Example 9.1-1  (Project Selection)

Five projects are being evaluated over a 3-year planning horizon. The following table gives the expected returns for each project and the associated yearly expenditures.

<table>
<thead>
<tr>
<th>Project</th>
<th>Expenditures (million $)/yr</th>
<th>Returns (million $)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>6</td>
</tr>
</tbody>
</table>

Available funds (million $) 25 25 25

Which projects should be selected over the 3-year horizon?
The problem reduces to a "yes-no" decision for each project. Define the binary variable $x_j$ as

$$ x_j = \begin{cases} 1, & \text{if project } j \text{ is selected} \\ 0, & \text{if project } j \text{ is not selected} \end{cases} $$

The ILP model is

Maximize $z = 20x_1 + 40x_2 + 20x_3 + 15x_4 + 30x_5$

subject to

$$ 5x_1 + 4x_2 + 3x_3 + 7x_4 + 8x_5 \leq 25 $$
$$ x_1 + 7x_2 + 9x_3 + 4x_4 + 6x_5 \leq 25 $$
$$ 8x_1 + 10x_2 + 2x_3 + x_4 + 10x_5 \leq 25 $$

$x_1, x_2, x_3, x_4, x_5 = (0, 1)$

The optimum integer solution (obtained by AMPL, Solver, or TORA) is $x_1 = x_2 = x_3 = x_4 = 1, x_5 = 0$, with $z = 95$ (million $\$$). The solution shows that all but project 5 must be selected.

Remarks. It is interesting to compare the continuous LP solution with the ILP solution. The LP optimum, obtained by replacing $x_j = (0, 1)$ with $0 \leq x_j \leq 1$ for all $j$, yields $x_1 = .5789$, $x_2 = x_3 = x_4 = 1, x_5 = .7368$, and $z = 108.68$ (million $\$$). The solution is meaningless because two of the variables assume fractional values. We may round the solution to the closest integer values, which yields $x_1 = x_2 = 1$. However, the resulting solution is infeasible because the constraints are violated. More important, the concept of rounding is meaningless here because $x_j$ represents a "yes-no" decision.
Example 9.1-2 (Installing Security Telephones)

To promote on-campus safety, the U of A Security Department is in the process of installing emergency telephones at selected locations. The department wants to install the minimum number of telephones, provided that each of the campus main streets is served by at least one telephone. Figure 9.1 maps the principal streets (A to K) on campus.

It is logical to place the telephones at street intersections so that each telephone will serve at least two streets. Figure 9.1 shows that the layout of the streets requires a maximum of eight telephone locations.

Define

\[ x_j = \begin{cases} 1, & \text{a telephone is installed in location } j \\ 0, & \text{otherwise} \end{cases} \]

The constraints of the problem require installing at least one telephone on each of the 11 streets (A to K). Thus, the model becomes

\[
\text{Minimize } z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8
\]

subject to

\[
x_1 + x_2 \geq 1 \quad \text{(Street A)}
\]
\[
x_2 + x_3 \geq 1 \quad \text{(Street B)}
\]
\[
x_4 + x_5 \geq 1 \quad \text{(Street C)}
\]

**FIGURE 9.1**
Street Map of the U of A Campus

\[
x_7 + x_8 \geq 1 \quad \text{(Street D)}
\]
\[
x_6 + x_7 \geq 1 \quad \text{(Street E)}
\]
\[
x_2 + x_6 \geq 1 \quad \text{(Street F)}
\]
\[
x_1 + x_6 \geq 1 \quad \text{(Street G)}
\]
\[
x_4 + x_7 \geq 1 \quad \text{(Street H)}
\]
\[
x_2 + x_4 \geq 1 \quad \text{(Street I)}
\]
\[
x_5 + x_8 \geq 1 \quad \text{(Street J)}
\]
\[
x_3 + x_5 \geq 1 \quad \text{(Street K)}
\]

The optimum solution of the problem requires installing four telephones at intersections 1, 2, 5, and 7.

**Remarks.** In the strict sense, set-covering problems are characterized by (1) the variables \( x_j, j = 1, 2, \ldots, n \), are binary, (2) the left-hand-side coefficients of the constraints are 0 or 1, (3) the right-hand side of each constraint is of the form \( \geq 1 \), and (4) the objective function minimizes \( c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \), where \( c_j > 0 \) for all \( j = 1, 2, \ldots, n \). In the present example, \( c_j = 1 \) for all \( j \). If \( c_j \) represents the installation cost in location \( j \), then these coefficients may assume values other than 1. Variations of the set-covering problem include additional side conditions, as some of the situations in Problem Set 9.1b show.
Example 9.1-3 (Choosing a Telephone Company)

I have been approached by three telephone companies to subscribe to their long distance service in the United States. MaBell will charge a flat $16 per month plus $.25 a minute. PaBell will charge $25 a month but will reduce the per-minute cost to $.21. As for BabyBell, the flat monthly charge is $18, and the cost per minute is $.22. I usually make an average of 200 minutes of long-distance calls a month. Assuming that I do not pay the flat monthly fee unless I make calls and that I can apportion my calls among all three companies as I please, how should I use the three companies to minimize my monthly telephone bill?

This problem can be solved readily without ILP. Nevertheless, it is instructive to formulate it as an integer program.

Define

\[ x_1 = \text{MaBell long-distance minutes per month} \]
\[ x_2 = \text{PaBell long-distance minutes per month} \]
\[ x_3 = \text{BabyBell long-distance minutes per month} \]
\[ y_j = 1 \text{ if } x_j > 0 \text{ and } 0 \text{ if } x_j = 0 \]
\[ y_j = 1 \text{ if } x_j > 0 \text{ and } 0 \text{ if } x_j = 0 \]
\[ y_j = 1 \text{ if } x_j > 0 \text{ and } 0 \text{ if } x_j = 0 \]

We can ensure that \( y_j \) will equal 1 if \( x_j \) is positive by using the constraint

\[ x_j \leq M y_j, j = 1, 2, 3 \]

The value of \( M \) should be selected sufficiently large so as not restrict the variable \( x_j \) artificially. Because I make about 200 minutes of calls a month, then \( x_j \leq 200 \) for all \( j \), and it is safe to select \( M = 200 \).

The complete model is

Minimize \( z = .25x_1 + .21x_2 + .22x_3 + 16y_1 + 25y_2 + 18y_3 \)

subject to

\[ x_1 + x_2 + x_3 = 200 \]
\[ x_1 \leq 200 y_1 \]
\[ x_2 \leq 200 y_2 \]
\[ x_3 \leq 200 y_3 \]
\[ x_1, x_2, x_3 \geq 0 \]
\[ y_1, y_2, y_3 = (0, 1) \]

The formulation shows that the \( j \)th monthly flat fee will be part of the objective function \( z \) only if \( y_j = 1 \), which can happen only if \( x_j > 0 \) (per the last three constraints of the model). If \( x_j = 0 \) at the optimum, then the minimization of \( z \), together with the fact that the objective coefficient of \( y_j \) is strictly positive, will force \( y_j \) to equal zero, as desired.

The optimum solution yields \( x_1 = 200 \), \( y_1 = 1 \), and all the remaining variables equal to zero, which shows that BabyBell should be selected as my long-distance carrier. Remember that the information conveyed by \( y_1 = 1 \) is redundant because the same result is implied by \( x_3 > 0 \) (\( = 200 \)). Actually, the main reason for using \( y_1, y_2 \), and \( y_3 \) is to account for the monthly flat fee. In effect, the three binary variables convert an ill-behaved (nonlinear) model into an analytically tractable formulation. This conversion has resulted in introducing the integer (binary) variables in an otherwise continuous problem.
**Example 9.1-4 (Job-Sequencing Model)**

Jobco uses a single machine to process three jobs. Both the processing time and the due date (in days) for each job are given in the following table. The due dates are measured from zero, the assumed start time of the first job.

<table>
<thead>
<tr>
<th>Job</th>
<th>Processing time (days)</th>
<th>Due date (days)</th>
<th>Late penalty $/day</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>25</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>22</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>35</td>
<td>34</td>
</tr>
</tbody>
</table>

The objective of the problem is to determine the minimum late-penalty sequence for processing the three jobs.

Define

\[ x_j = \text{Start date in days for job } j \text{ (measured from zero)} \]

The problem has two types of constraints: the noninterference constraints (guaranteeing that no two jobs are processed concurrently) and the due-date constraints. Consider the noninterference constraints first.

Two jobs \( i \) and \( j \) with processing time \( p_i \) and \( p_j \) will not be processed concurrently if either \( x_i \geq x_j + p_j \) or \( x_j \geq x_i + p_i \), depending on whether job \( j \) precedes job \( i \), or vice versa. Because all mathematical programs deal with *simultaneous* constraints only, we transform the either-or constraints by introducing the following auxiliary binary variable:

\[
y_{ij} = \begin{cases} 
1, & \text{if } i \text{ precedes } j \\
0, & \text{if } j \text{ precedes } i
\end{cases}
\]

For \( M \) sufficiently large, the either-or constraint is converted to the following two simultaneous constraints

\[ M y_{ij} + (x_i - x_j) \geq p_j \text{ and } M(1 - y_{ij}) + (x_j - x_i) \geq p_i \]

The conversion guarantees that only one of the two constraints can be active at any one time. If \( y_{ij} = 0 \), the first constraint is active, and the second is redundant (because its left-hand side will include \( M \), which is much larger than \( p_i \)). If \( y_{ij} = 1 \), the first constraint is redundant, and the second is active.

Next, the due-date constraint is considered. Given that \( d_j \) is the due date for job \( j \), let \( s_j \) be an unrestricted variable. Then, the associated constraint is

\[ x_j + p_j + s_j = d_j \]

If \( s_j \geq 0 \), the due date is met, and if \( s_j < 0 \), a late penalty applies. Using the substitution

\[ s_j = s^*_j - s^+_j, s^+_j, s^-_j \geq 0 \]

the constraint becomes

\[ x_j + s^-_j - s^+_j = d_j - p_j \]

The late-penalty cost is proportional to \( s^+_j \).

The model for the given problem is
Minimize \( z = 19s_1^+ + 12s_2^+ + 34s_3^+ \)

subject to

\[
\begin{align*}
    x_1 - x_2 &+ M\gamma_{12} & \geq 20 \\
    -x_1 + x_2 &- M\gamma_{12} & \geq 5 - M \\
    x_1 - x_3 &+ M\gamma_{13} & \geq 15 \\
    -x_1 + x_3 &- M\gamma_{13} & \geq 5 - M \\
    x_2 - x_3 &+ M\gamma_{23} & \geq 15 \\
    -x_2 + x_3 &- M\gamma_{23} & \geq 20 - M \\
    x_1 &+ s_1^- - s_1^0 & = 25 - 5 \\
    x_2 &+ s_2^- - s_2^0 & = 22 - 20 \\
    x_3 &+ s_3^- - s_3^0 & = 35 - 15 \\
    x_1, x_2, x_3, s_1^+, s_2^+, s_3^+, s_1^-, s_2^-, s_3^- & \geq 0 \\
    \gamma_{12}, \gamma_{13}, \gamma_{23} & \in (0, 1)
\end{align*}
\]

The integer variables, \( \gamma_{12}, \gamma_{13}, \) and \( \gamma_{23} \) are introduced to convert the either-or constraints into simultaneous constraints. The resulting model is a mixed ILP.

To solve the model, we choose \( M = 100 \), a value that is larger than the sum of the processing times for all three activities.

The optimal solution is \( x_1 = 20 \), \( x_2 = 0 \), and \( x_3 = 25 \). This means that job 2 starts at time 0, job 1 starts at time 20, and job 3 starts at time 25, thus yielding the optimal processing sequence \( 2 \to 1 \to 3 \). The solution calls for completing job 2 at time \( 0 + 20 = 20 \), job 1 at time \( 20 + 5 = 25 \), and job 3 at \( 25 + 15 = 40 \) days. Job 3 is delayed by \( 40 - 35 = 5 \) days past its due date at a cost of \( 5 \times $34 = $170 \).