A related problem for airline companies is that their crew schedules occasionally need to be revised quickly when flight delays or cancellations occur because of inclement weather, aircraft mechanical problems, or crew unavailability. As described in an application vignette in Sec. 2.2 (as well as in Selected Reference A12), *Continental Airlines* (now merged with United Airlines) achieved savings of $40 million in the first year of using an elaborate decision support system based on BIP for optimizing the *reassignment* of crews to flights when such emergencies occur.

Many of the problems that face airline companies also arise in other segments of the transportation industry. Therefore, some of the airline applications of OR are being extended to these other segments, including extensive use now by the railroad industry. For example, the second application vignette in this section describes how *Netherlands Railways* won a prestigious award for its applications of operations research, including integer programming and constraint programming (the subject of Sec. 12.9), throughout its operations.

12.3 INNOVATIVE USES OF BINARY VARIABLES IN MODEL FORMULATION

You have just seen a number of examples where the *basic decisions* of the problem are of the *yes-or-no type,* so that *binary variables* are introduced to represent these decisions. We now will look at some other ways in which binary variables can be very useful. In particular, we will see that these variables sometimes enable us to take a problem whose natural formulation is intractable and *reformulate* it as a pure or mixed IP problem.

This kind of situation arises when the original formulation of the problem fits either an IP or a linear programming format *except* for minor disparities involving combinatorial relationships in the model. By expressing these combinatorial relationships in terms of questions that must be answered yes or no, **auxiliary binary variables** can be introduced to the model to represent these yes-or-no decisions. (Rather than being a decision variable for the original problem under consideration, an *auxiliary* binary variable is a binary variable that is introduced into the model of the problem simply to help formulate the model as a pure or mixed BIP model.) Introducing these variables reduces the problem to an MIP problem (or a *pure* IP problem if all the original variables also are required to have integer values).

Some cases that can be handled by this approach are discussed next, where the x_j denote the *original* variables of the problem (they may be either continuous or integer variables) and the y_i denote the *auxiliary* binary variables that are introduced for the reformulation.

Either-Or Constraints

Consider the important case where a choice can be made between two constraints, so that *only one* (either one) must hold (whereas the other one can hold but is not required to do so). For example, there may be a choice as to which of two resources to use for a certain purpose, so that it is necessary for only one of the two resource availability constraints to hold mathematically. To illustrate the approach to such situations, suppose that one of the requirements in the overall problem is that

$$\begin{array}{ll} \text{Either} & 3x_1 + 2x_2 \leq 18 \\ \text{or} & x_1 + 4x_2 \leq 16, \end{array}$$

i.e., at least one of these two inequalities must hold but not necessarily both. This requirement must be reformulated to fit it into the linear programming format where *all* specified constraints must hold. Let M symbolize a very large positive number. Then this requirement can be rewritten as

$$\text{Either} \quad \begin{aligned} 3x_1 + 2x_2 &\le 18 \\ x_1 + 4x_2 &\le 16 + M \end{aligned}$$

$$\text{or} \quad \begin{aligned} 3x_1 + 2x_2 &\le 18 + M \\ x_1 + 4x_2 &\le 16. \end{aligned}$$

The key is that adding M to the right-hand side of such constraints has the effect of eliminating them, because they would be satisfied automatically by any solutions that satisfy the other constraints of the problem. (This formulation assumes that the set of feasible solutions for the overall problem is a bounded set and that M is large enough that it will not eliminate any feasible solutions.) This formulation is equivalent to the set of constraints

$$\begin{aligned} 3x_1 + 2x_2 &\le 18 + My \\ x_1 + 4x_2 &\le 16 + M(1 - y). \end{aligned}$$

Because the *auxiliary variable* y must be either 0 or 1, this formulation guarantees that one of the original constraints must hold while the other is, in effect, eliminated. This new set of constraints would then be appended to the other constraints in the overall model to give a pure or mixed IP problem (depending upon whether the x_j are integer or continuous variables).

This approach is related directly to our earlier discussion about expressing combinatorial relationships in terms of questions that must be answered yes or no. The combinatorial relationship involved concerns the combination of the *other* constraints of the model with the *first* of the two *alternative* constraints and then with the *second.* Which of these two combinations of constraints is *better* (in terms of the value of the objective function that then can be achieved)? To rephrase this question in yes-or-no terms, we ask two complementary questions:

1. Should $x_1 + 4x_2 \le 16$ be selected as the constraint that must hold?
2. Should $3x_1 + 2x_2 \le 18$ be selected as the constraint that must hold?

Because exactly one of these questions is to be answered affirmatively, we let the binary terms y and $1 - y$, respectively, represent these yes-or-no decisions. Thus, $y = 1$ if the answer is yes to the first question (and no to the second), whereas $1 - y = 1$ (that is, $y = 0$) if the answer is yes to the second question (and no to the first). Since $y + 1 - y = 1$ (one yes) automatically, there is no need to add another constraint to force these two decisions to be mutually exclusive. (If separate binary variables y_1 and y_2 had been used instead to represent these yes-or-no decisions, then an additional constraint $y_1 + y_2 = 1$ would have been needed to make them mutually exclusive.)

A formal presentation of this approach is given next for a more general case.

K out of *N* Constraints Must Hold

Consider the case where the overall model includes a set of N possible constraints such that only some K of these constraints *must* hold. (Assume that $K < N$.) Part of the optimization process is to choose the *combination* of K constraints that permits the objective function to reach its best possible value. The $N - K$ constraints *not* chosen are, in effect, eliminated from the problem, although feasible solutions might coincidentally still satisfy some of them.

This case is a direct generalization of the preceding case, which had $K = 1$ and $N = 2$. Denote the N possible constraints by

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &\le d_1 \\ f_2(x_1, x_2, \ldots, x_n) &\le d_2 \\ &\vdots \\ f_N(x_1, x_2, \ldots, x_n) &\le d_N. \end{aligned}$$

Then, applying the same logic as for the preceding case, we find that an equivalent formulation of the requirement that some K of these constraints *must* hold is

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &\le d_1 + My_1 \\ f_2(x_1, x_2, \ldots, x_n) &\le d_2 + My_2 \\ &\vdots \\ f_N(x_1, x_2, \ldots, x_n) &\le d_N + My_N \\ \sum_{i=1}^{N} y_i &= N - K, \end{aligned}$$

and

$$y_i \text{ is binary}, \qquad \text{for } i = 1, 2, \ldots, N,$$

where M is an extremely large positive number. For each binary variable y_i ($i = 1, 2, \ldots, N$), note that $y_i = 0$ makes $My_i = 0$, which reduces the new constraint i to the original constraint i. On the other hand, $y_i = 1$ makes $(d_i + My_i)$ so large that (again assuming a bounded feasible region) the new constraint i is automatically satisfied by any solution that satisfies the other new constraints, which has the effect of eliminating the original constraint i. Therefore, because the constraints on the y_i guarantee that K of these variables will equal 0 and those remaining will equal 1, K of the original constraints will be unchanged and the other $(N - K)$ original constraints will, in effect, be eliminated. The choice of *which* K constraints should be retained is made by applying the appropriate algorithm to the overall problem so it finds an optimal solution for *all* the variables simultaneously.

Functions with *N* Possible Values

Consider the situation where a given function is required to take on any one of N given values. Denote this requirement by

$$f(x_1, x_2, \ldots, x_n) = d_1 \qquad \text{or} \qquad d_2, \ldots, \qquad \text{or} \qquad d_N.$$

One special case is where this function is

$$f(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{n} a_j x_j,$$

as on the left-hand side of a linear programming constraint. Another special case is where $f(x_1, x_2, \ldots, x_n) = x_j$ for a given value of j, so the requirement becomes that x_j must take on any one of N given values.

The equivalent IP formulation of this requirement is the following:

$$\begin{aligned} f(x_1, x_2, \ldots, x_n) &= \sum_{i=1}^{N} d_i y_i \\ \sum_{i=1}^{N} y_i &= 1 \end{aligned}$$

and

$$y_i \text{ is binary}, \qquad \text{for } i = 1, 2, \ldots, N.$$

so this new set of constraints would replace this requirement in the statement of the overall problem. This set of constraints provides an *equivalent* formulation because exactly one y_i must equal 1 and the others must equal 0, so exactly one d_i is being chosen as the value of the function. In this case, there are N yes-or-no questions being asked, namely, should d_i be the value chosen ($i = 1, 2, \ldots, N$)? Because the y_i respectively represent these *yes-or-no decisions,* the second constraint makes them *mutually exclusive alternatives.*

To illustrate how this case can arise, reconsider the Wyndor Glass Co. problem presented in Sec. 3.1. Eighteen hours of production time per week in Plant 3 currently is unused and available for the two new products *or* for certain future products that will be ready for production soon. In order to leave any remaining capacity in usable blocks for these future products, management now wants to impose the restriction that the production time used by the two current new products be 6 *or* 12 *or* 18 hours per week. Thus, the third constraint of the original model ($3x_1 + 2x_2 \leq 18$) now becomes

$$3x_1 + 2x_2 = 6 \qquad \text{or} \qquad 12 \qquad \text{or} \qquad 18.$$

In the preceding notation, $N = 3$ with $d_1 = 6$, $d_2 = 12$, and $d_3 = 18$. Consequently, management's new requirement should be formulated as follows:

$$3x_1 + 2x_2 = 6y_1 + 12y_2 + 18y_3$$
$$y_1 + y_2 + y_3 = 1$$

and

$$y_1, y_2, y_3 \text{ are binary.}$$

The overall model for this new version of the problem then consists of the original model (see Sec. 3.1) plus this new set of constraints that replaces the original third constraint. This replacement yields a very tractable MIP formulation.

The Fixed-Charge Problem

It is quite common to incur a fixed charge or setup cost when undertaking an activity. For example, such a charge occurs when a production run to produce a batch of a particular product is undertaken and the required production facilities must be set up to initiate the run. In such cases, the total cost of the activity is the sum of a variable cost related to the level of the activity and the setup cost required to initiate the activity. Frequently the variable cost will be at least roughly proportional to the level of the activity. If this is the case, the *total cost* of the activity (say, activity j) can be represented by a function of the form

$$f_j(x_j) = \begin{cases} k_j + c_j x_j & \text{if } x_j > 0 \\ 0 & \text{if } x_j = 0, \end{cases}$$

where x_j denotes the level of activity j ($x_j \geq 0$), k_j denotes the setup cost, and c_j denotes the cost for each incremental unit. Were it not for the setup cost k_j, this cost structure would suggest the possibility of a *linear programming* formulation to determine the optimal levels of the competing activities. Fortunately, even with the k_j, MIP can still be used.

To formulate the overall model, suppose that there are n activities, each with the preceding cost structure (with $k_j \geq 0$ in every case and $k_j > 0$ for some $j = 1, 2, \ldots, n$), and that the problem is to

$$\text{Minimize} \qquad Z = f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n),$$

subject to

given linear programming constraints.

To convert this problem to an MIP format, we begin by posing n questions that must be answered yes or no; namely, for each $j = 1, 2, \ldots, n$, should activity j be undertaken ($x_j > 0$)? Each of these *yes-or-no decisions* is then represented by an auxiliary *binary variable* y_j, so that

$$Z = \sum_{j=1}^{n} (c_j x_j + k_j y_j),$$

where

$$y_j = \begin{cases} 1 & \text{if } x_j > 0 \\ 0 & \text{if } x_j = 0. \end{cases}$$

Therefore, the y_j can be viewed as *contingent decisions* similar to (but not identical to) the type considered in Sec. 12.1. Let M be an extremely large positive number that exceeds the maximum feasible value of any x_j ($j = 1, 2, \ldots, n$). Then the constraints

$$x_j \le M y_j \qquad \text{for } j = 1, 2, \ldots, n$$

will ensure that $y_j = 1$ rather than 0 whenever $x_j > 0$. The one difficulty remaining is that these constraints leave y_j free to be either 0 or 1 when $x_j = 0$. Fortunately, this difficulty is automatically resolved because of the nature of the objective function. The case where $k_j = 0$ can be ignored because y_j can then be deleted from the formulation. So we consider the only other case, namely, where $k_j > 0$. When $x_j = 0$, so that the constraints permit a choice between $y_j = 0$ and $y_j = 1$, $y_j = 0$ must yield a smaller value of Z than $y_j = 1$. Therefore, because the objective is to minimize Z, an algorithm yielding an optimal solution would always choose $y_j = 0$ when $x_j = 0$.

To summarize, the MIP formulation of the fixed-charge problem is

$$\text{Minimize} \qquad Z = \sum_{j=1}^{n} (c_j x_j + k_j y_j),$$

subject to

the original constraints, plus
$x_j - M y_j \le 0$

and

y_j is binary, $\qquad$ for $j = 1, 2, \ldots, n$.

If the x_j also had been restricted to be integer, then this would be a *pure* IP problem.

To illustrate this approach, look again at the Nori & Leets Co. air pollution problem described in Sec. 3.4. The first of the abatement methods considered—increasing the height of the smokestacks—actually would involve a substantial *fixed charge* to get ready for *any* increase in addition to a variable cost that would be roughly proportional to the amount of increase. After conversion to the equivalent annual costs used in the formulation, this fixed charge would be \$2 million each for the blast furnaces and the open-hearth furnaces, whereas the variable costs are those identified in Table 3.14. Thus, in the preceding notation, $k_1 = 2$, $k_2 = 2$, $c_1 = 8$, and $c_2 = 10$, where the objective function is expressed in units of *millions* of dollars. Because the other abatement methods do not involve any fixed charges, $k_j = 0$ for $j = 3, 4, 5, 6$. Consequently, the new MIP formulation of this problem is

$$\text{Minimize} \qquad Z = 8x_1 + 10x_2 + 7x_3 + 6x_4 + 11x_5 + 9x_6 + 2y_1 + 2y_2,$$

subject to

the constraints given in Sec. 3.4, plus
$x_1 - My_1 \leq 0,$
$x_2 - My_2 \leq 0,$

and

y_1, y_2 are binary.

Binary Representation of General Integer Variables

Suppose that you have a pure IP problem where most of the variables are *binary* variables, but the presence of a few *general* integer variables prevents you from solving the problem by one of the very efficient BIP algorithms now available. A nice way to circumvent this difficulty is to use the *binary representation* for each of these general integer variables. Specifically, if the bounds on an integer variable x are

$$0 \leq x \leq u$$

and if N is defined as the integer such that

$$2^N \leq u < 2^{N+1},$$

then the **binary representation** of x is

$$x = \sum_{i=0}^{N} 2^i y_i,$$

where the y_i variables are (auxiliary) binary variables. Substituting this binary representation for each of the general integer variables (with a different set of auxiliary binary variables for each) thereby reduces the entire problem to a BIP model.

For example, suppose that an IP problem has just two general integer variables x_1 and x_2 along with many binary variables. Also suppose that the problem has nonnegativity constraints for both x_1 and x_2 and that the functional constraints include

$$\begin{aligned} x_1 &\leq 5 \\ 2x_1 + 3x_2 &\leq 30. \end{aligned}$$

These constraints imply that $u = 5$ for x_1 and $u = 10$ for x_2, so the above definition of N gives $N = 2$ for x_1 (since $2^2 \leq 5 < 2^3$) and $N = 3$ for x_2 (since $2^3 \leq 10 < 2^4$). Therefore, the binary representations of these variables are

$$\begin{aligned} x_1 &= y_0 + 2y_1 + 4y_2 \\ x_2 &= y_3 + 2y_4 + 4y_5 + 8y_6. \end{aligned}$$

After we substitute these expressions for the respective variables throughout all the functional constraints and the objective function, the two functional constraints noted above become

$$\begin{aligned} y_0 + 2y_1 + 4y_2 &\leq 5 \\ 2y_0 + 4y_1 + 8y_2 + 3y_3 + 6y_4 + 12y_5 + 24y_6 &\leq 30. \end{aligned}$$

Observe that each feasible value of x_1 corresponds to one of the feasible values of the vector (y_0, y_1, y_2), and similarly for x_2 and (y_3, y_4, y_5, y_6). For example, $x_1 = 3$ corresponds to $(y_0, y_1, y_2) = (1, 1, 0)$, and $x_2 = 5$ corresponds to $(y_3, y_4, y_5, y_6) = (1, 0, 1, 0)$.

For an IP problem where *all* the variables are (bounded) general integer variables, it is possible to use this same technique to reduce the problem to a BIP model. However, this is not advisable for most cases because of the explosion in the number of variables involved. Applying a good IP algorithm to the original IP model generally should be more efficient than applying a good BIP algorithm to the much larger BIP model.[1]

In general terms, for *all* the formulation possibilities with auxiliary binary variables discussed in this section, we need to strike the same note of caution. This approach sometimes requires adding a relatively large number of such variables, which can make the model *computationally infeasible.* (Section 12.5 will provide some perspective on the sizes of IP problems that can be solved.)

12.4 SOME FORMULATION EXAMPLES

We now present a series of examples that illustrate a variety of formulation techniques with binary variables, including those discussed in the preceding sections. For the sake of clarity, these examples have been kept very small. (**A somewhat larger formulation example**, with dozens of binary variables and constraints, is included in the Solved Examples section of the book's website.) In actual applications, these formulations typically would be just a small part of a vastly larger model.

EXAMPLE 1 Making Choices When the Decision Variables Are Continuous

The Research and Development Division of the GOOD PRODUCTS COMPANY has developed three possible new products. However, to avoid undue diversification of the company's product line, management has imposed the following restriction:

> **Restriction 1:** From the three possible new products, *at most two* should be chosen to be produced.

Each of these products can be produced in either of two plants. For administrative reasons, management has imposed a second restriction in this regard.

> **Restriction 2:** Just one of the two plants should be chosen to be the sole producer of the new products.

The production cost per unit of each product would be essentially the same in the two plants. However, because of differences in their production facilities, the number of hours of production time needed per unit of each product might differ between the two plants. These data are given in Table 12.2, along with other relevant information, including marketing

TABLE 12.2 Data for Example 1 (the Good Products Co. problem)

	Production Time Used for Each Unit Produced			Production Time Available per Week
	Product 1	Product 2	Product 3	
Plant 1	3 hours	4 hours	2 hours	30 hours
Plant 2	4 hours	6 hours	2 hours	40 hours
Unit profit	5	7	3	(thousands of dollars)
Sales potential	7	5	9	(units per week)

[1]For evidence supporting this conclusion, see J. H. Owen and S. Mehrotra, "On the Value of Binary Expansions for General Mixed Integer Linear Programs," *Operations Research,* **50:** 810–819, 2002.

estimates of the number of units of each product that could be sold per week if it is produced. The objective is to choose the products, the plant, and the production rates of the chosen products so as to maximize total profit.

In some ways, this problem resembles a standard *product mix problem* such as the Wyndor Glass Co. example described in Sec. 3.1. In fact, if we changed the problem by dropping the two restrictions *and* by requiring each unit of a product to use the production hours given in Table 12.2 in *both plants* (so the two plants now perform different operations needed by the products), it would become just such a problem. In particular, if we let x_1, x_2, x_3 be the production rates of the respective products, the model then becomes

$$\text{Maximize} \quad Z = 5x_1 + 7x_2 + 3x_3,$$

subject to

$$\begin{aligned} 3x_1 + 4x_2 + 2x_3 &\leq 30 \\ 4x_1 + 6x_2 + 2x_3 &\leq 40 \\ x_1 \qquad\qquad &\leq 7 \\ x_2 \qquad &\leq 5 \\ x_3 &\leq 9 \end{aligned}$$

and

$$x_1 \geq 0, \qquad x_2 \geq 0, \qquad x_3 \geq 0.$$

For the real problem, however, restriction 1 necessitates adding to the model the constraint

The number of strictly positive decision variables (x_1, x_2, x_3) must be ≤ 2.

This constraint does not fit into a linear or an integer programming format, so the key question is how to convert it to such a format so that a corresponding algorithm can be used to solve the overall model. If the decision variables were binary variables, then the constraint would be expressed in this format as $x_1 + x_2 + x_3 \leq 2$. However, with *continuous* decision variables, a more complicated approach involving the introduction of auxiliary binary variables is needed.

Requirement 2 necessitates replacing the first two functional constraints ($3x_1 + 4x_2 + 2x_3 \leq 30$ and $4x_1 + 6x_2 + 2x_3 \leq 40$) by the restriction

$$\begin{aligned} &\text{Either} \quad 3x_1 + 4x_2 + 2x_3 \leq 30 \\ &\text{or} \qquad\;\; 4x_1 + 6x_2 + 2x_3 \leq 40 \end{aligned}$$

must hold, where the choice of which constraint must hold corresponds to the choice of which plant will be used to produce the new products. We discussed in the preceding section how such an either-or constraint can be converted to a linear or an integer programming format, again with the help of an auxiliary binary variable.

Formulation with Auxiliary Binary Variables. To deal with requirement 1, we introduce three auxiliary binary variables (y_1, y_2, y_3) with the interpretation

$$y_j = \begin{cases} 1 & \text{if } x_j > 0 \text{ can hold (can produce product } j) \\ 0 & \text{if } x_j = 0 \text{ must hold (cannot produce product } j), \end{cases}$$

for $j = 1, 2, 3$. To enforce this interpretation in the model with the help of M (an extremely large positive number), we add the constraints

$$\begin{aligned}
&x_1 \le My_1\\
&x_2 \le My_2\\
&x_3 \le My_3\\
&y_1 + y_2 + y_3 \le 2\\
&y_j \text{ is binary}, \qquad \text{for } j = 1, 2, 3.
\end{aligned}$$

The either-or constraint and nonnegativity constraints give a *bounded* feasible region for the decision variables (so each $x_j \le M$ throughout this region). Therefore, in each $x_j \le My_j$ constraint, $y_j = 1$ allows any value of x_j in the feasible region, whereas $y_j = 0$ forces $x_j = 0$. (Conversely, $x_j > 0$ forces $y_j = 1$, whereas $x_j = 0$ allows either value of y_j.) Consequently, when the fourth constraint forces choosing at most two of the y_j to equal 1, this amounts to choosing at most two of the new products as the ones that can be produced.

To deal with requirement 2, we introduce another auxiliary binary variable y_4 with the interpretation

$$y_4 = \begin{cases} 1 & \text{if } 4x_1 + 6x_2 + 2x_3 \le 40 \text{ must hold (choose Plant 2)} \\ 0 & \text{if } 3x_1 + 4x_2 + 2x_3 \le 30 \text{ must hold (choose Plant 1).} \end{cases}$$

As discussed in Sec. 12.3, this interpretation is enforced by adding the constraints,

$$\begin{aligned}
&3x_1 + 4x_2 + 2x_3 \le 30 + My_4\\
&4x_1 + 6x_2 + 2x_3 \le 40 + M(1 - y_4)\\
&y_4 \text{ is binary.}
\end{aligned}$$

Consequently, after we move all variables to the left-hand side of the constraints, the complete model is

$$\text{Maximize} \qquad Z = 5x_1 + 7x_2 + 3x_3,$$

subject to

$$\begin{aligned}
x_1 &\le 7\\
x_2 &\le 5\\
x_3 &\le 9\\
x_1 - My_1 &\le 0\\
x_2 - My_2 &\le 0\\
x_3 - My_3 &\le 0\\
y_1 + y_2 + y_3 &\le 2\\
3x_1 + 4x_2 + 2x_3 - My_4 &\le 30\\
4x_1 + 6x_2 + 2x_3 + My_4 &\le 40 + M
\end{aligned}$$

and

$$\begin{aligned}
&x_1 \ge 0, \qquad x_2 \ge 0, \qquad x_3 \ge 0\\
&y_j \text{ is binary}, \qquad \text{for } j = 1, 2, 3, 4.
\end{aligned}$$

This now is an MIP model, with three variables (the x_j) not required to be integer and four binary variables, so an MIP algorithm can be used to solve the model. When this is done (after substituting a large numerical value for M),[2] the optimal solution is $y_1 = 1$,

[2]In practice, some care is taken to choose a value for M that definitely is large enough to avoid eliminating any feasible solutions, but as small as possible otherwise in order to avoid unduly enlarging the feasible region for the LP relaxation (described in the next section) and to avoid numerical instability. For this example, a careful examination of the constraints reveals that the minimum feasible value of M is $M = 9$.

$y_2 = 0$, $y_3 = 1$, $y_4 = 1$, $x_1 = 5\frac{1}{2}$, $x_2 = 0$, and $x_3 = 9$; that is, choose products 1 and 3 to produce, choose Plant 2 for the production, and choose the production rates of $5\frac{1}{2}$ units per week for product 1 and 9 units per week for product 3. The resulting total profit is \$54,500 per week.

EXAMPLE 2 Violating Proportionality

The SUPERSUDS CORPORATION is developing its marketing plans for next year's new products. For three of these products, the decision has been made to purchase a total of five TV spots for commercials on national television networks. The problem we will focus on is how to allocate the five spots to these three products, with a maximum of three spots (and a minimum of zero) for each product.

Table 12.3 shows the estimated impact of allocating zero, one, two, or three spots to each product. This impact is measured in terms of the *profit* (in units of millions of dollars) from the *additional sales* that would result from the spots, considering also the cost of producing the commercial and purchasing the spots. The objective is to allocate five spots to the products so as to maximize the total profit.

This small problem can be solved easily by dynamic programming (Chap. 11) or even by inspection. (The optimal solution is to allocate two spots to product 1, no spots to product 2, and three spots to product 3.) However, we will show two different BIP formulations for illustrative purposes. Such a formulation would become necessary if this small problem needed to be incorporated into a larger IP model involving the allocation of resources to marketing activities for all the corporation's new products.

One Formulation with Auxiliary Binary Variables. A natural formulation would be to let x_1, x_2, x_3 be the number of TV spots allocated to the respective products. The contribution of each x_j to the objective function then would be given by the corresponding column in Table 12.3. However, each of these columns violates the assumption of proportionality described in Sec. 3.3. Therefore, we cannot write a *linear* objective function in terms of these integer decision variables.

Now see what happens when we introduce an *auxiliary binary variable* y_{ij} for each positive integer value of $x_i = j$ ($j = 1, 2, 3$), where y_{ij} has the interpretation

$$y_{ij} = \begin{cases} 1 & \text{if } x_i = j \\ 0 & \text{otherwise.} \end{cases}$$

(For example, $y_{21} = 0$, $y_{22} = 0$, and $y_{23} = 1$ mean that $x_2 = 3$.) The resulting *linear* BIP model is

$$\text{Maximize} \quad Z = y_{11} + 3y_{12} + 3y_{13} + 2y_{22} + 3y_{23} - y_{31} + 2y_{32} + 4y_{33},$$

TABLE 12.3 Data for Example 2 (the Supersuds Corp. problem)

Number of TV Spots	Profit: Product 1	2	3
0	0	0	0
1	1	0	−1
2	3	2	2
3	3	3	4

subject to

$$
\begin{aligned}
y_{11} + y_{12} + y_{13} &\le 1\\
y_{21} + y_{22} + y_{23} &\le 1\\
y_{31} + y_{32} + y_{33} &\le 1\\
y_{11} + 2y_{12} + 3y_{13} + y_{21} + 2y_{22} + 3y_{23} + y_{31} + 2y_{32} + 3y_{33} &= 5
\end{aligned}
$$

and

each y_{ij} is binary.

Note that the first three functional constraints ensure that each x_i will be assigned just one of its possible values. (Here $y_{i1} + y_{i2} + y_{i3} = 0$ corresponds to $x_i = 0$, which contributes nothing to the objective function.) The last functional constraint ensures that $x_1 + x_2 + x_3 = 5$. The *linear* objective function then gives the total profit according to Table 12.3.

Solving this BIP model gives an optimal solution of

$$
\begin{array}{lllll}
y_{11} = 0, & y_{12} = 1, & y_{13} = 0, & \text{so} & x_1 = 2\\
y_{21} = 0, & y_{22} = 0, & y_{23} = 0, & \text{so} & x_2 = 0\\
y_{31} = 0, & y_{32} = 0, & y_{33} = 1, & \text{so} & x_3 = 3.
\end{array}
$$

Another Formulation with Auxiliary Binary Variables. We now redefine the above auxiliary binary variables y_{ij} as follows:

$$
y_{ij} = \begin{cases} 1 & \text{if } x_i \ge j\\ 0 & \text{otherwise.} \end{cases}
$$

Thus, the difference is that $y_{ij} = 1$ now if $x_i \ge j$ instead of $x_i = j$. Therefore,

$$
\begin{array}{lllll}
x_i = 0 & \Rightarrow & y_{i1} = 0, & y_{i2} = 0, & y_{i3} = 0,\\
x_i = 1 & \Rightarrow & y_{i1} = 1, & y_{i2} = 0, & y_{i3} = 0,\\
x_i = 2 & \Rightarrow & y_{i1} = 1, & y_{i2} = 1, & y_{i3} = 0,\\
x_i = 3 & \Rightarrow & y_{i1} = 1, & y_{i2} = 1, & y_{i3} = 1,\\
\multicolumn{5}{l}{\text{so } x_i = y_{i1} + y_{i2} + y_{i3}}
\end{array}
$$

for $i = 1, 2, 3$. Because allowing $y_{i2} = 1$ is contingent upon $y_{i1} = 1$ and allowing $y_{i3} = 1$ is contingent upon $y_{i2} = 1$, these definitions are enforced by adding the constraints

$$
y_{i2} \le y_{i1} \quad \text{and} \quad y_{i3} \le y_{i2}, \qquad \text{for } i = 1, 2, 3.
$$

The new definition of the y_{ij} also changes the objective function, as illustrated in Fig. 12.1 for the product 1 portion of the objective function. Since y_{11}, y_{12}, y_{13} provide the successive increments (if any) in the value of x_1 (starting from a value of 0), the coefficients of y_{11}, y_{12}, y_{13} are given by the respective *increments* in the product 1 column of Table 12.3 ($1 - 0 = 1$, $3 - 1 = 2$, $3 - 3 = 0$). These *increments* are the *slopes* in Fig. 12.1, yielding $1y_{11} + 2y_{12} + 0y_{13}$ for the product 1 portion of the objective function. Note that applying this approach to all three products still must lead to a *linear* objective function.

After we bring all variables to the left-hand side of the constraints, the resulting complete BIP model is

$$
\text{Maximize} \qquad Z = y_{11} + 2y_{12} + 2y_{22} + y_{23} - y_{31} + 3y_{32} + 2y_{33},
$$

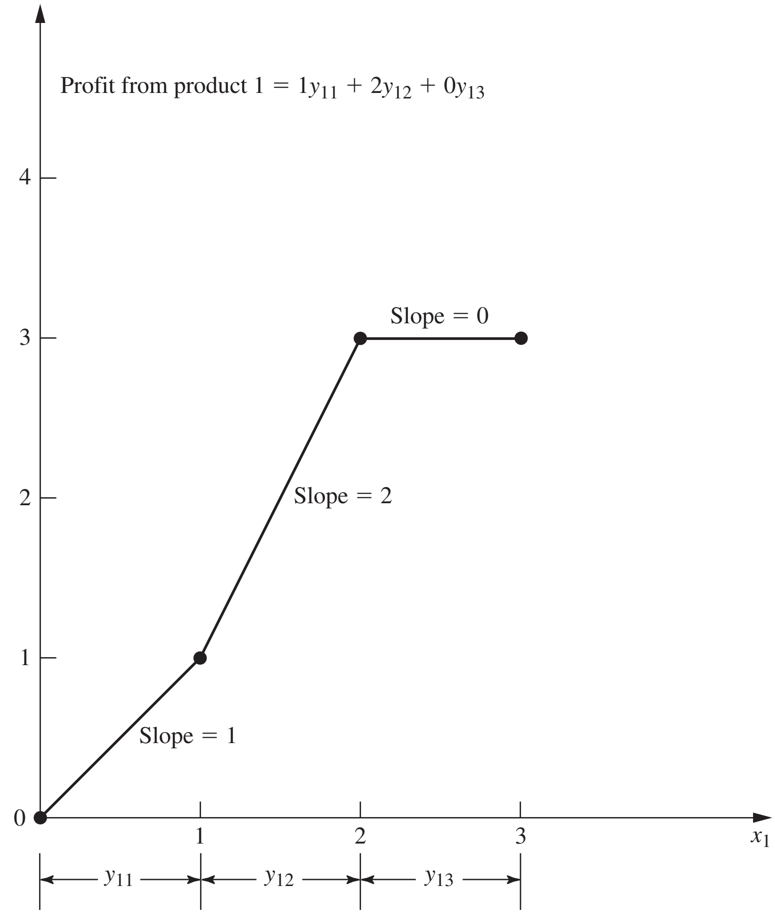

FIGURE 12.1
The profit from the additional sales of product 1 that would result from x_1 TV spots, where the slopes give the corresponding coefficients in the objective function for the second BIP formulation for Example 2 (the Supersuds Corp. problem).

subject to

$$
\begin{aligned}
&y_{12} - y_{11} \leq 0 \\
&y_{13} - y_{12} \leq 0 \\
&y_{22} - y_{21} \leq 0 \\
&y_{23} - y_{22} \leq 0 \\
&y_{32} - y_{31} \leq 0 \\
&y_{33} - y_{32} \leq 0 \\
&y_{11} + y_{12} + y_{13} + y_{21} + y_{22} + y_{23} + y_{31} + y_{32} + y_{33} = 5
\end{aligned}
$$

and

each y_{ij} is binary.

Solving this BIP model gives an optimal solution of

$$
\begin{array}{lllll}
y_{11} = 1, & y_{12} = 1, & y_{13} = 0, & \text{so} & x_1 = 2 \\
y_{21} = 0, & y_{22} = 0, & y_{23} = 0, & \text{so} & x_2 = 0 \\
y_{31} = 1, & y_{32} = 1, & y_{33} = 1, & \text{so} & x_3 = 3.
\end{array}
$$

There is little to choose between this BIP model and the preceding one other than personal taste. They have the same number of binary variables (the prime consideration in determining computational effort for BIP problems). They also both have some *special* *structure* (constraints for *mutually exclusive alternatives* in the first model and constraints for *contingent decisions* in the second) that can lead to speedup. The second model does have more functional constraints than the first.

EXAMPLE 3 Covering All Characteristics

SOUTHWESTERN AIRWAYS needs to assign its crews to cover all its upcoming flights. We will focus on the problem of assigning three crews based in San Francisco to the flights listed in the first column of Table 12.4. The other 12 columns show the 12 feasible sequences of flights for a crew. (The numbers in each column indicate the order of the flights.) Exactly three of the sequences need to be chosen (one per crew) in such a way that every flight is covered. (It is permissible to have more than one crew on a flight, where the extra crews would fly as passengers, but union contracts require that the extra crews would still need to be paid for their time as if they were working.) The cost of assigning a crew to a particular sequence of flights is given (in thousands of dollars) in the bottom row of the table. The objective is to minimize the total cost of the three crew assignments that cover all the flights.

Formulation with Binary Variables. With 12 feasible sequences of flights, we have 12 yes-or-no decisions:

Should sequence j be assigned to a crew? $(j = 1, 2, \ldots, 12)$

Therefore, we use 12 binary variables to represent these respective decisions:

$$x_j = \begin{cases} 1 & \text{if sequence } j \text{ is assigned to a crew} \\ 0 & \text{otherwise.} \end{cases}$$

The most interesting part of this formulation is the nature of each constraint that ensures that a corresponding flight is covered. For example, consider the last flight in Table 12.4 [Seattle to Los Angeles (LA)]. Five sequences (namely, sequences 6, 9, 10, 11, and 12) include this flight. Therefore, at least one of these five sequences must be chosen. The resulting constraint is

$$x_6 + x_9 + x_{10} + x_{11} + x_{12} \geq 1.$$

Using similar constraints for the other 10 flights, the complete BIP model is

$$\text{Minimize} \quad Z = 2x_1 + 3x_2 + 4x_3 + 6x_4 + 7x_5 + 5x_6 + 7x_7 + 8x_8 + 9x_9 + 9x_{10} + 8x_{11} + 9x_{12},$$

TABLE 12.4 Data for Example 3 (the Southwestern Airways problem)

	Feasible Sequence of Flights											
Flight	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1. San Francisco to Los Angeles	1			1			1			1		
2. San Francisco to Denver		1			1			1			1	
3. San Francisco to Seattle			1			1			1			1
4. Los Angeles to Chicago				2			2		3	2		3
5. Los Angeles to San Francisco	2					3				5	5	
6. Chicago to Denver				3	3				4			
7. Chicago to Seattle							3	3		3	3	4
8. Denver to San Francisco		2		4	4				5			
9. Denver to Chicago					2			2			2	
10. Seattle to San Francisco			2				4	4				5
11. Seattle to Los Angeles						2			2	4	4	2
Cost, $1,000's	2	3	4	6	7	5	7	8	9	9	8	9

subject to

$$
\begin{aligned}
x_1 + x_4 + x_7 + x_{10} &\geq 1 && \text{(SF to LA)}\\
x_2 + x_5 + x_8 + x_{11} &\geq 1 && \text{(SF to Denver)}\\
x_3 + x_6 + x_9 + x_{12} &\geq 1 && \text{(SF to Seattle)}\\
x_4 + x_7 + x_9 + x_{10} + x_{12} &\geq 1 && \text{(LA to Chicago)}\\
x_1 + x_6 + x_{10} + x_{11} &\geq 1 && \text{(LA to SF)}\\
x_4 + x_5 + x_9 &\geq 1 && \text{(Chicago to Denver)}\\
x_7 + x_8 + x_{10} + x_{11} + x_{12} &\geq 1 && \text{(Chicago to Seattle)}\\
x_2 + x_4 + x_5 + x_9 &\geq 1 && \text{(Denver to SF)}\\
x_5 + x_8 + x_{11} &\geq 1 && \text{(Denver to Chicago)}\\
x_3 + x_7 + x_8 + x_{12} &\geq 1 && \text{(Seattle to SF)}\\
x_6 + x_9 + x_{10} + x_{11} + x_{12} &\geq 1 && \text{(Seattle to LA)}\\
\sum_{j=1}^{12} x_j &= 3 && \text{(assign three crews)}
\end{aligned}
$$

and

$$x_j \text{ is binary,} \qquad \text{for } j = 1, 2, \ldots, 12.$$

One optimal solution for this BIP model is

$$
\begin{aligned}
x_3 &= 1 && \text{(assign sequence 3 to a crew)}\\
x_4 &= 1 && \text{(assign sequence 4 to a crew)}\\
x_{11} &= 1 && \text{(assign sequence 11 to a crew)}
\end{aligned}
$$

and all other $x_j = 0$, for a total cost of \$18,000. (Another optimal solution is $x_1 = 1$, $x_5 = 1$, $x_{12} = 1$, and all other $x_j = 0$.)

This example illustrates a broader class of problems called **set covering problems**.[3] Any set covering problem can be described in general terms as involving a number of potential *activities* (such as flight sequences) and *characteristics* (such as flights). Each activity possesses some but not all of the characteristics. The objective is to determine the least costly combination of activities that collectively possess (cover) each characteristic at least once. Thus, let S_i be the set of all activities that possess characteristic i. At least one member of the set S_i must be included among the chosen activities, so a constraint,

$$\sum_{j \in S_i} x_j \geq 1,$$

is included for each characteristic i.

A related class of problems, called **set partitioning problems,** changes each such constraint to

$$\sum_{j \in S_i} x_j = 1,$$

so now *exactly* one member of each set S_i must be included among the chosen activities. For the crew scheduling example, this means that each flight must be included *exactly* once among the chosen flight sequences, which rules out having extra crews (as passengers) on any flight.

[3]Strictly speaking, a set covering problem does not include any *other* functional constraints such as the last functional constraint in the above crew scheduling example. It also is sometimes assumed that every coefficient in the objective function being minimized equals *one,* and then the name *weighted set covering problem* is used when this assumption does not hold.