_1 units of the resource, we have the modified problem:

$$\text{Maximize } 22\lambda_1 + 17\lambda_2 + p_1\lambda_3,$$

subject to:

$$\begin{aligned} 18\lambda_1 + 13\lambda_2 + r_1\lambda_3 &\le 17,\\ \lambda_1 + \lambda_2 + \lambda_3 &= 1, \end{aligned}$$

$$\lambda_1 \ge 0, \qquad \lambda_2 \ge 0, \qquad \lambda_3 \ge 0.$$

To discover whether any new proposal would aid the maximization, we price out the general new activity to determine its reduced cost coefficient $\bar{p}_1$. In this case, applying the shadow prices gives:

$$\bar{p}_1 = p_1 - (1)r_1 - (4)1. \tag{2}$$

Note that we must specify the new proposal by giving numerical values to p_1 and r_1 before $\bar{p}_1$ can be determined.

By the simplex optimality criterion, the weighting problem cannot be improved if $\bar{p}_1 \le 0$ for every new proposal that the subproblem can submit. Moreover, if $\bar{p}_1 > 0$, then the proposal that gives $\bar{p}_1$ improves the objective value. We can check both conditions by solving max $\bar{p}_1$ over all potential proposals.

Recall, from the original problem statement, that, for any proposal x_1, x_2, x_3,

$$p_1 \text{ is given by } 4x_1 + x_2 + 6x_3, \tag{3}$$

and

$$r_1 \text{ is given by } 3x_1 + 2x_2 + 4x_3.$$

Substituting in (2), we obtain:

$$\begin{aligned} \bar{p}_1 &= (4x_1 + x_2 + 6x_3) - 1(3x_1 + 2x_2 + 4x_3) - 4(1),\\ &= x_1 - x_2 + 2x_3 - 4, \end{aligned}$$

and

$$\text{Max } \bar{p}_1 = \text{Max } (x_1 - x_2 + 2x_3 - 4).$$

Checking potential proposals by using this objective in the subproblem, we find that the solution is $x_1 = 2$, $x_2 = 1$, $x_3 = 2$, and $\bar{p}_1 = (2) - (1) + 2(2) - 4 = 1 > 0$. Equation (3) gives $p_1 = 21$ and $r_1 = 16$.

Table 12.1 Weighting subproblem proposals.

	Activity levels			*Resource usage*	*Objective value*	*Weights*
	x_1	x_2	x_3			
Proposal 1	2	2	2	18	22	λ_1
Proposal 2	1	1	2	13	17	λ_2
Weighted proposal	$2\lambda_1 + \lambda_2$	$2\lambda_1 + \lambda_2$	$2\lambda_1 + 2\lambda_2$	$18\lambda_1 + 13\lambda_2$	$22\lambda_1 + 17\lambda_2$	

Consequently, the new proposal is useful, and the weighting problem becomes:

$$
\begin{array}{lll}
\text{Maximize } z = 22\lambda_1 + 17\lambda_2 + 21\lambda_3, & & \textit{Optimal shadow prices} \\
\text{subject to:} & & \\
18\lambda_1 + 13\lambda_2 + 16\lambda_3 \leq 17, & & \frac{1}{2} \\
\lambda_1 + \lambda_2 + \lambda_3 = 1, & & 13 \\
\lambda_1 \geq 0, \quad \lambda_2 \geq 0, \quad \lambda_3 \geq 0. & &
\end{array}
\tag{4}
$$

The solution is $\lambda_1 = \lambda_3 = \frac{1}{2}$, and z has increased from 21 to $21\frac{1}{2}$. Introducing a new proposal with contribution p_2, resource usage r_2, and weight λ_4, we now may repeat the same procedure. Using the new shadow prices and pricing out this proposal to determine its reduced cost, we find that:

$$
\begin{aligned}
\bar{p}_2 &= p_2 - \tfrac{1}{2}r_2 - 13(1), \\
&= (4x_1 + x_2 + 6x_3) - \tfrac{1}{2}(3x_1 + 2x_2 + 4x_3) - 13, \\
&= \tfrac{5}{2}x_1 + 4x_3 - 13.
\end{aligned}
\tag{5}
$$

Solving the subproblem again, but now with expression (5) as an objective function, gives $x_1 = 2, x_2 = 1, x_3 = 2$, and

$$\bar{p}_2 = \tfrac{5}{2}(2) + 4(2) - 13 = 0.$$

Consequently, no new proposal improves the current solution to the weighting problem (4). The optimal solution to the overall problem is given by weighting the first and third proposals each by $\frac{1}{2}$; see Table 12.2.

Table 12.2 Optimal weighting of proposals.

	Activity levels			*Resource*	*Objective*	
	x_1	x_2	x_3	*usage*	*value*	*Weights*
Proposal 1	2	2	2	18	22	$\frac{1}{2}$
Proposal 3	2	1	2	16	21	$\frac{1}{2}$
Optimal solution	2	$\frac{3}{2}$	2	17	$21\frac{1}{2}$	

The algorithm determines an optimal solution by successively generating a new proposal from the subproblem at each iteration, and then finding weights that maximize the objective function among all combinations of the proposals generated thus far. Each proposal is an extreme point of the subproblem feasible region; because this region contains a finite number of extreme points, at most a finite number of subproblem solutions will be required.

The following sections discuss the algorithm more fully in terms of its geometry, formal theory, and economic interpretation.

12.3 GEOMETRICAL INTERPRETATION OF DECOMPOSITION

The geometry of the decomposition procedure can be illustrated by the problem solved in the previous section:

$$\text{Maximize } 4x_1 + x_2 + 6x_3,$$

subject to:

$$
\begin{array}{ll}
3x_1 + 2x_2 + 4x_3 \leq 17, & \text{(Complicating resource constraint)} \\
\hline
1 \leq x_j \leq 2 \ (j = 1, 2, 3). & \text{(Subproblem)}
\end{array}
$$

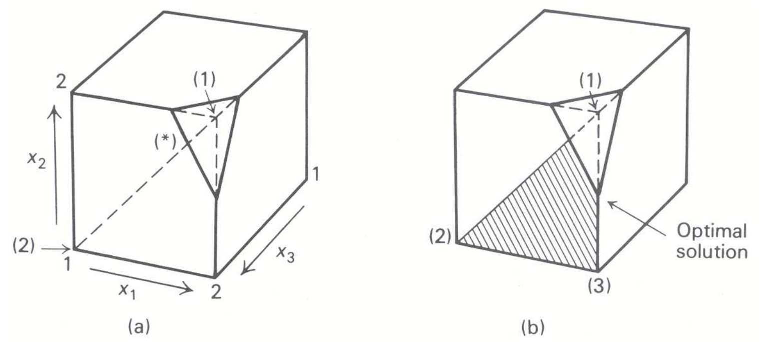

Figure 12.2 Geometry of the decomposition method. (a) First approximation to feasible region; (b) final approximation to feasible region.

The feasible region is plotted in Fig. 12.2.

The feasible region to the subproblem is the cube $1 \leq x_j \leq 2$, and the resource constraint

$$3x_1 + 2x_2 + 4x_3 \leq 17$$

cuts away one corner from this cube.

The decomposition solution in Section 12.2 started with proposals (1) and (2) indicated in Fig. 12.2(a). Note that proposal (1) is not feasible since it violates the resource constraint. The initial weighting problem considers all combinations of proposals (1) and (2); these combinations correspond to the line segment joining points (1) and (2). The solution lies at $(*)$ on the intersection of this line segment and the resource constraint.

Using the shadow prices from the weighting problem, the subproblem next generates the proposal (3). The new weighting problem considers all weighted combinations of (1), (2) and (3). These combinations correspond to the triangle determined by these points, as depicted in Fig. 12.2(b). The optimal solution lies on the midpoint of the line segment joining (1) and (3), or at the point $x_1 = 2$, $x_2 = \frac{3}{2}$, and $x_3 = 2$. Solving the subproblem indicates that no proposal can improve upon this point and so it is optimal.

Note that the first solution to the weighting problem at $(*)$ is not an extreme point of the feasible region. This is a general characteristic of the decomposition algorithm that distinguishes it from the simplex method. In most applications, the method will consider many nonextreme points while progressing toward the optimal solution.

Also observe that the weighting problem approximates the feasible region of the overall problem. As more subproblem proposals are added, the approximation improves by including more of the feasible region. The efficiency of the method is predicated on solving the problem before the approximation becomes too fine and many proposals are generated. In practice, the algorithm usually develops a fairly good approximation quickly, but then expends considerable effort refining it. Consequently, when decomposition is applied, the objective value usually increases rapidly and then ''tails off'' by approaching the optimal objective value very slowly. This phenomenon is illustrated in Fig. 12.3 which plots the progress of the objective function for a typical application of the decomposition method.

Fortunately, as discussed in Section 12.7, one feature of the decomposition algorithm is that it provides an upper bound on the value of the objective function at each iteration (see Fig. 12.3). As a result, the procedure can be terminated, prior to finding an optimal solution, with a conservative estimate of how far the current value of the objective function can be from its optimal value. In practice, since the convergence of the algorithm has proved to be slow in the final stages, such a termination procedure is employed fairly often.

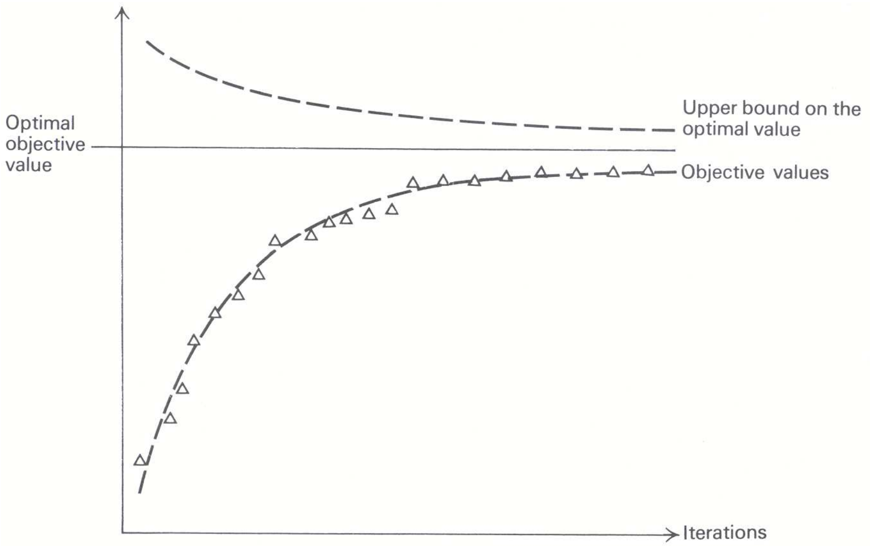

Figure 12.3 Objective progress for a typical application of decomposition.

12.4 THE DECOMPOSITION ALGORITHM

This section formalizes the decomposition algorithm, discusses implementation issues, and introduces a variation of the method applicable to primal block angular problems.

Formal Algorithm

Decomposition is applied to problems with the following structure.

$$\text{Maximize } z = c_1x_1 + c_2x_2 + \cdots + c_nx_n,$$

subject to:

$$\left.\begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ \vdots \qquad\qquad\qquad\qquad\qquad\quad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{array}\right\} \begin{array}{l}\text{Complicating}\\ \text{resource}\\ \text{constraints}\end{array}$$

$$\left.\begin{array}{l} e_{11}x_1 + e_{12}x_2 + \cdots + e_{1n}x_n = d_1 \\ \vdots \qquad\qquad\qquad\qquad\qquad\quad \vdots \\ e_{q1}x_1 + e_{q2}x_2 + \cdots + e_{qn}x_n = d_q \\ \qquad x_j \geq 0 \qquad (j = 1, 2, \ldots, n). \end{array}\right\} \begin{array}{l}\text{Subproblem}\\ \text{constraints}\end{array}$$

The constraints are divided into two groups. Usually the problem is much easier to solve if the complicating a_{ij} constraints are omitted, leaving only the ‘‘easy’’ e_{ij} constraints.

Given any subproblem proposal $x_1, x_2, \ldots, x_n$ (i.e., a feasible solution to the subproblem constraints), we may compute:

$$r_i = a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \quad (i = 1, 2, \ldots, m), \tag{6}$$

and

$$p = c_1x_1 + c_2x_2 + \cdots + c_nx_n,$$

which are, respectively, the amount of resource r_i used in the ith complicating constraint and the profit p associated with the proposal.

When k proposals to the subproblem are known, the procedure acts to weight these proposals optimally. Let superscripts distinguish between proposals so that r_i^j is the use of resource i by the jth proposal and p^j is the profit for the jth proposal. Then the weighting problem is written as:

$$\text{Maximize } \left[p^1\lambda_1 + p^2\lambda_2 + \cdots + p^k\lambda_k\right], \qquad \underline{\begin{array}{c}\textit{Optimal}\\ \textit{shadow}\\ \textit{prices}\end{array}}$$

subject to:

$$\begin{array}{llll}
r_1^1\lambda_1 + r_1^2\lambda_2 + \cdots + r_1^k\lambda_k & = b_1, & & \pi_1 \\
r_2^1\lambda_1 + r_2^2\lambda_2 + \cdots + r_2^k\lambda_k & = b_2, & & \pi_1 \\
\vdots & \vdots & & \\
r_m^1\lambda_1 + r_m^2\lambda_2 + \cdots + r_m^k\lambda_k & = b_m, & & \pi_m \\
\lambda_1 + \quad \lambda_2 + \cdots + \quad \lambda_k & = 1, & & \sigma \\
\lambda_j \geq 0 \qquad (j = 1, 2, \ldots, k). & & &
\end{array} \tag{7}$$

The weights $\lambda_1, \lambda_2, \ldots, \lambda_k$ are variables and r_i^j and p^j are known data in this problem.

Having solved the weighting problem and determined optimal shadow prices, we next consider adding new proposals. As we saw in Chapters 3 and 4, the reduced cost for a new proposal in the weighting linear program is given by:

$$\overline{p} = p - \sum_{i=1}^{m} \pi_i r_i - \sigma.$$

Substituting from the expressions in (6) for p and the r_i, we have

$$\overline{p} = \sum_{j=1}^{n} c_j x_j - \sum_{i=1}^{m} \pi_i \left(\sum_{j=1}^{n} a_{ij} x_j \right) - \sigma$$

or, rearranging,

$$\overline{p} = \sum_{j=1}^{n} \left(c_j - \sum_{i=1}^{m} \pi_i a_{ij} \right) x_j - \sigma.$$

Observe that the coefficient for x_j is the same reduced cost that was used in normal linear programming when applied to the complicating constraints. The additional term σ, introduced for the weighting constraint in problem (7), is added because of the subproblem constraints.

To determine whether any new proposal will improve the weighting linear program, we seek max $\overline{p}$ by solving the subproblem

$$v^k = \text{Max} \sum_{j=1}^{n} \left(c_j - \sum_{i=1}^{m} \pi_i a_{ij} \right) x_j, \tag{8}$$

subject to the subproblem constraints. There are two possible outcomes:

i) If $v^k \leq \sigma$, then max $\overline{p} \leq 0$. No new proposal improves the weighting linear program, and the procedure terminates. The solution is specified by weighting the subproblem proposals by the optimal weights $\lambda_1^*, \lambda_2^*, \ldots, \lambda_k^*$ to problem (7).

ii) If $v^k > \sigma$, then the optimal solution $x_1^*, x_2^*, \ldots, x_n^*$ to the subproblem is used in the weighting problem, by calculating the resource usages $r_1^{k+1}, r_2^{k+1}, \ldots, r_m^{k+1}$ and profit p^{k+1} for this proposal from the expressions in (6), and adding these coefficients with weight λ_{k+1}. The weighting problem is solved with this additional proposal and the procedure is repeated.

Section 12.7 develops the theory of this method and shows that it solves the original problem after a *finite* number of steps. This property uses the fact that the subproblem is a linear program, so that the simplex method for its solution determines each new proposal as an extreme point of the subproblem feasible region. Finite convergence then results, since there are only a finite number of potential proposals (i.e., extreme points) for the subproblem.

Computation Considerations

Initial Solutions

When solving linear programs, initial feasible solutions are determined by Phase I of the simplex method. Since the weighting problem is a linear program, the same technique can be used to find an initial solution for the decomposition method. Assuming that each righthand-side coefficient b_i is nonnegative, we introduce artificial variables $a_1, a_2, \ldots, a_m$ and solve the Phase I problem:

$$w = \text{Maximize } (-a_1 - a_2 - \cdots - a_m),$$

subject to:

$$\begin{aligned} a_i + r_i^1\lambda_1 + r_i^2\lambda_2 + \cdots + r_i^k\lambda_k &= b_i \quad (i = 1, 2, \ldots, m),\\ \lambda_1 + \lambda_2 + \cdots + \lambda_k &= 1,\\ \lambda_j &\geq 0 \quad (j = 1, 2, \ldots, k),\\ a_i &\geq 0 \quad (i = 1, 2, \ldots, m). \end{aligned}$$

This problem weights subproblem proposals as in the original problem and decomposition can be used in its solution. To initiate the procedure, we might include only the artificial variables $a_1, a_2, \ldots, a_m$ and any known subproblem proposals. If no subproblem proposals are known, one can be found by ignoring the complicating constraints and solving a linear program with only the subproblem constraints. New subproblem proposals are generated by the usual decomposition procedure. In this case, though, the profit contribution of every proposal is zero for the Phase I objective; i.e., the pricing calculation is:

$$\overline{p} = 0 - \sum \pi_i r_i - \sigma.$$

Otherwise, the details are the same as described previously.

If the optimal objective value $w^* < 0$, then the originalconstraints are infeasible and the procedure terminates. If $w^* = 0$, the final solution to the phase I problem identifiesproposals and weights that are feasible in the weighting problem. We continue by applying decomposition with the phase II objective function, starting with these proposals.

The next section illustrates this phase I procedure in a numerical example.

Resolving Problems

Solving the weighting problem determines an optimal basis. After a new column (proposal) is added from the subproblem, this basis can be used as a starting point to solve the new weighting problem by the revised simplex method (see Appendix B). Usually, the old basis is near-optimal and few iterations are required for the new problem. Similarly, the optimal basis for the last subproblem can be used to initiate the solution to that problem when it is considered next.

Dropping Nonbasic Columns

After many iterations the number of columns in the weighting problem may become large. Any nonbasic proposal to that problem can be dropped to save storage. If it is required, it is generated again by the subproblem.

Variation of the Method

The decomposition approach can be modified slightly for treating primal block-angular structures. For notational convenience, let us consider the problem with only two subsystems:

$$\text{Maximize } z = c_1x_1 + c_2x_2 + \cdots + c_tx_t + c_{t+1}x_{t+1} + \cdots + c_nx_n,$$

subject to:

$$\begin{aligned}
a_{i1}x_1 + a_{i2}x_2 \cdots + a_{it}x_t + a_{i,t+1}x_{t+1} + \cdots + a_{in}x_n &= b_i \quad (i = 1, 2, \ldots, m),\\
e_{s1}x_1 + e_{s2}x_2 \cdots + e_{st}x_t \qquad\qquad\qquad\qquad &= d_s \quad (s = 1, 2, \ldots, \overline{q}),\\
e_{s,t+1}x_{t+1} + \cdots + e_{sn}x_n &= d_s \quad (s = \overline{q} + 1, \overline{q} + 2, \ldots, q)\\
x_j &\geq 0, \quad (j = 1, 2, \ldots, n).
\end{aligned}$$

The easy e_{ij} constraints in this case are composed of two independent subsystems, one containing the variables $x_1, x_2, \ldots, x_t$ and the other containing the variables $x_{t+1}, x_{t+2}, \ldots, x_n$.

Decomposition may be applied by viewing the e_{ij} constraints as a single subproblem. Alternately, each subsystem may be viewed as a separate subproblem. Each will submit its own proposals and the weighting problem will act to coordinate these proposals in the following way. For any proposal $x_1, x_2, \ldots, x_t$ from subproblem 1, let

$$r_{i1} = a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{it}x_t \quad (i = 1, 2, \ldots, m),$$

and

$$p_1 = c_1x_1 + c_2x_2 + \cdots + c_tx_t$$

denote the resource usage r_{i1} in the ith a_{ij} constraint and profit contribution p_1 for this proposal. Similarly, for any proposal $x_{t+1}, x_{t+2}, \ldots, x_n$ from subproblem 2, let

$$r_{i2} = a_{i,t+1}x_{t+1} + a_{i,t+2}x_{t+2} + \cdots + a_{in}x_n \quad (i = 1, 2, \ldots, m),$$

and

$$p_2 = c_{t+1}x_{t+1} + c_{t+2}x_{t+2} + \cdots + c_nx_n$$

denote its corresponding resource usage and profit contribution.

Suppose that, at any stage in the algorithm, k proposals are available from subproblem 1, and ℓ proposals are available from subproblem 2. Again, letting superscripts distinguish between proposals, we have the weighting problem:

$$\text{Max } p_1^1\lambda_1 + p_1^2\lambda_2 + \cdots + p_1^k\lambda_k + p_2^1\mu_1 + p_2^2\mu_2 + \cdots + p_2^\ell\mu_\ell,$$

subject to:

$$\begin{array}{ll}
 & \textit{Optimal shadow prices}\\
r_{i1}^1\lambda_1 + r_{i1}^2\lambda_2 + \cdots + r_{i1}^k\lambda_k + r_{i2}^1\mu_1 + r_{i2}^2\mu_2 + \cdots + r_{i2}^\ell\mu_\ell = b_i & \\
\qquad\qquad\qquad\qquad (i = 1, \ldots, m) & \pi_i\\
\lambda_1 + \lambda_2 + \cdots + \lambda_k \qquad\qquad\qquad = 1, & \sigma_1\\
\qquad\qquad\qquad \mu_1 + \mu_2 + \cdots + \mu_\ell = 1, & \sigma_2\\
\lambda_j \geq 0, \quad \mu_s \geq 0 \quad (j = 1, 2, \ldots, k; s = 1, 2, \ldots, \ell). &
\end{array}$$

The variables $\lambda_1, \lambda_2, \ldots, \lambda_k$ weight subproblem 1 proposals and the variables $\mu_1, \mu_2, \ldots, \mu_\ell$ weight subproblem 2 proposals. The objective function adds the contribution from both subproblems and the ith constraint states that the total resource usage from both subsystems should equal the resource availability b_i of the ith resource.

After solving this linear program and determining the optimal shadow prices $\pi_1, \pi_2, \ldots, \pi_m$ and σ_1, σ_2, optimality is assessed by pricing out potential proposals from each subproblem:

$$\overline{p}_1 = p_1 - \sum_{i=1}^{m} \pi_i r_{i1} - \sigma_1 \quad \text{for subproblem 1,}$$

$$\overline{p}_2 = p_2 - \sum_{i=1}^{m} \pi_i r_{i2} - \sigma_2 \quad \text{for subproblem 2.}$$

Substituting for p_1, p_2, r_{i1}, and r_{i2} in terms of the variable x_j, we make these assessments, as in the usual decomposition procedure, by solving the subproblems:

Subproblem 1

$$v_1 = \text{Max} \sum_{j=1}^{t} \left(c_j - \sum_{i=1}^{m} \pi_i a_{ij} \right) x_j,$$

subject to:

$$\begin{aligned} e_{s1}x_1 + e_{s2}x_2 + \cdots + e_{st}x_t &= d_s \qquad (s = 1, 2, \ldots, \overline{q}), \\ x_j &\geq 0, \qquad (j = 1, 2, \ldots, t). \end{aligned}$$

Subproblem 2

$$v_2 = \text{Max} \sum_{j=t+1}^{n} \left(c_j - \sum_{i=1}^{m} \pi_i a_{ij} \right) x_j,$$

subject to:

$$\begin{aligned} e_{s,t+1}x_{t+1} + e_{s,t+2}x_{t+2} + \cdots + e_{sn}x_n &= d_s \qquad (s = \overline{q}+1, \overline{q}+2, \ldots, q), \\ x_j &\geq 0, \qquad (j = t+1, t+2, \ldots, n). \end{aligned}$$

If $v_i \leq \sigma_i$ for $i = 1$ and 2, then $\overline{p}_1 \leq 0$ for every proposal from subproblem 1 and $\overline{p}_2 \leq 0$ for every proposal from subproblem 2; the optimal solution has been obtained. If $v_1 > \sigma_1$, the optimal proposal to the first subproblem is added to the weighting problem; if $v_2 > \sigma_2$, the optimal proposal to the second subproblem is added to the weighting problem. The procedure then is repeated.

This modified algorithm easily generalizes when the primal block-angular system contains more than two subsystems. There then will be one weighting constraint for each subsystem. We should point out that it is not necessary to determine a new proposal from each subproblem at every iteration. Consequently, it is not necessary to solve each subproblem at every iteration, but rather subproblems must be solved until the condition $v_i > \sigma_i$ (that, is, $\overline{p}_i > 0$) is achieved for one solution, so that at least one new proposal is added to the weighting problem at each iteration.

12.5 AN EXAMPLE OF THE DECOMPOSITION PROCEDURE

To illustrate the decomposition procedure with an example that indicates some of its computational advantages, we consider a special case of the multicommodity-flow problem introduced as an example in Section 12.1.

An automobile company produces luxury and compact cars at two of its regional plants, for distribution to three local markets. Tables 12.3 and 12.4 specify the transportation characteristics of the problem on a per-month basis, including the transportation solution. The problem is formulated in terms of profit maximization.

One complicating factor is introduced by the company's delivery system. The company has contracted to ship from plants to destinations with a trucking company.

Table 12.3 Luxury cars

Plant	*Market* 1	2	3	*Supply*
1	$\boxed{100}$ 15	$\boxed{120}$ 10	$\boxed{90}$	25
2	$\boxed{80}$ 5	$\boxed{70}$	$\boxed{140}$ 10	15
Demand	20	10	10	

Profit = 4500

Table 12.4 Compact cars

Plant	*Market* 1	2	3	*Supply*
1	$\boxed{40}$ 20	$\boxed{20}$ 10	$\boxed{30}$ 20	50
2	$\boxed{20}$	$\boxed{40}$ 30	$\boxed{10}$	30
Demand	20	40	20	

Profit = 2800

The routes from plant 1 to both markets 1 and 3 are hazardous, however; for this reason, the trucking contract specifies that no more than 30 cars in total should be sent along either of these routes in any single month. The above solutions sending 35 cars (15 luxury and 20 compact) from plant 1 to market 1 does not satisfy this restriction, and must be modified.

Let superscript 1 denote luxury cars, superscript 2 denote compact cars, and let x^k_{ij} be the number of cars of type k sent from plant i to market to j. The model is formulated as a primal block-angular problem with objective function

$$\text{Maximize}\Big[100x^1_{11} + 120x^1_{12} + 90x^1_{13} + 80x^1_{21} + 70x^1_{22} + 140x^1_{23}$$
$$+40x^2_{11} + 20x^2_{12} + 30x^2_{13} + 20x^2_{21} + 40x^2_{22} + 10x^2_{23}\Big].$$

The five supply and demand constraints of each transportation table and the following two trucking restrictions must be satisfied.

$$\begin{aligned} x^1_{11} + x^2_{11} \qquad\qquad &\le 30 \qquad \text{(Resource 1)},\\ x^1_{13} + x^2_{13} &\le 30 \qquad \text{(Resource 2)}. \end{aligned}$$

This linear program is easy to solve without the last two constraints, since it then reduces to two separate transportation problems. Consequently, it is attractive to use decomposition, with the transportation problems as two separate subproblems.

The initial weighting problem considers the transportation solutions as one proposal from each subproblem. Since these proposals are infeasible, a Phase I version of the weighting problem with artificial variable a_1 must be solved first:

$$\begin{array}{lr} \text{Maximize } (-a_1), & \textit{Optimal shadow prices} \\ \text{subject to:} & \\ -a_1 + 15\lambda_1 + 20\mu_1 + s_1 \qquad = 30, & \pi_1 = 1 \\ 0\lambda_1 + 20\mu_1 \qquad + s_2 = 30, & \pi_2 = 0 \\ \lambda_1 \qquad\qquad = 1, & \sigma_1 = -15 \\ \mu_1 \qquad\qquad = 1, & \sigma_2 = -20 \\ \lambda_1 \ge 0, \quad \mu_1 \ge 0, \quad s_1 \ge 0, \quad s_2 \ge 0. & \end{array}$$

In this problem, s_1 and s_2 are slack variables for the complicating resource constraints. Since the two initial proposals ship $15 + 20 = 35$ cars on route $1 - 1$, the first constraint is infeasible, and we must introduce an artificial variable in this constraint. Only 20 cars are shipped on route $1 - 3$, so the second constraint is feasible, and the slack variable s_2 can serve as an initial basic variable in this constraint. No artificial variable is required.

Table 12.5

Plant	*Market* 1	2	3	*Supply*
1	$\boxed{-1}$ 5	$\boxed{0}$ 10	$\boxed{0}$ 10	25
2	$\boxed{0}$ 15	$\boxed{0}$	$\boxed{0}$	15
Demand	20	10	10	

Phase II profit contribution = 3800

The solution to this problem is $a_1 = 5$, $\lambda_1 = 1$, $\mu_1 = 1$, $s_1 = 0$, $s_2 = 10$, with the optimal shadow prices indicated above. Potential new luxury-car proposals are assessed by using the Phase I objective function and pricing out:

$$\begin{aligned} \overline{p}_1 &= 0 - \pi_1 r_{11} + \pi_2 r_{21} - \sigma_1 \\ &= 0 - (1)r_{11} - (0)r_{21} + 15. \end{aligned}$$

Since the two resources for the problem are the shipping capacities from plant 1 to markets 1 and 3, $r_{11} = x^1_{11}$ and $r_{21} = x^1_{13}$, and this expression reduces to:

$$\overline{p}_1 = -x^1_{11} + 15.$$

The subproblem becomes the transportation problem for luxury cars with objective coefficients as shown in Table 12.5. Note that this problem imposes a penalty of \$1 for sending a car along route $1 - 1$.
The solution indicated in the transportation tableau has an optimal objective value $v_1 = -5$. Since $\overline{p}_1 = v_1 + 15 > 0$, this proposal, using 5 units of resource 1 and 10 units of resource 2, is added to the weighting problem. The inclusion of this proposal causes a_1 to leave the basis, so that Phase I is completed.

Using the proposals now available, we may formulate the Phase II weighting problem as:

$$\begin{array}{llr} \text{Maximize } 4500\lambda_1 + 3800\lambda_2 + 2800\mu_1, & & \begin{array}{c}\textit{Optimal}\\ \textit{shadow}\\ \textit{prices}\end{array} \\ \text{subject to:} & & \\ 15\lambda_1 + 5\lambda_2 + 20\mu_1 + s_1 \qquad\quad = 30, & & \pi_1 = 70 \\ 0\lambda_1 + 10\lambda_2 + 20\mu_1 \qquad + s_2 = 30, & & \pi_2 = 0 \\ \lambda_1 + \lambda_2 \qquad\qquad\qquad\qquad = 1, & & \sigma_1 = 3450 \\ \qquad\qquad \mu_1 \qquad\qquad\quad = 1, & & \sigma_2 = 1400 \end{array}$$

$$\lambda_1 \geq 0, \quad \lambda_2 \geq 0, \quad \mu_1 \geq 0, \quad s_1 \geq 0, \quad s_2 \geq 0.$$

The optimal solution is given by $\lambda_1 = \lambda_2 = \frac{1}{2}$, $\mu_1 = 1$, $s_1 = 0$, and $s_2 = 5$, with an objective value of \$6950. Using the shadow prices to price out potential proposals gives:

$$\begin{aligned} \overline{p}^j &= p^j - \pi_1 r_{1j} - \pi_2 r_{2j} - \sigma_j \\ &= p^j - \pi_1(x^j_{11}) - \pi_2(x^j_{13}) - \sigma_j, \end{aligned}$$

or

$$\begin{aligned} \overline{p}^1 &= p^1 - 70(x^1_{11}) - 0(x^1_{13}) - 3450 = p^1 - 70x^1_{11} - 3450, \\ \overline{p}^2 &= p^2 - 70(x^2_{11}) - 0(x^2_{13}) - 1400 = p^2 - 70x^2_{11} - 1400. \end{aligned}$$

In each case, the per-unit profit for producing in plant 1 for market 1 has decreased by \$70. The decomposition algorithm has imposed a penalty of \$70 on route $1-1$ shipments, in order to divert shipments to an alternative route. The solution for luxury cars is given in Table 12.6. Since $v_1 - \sigma_1 = 3450 = 0$, no new luxury-car

Table 12.6 Luxury cars

Plant	*Market* 1	2	3	*Supply*
1	$\boxed{30}$ 15	$\boxed{120}$ 10	$\boxed{90}$	25
2	$\boxed{80}$ 5	$\boxed{70}$	$\boxed{140}$ 10	15
Demand	20	10	10	

$v_1 = 30(15) + 120(10) + 80(5) + 140(10) = 3450$

proposal is profitable, and we must consider compact cars, as in Table 12.7.

Table 12.7 Compact cars

Plant	*Market* 1	2	3	*Supply*
1	$\boxed{-30}$	$\boxed{20}$ 30	$\boxed{30}$ 20	50
2	$\boxed{20}$ 20	$\boxed{40}$ 10	$\boxed{10}$	30
Demand	20	40	20	

$v_2 = 20(30) + 30(20) + 20(20) + 40(10) = 2000$

Here $v_2 - \sigma_2 = 2000 - 1400 > 0$, so that the given proposal improves the weighting problem. It uses no units of resource 1, 20 units of resource 2, and its profit contribution is 2000, which in this case happens to equal v_2. Inserting this proposal in the weighting problem, we have:

Maximize $4500\lambda_1 + 3800\lambda_2 + 2800\mu_1 + 2000\mu_2$, *Optimal shadow prices*

subject to:

$$
\begin{array}{lllllllll}
15\lambda_1 + & 5\lambda_2 + & 20\mu_1 + & 0\mu_2 + & s_1 & & = 30, & \pi_1 = & 40 \\
0\lambda_1 + & 10\lambda_2 + & 20\mu_1 + & 20\mu_2 + & & +s_2 & = 30, & \pi_2 = & 0 \\
\lambda_1 + & \lambda_2 & & & & & = \ 1, & \sigma_1 = & 3900 \\
 & & \mu_1 + & & & & = \ 1, & \sigma_2 = & 2000
\end{array}
$$

$$\lambda_1 \geq 0, \quad \lambda_2 \geq 0, \quad \mu_1 \geq 0, \quad \mu_2 \geq 0, s_1 \geq 0, s_2 \geq \ 0.$$

The optimal basic variables are $\lambda_1 = 1$, $\mu_1 = \frac{3}{4}$, $\mu_2 = \frac{1}{4}$,and $s_2 = 10$, with objective value \$7100, and the pricing-out operations become:

$$\overline{p}^1 = p^1 - 40(x^1_{11}) - 0(x^1_{13}) - 3900 \quad \text{for luxury cars,}$$

and

$$\overline{p}^2 = p^2 - 40(x^2_{11}) - 0(x^2_{13}) - 2000 \quad \text{for compact cars.}$$

The profit contribution for producing in plant 1 for market 1 now is penalized by \$40 per unit for both types of cars. The transportation solutions are given by Tables 12.8 and 12.9.

Table 12.8 Luxury cars

Plant	*Market* 1	2	3	*Supply*
1	$\boxed{60}$ 15	$\boxed{120}$ 10	$\boxed{90}$	25
2	$\boxed{80}$ 5	$\boxed{70}$	$\boxed{140}$ 10	15
Demand	20	10	10	

$v_1 = 60(15) + 120(10) + 80(5) + 140(10) = 3900$

Table 12.9 Compact cars

Plant	*Market* 1	2	3	*Supply*
1	$\boxed{0}$ 20	$\boxed{20}$ 10	$\boxed{30}$ 20	50
2	$\boxed{20}$	$\boxed{40}$ 30	$\boxed{10}$	30
Demand	20	40	20	

$v_2 = 0(20) + 20(10) + 30(20) + 40(30) = 2000$

Since $v_1 - \sigma_1 = 0$ and $v_2 - \sigma_2 = 0$, neither subproblem can submit proposals to improve the last weighting problem and the optimal solution uses the first luxury car proposal, since $\lambda_1 = 1$, and weights the two compact car proposals with $\mu_1 = \frac{3}{4}$, $\mu_2 = \frac{1}{4}$, giving the composite proposal shown in Table 12.10.

Table 12.10 Compact cars

$$\begin{array}{c|ccc} & & \textit{Market} & \\ \textit{Plant} & 1 & 2 & 3 \\ \hline 1 & 15 & 15 & 20 \\ 2 & 5 & 25 & 0 \end{array} = \frac{3}{4} \begin{array}{ccc} & \textit{Market} & \\ 1 & 2 & 3 \\ \hline 20 & 10 & 20 \\ 0 & 30 & 0 \end{array} + \frac{1}{4} \begin{array}{ccc} & \textit{Market} & \\ 1 & 2 & 3 \\ \hline 0 & 30 & 20 \\ 20 & 10 & 0 \end{array}$$

Observe that, although both of the transportation proposals shown on the righthand side of this expression solve the final transportation subproblem for compact cars with value $v_2 = 2000$, neither is an optimal solution to the overall problem. The unique solution to the overall problem is the composite compact-car proposal shown on the left together with the first luxury-car proposal.

12.6 ECONOMIC INTERPRETATION OF DECOMPOSITION

The connection between prices and resource allocation has been a dominant theme in economics for some time. The analytic basis for pricing systems is rather new, however, and owes much to the development of mathematical programming. Chapter 4 established a first connection between pricing theory and mathematical programming by introducing an economic interpretation of linear-programming duality theory. Decomposition extends this interpretation to decentralized decision making. It provides a mechanism by which prices can be used to coordinate the activities of several decision makers.

For convenience, let us adopt the notation of the previous section and discuss primal block-angular systems with two subsystems. We interpret the problem as a profit maximization for a firm with two divisions. There are two levels of decision making—corporate and subdivision. Subsystem constraints reflect the divisions' allocation of their own resources, assuming that these resources are not shared. The complicating constraints limit corporate resources, which are shared and used in any proposal from either division.

Frequently, it is very expensive to gather detailed information about the divisions in a form usable by either corporate headquarters or other divisions, or to gather detailed corporate information for the divisions. Furthermore, each level of decision making usually requires its own managerial skills with separate responsibilities. For these reasons, it is often best for each division and corporate headquarters to operate somewhat in isolation, passing on only that information required to coordinate the firm's activities properly.

As indicated in Fig. 12.4, in a decomposition approach the information passed on are *prices*, from corporate headquarters to the divisions, and *proposals*, from the divisions to the corporate coordinator. Only the coordinator knows the full corporate constraints and each division knows its own operating constraints. The corporate coordinator acts to weight subproblem proposals by linear programming, to maximize profits. From the interpretation of duality given in Chapter 4, the optimal shadow prices from its solution establish a

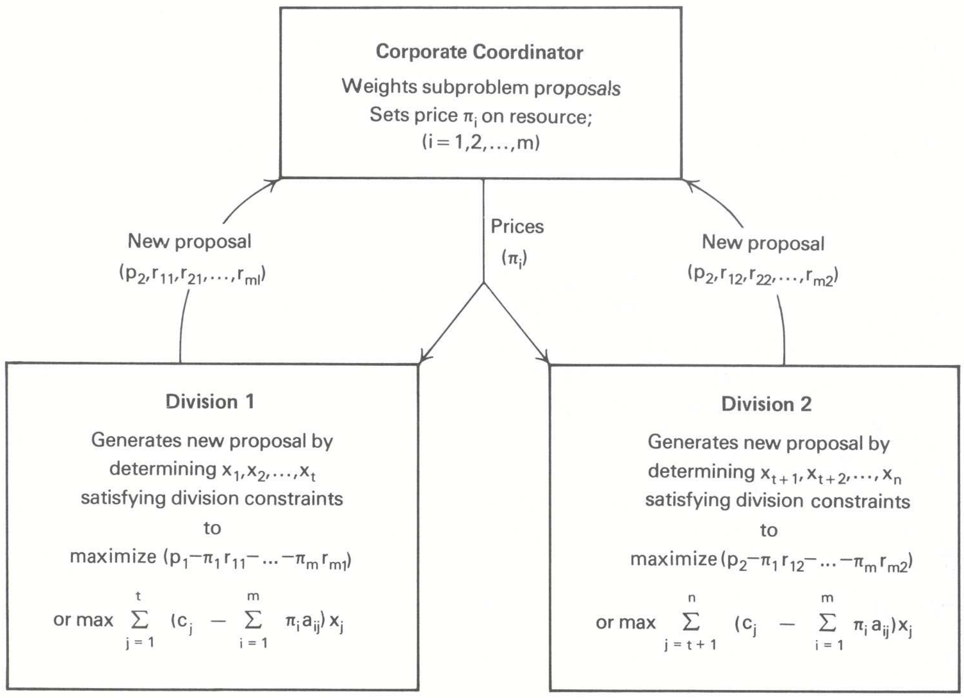

Figure 12.4 Information transfer in decomposition.

per-unit value for each resource. These prices are an internal evaluation of resources by the firm, indicating how profit will be affected by changes in the resource levels.

To ensure that the divisions are cognizant of the firm's evaluation of resources, the coordinator ''charges'' the divisions for their use of corporate resources. That is, whatever decisions $x_1, x_2, \ldots, x_t$ the first division makes, its gross revenue is

$$p_1 = c_1 x_1 + c_2 x_2 + \cdots + c_t x_t,$$

and its use of resource i is given by

$$r_{i1} = a_{i1} x_1 + a_{i2} x_2 + \cdots + a_{it} x_t.$$

Consequently, its net profit is computed by:

$$\begin{aligned}(\text{Net profit}) &= (\text{Gross revenue}) - (\text{Resource cost}) \\ &= p_1 - \pi_1 r_{11} - \pi_2 r_{21} - \cdots - \pi_m r_{m1},\end{aligned}$$

or, substituting in terms of the x_j's and rearranging,

$$(\text{Net profit}) = \sum_{j=1}^{t} \left(c_j - \sum_{i=1}^{m} \pi_i a_{ij} \right) x_j.$$

In this objective, c_j is the per-unit gross profit for activity x_j. The shadow price π_i is the value of the ith corporate resource, $\pi_i a_{ij}$ is the cost of resource i for activity j, and $\sum_{i=1}^{m} \pi_i a_{ij}$ is the total corporate resource cost, or opportunity cost, to produce each unit of this activity.

The cost $\sum_{i=1}^{m} \pi_i a_{ij}$ imposes a penalty upon activity j that reflects impacts resulting from this activity that are external to the division. That is, by engaging in this activity, the firm uses additional units a_{ij} of the corporate resources. Because the firm has limited resources, the activities of other divisions must be modified to compensate for this resource usage. The term $\sum_{i=1}^{m} \pi_i a_{ij}$ is the revenue lost by the firm as a result of the modifications that the other divisions must make in their use of resources.

Once each division has determined its optimal policy with respect to its net profit objective, it conveys this information in the form of a proposal to the coordinator. If the coordinator finds that no *new* proposals are better than those currently in use in the weighting problem, then prices π_i have stabilized (since the former linear-programming solution remains optimal), and the procedure terminates. Otherwise, new prices are determined, they are transmitted to the divisions, and the process continues.

Finally, the coordinator assesses optimality by pricing out the newly generated proposal in the weighting problem. For example, for a new proposal from division 1, the calculation is:

$$\overline{p}_1 = (p_1 - \pi_1 r_{11} - \pi_2 r_{21} - \cdots - \pi_m r_{m1}) - \sigma_1,$$

where σ_1 is the shadow price of the weighting constraint for division 1 proposals. The first term is the net profit of the new proposal as just calculated by the division. The term σ_1 is interpreted as the value (gross profit − resource cost) of the optimal composite or weighted proposal from the previous weighting problem. If $\overline{p}_1 > 0$, the new proposal's profit exceeds that of the composite proposal, and the coordinator alters the plan. The termination condition is that $\overline{p}_1 \leq 0$ and $\overline{p}_2 \leq 0$, when no new proposal is better than the current composite proposals of the weighting problem.

Example: The final weighting problem to the automobile example of the previous section was:

Maximize $4500\lambda_1 + 3800\lambda_2 + 2800\mu_1, + 2000\mu_2,$

subject to:

$$\begin{array}{rrrrrrl|l}
 & & & & & & & \textit{Optimal shadow prices} \\
15\lambda_1 + & 5\lambda_2 + & 20\mu_1 + & 0\mu_2 + & s_1 & & = 30, & \pi_1 = 40 \\
0\lambda_1 + & 10\lambda_2 + & 20\mu_1 + & 20\mu_2 + & & s_2 & = 30, & \pi_2 = 0 \\
\lambda_1 + & \lambda_2 & & & & & = 1, & \sigma_1 = 3900 \\
 & & \mu_1 + & \mu_2 & & & = 1, & \sigma_2 = 2000
\end{array}$$

$$\lambda_j,\ \mu_j,\ s_j \geq 0 \quad (j = 1, 2),$$

with optimal basic variables $\lambda_1 = 1$, $\mu_1 = \frac{3}{4}$, $\mu_2 = \frac{1}{4}$, and $s_2 = 10$. The first truck route from plant 1 to market 1 (constraint 1) is used to capacity, and the firm evaluates sending another car along this route at \$40. The second truck route from plant 1 to market 3 is not used to capacity, and accordingly its internal evaluation is $\pi_2 = \$0$.

The composite proposal for subproblem 1 is simply its first proposal with $\lambda_1 = 1$. Since this proposal sends 15 cars on the first route at \$40 each, its net profit is given by:

$$\begin{aligned}
\sigma_1 &= \text{(Gross profit)} - \text{(Resource cost)} \\
&= \$4500 - \$40(15) = \$3900.
\end{aligned}$$

Similarly, the composite proposal for compact cars sends

$$20(\tfrac{3}{4}) + 0(\tfrac{1}{4}) = 15$$

cars along the first route. Its net profit is given by weighting its gross profit coefficients with $\mu_1 = \frac{3}{4}$ and $\mu_2 = \frac{1}{4}$ and subtracting resource costs, that is, as

$$\sigma_2 = [\$2800(\tfrac{3}{4}) + \$2000(\tfrac{1}{4})] - \$40(15) = \$2000.$$

By evaluating its resources, the corporate weighting problem places a cost of \$40 on each car sent along route 1–1. Consequently, when solving the subproblems, the gross revenue in the transportation array must be decreased by \$40 along the route 1–1; the profit of luxury cars along route 1–1 changes from \$100 to $\$60(= \$100 - \$40)$, and the profit of compact cars changes from \$40 to $\$0(= \$40 - \$40)$.

To exhibit the effect of externalities between the luxury and compact divisions, suppose that the firm ships 1 additional luxury car along route 1–1. Then the capacity along this route decreases from 30 to 29 cars. Since $\lambda_1 = 1$ is fixed in the optimal basic solution, μ_1 must decrease by $\frac{1}{20}$ to preserve equality in the first resource constraint of the basic solution. Since $\mu_1 + \mu_2 = 1$, this means that μ_2 must increase by $\frac{1}{20}$. Profit from the compact-car proposals then changes by $\$2800(-\frac{1}{20}) + \$2000(+\frac{1}{20}) = -\$40$, as required. Note that the 1-unit change in luxury-car operations induces a change in the composite proposal of compact cars for subproblem 2. The decomposition algorithm allows the luxury-car managers to be aware of this external impact through the price information passed on from the coordinator.

Finally, note that, although the price concept introduced in this economic interpretation provides an internal evaluation of resources that permits the firm to coordinate subdivision activities, the prices by themselves do not determine the optimal production plan at each subdivision. As we observed in the last section, the compact-car subdivision for this example has several optimal solutions to its subproblem transportation problem with respect to the optimal resource prices of $\pi_1 = \$40$ and $\pi_2 = \$0$. Only one of these solutions however, is optimal for the *overall* corporate plan problem. Consequently, the coordinator must negotiate the final solution used by the subdivision; merely passing on the optimal resources will not suffice.

12.7 DECOMPOSITION THEORY

In this section, we assume that decomposition is applied to a problem with only one subproblem:

$$\text{Maximize } z = c_1x_1 + c_2x_2 + \cdots + c_nx_n,$$

subject to:

$$\begin{aligned}
a_{i1}x_1 + a_{i2}x_2 \quad &+ \cdots + a_{in}x_n = b_i \\
&(i = 1, 2, \ldots, m), \quad \text{(Complicating constants)} \\
e_{s1}x_1 + e_{s2}x_2 \quad &+ \cdots + e_{sn}x_n = d_s \\
&(s = 1, 2, \ldots, q), \quad \text{(Subproblem)} \\
x_j \geq 0 \qquad &(j = 1, 2, \ldots, n).
\end{aligned}$$

The discussion extends to primal block-angular problems in a straightforward manner.

The theoretical justification for decomposition depends upon a fundamental result from convex analysis. This result is illustrated in Fig. 12.5 for a feasible region determined by linear inequalities. The region is bounded and has five extreme points denoted $x^1, x^2, \ldots, x^5$. Note that any point y in the feasible region can be expressed as a weighted (convex) combination of extreme points. For example, the weighted combination of the extreme points $x^1,\ x^2$, and x^5 given by

$$y = \lambda_1x^1 + \lambda_2x^2 + \lambda_3x^5,$$

for some selection of

$$\lambda_1 \geq 0, \qquad \lambda_2 \geq 0, \qquad \text{and} \qquad \lambda_3 \geq 0,$$

with

$$\lambda_1 + \lambda_2 + \lambda_3 = 1,$$

determines the shaded triangle in Fig. 12.5. Note that the representation of y as an extreme point is not unique; y can also be expressed as a weighted combination of $x^1,\ x^4$, and x^5, or $x^1,\ x^3$, and x^5.

The general result that we wish to apply is stated as the Representation Property, defined as:

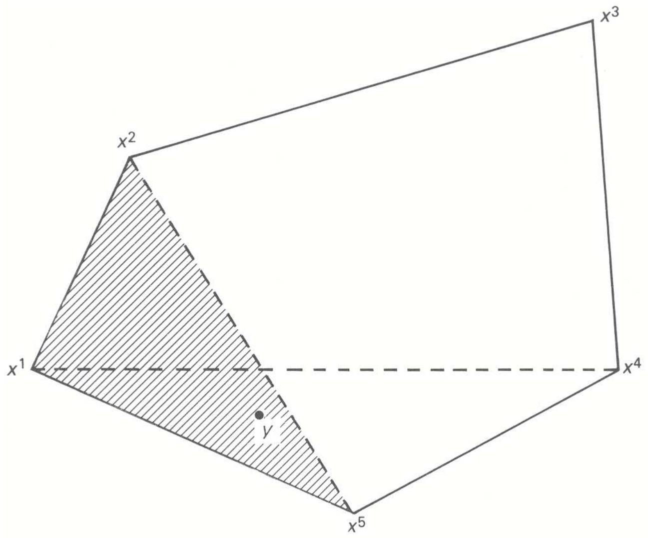

Figure 12.5 Extreme point representation.

Representation Property. Let $x^1, x^2, \ldots, x^K$ be the extreme points [each x^k specifies values for each variable x_j as $(x_1^k, x_2^k, \ldots, x_n^k)$] of a feasible region determined by the constraints

$$\begin{aligned} e_{s1}x_1 + e_{s2}x_2 + \cdots + e_{sn}x_n &= d_s \qquad (s = 1, 2, \ldots, q), \\ x_j &\geq 0 \qquad (j = 1, 2, \ldots, n), \end{aligned}$$

and assume that the points in this feasible region are bounded. Then any feasible point $x = (x_1, x_2, \ldots, x_n)$ can be expressed as a convex (weighted) combination of the points $x^1, x^2, \ldots, x^K$, as

$$x_j = \lambda_1 x_j^1 + \lambda_2 x_j^2 + \cdots + \lambda_K x_j^K \qquad (j = 1, 2, \ldots, n)$$

with

$$\begin{aligned} \lambda_1 + \lambda_2 + \cdots + \lambda_K &= 1, \\ \lambda_k &\geq 0 \qquad (k = 1, 2, \ldots, K). \end{aligned}$$

By applying this result, we can express the overall problem in terms of the extreme points $x^1, x^2, \ldots, x^K$ of the subproblem. Since *every* feasible point to the subproblem is generated as the coefficients λ_k vary, the original problem can be re-expressed as a linear program in the variables λ_k:

$$\text{Max } z = c_1(\lambda_1 x_1^1 + \lambda_2 x_1^2 + \cdots + \lambda_K x_1^K) + \cdots + c_n(\lambda_1 x_n^1 + \lambda_2 x_n^2 + \cdots + \lambda_K x_n^K),$$

subject to:

$$\begin{aligned} a_{i1}(\lambda_1 x_1^1 + \lambda_2 x_1^2 + \cdots + \lambda_K x_1^K) + \cdots + a_{in}(\lambda_1 x_n^1 + \lambda_2 x_n^2 + \cdots + \lambda_K x_n^K) & \\ = b_i &\qquad (i = 1, 2, \ldots, m), \\ \lambda_1 + \lambda_2 + \cdots + \lambda_K = 1, & \\ \lambda_k \geq 0 &\qquad (k = 1, 2, \ldots, K). \end{aligned}$$

or, equivalently, by collecting coefficients for the λ_k:

$$\text{Max z } = p_1\lambda_1 + p_2\lambda_2 + \cdots + p_K\lambda_K,$$

subject to:

$$\begin{aligned} r_i^1\lambda_1 + r_i^2\lambda_2 + \cdots + r_i^K\lambda_K &= b_i && \text{(Resource constraint)} \\ & && (i = 1, 2, \ldots, m) \\ \lambda_1 + \lambda_2 + \cdots + \lambda_K &= 1, && \text{(Weighting constraint)} \\ \lambda_k &\ge 0 && (k = 1, 2, \ldots, K), \end{aligned}$$

where

$$p_k = c_1x_1^k + c_2x_2^k + \cdots + c_nx_n^k,$$

and

$$r_i^k = a_{i1}x_1^k + a_{i2}x_2^k + \cdots + a_{in}x_n^k \qquad (i = 1, 2, \ldots, m)$$

indicate, respectively, the profit and resource usage for the kth extreme point $x^k = (x_1^k, x_2^k, \ldots, x_n^k)$.Observe that this notation corresponds to that used in Section 12.4. Extreme points here play the role of proposals in that discussion.

It is important to recognize that the new problem is equivalent to the original problem. The weights ensure that the solution $x_j = \lambda_1x_j^1 + \lambda_2x_j^2 + \cdots + \lambda_Kx_j^K$ satisfies the subproblem constraints, and the resource constraints for b_i are equivalent to the original complicating constraints. The new form of the problem includes all the characteristics of the original formulation and is often referred to as the *master problem*.

Note that the reformulation has reduced the number of constraints by replacing the subproblem constraints with the single weighting constraint. At the same time, the new version of the problem usually has many more variables, since the number of extreme points in the subproblem may be enormous (hundreds of thousands). For this reason, it seldom would be tractable to generate all the subproblem extreme points in order to solve the master problem directly by linear programming.

Decomposition avoids solving the full master problem. Instead, it starts with a subset of the subproblem extreme points and generates the remaining extreme points *only as needed*. That is, it starts with the *restricted master problem*

$$\begin{array}{lll} & & \textit{Optimal shadow prices} \\ z^J = \text{Max } z = p_1\lambda_1 + p_2\lambda_2 + \cdots + p_J\lambda_J, & & \\ \text{subject to:} & & \\ r_i^1\lambda_1 + r_i^2\lambda_2 + \cdots + r_i^J\lambda_J = b_i & (i = 1, 2, \ldots, m), & \pi_i \\ \lambda_1 + \lambda_2 + \cdots + \lambda_J = 1, & & \sigma \\ \lambda_k \ge 0 & (k = 1, 2, \ldots, J), & \end{array}$$

where J is usually so much less than K that the simplex method can be employed for its solution.

Any feasible solution to the restricted master problem is feasible for the master problem by taking $\lambda_{J+1} = \lambda_{J+2} = \cdots = \lambda_K = 0$. The theory of the simplex method shows that the solution to the restricted master problem is optimal for the overall problem if every column in the master problem prices out to be nonnegative; that is, if

$$p_k - \pi_1r_1^k - \pi_2r_2^k - \cdots - \pi_mr_m^k - \sigma \le 0 \qquad (k = 1, 2, \ldots, K)$$

or, equivalently, in terms of the variables x_j^k generating p_k and the r_i^k's,

$$\sum_{j=1}^n \left[c_j - \sum_{i=1}^m \pi_ia_{ij}\right] x_j^k - \sigma \le 0 \qquad (k = 1, 2, \ldots, K). \tag{9}$$

This condition can be checked easily without enumerating every extreme point. We must solve only the linear-programming subproblem

$$v^J = \text{Max} \sum_{j=1}^n \left[c_j - \sum_{i=1}^m \pi_ia_{ij}\right] x_j,$$

subject to:

$$
\begin{aligned}
e_{s1}x_1 + e_{s2}x_2 + \cdots + e_{sn}x_n &= d_s \quad (s = 1, 2, \ldots, q),\\
x_j &\geq 0 \quad (j = 1, 2, \ldots, n).
\end{aligned}
$$

If $v^J - \sigma \leq 0$, then the optimality condition (9) is satisfied, and the problem has been solved. The optimal solution $x_1^*, x_2^*, \ldots, x_n^*$ is given by weighting the extreme points $x_j^1, x_j^2, \ldots, x_j^J$ used in the restricted master problem by the optimal weights $\lambda_1^*, \lambda_2^*, \ldots, \lambda_J^*$ to that problem, that is,

$$
x_j^* = \lambda_1^* x_j^1 + \lambda_2^* x_j^2 + \cdots + \lambda_j^* x_j^J \quad (j = 1, 2, \ldots, n).
$$

If $v^J - \sigma > 0$, then the optimal extreme point solution to the subproblem x_1^{J+1}, $x_2^{J+1}, \ldots, x_n^{J+1}$ is usedat the $(J+1)$st extreme point to improve the restricted master. A new weight λ_{J+1} is added to the restricted master problem with coefficients

$$
\begin{aligned}
p_{J+1} &= c_1 x_1^{J+1} + c_2 x_2^{J+1} + \cdots + c_n x_n^{J+1},\\
r_i^{J+1} &= a_{i1} x_1^{J+1} + a_{i2} x_2^{J+1} + \cdots + a_{in} x_n^{J+1} \quad (i = 1, 2, \ldots, m),
\end{aligned}
$$

and the process is then repeated.

Convergence Property. The representation property has shown that decomposition is solving the master problem by generating coefficient data as needed. Since the master problem is a linear program, the decomposition algorithm inherits finite convergence from the simplex method. Recall that the simplex method solves linear programs in a finite number of steps. For decomposition, the subproblem calculation ensures that the variable introduced into the basis has a positive reduced cost, just as in applying the simplex method to the master problem. Consequently, from the linear-programming theory, the master problem is solved in a finite number of steps; the procedure thus determines an optimal solution by solving the restricted master problem and subproblem alternately a *finite* number of times.

Bounds on the Objective value

We previously observed that the value z^J to the restricted master problem tends to tail off and approach z^*, the optimal value to the overall problem, very slowly. As a result, we may wish to terminate the algorithm before an optimal solution has been obtained, rather than paying the added computational expense to improve the current solution only slightly. An important feature of the decomposition approach is that it permits us to assess the effect of terminating with a suboptimal solution by indicating how far z^J is removed from z^*.

For notation let $\pi_1^J, \pi_2^J, \ldots, \pi_m^J$, and σ^J denote the optimal shadow prices for the m resource constraints and the weighting constraint in the current restricted master problem. The current subproblem is:

$$
v^J = \text{Max} \sum_{j=1}^{n} \left(c_j - \sum_{i=1}^{m} \pi_i^J a_{ij} \right) x_j,
$$

subject to:

	Optimal shadow prices
$\sum_{j=1}^{n} e_{sj} x_j = d_s \quad (s = 1, 2, \ldots, q),$	α_s
$x_j \geq 0 \quad (j = 1, 2, \ldots, n),$	

with optimal shadow prices $\alpha_1,\ \alpha_2, \ldots, \alpha_q$. By linear programming duality theory these shadow prices solve the dual to the subproblem, so that

$$c_j - \sum_{i=1}^{m} \pi_i^J a_{ij} - \sum_{s=1}^{q} \alpha_s e_{sj} \leq 0 \qquad (j = 1, 2, \ldots, n)$$

and

$$\sum_{s=1}^{q} \alpha_s d_s = v^J. \tag{10}$$

But these inequalities are precisely the dual feasibility conditions of the original problem, and so the solution to every subproblem provides a dual feasible solution to that problem. The weak duality property of linear programming, though, shows that every feasible solution to the dual gives an upper bound to the primal objective value z^*. Thus

$$\sum_{i=1}^{m} \pi_i^J b_i + \sum_{s=1}^{q} \alpha_s d_s \geq z^*. \tag{11}$$

Since the solution to every restricted master problem determines a feasible solution to the original problem (via the master problem), we also know that

$$z^* \geq z^J.$$

As the algorithm proceeds, the lower bounds z^J increase and approach z^*. There is, however, no guarantee that the dual feasible solutions are improving. Consequently, theupper bound generated at any step may be worse than those generated at previous steps, and we always record the *best* upper bound generated thus far.

The upper bound can be expressed in an alternative form. Since the variables π_i^J and σ^J are optimal shadow prices to the restricted master problem, they solve its dual problem, so that:

$$z^J = \sum_{i=1}^{m} \pi_i^J b_i + \sigma^J = \text{ Dual objective value.}$$

Substituting this value together with the equality (10), in expression (11), gives the alternative form for the bounds:

$$z^J \leq z^* \leq z^J - \sigma^J + v^J.$$

This form is convenient since it specifies the bounds in terms of the objective values of the subproblem and restricted master problem. The only dual variable used corresponds to the weighting constraint in the restricted master.

To illustrate these bounds, reconsider the preview problem introduced in Section 12.2. The first restricted master problem used two subproblem extreme points and was given by:

	Optimal shadow prices
$z^2 = \text{Max z} = 22\lambda_1 + 17\lambda_2,$	
subject to:	
$18\lambda_1 + 13\lambda_2 \leq 17,$	$\pi_1^2 = 1$
$\lambda_1 + \quad \lambda_2 = \ 1,$	$\sigma^2 = 4$
$\lambda_1 \geq 0, \quad \lambda_2 \geq 0.$	

Here $z^2 = 21$ and the subproblem

$$v^2 = \text{Max}(x_1 - x_2 + 2x_3),$$

subject to:

$$1 \leq x_j \leq 2 \quad (j = 1, 2, 3),$$

has the solution $v^2 = 5$. Thus,

$$21 \leq z^* \leq 21 - 4 + 5 = 22.$$

At this point, computations could have been terminated, with the assurance that the solution of the current restricted master problem is within 5 percent of the optimal objective value, which in this case is $z^* = 21\frac{1}{2}$.

Unbounded Solution to the Subproblem

For expositional purposes, we have assumed that every subproblem encountered has an optimal solution, even though its objective value might be unbounded. First, we should note that an unbounded objective value to a subproblem does not necessarily imply that the overall problem is unbounded, since the constraints that the subproblem ignores may prohibit the activity levels leading to an unbounded subproblem solution from being feasible to the full problem. Therefore, we cannot simply terminate computations when the subproblem is unbounded; a more extensive procedure is required; reconsidering the representation property underlying the theory will suggest the appropriate procedure.

When the subproblem is unbounded, the representation property becomes more delicate. For example, the feasible region in Fig. 12.6 contains three extreme points. Taking the weighted combination of these points generates only the shaded portion of the feasible region. Observe, though, that by moving from the shaded region in a *direction parallel to either of the unbounded edges*, every feasible point can be generated. This suggests that the general representation property should include *directions* as well as extreme points. Actually, we do not require all possible movement directions, but only those that are analogous to extreme points.

Before exploring this idea, let us introduce a definition.

Definition.

i) A direction $d = (d_1, d_2, \ldots, d_n)$ is called a *ray* for the subproblem if, whenever $x_1, x_2, \ldots, x_n$ is a feasible solution, then the point

$$x_1 + \theta d_1, x_2 + \theta d_2, \ldots, x_n + \theta d_n$$

also is feasible for any choice of $\theta \geq 0$.

ii) A ray $d = (d_1, d_2, \ldots, d_n)$ is called an *extreme ray* if it cannot be expressed as a weighted combination of two other rays; that is, if there are no two rays $d' = (d'_1, d'_2, \ldots, d'_n)$ and $d'' = (d''_1, d''_2, \ldots, d''_n)$ and

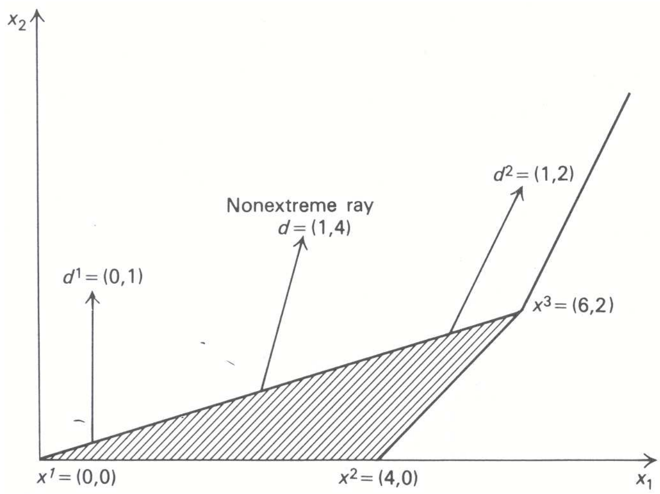

Figure 12.6 Representing an unbounded region.

weight $0 < \lambda < 1$ such that

$$d_j = \lambda d'_j + (1 - \lambda) d''_j \qquad (j = 1, 2, \ldots, n).$$

A ray is a direction that points from any feasible point only toward other feasible points. An extreme ray is an unbounded edge of the feasible region. In Fig. 12.6, d^1 and d^2 are the only two extreme rays. Any other ray such as d can be expressed as a weighted combination of these two rays; for example,

$$d = (1, 4) = 2d^1 + d^2 = 2(0, 1) + (1, 2).$$

The extended representation result states that there are only a finite number of extreme rays $d^1, d^2, \ldots, d^L$ to the subproblem and that any feasible solution $x = (x_1, x_2, \ldots, x_n)$ to the subproblem can be expressed as a weighted combination of extreme points plus a nonnegative combination of extreme rays, as:

$$\begin{aligned} x_j &= \lambda_1 x_j^1 + \lambda_2 x_j^2 + \cdots + \lambda_k x_j^k + \theta_1 d_j^1 + \theta_2 d_j^2 + \cdots + \theta_L d_j^L \qquad && (j = 1, 2, \ldots, n), \\ &\lambda_1 + \lambda_2 + \cdots + \lambda_k && = 1, \\ &\lambda_k \geq 0, \qquad \theta_\ell \geq 0 \qquad && (k = 1, 2, \ldots, K; \ell = 1, 2, \ldots, L). \end{aligned}$$

Observe that the θ_j need not sum to one.

Let $\hat{p}_k$ and $\hat{r}_i^k$ denote, respectively, the per-unit profit and the ith resource usage of the kth extreme ray; that is,

$$\begin{aligned} \hat{p}_k &= c_1 d_1^k + c_2 d_2^k + \cdots + c_n d_n^k, \\ \hat{r}_i^k &= a_{i1} d_1^k + a_{i2} d_2^k + \cdots + a_{in} d_n^k \quad (i = 1, 2, \ldots, m). \end{aligned}$$

Substituting as before for x_j in the complicating constraints in terms of extreme points and extreme rays gives the master problem:

$$\text{Max } z = p_1\lambda_1 + p_2\lambda_2 + \cdots + p_K\lambda_K + \hat{p}_1\theta_1 + \hat{p}_2\theta_2 + \cdots + \hat{p}_L\theta_L,$$

subject to:

$$\begin{aligned} r_i^1\lambda_1 + r_i^2\lambda_2 + \cdots + r_i^K\lambda_K + \hat{r}_i^1\theta_1 + \hat{r}_i^2\theta_2 + \cdots + \hat{r}_i^L\theta_L &= b_i \qquad (i = 1, 2, \ldots, m), \\ \lambda_1 + \lambda_2 + \cdots + \lambda_K &= 1, \\ \lambda_k \geq 0 \quad \theta_\ell \geq 0 \qquad (k = 1, 2, \ldots, K; \ell = 1, 2, \ldots, L).& \end{aligned}$$

The solution strategy parallels that given previously. At each step, we solve a restricted master problem containing only a subset of the extreme points and extreme rays, and use the optimal shadow prices to define a subproblem. If the subproblem has an optimal solution, a new extreme point is added to the restricted master problem and it is solved again. When the subproblem is unbounded, though, an extreme ray is added to the restricted master problem. To be precise, we must specify how an extreme ray is identified. It turns out that an extreme ray is determined easily as a byproduct of the simplex method, as illustrated by the following example.

Example

$$\text{Maximize } z = 5x_1 - x_2,$$

subject to:

$$\begin{array}{lll} x_1 \leq 8, & & \text{(Complicating constraint)} \\ \left.\begin{array}{l} x_1 - x_2 + x_3 = 4 \\ 2x_1 - x_2 + x_4 = 10 \\ x_j \geq 0 \quad (j = 1, 2, 3, 4). \end{array}\right\} & & \text{(Subproblem)} \end{array}$$

The subproblem has been identified as above solely for purposes of illustration. The feasible region to the subproblem was given in terms of x_1 and x_2 in Fig. 12.6 by viewing x_3 and x_4 as slack variables.

As an initial restricted master problem, let us use the extreme points $(x_1, x_2, x_3, x_4) = (4, 0, 0, 2)$, $(x_1, x_2, x_3, x_4) = (6, 2, 0, 0)$, and no extreme rays. These extreme points, respectively, use 4 and 6 units of the complicating resource and contribute 20 and 28 units to the objective function. The restricted master problem is given by:

$$
\begin{array}{lr}
z^2 = \text{Max } 20\lambda_1 + 28\lambda_2, & \textit{Optimal shadow prices} \\
\text{subject to:} & \\
4\lambda_1 + 6\lambda_2 \le 8, & 0 \\
\lambda_1 + \lambda_2 = 1, & 28 \\
\lambda_1 \ge 0, \quad \lambda_2 \ge 0. &
\end{array}
$$

The solution is $\lambda_1 = 0,\ \lambda_2 = 1,\ z^2 = 28$, with a price of 0 on the complicating constraint.

The subproblem is

$$v^2 = \text{Max } 5x_1 - x_2,$$

subject to:

$$
\begin{aligned}
x_1 - x_2 + x_3 \quad &= 4, \\
2x_1 - x_2 \quad + x_4 &= 10, \\
x_j &\ge 0 \qquad (j = 1, 2, 3, 4).
\end{aligned}
$$

Solving by the simplex method leads to the canonical form:

$$\text{Maximize } z = \qquad 3x_3 - 4x_4 + 28,$$

subject to:

$$
\begin{aligned}
x_1 \quad - x_3 + x_4 &= 6, \\
x_2 - 2x_3 + x_4 &= 2, \\
x_j &\ge 0 \qquad (j = 1, 2, 3, 4).
\end{aligned}
$$

Since the objective coefficient for x_3 is positive and x_3 does not appear in any constraint with a positive coefficient, the solution is unbounded. In fact, as we observed when developing the simplex method, by taking $x_3 = \theta$, the solution approaches $+\infty$ by increasing θ and setting

$$
\begin{aligned}
z &= 28 + 3\theta, \\
x_1 &= 6 + \theta, \\
x_2 &= 2 + 2\theta.
\end{aligned}
$$

This serves to alter $x_1,\ x_2,\ x_3,\ x_4$, from $x_1 = 6,\ x_2 = 2,\ x_3 = 0,\ x_4 = 0$, to $x_1 = 6+\theta,\ x_2 = 2+2\theta,\ x_3 = \theta,\ x_4 = 0$, so that we move in the direction $d = (1, 2, 1, 0)$ by a multiple of θ. This direction has a per-unit profit of 3 and uses 1 unit of the complicating resource. It is the extreme ray added to the restricted master problem, which becomes:

$$
\begin{array}{lr}
z^3 = \text{Max } 20\lambda_1 + 28\lambda_2 + 3\theta_1, & \textit{Optimal shadow prices} \\
\text{subject to:} & \\
4\lambda_1 + 6\lambda_2 + \theta_1 \le 8, & 3 \\
\lambda_1 + \lambda_2 \qquad = 1, & 10 \\
\lambda_1 \ge 0, \quad \lambda_2 \ge 0, \quad \theta_1 \ge 0, &
\end{array}
$$

and has optimal solution $\lambda_1 = 0,\ \lambda_2 = 1,\ \theta_1 = 2$, and $z^3 = 34$.

Since the price of the complicating resource is 3, the new subproblem objective function becomes:

$$v^3 = \text{Max } 5x_1 - x_2 - 3x_1 = \text{Max } 2x_1 - x_2.$$

Graphically we see from Fig. 12.6, that an optimal solution is $x_1 = 6,\ x_2 = 2,\ x_3 = x_4 = 0,\ v^3 = 10$. Since $v^3 \le \sigma^3 = 10$, the last solution solves the full master problem and the procedure terminates. The optimal solution uses the extreme point $(x_1,\ x_2,\ x_3,\ x_4) = (6, 2, 0, 0)$, plus two times the extreme ray $d = (1, 2, 1, 0)$; that is,

$$\begin{aligned} x_1 &= 6 + 2(1) = 8, \quad & x_2 &= 2 + 2(2) = 6, \\ x_3 &= 0 + 2(1) = 2, \quad & x_4 &= 0 + 2(0) = 0. \end{aligned}$$

In general, whenever the subproblem is unbounded, the simplex method determines a canonical form with $\overline{c}_j > 0$ and $\overline{a}_{ij} \le 0$ for each coefficient of some non-basic variable x_j. As above, the extreme ray $d = (d_1, d_2, \ldots, d_n)$ to be submitted to the restricted master problem has a profit coefficient $\overline{c}_j$ and coefficients d_k given by

$$d_k = \begin{cases} 1 & \text{if } k = s \text{ (increasing nonbasic } x_s\text{);} \\ -\overline{a}_{ij} & \text{if } x_k \text{ is the } i\text{th basic variable (changing the basis to compensate} \\ & \qquad \text{for } x_s\text{);} \\ 0 & \text{if } x_k \text{ is nonbasic and } k \ne s \text{ (hold other nonbasics at 0).} \end{cases}$$

The coefficients of this extreme ray simply specify how the values of the basic variables change per unit change in the nonbasic variable x_s being increased.

12.8 COLUMN GENERATION

Large-scale systems frequently result in linear programs with enormous numbers of variables, that is, linear programs such as:

$$z^* = \text{Max } z = c_1x_1 + c_2x_2 \ + \cdots + c_nx_n,$$

subject to:

$$\begin{aligned} a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n &= b_i \qquad (i = 1, 2, \ldots, m), \\ x_j &\ge 0 \qquad (j = 1, 2, \ldots, n), \end{aligned} \tag{12}$$

where n is very large. These problems arise directly from applications such as the multi-item production scheduling example from Section 12.1, or the cutting-stock problem to be introduced below. They may arise in other situations as well. For example, the master problem in decomposition has this form; in this case, problem variables are the weights associated with extreme points and extreme rays.

Because of the large number of variables, direct solution by the simplex method may be inappropriate. Simply generating all the coefficient data a_{ij} usually will prohibit this approach. Column generation extends the technique introduced in the decomposition algorithm, of using the simplex method, but generating the coefficient data only as needed. The method is applicable when the data has inherent structural properties that allow numerical values to be specified easily. In decomposition, for example, we exploited the fact that the data for any variable corresponds to an extreme point or extreme ray of another linear program. Consequently, new data could be generated by solving this linear program with an appropriate objective function.

The column-generation procedure very closely parallels the mechanics of the decomposition algorithm. The added wrinkle concerns the subproblem, which now need not be a linear program, but can be any type of optimization problem, including nonlinear, dynamic, or integer programming problems. As in decomposition, we assume *a priori* that certain variables, say $x_{J+1},\ x_{J+2}, \ldots, x_n$ are nonbasic and restrict their values to zero. The resulting problem is:

$$
\begin{array}{llll}
z^J = \text{Max } c_1x_1 + c_2x_2 + \cdots + c_Jx_J, & & & \textit{Optimal shadow prices} \\
\text{subject to:} & & & \\
a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{iJ}x_J = b_i & (i = 1, 2, \ldots, m), & & \pi_i^J \\
x_j \geq 0 & (j = 1, 2, \ldots, J); & &
\end{array}
\tag{13}
$$

this is now small enough so that the simplex method can be employed for its solution. The original problem (12) includes all of the problem characteristics and again is called a *master problem*, whereas problem (13) is called the *restricted master problem*.

Suppose that the restricted master problem has been solved by the simplex method and that $\pi_1^J, \pi_2^J, \ldots, \pi_m^J$ are the optimal shadow prices. The optimal solution together with $x_{J+1} = x_{J+2} = \cdots = x_n = 0$ is feasible and so potentially optimal for the master problem (12). It is optimal if the simplex optimality condition holds, that is, if $\overline{c}_j = c_j - \sum_{i=1}^m \pi_i^J a_{ij} \leq 0$ for every variable x_j. Stated in another way, the solution to the restricted master problem is optimal if $v^J \leq 0$ where:

$$
v^J = \max_{1 \leq j \leq n} \left[c_j - \sum_{i=1}^m \pi_i^J a_{ij} \right]. \tag{14}
$$

If this condition is satisfied, the original problem has been solved without specifying all of the a_{ij} data or solving the full master problem.

If $v^J = c_s - \sum_{i=1}^m \pi_i^J a_{is} > 0$, then the simplex method, when applied to the master problem, would introduce variable x_s into the basis. Column generation accounts for this possibility by adding variable x_s as a new variable to the restricted master problem. The new restricted master can be solved by the simplex method and the entire procedure can be repeated.

This procedure avoids solving the full master problem; instead it alternately solves a restricted master problem and makes the computations (14) to generate data $a_{1s}, a_{2s}, \ldots, a_{ms}$ for a new variable x_s. Observe that (14) is itself an optimization problem, with variables $j = 1, 2, \ldots, n$. It is usually referred to as a *subproblem*.

The method is specified in flow-chart form in Fig. 12.7. Its efficiency is predicated upon:

i) Obtaining an optimal solution before many columns have been added to the restricted master problem. Otherwise the problems inherent in the original formulation are encountered.

ii) Being able to solve the subproblem effectively.

Details concerning the subproblem depend upon the structural characteristics of the problem being studied. By considering a specific example, we can illustrate how the subproblem can be an optimization problem other than a linear program.

Example.

(Cutting-stock problem) A paper (textile) company must produce various sizes of its paper products to meet demand. For most grades of paper, the production technology makes it much easier to first produce the paper on large rolls, which are then cut into smaller rolls of the required sizes. Invariably, the cutting process involves some waste. The company would like to minimize waste or, equivalently, to meet demand using the fewest number of rolls.

For notational purposes, assume that we are interested in one grade of paper and that this paper is produced only in rolls of length ℓ for cutting. Assume, further, that the demand requires d_i rolls of size ℓ_i $(i = 1, 2, \ldots, m)$ to be cut. In order for a feasible solution to be possible, we of course need $\ell_i \leq \ell$.

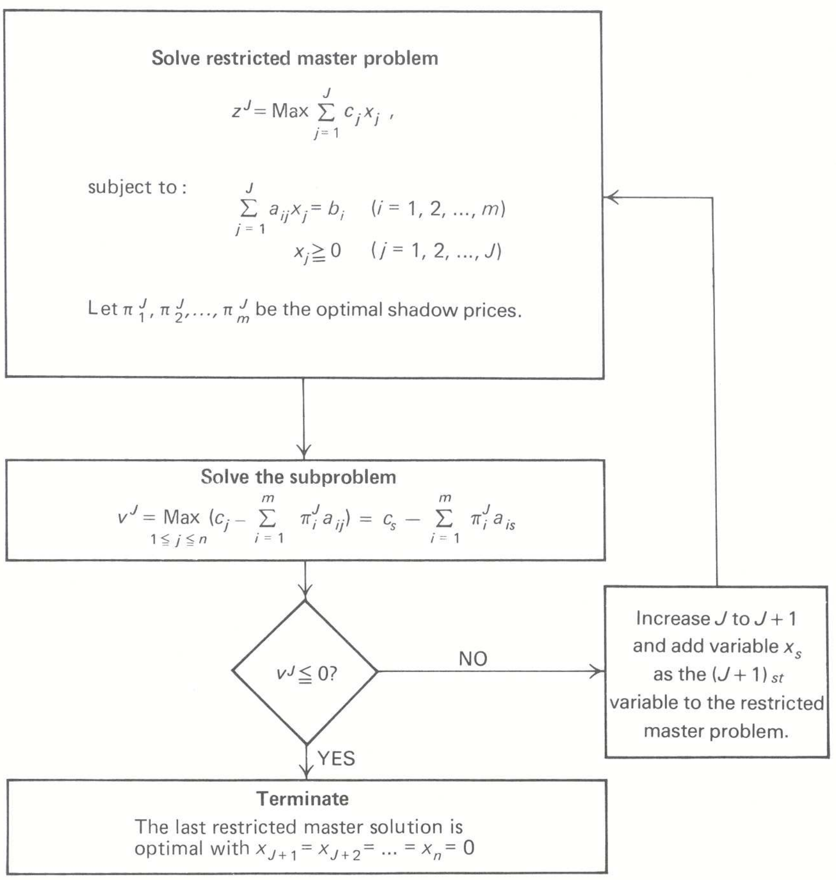

Figure 12.7 Column generation.

One approach to the problem is to use the possible cutting patterns for the rolls as decision variables. Consider, for example, $\ell = 200$ inches and rolls required for 40 different lengths ℓ_i ranging from 20 to 80 inches. One possible cutting pattern produces lengths of

$$35'', \quad 40'', \quad 40'', \quad 70'',$$

with a waste of 15 inches. Another is

$$20'', \quad 25'', \quad 30'', \quad 50'', \quad 70'',$$

with a waste of 5 inches. In general, let

n = Number of possible cutting patterns,
x_j = Number of times cutting pattern j is used,
a_{ij} = Number of rolls of size ℓ_i used on the jth cutting pattern.

Then $a_{ij}x_j$ is the number of rolls of size ℓ_i cut using pattern j, and the problem of minimizing total rolls used to fulfill demand becomes:

$$\text{Minimize } x_1 + \quad x_2 + \cdots + \quad x_n,$$

subject to:

$$\begin{aligned} a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n &\geq d_i & (i = 1, 2, \ldots, m),\\ x_j &\geq 0 \quad \text{and integer} & (j = 1, 2, \ldots, n). \end{aligned}$$

For the above illustration, the number of possible cutting patterns, n, exceeds 10 million, and this problem is alarge-scale integer-programming problem. Fortunately, the demands d_i are usually high, so that rounding optimal linear-programming solutions to integers leads to good solutions.

If we drop the integer restrictions, the problem becomes a linear program suited for the column-generation algorithm. The subproblem becomes:

$$v^J = \underset{1 \leq j \leq n}{\text{Min}} \left[1 - \sum_{i=1}^{m} \pi_i^J a_{ij} \right], \tag{15}$$

since each objective coefficient is equal to one. Note that the subproblem is a minimization, since the restricted master problem is a minimization problem seeking variables x_j with $\overline{c}_j < 0$ (as opposed to seeking $\overline{c}_j > 0$ for maximization).

The subproblem considers all potential cutting plans. Since a cutting plan j is feasible whenever

$$\sum_{i=1}^{m} \ell_i a_{ij} \leq \ell, \tag{16}$$

$$a_{ij} \geq 0 \text{ and integer},$$

the subproblem must determine the coefficients a_{ij} of a new plan to minimize (15). For example, if the roll length is given by $\ell = 100''$ and the various lengths ℓ_i to be cut are 25, 30, 35, 40, 45, and 50$''$, then the subproblem constraints become:

$$\begin{aligned} &25a_{1j} + 30a_{2j} + 35a_{3j} + 40a_{4j} + 45a_{5j} + 50a_{6j} \leq 100,\\ &a_{ij} \geq 0 \quad \text{and integer} \qquad (i = 1, 2, \ldots, 6). \end{aligned}$$

The optimal values for a_{ij} indicate how many of each length ℓ_i should be included in the new cutting pattern j. Because subproblem (15) and (16) is a one-constraint integer-programming problem (called a *knapsack* problem), efficient special-purpose dynamic-programming algorithms can be used for its solution.

As this example illustrates, column generation is a flexible approach for solving linear programs with many columns. To be effective, the algorithm requires that the subproblem can be solved efficiently, as in decomposition or the cutting-stock problem, to generate a new column or to show that the current restricted master problem is optimal. In the next chapter, we discuss another important application by using column generation to solve *nonlinear* programs.

EXERCISES

1. Consider the following linear program:

$$\text{Maximize } 9x_1 + \quad x_2 \;- 15x_3 - \; x_4,$$

subject to:

$$\begin{aligned} -\,3x_1 + \;\; 2x_2 + \;\; 9x_3 + \;\; x_4 &\le 7, \\ 6x_1 + 16x_2 - 12x_3 - 2x_4 &\le 10, \\ 0 \le x_j \le 1 \qquad (j &= 1, 2, 3, 4). \end{aligned}$$

Assuming no bounded-variable algorithm is available, solve by the decomposition procedure, using $0 \le x_j \le 1$ $(j = 1, 2, 3, 4)$ as the subproblem constraints.

Initiate the algorithm with two proposals: the optimum solution to the subproblem and the proposal $x_1 = 1$, $x_2 = x_3 = x_4 = 0$.

Consider the following linear-programming problem with special structure:

$$\text{Maximize } z = 15x_1 + \; 7x_2 \; + \; 15x_3 \; + \; 20y_1 \; + \; 12y_2,$$

subject to:

$$\left.\begin{aligned} x_1 + \;\; x_2 + \;\; x_3 + \;\; y_1 + \; y_2 &\le 5 \\ 3x_1 + 2x_2 + 4x_3 + 5y_1 + 2y_2 &\le 16 \end{aligned}\right\} \text{Master problem}$$

$$\left.\begin{aligned} 4x_1 + 4x_2 + 5x_3 &\le 20 \\ 2x_1 + \;\; x_2 &\le 4 \\ x_1 \ge 0, x_2 \ge 0, x_3 &\ge 0 \end{aligned}\right\} \text{Subproblem I}$$

$$\left.\begin{aligned} y_1 + \tfrac{1}{2}y_2 &\le 3 \\ \tfrac{1}{2}y_1 + \tfrac{1}{2}y_2 &\le 2 \\ y_1 \ge 0, \; y_2 &\ge 0. \end{aligned}\right\} \text{Subproblem I}$$

Tableau 1 represents the solution of this problem by the decomposition algorithm in the midst of the calculations. The variables s_1 and s_2 are slack variables for the first two constraints of the master problem; the variables a_1 and a_2 are artificial variables for the weighting constraints of the master problem.

Tableau 1 Data at an iteration of the decomposition method

Basic variables	Current values	*Subproblem I* λ_1	λ_2	*Subproblem II* μ_1	μ_2	*Slacks* s_1	s_2	*Artificials* a_1	a_2
λ_1	$\frac{1}{2}$	1				$\frac{1}{3}$	$-\frac{1}{6}$	$\frac{4}{3}$	
λ_2	$\frac{2}{3}$		1			$-\frac{1}{3}$	$\frac{1}{6}$	$-\frac{1}{3}$	
μ_1	$\frac{5}{12}$			1		$\frac{5}{12}$	$-\frac{1}{12}$	$-\frac{1}{3}$	
μ_2	$\frac{7}{12}$				1	$-\frac{5}{12}$	$\frac{1}{12}$	$\frac{1}{3}$	1
$(-z)$	$-\frac{80}{3}$					-10	-1	-4	

The extreme points generated thus far are:

x_1	x_2	x_3	*Weights*
2	0	0	λ_1
0	0	4	λ_2

for subproblem I and

y_1	y_2	*Weights*
0	4	μ_1
0	0	μ_2

for subproblem II

a) What are the shadow prices associated with each constraint of the restricted master?
b) Formulate the two subproblems that need to be solved at this stage using the shadow prices determined in part (a).
c) Solve each of the subproblems graphically.
d) Add any newly generated extreme points of the subproblems to the restricted master.
e) Solve the new restricted master by the simplex method continuing from the previous solution. (See Exercise 29 in Chapter 4.)
f) How do we know whether the current solution is optimal?

3. Consider a transportation problem for profit maximization:

$$\text{Maximize } z = c_{11}x_{11} + c_{12}x_{12} + c_{13}x_{13} + c_{21}x_{21} + c_{22}x_{22} + c_{23}x_{23},$$

subject to:

$$\begin{aligned}
x_{11} + x_{12} + x_{13} \qquad\qquad\qquad\qquad\qquad &= a_1,\\
x_{21} + x_{22} + x_{23} &= a_2,\\
x_{11} \qquad\qquad + x_{21} \qquad\qquad &= b_1,\\
x_{12} \qquad\qquad + x_{22} \qquad &= b_2,\\
x_{13} \qquad\qquad + x_{23} &= b_3,
\end{aligned}$$

$$x_{ij} \geq 0 \qquad (i = 1, 2;\ j = 1, 2, 3).$$

a) Suppose that we solve this problem by decomposition, letting the requirement b_i constraints and nonnegativity $x_{ij} \geq 0$ constraints compose the subproblem. Is it easy to solve the subproblem at each iteration? Does the restricted master problem inherit the network structure of the problem, or is the network structure ''lost'' at the master-problem level?
b) Use the decomposition procedure to solve for the data specified in the following table:

	Distribution profits (c_{ij})			*Availabilities* a_i
	100	150	200	20
	50	50	75	40
Requirements b_i	10	30	20	

Initiate the algorithm with two proposals:

Activity levels

	x_{11}	x_{12}	x_{13}	x_{21}	x_{22}	x_{23}	*Profit*
Proposal 1	0	0	0	10	30	20	9500
Proposal 2	10	30	20	0	0	0	3500

To simplify calculations, you may wish to use the fact that a transportation problem contains a redundant equation and remove the second supply equation from the problem.

4. A small city in the northeast must racially balance its 10 elementary schools or sacrifice all federal aid being issued to the school system. Since the recent census indicated that approximately 28% of the city's population is composed of minorities, it has been determined that each school in the city must have a minority student population of 25% to 30% to satisfy the federal definition of ''racial balance.'' The decision has been made to bus children in order to meet this goal. The parents of the children in the 10 schools are very concerned about the additional travel time for the children who will be transferred to new schools. The School Committee has promised these parents that the busing plan will minimize the total time that the children of the city have to travel. Each school district is divided into 2 zones, one which is close to the local school and one which is far from the school, as shown in Fig. E12.1.

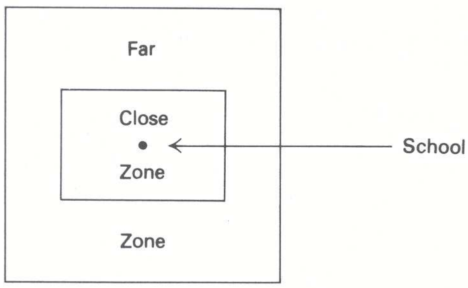

Figure E12.1

The School Committee has also promised the parents of children who live in a ''close zone'' that they will attempt to discourage the busing of this group of children (minority and nonminority) away from their present neighborhood school. The School Committee members are intent on keeping their promises to this group of parents.

An additional problem plaguing the Committee is that any school whose enrollment drops below 200 students must be closed; this situation would be unacceptable to the Mayor and to the taxpayers who would still be supporting a ''closed school'' serving no one.

The available data include the following:

For each district $i = 1, 2, \ldots, 10$, we have

$$
\begin{aligned}
N_i^{\text{NONc}} &= \text{Number of nonminority children in the close zone of school district } i.\\
N_i^{\text{MINc}} &= \text{Number of minority children in the close zone of school district } i.\\
N_i^{\text{NONf}} &= \text{Number of nonminority children in the far zone of school district } i.\\
N_i^{\text{MINf}} &= \text{Number of minority children in the far zone of school district } i.
\end{aligned}
$$

For each pair (i, j), of school districts, we have the travel time t_{ij}.

For each school i, the capacity D_i is known (all $D_i > 200$ and there is enough school capacity to accommodate all children).

a) Formulate the problem as a linear program. [*Hint.* To discourage the busing of students who live close to their neighborhood school, you may add a penalty, p, to the travel time to any student who lives in the close zone of school district i and is assigned to school district j $(i \neq j)$. Assume that a student who lives in the close zone of school i and is assigned to school i does not have to be bused.]

b) There is only a small-capacity minicomputer in the city, which cannot solve the linear program in its entirety. Hence, the decomposition procedure could be applied to solve the problem. If you were a mathematical programming specialist hired by the School Committee, how would you decompose the program formulated in part (a)? Identify the subproblem, the weighting program, and the proposal-generating program. Do not attempt to solve the problem.

5. A food company blames seasonality in production for difficulties that it has encountered in scheduling its activities efficiently. The company has to cope with three major difficulties:

I) Its food products are perishable. On the average, one unit spoils for every seven units kept in inventory from one month to another.

II) It is costly to change the level of the work force to coincide with requirements imposed by seasonal demands. It costs $750 to hire and train a new worker, and $500 to fire a worker.
III) On the average, one out of eight workers left idle in any month decides to leave the firm.

Because of the ever-increasing price of raw materials, the company feels that it should design a better scheduling plan to reduce production costs, rather than lose customers by increasing prices of its products.

The task of the team hired to study this problem is made easier by the following operating characteristics of firm:

i) Practically, the firm has no problems procuring any raw materials that it requires;
ii) Storage capacity is practically unlimited at the current demand level; and
iii) The products are rather homogeneous, so that all output can be expressed in standard units (by using certain equivalence coefficients).

The pertinent information for decision-making purposes is:

iv) The planning horizon has $T = 12$ months (one period = one month);
v) Demand D_i is known for each period ($i = 1, 2, \ldots, 12$);
vi) Average productivity is 1100 units per worker per month;
vii) The level of the work force at the start of period 1 is L_1; S_0 units of the product are available in stock at the start of period 1;
viii) An employed worker is paid W_t as wages per month in period t;
ix) An idle worker is paid a minimum wage of M_t in month t, to be motivated not to leave;
x) It costs I dollars to keep one unit of the product in inventory for one month.

With the above information, the company has decided to construct a pilot linear program to determine work-force level, hirings, firings, inventory levels, and idle workers.

a) Formulate the linear program based on the data above. Show that the model has a staircase structure.
b) Restate the constraints in terms of cumulative demand and work force; show that the model now has block triangular structure.

6. A firm wishing to operate with as decentralized an organizational structure as possible has two separate operating divisions. The divisions can operate independently except that they compete for the firm's two scarce resources—working capital and a particular raw material. Corporate headquarters would like to set prices for the scarce resources that would be paid by the divisions, in order to ration the scarce resources. The goal of the program is to let each division operate independently with as little interference from corporate headquarters as possible.

Division #1 produces 3 products and faces capacity constraints as follows:

$$\begin{aligned} 4x_1 + 4x_2 + 5x_3 &\le 20, \\ 4x_1 + 2x_2 &\le 8, \\ x_1 \ge 0, \quad x_2 \ge 0, \quad x_3 &\ge 0. \end{aligned}$$

The contribution to the firm per unit from this division's products are 2.50, 1.75, and 0.75, respectively. Division #2 produces 2 different products and faces its own capacity constraints as follows:

$$\begin{aligned} 2y_1 + y_2 &\le 6, \\ y_1 + y_2 &\le 4, \\ y_1 \ge 0, \quad y_2 &\ge 0. \end{aligned}$$

The contribution to the firm per unit from this division's products are 3 and 2, respectively. The joint constraints that require coordination among the divisions involve working capital and one raw material. The constraint on working capital is

$$x_1 + x_2 + x_3 + y_1 + y_2 \le 7,$$

and the constraint on the raw material is

$$3x_1 + 2x_2 + 4x_3 + 5y_1 + 2y_2 \le 16.$$

Corporate headquarters has decided to use decomposition to set the prices for the scarce resources. The optimal solution using the decomposition algorithm indicated that division #1 should produce $x_1 = 1$, $x_2 = 2$, and $x_3 = 0$, while division #2 should produce $y_1 = \frac{1}{2}$ and $y_2 = 3\frac{1}{2}$. The shadow prices turned out to be $\frac{2}{3}$ and $\frac{1}{3}$ for working capital and raw material, respectively. Corporate headquarters congratulated itself for a fine price of analysis. They then announced these prices to the divisions and told the divisions to optimize their own operations independently. Division #1 solved its subproblem and reported an operating schedule of $x_1 = 0$, $x_2 = 4$, $x_3 = 0$. Similarly, division #2 solved its subproblem and reported an operating schedule of $y_1 = 2$, $y_2 = 2$.

Corporate headquarters was aghast—together the divisions requested more of both working capital and the raw material than the firm had available!

a) Did the divisions cheat on the instructions given them by corporate headquarters?
b) Were the shadow prices calculated correctly?
c) Explain the paradox.
d) What can the corporate headquarters do with the output of the decomposition algorithm to produce overall optimal operations?

7. For the firm described in Exercise 6, analyze the decomposition approach in detail.

a) Graph the constraints of each subproblem, division #1 in three dimensions and division #2 in two dimensions.
b) List *all* the extreme points for each set of constraints.
c) Write out the full master problem, including all the extreme points.
d) The optimal shadow prices are $\frac{2}{3}$ and $\frac{1}{3}$ on the working capital and raw material, respectively. The shadow prices on the weighting constraints are $\frac{5}{8}$ and $\frac{8}{3}$ for divisions #1 and #2, respectively. Calculate the reduced costs of all variables.
e) Identify the basic variables and determine the weights on each extreme point that form the optimal solution.
f) Solve the subproblems using the above shadow prices. How do you know that the solution is optimal after solving the subproblems?
g) Show graphically that the optimal solution to the overall problem is not an extreme solution to either subproblem.

8. To plan for long-range energy needs, a federal agency is modeling electrical-power investments as a linear program. The agency has divided its time horizon of 30 years into six periods $t = 1, 2, \ldots, 6$, of five years each. By the end of each of these intervals, the government can construct a number of plants (hydro, fossil, gas turbine, nuclear, and so forth). Let x_{ij} denote the capacity of plant j when initiated at the end of interval i, with per-unit construction cost of c_{ij}. Quantities x_{0j} denote capacities of plants currently in use.

Given the decisions x_{ij} on plant capacity, the agency must decide how to operate the plants to meet energy needs. Since these decisions require more detailed information to account for seasonal variations in energy demand, the agency has further divided each of the time intervals $t = 1, 2, \ldots, 6$ into 20 subintervals $s = 1, 2, \ldots, 20$. The agency has estimated the electrical demand in each (interval t, subinterval s) combination as d_{ts}. Let o_{ijts} denote the operating level during the time period ts of plant j that has been constructed in interval i. The plants must be used to meet demand requirements and incur per-unit operating costs of v_{ijts}. Because of operating limitations and aging, the plants cannot always operate at full construction capacity. Let a_{ijt} denote the availability during time period t of plant j that was constructed in time interval i. Typically, the coefficient a_{ijt} will be about 0.9. Note that $a_{ijt} = 0$ for $t \leq i$, since the plant is not available until after the end of its construction interval i.

To model uncertainties in its demand forecasts, the agency will further constrain its construction decisions by introducing a margin m of reserve capacity; in each period the total operating capacity from all plants must be at least as large as $d_{ts}(1 + m)$.

Finally, the total output of hydroelectric power in any time interval t cannot exceed the capacity H_{it} imposed by availability of water sources. (In a more elaborate model, we might incorporate H_{it} as a decision variable.)

The linear-programming model developed by the agency is:

$$\text{Minimize } \sum_{j=1}^{20} \sum_{i=1}^{6} c_{ij} x_{ij} + \sum_{j=1}^{20} \sum_{t=1}^{6} \sum_{i=0}^{6} \sum_{s=1}^{20} v_{ijts} o_{ijts} \theta_s,$$

subject to:

$$\begin{array}{lll}
\sum_{j=1}^{20}\sum_{i=0}^{6} o_{ijts} \geq d_{ts} & & (t=1,2,\ldots,6; \quad s=1,2,\ldots,20), \\
o_{ijts} \leq a_{ijt}x_{ij} & & (i=0,1,\ldots,6; \quad t=1,2,\ldots,6; \\
 & & \; j=1,2,\ldots,20; \; s=1,2,\ldots,20), \\
\sum_{s=1}^{20} o_{ihts}\,\theta_s \leq H_{it} & & (t=1,2,\ldots,6; \quad i=0,1,\ldots,6), \\
\sum_{j=1}^{20}\sum_{i=0}^{6} x_{ij} \geq d_{ts}(1+m) & & (t=1,2,\ldots,6; \quad s=1,2,\ldots,20), \\
x_{ij} \geq 0, \quad o_{ijts} \geq 0 & & (i=0,1,\ldots,6; \quad t=1,2,\ldots,6; \\
 & & \; j=1,2,\ldots,20; \; s=1,2,\ldots,20).
\end{array}$$

In this formulation, θ_s denotes the length of time period s; the values of x_{ij} are given. The subscript h denotes hydroelectric.

a) Interpret the objective function and each of the constraints in this model. How large is the model?
b) What is the structure of the constraint coefficients for this problem?
c) Suppose that we apply the decomposition algorithm to solve this problem; for each plant j and time period t, let the constraints

$$\begin{array}{ll}
o_{ijts} \leq a_{ijt}x_{ij} & (i=0,1,\ldots,6; \quad s=1,2,\ldots,20), \\
o_{ijts} \geq 0, \quad x_{ij} \geq 0 & (i=0,1,\ldots,6; \quad s=1,2,\ldots,20),
\end{array}$$

form a subproblem. What is the objective function to the subproblem at each step? Show that each subproblem either solves at $o_{ijts} = 0$ and $x_{ij} = 0$ for all i and s, or is unbounded. Specify the steps for applying the decomposition algorithm with this choice of subproblems.
d) How would the application of decomposition discussed in part (c) change if the constraints

$$\sum_{s=1}^{20} o_{ihts}\theta_s \leq H_{it}, \qquad (i=0,1,\ldots,6),$$

are added to each subproblem in which $j = h$ denotes a hydroelectric plant?

9. The decomposition method can be interpreted as a ''cutting-plane'' algorithm. To illustrate this viewpoint, consider the example:

$$\text{Maximize } z = 3x_1 + 8x_2,$$

subject to:

$$\begin{aligned}
2x_1 + 4x_2 &\leq 3, \\
0 \leq x_1 &\leq 1, \\
0 \leq x_2 &\leq 1.
\end{aligned}$$

Applying decomposition with the constraints $0 \leq x_1 \leq 1$ and $0 \leq x_2 \leq 1$ as the subproblem, we have four extreme points to the subproblem:

	Weights
Extreme point 1: $x_1 = 0$, $x_2 = 0$	λ_1
Extreme point 2: $x_1 = 0$, $x_2 = 1$	λ_2
Extreme point 3: $x_1 = 1$, $x_2 = 0$	λ_3
Extreme point 4: $x_1 = 1$, $x_2 = 1$	λ_4

Evaluating the objective function $3x_1 + 8x_2$ and resource usage $2x_1 + 4x_2$ at these extreme-point solutions gives the following master problem:

$$
\begin{array}{llr}
\text{Maximize } z = 0\lambda_1 + 8\lambda_2 + 3\lambda_3 + 11\lambda_4, & & \textit{Dual} \\
\text{subject to:} & & \underline{\textit{variables}} \\
& 0\lambda_1 + 4\lambda_2 + 2\lambda_3 + 6\lambda_4 \le 3, & \pi \\
& \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1, & \sigma \\
& \lambda_j \ge 0 \quad (j = 1, 2, 3, 4). &
\end{array}
$$

a) Let the variable w be defined in terms of the dual variables π and σ as $w = \sigma + 3\pi$. Show that the dual to the master problem in terms of w and π is:

Minimize w,

subject to:

$$
\begin{aligned}
w - 3\pi &\ge 0, \\
w + \pi &\ge 8, \\
w - \pi &\ge 3, \\
w + 3\pi &\ge 11.
\end{aligned}
$$

b) Figure E12.2 depicts the feasible region for the dual problem. Identify the optimal solution to the dual problem in this figure. What is the value of z^*, the optimal objective value of the original problem?

c) Suppose that we initiate the decomposition with a restricted master problem containing only the third extreme point $x_1 = 1$ and $x_2 = 0$. Illustrate the feasible region to the dual of this restricted master in terms of w and π, and identify its optimal solution w^* and π^*. Does this dual feasible region contain the feasible region to the full dual problem formulated in part (a)?

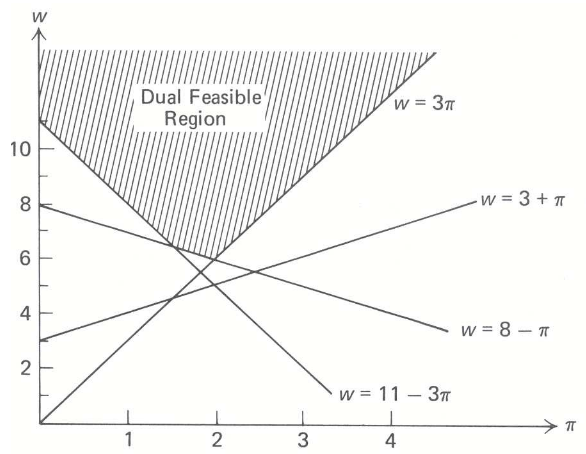

Figure E12.2 Dual feasible region.

d) Show that the next step of the decomposition algorithm adds a new constraint to the dual of the restricted master problem. Indicate which constraint in Fig. E12.2 is added next. Interpret the added constraint as ''cutting away'' the optimal solution w^* and π^* found in part (c) from the feasible region. What are the optimal values of the dual variables after the new constraint has been added?

e) Note that the added constraint is found by determining which constraint is most violated at $\pi = \pi^*$; that is, by moving vertically in Fig. E12.2 at $\pi = \pi^*$, crossing all violated constraints until we reach the dual feasible region at $w = \hat{w}$. Note that the optimal objective value z^* to the original problem satisfies the inequalities:

$$w^* \le z^* \le \hat{w}.$$

Relate this bound to the bounds discussed in this chapter.

f) Solve this problem to completion, using the decomposition algorithm. Interpret the solution in Fig. E12.2, indicating at each step the cut and the bounds on z^*.

g) How do extreme rays in the master problem alter the formulation of the dual problem? How would the cutting-plane interpretation discussed in this problem be modified when the subproblem is unbounded?

10. In this exercise we consider a two-dimensional version of the cutting stock problem.

a) Suppose that we have a W-by-L piece of cloth. The material can be cut into a number of smaller pieces and sold. Let π_{ij} denote the revenue for a smaller piece with dimensions w_i by ℓ_j $(i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$.

Operating policies dictate that we first cut the piece along its width into strips of size w_i. The strips are then cut into lengths of size ℓ_j. Any waste is scrapped, with no additional revenue.

For example, a possible cutting pattern for a 9-by-10 piece might be that shown in Fig. E12.3. The shaded regions correspond to trim losses. Formulate a (nonlinear) integer program for finding the maximum-revenue

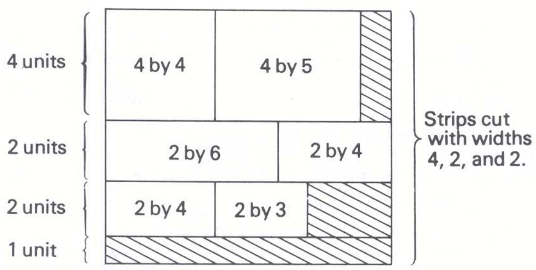

Figure E12.3

cutting pattern. Can we solve this integer program by solving several knapsack problems? [*Hint.* Can we use the same-length cuts in any strips with the same width? What is the optimal revenue v_i obtained from a strip of width w_i? What is the best way to choose the widths w_i to maximize the total value of the v_i's?]

b) A firm has unlimited availabilities of W-by-L pieces to cut in the manner described in part (a). It must cut these pieces into smaller pieces in order to meet its demand of d_{ij} units for a piece with width w_i and length $\ell_j (i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$. The firm wishes to use as few W-by-L pieces as possible to meet its sales commitments.

Formulate the firm's decision-making problem in terms of cutting patterns. How can column generation be used to solve the linear-programming approximation to the cutting-pattern formulation?

11. In Section 12.1 we provided a formulation for a large-scale multi-item production-scheduling problem. The purpose of this exercise (and of Exercises 12 and 13) is to explore the implications of the suggested formulation, as well as techniques that can be developed to solve the problem.

The more classical formulation of the multi-item scheduling problem can be stated as follows:

$$\text{Minimize } z = \sum_{j=1}^{J} \sum_{t=1}^{T} [s_{jt}\delta(x_{jt}) + v_{jt}x_{jt} + h_{jt}I_{jt}],$$

subject to:

$$\begin{aligned}
& x_{jt} + I_{j,t-1} - I_{jt} = d_{jt} && (t = 1, 2, \ldots, T;\ j = 1, 2, \ldots, J),\\
& \sum_{j=1}^{J} [\ell_j \delta(x_{jt}) + k_j x_{jt}] \leq b_t && (t = 1, 2, \ldots, T),\\
& x_{jt} \geq 0, \quad I_{jt} \geq 0 && (t = 1, 2, \ldots, T;\ j = 1, 2, \ldots, J),
\end{aligned}$$

where

$$\delta(x_{jt}) = \begin{cases} 0 & \text{if } x_{jt} = 0,\\ 1 & \text{if } x_{jt} > 0, \end{cases}$$

and x_{jt} = Units of item j to be produced in period t,
I_{jt} = Units of inventory of item j left over at the end of period t,
s_{jt} = Setup cost of item j in period t,
v_{jt} = Unit production cost of item j in period t,
h_{jt} = Inventory holding cost for item j in period t,
d_{jt} = Demand for item j in period t,
ℓ_j = Down time consumed in performing a setup for item j,
k_j = Man-hours required to produce one unit of item j,
b_t = Total man-hours available for period t.

a) Interpret the model formulation. What are the basic assumptions of the model? Is there any special structure to the model?
b) Formulate an equivalent (linear) mixed-integer program for the prescribed model. If $T = 12$ (that is, we are planning for twelve time periods) and $J = 10{,}000$ (that is, there are 10,000 items to schedule), how many integer variables, continuous variables, and constraints does the model have? Is it feasible to solve a mixed-integer programming model of this size?

12. Given the computational difficulties associated with solving the model presented in Exercise 11, A. S. Manne conceived of a way to approximate the mixed-integer programming model as a linear program. This transformation is based on defining for each item j a series of production sequences over the planning horizon T. Each sequence is a set of T nonnegative integers that identify the amount to be produced of item j at each time period t during the planning horizon, in such a way that demand requirements for the item are met. It is enough to consider production sequences such that, at a given time period, the production is either zero or the sum of consecutive demands for some number of periods into the future. This limits the number of production sequences to a total of 2^{T-1} for each item. Let

x_{jkt} = amount to be produced of item j in period t by means of production sequence k.

To illustrate how the production sequences are constructed, assume that $T = 3$. Then the total number of production sequences for item j is $2^{3-1} = 4$. The corresponding sequences are given in Table E12.1.

Table 12.11

Sequence number	*Time period* $t = 1$	$t = 2$	$t = 3$
$k = 1$	$x_{j11} = d_{j1} + d_{j2} + d_{j3}$	$x_{j12} = 0$	$x_{j13} = 0$
$k = 2$	$x_{j21} = d_{j1} + d_{j2}$	$x_{j22} = 0$	$x_{j23} = d_{j3}$
$k = 3$	$x_{j31} = d_{j1}$	$x_{j32} = d_{j2} + d_{j3}$	$x_{j33} = 0$
$k = 4$	$x_{j41} = d_{j1}$	$x_{j42} = d_{j2}$	$x_{j43} = d_{j3}$

The total cost associated with sequence k for the production of item j is given by

$$c_{jk} = \sum_{t=1}^{T} [s_{jt}\delta(x_{jkt}) + v_{jt}x_{jkt} + h_{jt}I_{jt}],$$

and the corresponding man-hours required for this sequence in period t is

$$a_{jkt} = \ell_j\delta(x_{jkt}) + k_j x_{jkt}.$$

a) Verify that, if the model presented in Exercise 11 is restricted to producing each item in production sequences, then it can be formulated as follows:

$$\text{Minimize } z = \sum_{j=1}^{J}\sum_{k=1}^{K} c_{jk}\theta_{jk},$$

subject to:

$$\sum_{j=1}^{J}\sum_{k=1}^{K} a_{jkt}\theta_{jk} \le b_t \qquad (t = 1, 2, \ldots, T),$$

$$\sum_{j=1}^{J} \theta_{jk} = 1 \qquad (k = 1, 2, \ldots, K),$$

$$\theta_{jk} \ge 0 \quad \text{and integer} \qquad (j = 1, 2, \ldots, J;\ k = 1, 2, \ldots, K).$$

b) Study the structure of the resulting model. How could you define the structure? For $T = 12$ and $J = 10{,}000$, how many rows and columns does the model have?

c) Under what conditions can we eliminate the integrality constraints imposed on variables θ_{jk} without significantly affecting the validity of the model? [*Hint.* Read the comment made on the multi-term scheduling problem in Section 12.1 of the text.]

d) Propose a decomposition approach to solve the resulting large-scale linear-programming model. What advantages and disadvantages are offered by this approach? (Assume that at this point the resulting subproblems are easy to solve. See Exercise 13 for details.)

13. Reconsider the large-scale linear program proposed in the previous exercise:

$$\text{Minimize } z = \sum_{j=1}^{J}\sum_{k=1}^{K} c_{jk}\theta_{jk},$$

subject to:

$$\sum_{j=1}^{J}\sum_{k=1}^{K} a_{jkt}\theta_{jk} \le b_t \qquad (t = 1, 2, \ldots, T), \tag{1}$$

$$\sum_{j=1}^{J} \theta_{jk} = 1 \qquad (k = 1, 2, \ldots, K), \tag{2}$$

$$\theta_{jk} \ge 0 \qquad (j = 1, 2, \ldots, J;\ k = 1, 2, \ldots, K), \tag{3}$$

a) Let us apply the column-generation algorithm to solve this problem. At some stage of the process, let π_t for $t = 1, 2, \ldots, T$ be the shadow prices associated with constraints (1), and let π_{T+k} for $k = 1, 2, \ldots, K$ be the shadow prices associated with constraints (2), in the restricted master problem. The reduced cost c_{jk} for variable θ_{jk} is given by the following expression:

$$\overline{c}_{jk} = c_{jk} - \sum_{t=1}^{T} \pi_t a_{jkt} - \pi_{T+k}.$$

Show, in terms of the original model formulation described in Exercise 11, that $\overline{c}_{jk}$ is defined as:

$$\overline{c}_{jk} = \sum_{t=1}^{T} \left[(s_{jt} - \pi_t \ell_j)\delta(x_{jkt}) + (v_{jt} - \pi_t k_j)x_{jkt} + h_{jt} I_{jt}\right] - \pi_{T+k}.$$

b) The subproblem has the form:

$$\underset{j}{\text{Minimize}}\ [\underset{k}{\text{minimize}}\ \overline{c}_{jk}].$$

The inner minimization can be interpreted as finding the minimum-cost production sequence for a specific item j. This problem can be interpreted as an uncapacitated single-item production problem under fluctuating demand requirements d_{jt} throughout the planning horizon $t = 1, 2, \ldots, T$. Suggest an effective dynamic-programming approach to determine the optimum production sequence under this condition.

c) How does the above approach eliminate the need to generate all the possible production sequences for a given item j? Explain the interactions between the master problem and the subproblem.

14. The ''traffic-assignment'' model concerns minimizing travel time over a large network, where traffic enters the network at a number of origins and must flow to a number of different destinations. We can consider this model as a multicommodity-flow problem by defining a commodity as the traffic that must flow between a particular origin–destination pair. As an alternative to the usual node–arc formulation, consider chain flows. A *chain* is merely a *directed* path through a network from an origin to a destination. In particular, let

$$a_{ij}^k = \begin{cases} 1 & \text{if arc } i \text{ is in chain } j\text{, which connects origin–destination pair } k, \\ 0 & \text{otherwise.} \end{cases}$$

In addition, define

$$z_j^k = \text{Flow over chain } j \text{ between origin–destination pair } k.$$

For example, the network in Fig. E12.4, shows the arc flows of one of the commodities, those vehicles entering node 1 and flowing to node 5. The chains connecting the origin–destination pair 1–5 can be used to express the flow in this network as:

	Chain 1	*Chain 2*	*Chain 3*	*Chain 4*	*Chain 5*
Chain j	1–2–5	1–2–4–5	1–4–5	1–3–5	1–3–4–5
Flow value z_j	3	1	2	3	2

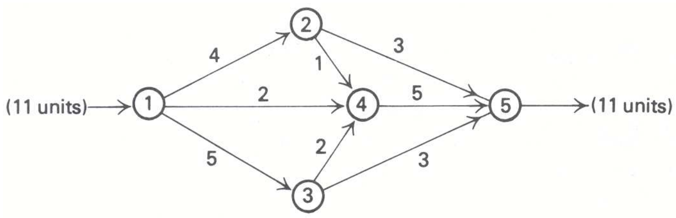

Figure E12.4

Frequently an upper bound u_i is imposed upon the total flow on each arc i. These restrictions are modeled as:

$$\sum_{j_1} a_{ij}^1 z_j^1 + \sum_{j_2} a_{ij}^2 z_j^2 + \cdots + \sum_{j_k} a_{ij}^K z_j^K \le u_i \quad (i = 1, 2, \ldots, I).$$

The summation indices j_k correspond to chains joining the kth origin–destination pair. The total number of arcs is I and the total number of origin–destination pairs is K. The requirement that certain levels of traffic must flow between origin–destination pairs can be formulated as follows:

$$\sum_{j_k} z_j^k = v_k \quad (k = 1, 2, \ldots, K),$$

where

$$v_k = \text{Required flow between origin–destination pair (commodity) } k.$$

Finally, suppose that the travel time over any arc is t_i, so that, if x_i units travel over arc i, the total travel time on arc on arc i is $t_i x_i$.

a) Complete the ''arc–chain'' formulation of the traffic-assignment problem by specifying an objective function that minimizes total travel time on the network. [*Hint*. Define the travel time over a chain, using the a_{ij} data.]

b) In reality, generating all the chains of a network is very difficult computationally. Suppose enough chains have been generated to determine a basic feasible solution to the linear program formulated in part (a). Show how to compute the reduced cost of the next chain to enter the basis from those generated thus far.

c) Now consider the chains not yet generated. In order for the current solution to be optimal, the minimum reduced costs of these chains must be nonnegative. How would you find the chain with the minimum reduced cost for each ''commodity''? [*Hint*. The reduced costs are, in general,

$$\overline{c}_j^k = \sum_i a_{ij}(t_i - \pi_i) - u_k,$$

where π_i and u_k are the shadow prices associated with the capacity restriction on the ith constraint and flow requirement between the kth origin–destination pair. What is the sign of π_i?]

d) Give an economic interpretation of π_i. In the reduced cost of part (c), do the values of π_i depend on which commodity flows over arc i?

15. Consider the node–arc formulation of the ''traffic-assignment'' model. Define a ''commodity'' as the flow from an origin to a destination. Let

$$x_{ij}^k = \text{Flow over arc } i - j \text{ of commodity } k.$$

The conservation-of-flow equations for each commodity are:

$$\sum_i x_{in}^k - \sum_j x_{nj}^k = \begin{cases} v_k & \text{if } n = \text{origin for commodity } k, \\ -v_k & \text{if } n = \text{destination for commodity } k, \\ 0 & \text{otherwise.} \end{cases}$$

The capacity restrictions on the arcs can be formulated as follows:

$$\sum_{k=1}^{K} x_{ij}^k \leq u_{ij} \qquad \text{for all arcs } i - j,$$

assuming that t_{ij} the travel time on arc $i - j$. To minimize total travel time on the network, we have the following objective function:

$$\text{Minimize} \sum_k \sum_i \sum_j t_{ij} x_{ij}^k.$$

a) Let the conservation-of-flow constraints for a commodity correspond to a subproblem, and the capacity restrictions on the arcs correspond to the master constraints in a decomposition approach. Formulate the restricted master, and the subproblem for the kth commodity. What is the objective function of this subproblem?

b) What is the relationship between solving the subproblems of the node–arc formulation and finding the minimum reduced cost for each commodity in the arc–chain formulation discussed in the previous exercise?

c) Show that the solution of the node–arc formulation by decomposition is identical to solving the arc–chain formulation discussed in the previous exercise. [*Hint*. In the arc–chain formulation, define new variables

$$\lambda_j^k = \frac{x_j^k}{v_k}.\Big]$$

16. In the node–arc formulation of the ''traffic-assignment'' problem given in Exercise 15, the subproblems correspond to finding the shortest path between the kth origin–destination pair. In general, there may be a large number of origin–destination pairs and hence a large number of such subproblems. However, in Chapter 11 on dynamic programming, we saw that we can solve simultaneously for the shortest paths from a particular origin to all destinations. We can then consolidate the subproblems by defining one subproblem for each node where traffic originates. The conservation-of-flow constraints become:

$$\sum_i y_{in}^s - \sum_j y_{nj}^s = \begin{cases} \sum v_k & \text{if } n = \text{origin node } s, \\ -v_k & \text{if } n = \text{a destination node in the origin–destination pair } k = (s, n) \\ 0 & \text{otherwise,} \end{cases}$$

where the summation $\sum v_k$ is the total flow emanating from origin s for all destination nodes. In this formulation, $y_{ij}^s = \sum x_{ij}^k$ denotes the total flow on arc $i - j$ that emanates from origin s; that is, the summation is carried over all origin–destination pairs $k = (s, t)$ whose origin is node s.

a) How does the decomposition formulation developed in Exercise 15 change with this change in definition of a subproblem? Specify the new formulation precisely.
b) Which formulation has more constraints in its restricted master?
c) Which restricted master is more restricted? [*Hint*. Which set of constraints implies the other?]
d) How does the choice of which subproblems to employ affect the decomposition algorithm? which choice would you expect to be more efficient? Why?

17. Consider a ''nested decomposition'' as applied to the problem

$$\text{Maximize } \sum_{j=1}^{n} c_j x_j,$$

subject to:

$$\begin{aligned}
\sum_{j=1}^{n} a_{ij} x_j &= b_i && (i = 1, 2, \ldots, k), && (1)\\
\sum_{j=1}^{n} d_{ij} x_j &= d_i && (i = k+1, k+2, \ldots, \ell), && (2)\\
\sum_{j=1}^{n} g_{ij} x_j &= g_i && (i = \ell+1, \ell+2, \ldots, m), && (3)\\
x_j &\geq 0 && (j = 1, 2, \ldots, n).
\end{aligned} \qquad \text{(P)}$$

Let (1) be the constraints of the (first) restricted master problem. If π_i $(i = 1, 2, \ldots, k)$ are shadow prices for the constraints (1) in the weighting problem, then

$$\text{Maximize } \sum_{j=1}^{n} \left(c_j - \sum_{i=1}^{k} \pi_i a_{ij} \right) x_j,$$

subject to:

$$\begin{aligned}
\sum_{j=1}^{n} d_{ij} x_j &= d_i && (i = k+1, k+2, \ldots, \ell), && (2')\\
\sum_{j=1}^{n} g_{ij} x_j &= g_i && (i = \ell+1, \ell+2, \ldots, m), && (3') && \text{(Subproblem 1)}\\
x_j &\geq 0, && (j = 1, 2, \ldots, n),
\end{aligned}$$

constitutes subproblem 1 (the proposal-generating problem).

Suppose, though, that the constraints (3′) complicate this problem and make it difficult to solve. Therefore, to solve the subproblem we further apply decomposition on subproblem 1. Constraints (2′) will be the constraints of the ''second'' restricted master. Given any shadow prices $\alpha_i (i = k+1, k+2, \ldots, \ell)$ for constraints (2′) in the weighting problem, the subproblem 2 will be:

$$\text{Maximize } \sum_{j=1}^{n} \left(c_j - \sum_{i=1}^{k} \pi_i a_{ij} - \sum_{i=k+1}^{\ell} \alpha_i d_{ij} \right) x_j,$$

subject to:

$$\begin{aligned}
\sum_{j=1}^{n} g_{ij} x_j &= g_i && (i = \ell+1, \ell+2, \ldots, m), && \text{(Subproblem 2)}\\
x_j &\geq 0 && (j = 1, 2, \ldots, n).
\end{aligned}$$

a) Consider the following decomposition approach: Given shadow prices π_i, solve subproblem (1) to completion by applying decomposition with subproblem (2). Use the solution to this problem to generate a new weighting variable to the first restricted master problem, or show that the original problem (P) [containing all constraints (1), (2), (3)] has been solved. Specify details of this approach.

b) Show finite convergence and bounds on the objective function to (P).

c) Now consider another approach: Subproblem 1 need not be solved to completion, but merely until a solution x_j $(j = 1, 2, \ldots, n)$ is found, so that

$$\sum_{j=1}^{n}\left(c_j - \sum_{i=1}^{k}\pi_i a_{ij}\right)x_j > \gamma,$$

where γ is the shadow price for the weighting constraint to the first restricted master. Indicate how to identify such a solution x_j $(j = 1, 2, \ldots, n)$ while solving the second restricted master problem; justify this approach.

d) Discuss convergence and objective bounds for the algorithm proposed in part (c).

ACKNOWLEDGMENTS

A number of the exercises in this chapter are based on or inspired by articles in the literature. Exercise 8: D. Anderson, ''Models for Determining Least-Cost Investments in Electricity Supply,'' *The Bell Journal of Economics and Management Science*, 3, No. 1, Spring 1972.

Exercise 10: P. E. Gilmore and R. E. Gomory, ''A Linear Programming Approach to the Cutting Stock Problem-II,'' *Operations Research*, 11, No. 6, November–December 1963.

Exercise 12: A. S. Manne, ''Programming of Economic Lot Sizes,'' *Management Science*, 4, No. 2, January 1958.

Exercise 13: B. P. Dzielinski and R. E. Gomory, ''Optimal Programming of Lot Sizes, Inventory, and Labor Allocations,'' *Management Science*, 11, No. 9, July 1965; and L. S. Lasdon and R. C. Terjung, ''An Efficient Algorithm for Multi-Item Scheduling,'' *Operation Research*, 19, No. 4, July–August 1971.

Exercises 14 through 16: S. P. Bradley, ''Solution Techniques for the Traffic Assignment Problem,'' Operations Research Center Report ORC 65–35, University of California, Berkeley. Exercise 17: R. Glassey, ''Nested Decomposition and Multi-Stage Linear Programs,'' *Management Science*, 20, No. 3, 1973.