C H A P T E R 9

Integer Linear Programming

Chapter Guide. Integer linear programs (ILPs) are linear programs with some or all the variables restricted to integer (or discrete) values. When you study ILP, you need to concentrate on three areas: application, theory, and computation. The chapter starts with a number of applications that demonstrate the rich use of ILP in practice. Then it presents the two prominent algorithms of ILP: branch and bound (B&B) and cutting plane. Of the two algorithms, B&B is decidedly more efficient computationally. Indeed, practically all commercial codes are rooted in B&B. The chapter closes with a presentation of the traveling salesperson problem (TSP), a problem that has important practical applications.

A drawback of ILP algorithms is their lack of consistency in solving integer problems. Although these algorithms are proven theoretically to converge in a finite number of iterations, their implementation on the computer (with its inherent machine roundoff error) is a different experience. You should keep this point in mind as you study the ILP algorithms.

The chapter shows how AMPL and Solver are used with ILP. You will find TORA's user-guided option useful in detailing the B&B computations.

This chapter includes a summary of 1 real-life application, 12 solved examples, 5 AMPL models, 1 Excel spreadsheet, 65 end-of-section problems, and 10 cases. The cases are in Appendix E on the CD. The AMPL/Excel/Solver/TORA programs are in folder ch9Files.

Real-Life Application—Optimizing Trailer Payloads at PFG Building Glass

PFG uses specially equipped (fifth-wheel) trailers to deliver packs of sheets of flat glass to customers. The packs vary in both size and weight, and a single trailer load may include different packs, depending on received orders. Government regulations set maximum limits on axle weights, and the actual positioning of the packs on the trailer is crucial in determining these weights. The problem deals with determining the optimal loading of the packs on the trailer bed to satisfy axle-weight limits. The problem is solved as an integer program. Case 7 in Chapter 24 on the CD provides the details of the study.

9.1 ILLUSTRATIVE APPLICATIONS

This section presents a number of ILP applications. The applications generally fall into two categories: *direct* and *transformed*. In the *direct* category, the variables are naturally integer and may assume binary (0 or 1) or general discrete values. For example, the problem may involve determining whether or not a project is selected for execution (binary) or finding the optimal number of machines needed to perform a task (general discrete value). In the *transformed* category, the original problem, which may not involve any integer variables, is analytically intractable. Auxiliary integer variables (usually binary) are used to make it tractable. For example, in sequencing two jobs, A and B, on a single machine, job A may precede job B *or* job B may precede job A. The "or" nature of the constraints is what makes the problem analytically intractable, because all mathematical programming algorithms deal with "and" constraints only. The situation is remedied by using auxiliary binary variables to transform the "or" constraints into *equivalent* "and" constraints.

For convenience, a **pure** integer problem is defined to have *all* integer variables. Otherwise, a problem is a **mixed** integer program if it deals with both continuous and integer variables.

9.1.1 Capital Budgeting

This section deals with decisions regarding whether or not investments should be made in individual projects. The decision is made under limited-budget considerations as well as priorities in the execution of the projects.

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.

	Expenditures (million $)/yr			
Project	*1*	*2*	*3*	Returns (million $)
1	5	1	8	20
2	4	7	10	40
3	3	9	2	20
4	7	4	1	15
5	8	6	10	30
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

$$\text{Maximize } z = 20x_1 + 40x_2 + 20x_3 + 15x_4 + 30x_5$$

subject to

$$\begin{aligned} 5x_1 + 4x_2 + 3x_3 + 7x_4 + 8x_5 &\le 25 \\ x_1 + 7x_2 + 9x_3 + 4x_4 + 6x_5 &\le 25 \\ 8x_1 + 10x_2 + 2x_3 + x_4 + 10x_5 &\le 25 \\ x_1, x_2, x_3, x_4, x_5 &= (0, 1) \end{aligned}$$

The optimum integer solution (obtained by AMPL, Solver, or TORA)[1] 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 $0 \le x_j \le 1$ 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 $x_1 = x_5 = 1$. 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.

PROBLEM SET 9.1A[2]

1. Modify and solve the capital budgeting model of Example 9.1-1 to account for the following additional restrictions:
   (a) Project 5 must be selected if either project 1 or project 3 is selected.
   (b) Projects 2 and 3 are mutually exclusive.
2. Five items are to be loaded in a vessel. The weight w_i, volume v_i, and value r_i for item i are tabulated below.

Item i	Unit weight, w_i (tons)	Unit volume, v_i (yd^3)	Unit worth, r_i (100 \$)
1	5	1	4
2	8	8	7
3	3	6	6
4	2	5	5
5	7	4	4

[1]To use TORA, select Integer Programming from Main Menu. After entering the problem data, go to output screen and select Automated B&B to obtain the optimum solution. Solver use is the same as in LP except that the targeted variables must be declared integer. The integer option (*int* or *bin*) is available in the **Solver Parameters** dialogue box when you add a new constraint. AMPL implementation for integer programming is the same as in linear programming, except that some or all the variables are declared integer by adding the key word `integer` (or `binary`) in the definition statement of the targeted variables. For example, the statement `var x{J}>=0,integer;` declares x_j as nonnegative integer for all $j \in J$. If x_j is binary, the statement is changed to `var x{J} binary;`. For execution, the statement `option solver cplex;` must precede `solve;`.
[2]Problems 3 to 6 are adapted from Malba Tahan, *El Hombre que Calculaba*, Editorial Limusa, Mexico City, pp. 39–182, 1994.

The maximum allowable cargo weight and volume are 112 tons and 109 yd^3, respectively. Formulate the ILP model, and find the most valuable cargo.

*3. Suppose that you have 7 full wine bottles, 7 half-full, and 7 empty. You would like to divide the 21 bottles among three individuals so that each will receive exactly 7. Additionally, each individual must receive the same quantity of wine. Express the problem as ILP constraints, and find a solution. (*Hint*: Use a dummy objective function in which all the objective coefficients are zeros.)

4. An eccentric sheikh left a will to distribute a herd of camels among his three children: Tarek receives at least one-half of the herd, Sharif gets at least one third, and Maisa gets at least one-ninth. The remainder goes to charity. The will does not specify the size of the herd except to say that it is an odd number of camels and that the named charity receives exactly one camel. Use ILP to determine how many camels the sheikh left in the estate and how many each child got.

5. A farm couple are sending their three children to the market to sell 90 apples with the objective of educating them about money and numbers. Karen, the oldest, carries 50 apples; Bill, the middle one, carries 30; and John, the youngest, carries only 10. The parents have stipulated five rules: (a) The selling price is either \$1 for 7 apples or \$3 for 1 apple, or a combination of the two prices. (b) Each child may exercise one or both options of the selling price. (c) Each of the three children must return with exactly the same amount of money. (d) Each child's income must be in whole dollars (no cents allowed). (e) The amount received by each child must be the largest possible under the stipulated conditions. Given that the three kids are able to sell all they have, use ILP to show how they can satisfy the parents' conditions.

*6. Once upon a time, there was a captain of a merchant ship who wanted to reward three crew members for their valiant effort in saving the ship's cargo during an unexpected storm in the high seas. The captain put aside a certain sum of money in the purser's office and instructed the first officer to distribute it equally among the three mariners after the ship had reached shore. One night, one of the sailors, unbeknown to the others, went to the purser's office and decided to claim (an equitable) one-third of the money in advance. After he had divided the money into three equal shares, an extra coin remained, which the mariner decided to keep (in addition to one-third of the money). The next night, the second mariner got the same idea and, repeating the same three-way division with what was left, ended up keeping an extra coin as well. The third night, the third mariner also took a third of what was left, plus an extra coin that could not be divided. When the ship reached shore, the first officer divided what was left of the money equally among the three mariners, again to be left with an extra coin. To simplify things, the first officer put the extra coin aside and gave the three mariners their allotted equal shares. How much money was in the safe to start with? Formulate the problem as an ILP, and find the solution. (*Hint*: The problem has a countably infinite number of integer solutions. For convenience, assume that we are interested in determining the smallest sum of money that satisfies the problem conditions. Then, boosting the resulting sum by 1, add it as a lower bound and obtain the next smallest sum. Continuing in this manner, a general solution pattern will evolve.)

7. (Weber, 1990) You have the following three-letter words: AFT, FAR, TVA, ADV, JOE, FIN, OSF, and KEN. Suppose that we assign numeric values to the alphabet starting with

$A = 1$ and ending with $Z = 26$. Each word is scored by adding numeric codes of its three letters. For example, AFT has a score of $1 + 6 + 20 = 27$. You are to select five of the given eight words that yield the maximum total score. Simultaneously, the selected five words must satisfy the following conditions:

$$\begin{pmatrix}\text{sum of letter 1}\\\text{scores}\end{pmatrix} < \begin{pmatrix}\text{sum of letter 2}\\\text{scores}\end{pmatrix} < \begin{pmatrix}\text{sum of letter 3}\\\text{scores}\end{pmatrix}$$

Formulate the problem as an ILP, and find the optimum solution.

8. Solve Problem 7 given that, in addition to the total sum being the largest, the sum of column 1 and the sum of column 2 will be the largest as well. Find the optimum solution.

9. (Weber, 1990) Consider the following two groups of words:

Group 1	Group 2
AREA	ERST
FORT	FOOT
HOPE	HEAT
SPAR	PAST
THAT	PROF
TREE	STOP

All the words in groups 1 and 2 can be formed from the nine letters A, E, F, H, O, P, R, S, and T. Develop a model to assign a unique numeric value from 1 through 9 to these letters such that the difference between the total scores of the two groups will be as small as possible. [*Note*: The score for a word is the sum of the numeric values assigned to its individual letters.]

***10.** The Record-a-Song Company has contracted with a rising star to record eight songs. The durations of the different songs are 8, 3, 5, 5, 9, 6, 7, and 12 minutes, respectively. Record-a-Song uses a two-sided cassette tape for the recording. Each side has a capacity of 30 minutes. The company would like to distribute the songs between the two sides such that the length of the songs on each side is about the same. Formulate the problem as an ILP, and find the optimum solution.

11. In Problem 10, suppose that the nature of the melodies dictates that songs 3 and 4 cannot be recorded on the same side. Formulate the problem as an ILP. Would it be possible to use a 25-minute tape (each side) to record the eight songs? If not, use ILP to determine the minimum tape capacity needed to make the recording.

***12.** (Graves and Associates, 1993) Ulern University uses a mathematical model that optimizes student preferences taking into account the limitation of classroom and faculty resources. To demonstrate the application of the model, consider the simplified case of 10 students who are required to select two courses out of six offered electives. The table below gives scores that represent each student's preference for individual courses, with a score of 100 being the highest. For simplicity, it is assumed that the preference score for a two-course selection is the sum of the individual score. Course capacity is the maximum number of students allowed to take the class.

	Preference score for course					
Student	*1*	*2*	*3*	*4*	*5*	*6*
1	20	40	50	30	90	100
2	90	100	80	70	10	40
3	25	40	30	80	95	90
4	80	50	60	80	30	40
5	75	60	90	100	50	40
6	60	40	90	10	80	80
7	45	40	70	60	55	60
8	30	100	40	70	90	55
9	80	60	100	70	65	80
10	40	60	80	100	90	10
Course capacity	6	8	5	5	6	5

Formulate the problem as an ILP and find the optimum solution.

9.1.2 Set-Covering Problem

In this class of problems, overlapping services are offered by a number of installations to a number of facilities. The objective is to determine the minimum number of installations that will *cover* (i.e., satisfy the service needs) of each facility. For example, water treatment plants can be constructed at various locations, with each plant serving different sets of cities. The overlapping arises when a given city can receive service from more than one plant.

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

$$\begin{aligned}
&x_1 + x_2 && \geq 1 \quad (\text{Street } A)\\
&\quad x_2 + x_3 && \geq 1 \quad (\text{Street } B)\\
&\qquad\quad x_4 + x_5 && \geq 1 \quad (\text{Street } C)
\end{aligned}$$

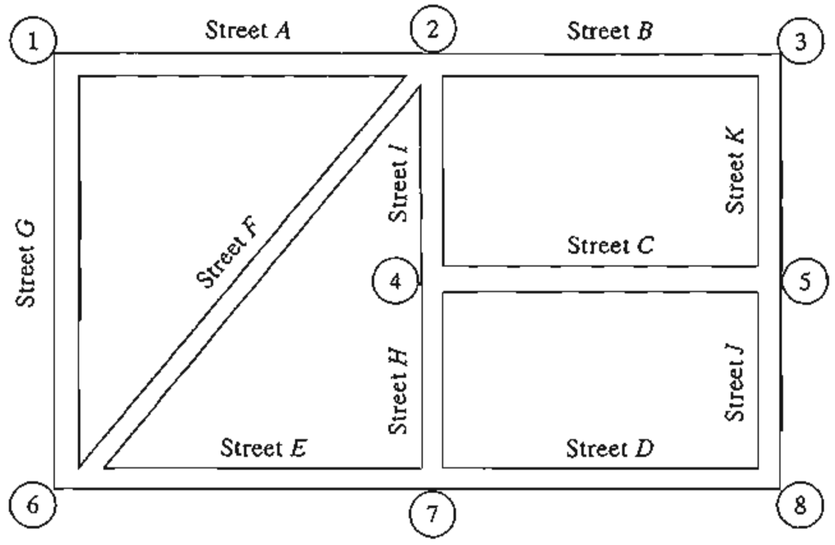


FIGURE 9.1
Street Map of the U of A Campus

$$
\begin{array}{rcl}
x_7 + x_8 \geq 1 & & \text{(Street } D\text{)} \\
x_6 + x_7 \geq 1 & & \text{(Street } E\text{)} \\
x_2 + x_6 \geq 1 & & \text{(Street } F\text{)} \\
x_1 + x_6 \geq 1 & & \text{(Street } G\text{)} \\
x_4 + x_7 \geq 1 & & \text{(Street } H\text{)} \\
x_2 + x_4 \geq 1 & & \text{(Street } I\text{)} \\
x_5 + x_8 \geq 1 & & \text{(Street } J\text{)} \\
x_3 + x_5 \geq 1 & & \text{(Street } K\text{)} \\
x_j = (0, 1), j = 1, 2, \ldots, 8 & &
\end{array}
$$

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 (≥ 1), and (4) the objective function minimizes $c_1x_1 + c_2x_2 + \cdots + c_nx_n$, 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.

AMPL Moment

Figure 9.2 presents a general AMPL model for any set-covering problem (file amplEx9.1-2.txt). The formulation is straightforward, once the use of *indexed set* is understood (see Section A.4). The model defines `street` as a (regular) set whose elements

```
#--------------Example 9.1-2-------------------
param n;    #maximum number of corners
set street;
set corner{street};
var x{1..n}binary;
minimize z: sum {j in 1..n} x[j];
subject to limit {i in street}:
        sum {j in corner[i]} x[j]>=1;
data;
param n:=8;
set street:=A B C D E F G H I J K;
set corner[A]:=1 2;
set corner[B]:=2 3;
set corner[C]:=4 5;
set corner[D]:=7 8;
set corner[E]:=6 7;
set corner[F]:=2 6;
set corner[G]:=1 6;
set corner[H]:=4 7;
set corner[I]:=2 4;
set corner[J]:=5 8;
set corner[K]:=3 5;

option solver cplex;
solve;
display z,x;
```

FIGURE 9.2

General AMPL model for the set-covering problem (file ampl Ex 9.1-2.txt)

are A through K. Next, the *indexed* set `corner{street}` defines the corners as a function of street. With these two sets, the constraints of the model can be formulated directly. The data of the model give the elements of the indexed sets that are specific to the situation in Example 9.1-2. Any other situation is handled by changing the data of the model.

PROBLEM SET 9.1B

*1. ABC is an LTL (less-than-truckload) trucking company that delivers loads on a daily basis to five customers. The following list provides the customers associated with each route:

Route	Customers served on the route
1	1, 2, 3, 4
2	4, 3, 5
3	1, 2, 5
4	2, 3, 5
5	1, 4, 2
6	1, 3, 5

The segments of each route are dictated by the capacity of the truck delivering the loads. For example, on route 1, the capacity of the truck is sufficient to deliver the loads to customers 1, 2, 3, and 4 only. The following table lists distances (in miles) among the truck terminal (ABC) and the customers.

	Miles from i to j					
i \ j	ABC	1	2	3	4	5
ABC	0	10	12	16	9	8
1	10	0	32	8	17	10
2	12	32	0	14	21	20
3	16	8	14	0	15	18
4	9	17	21	15	0	11
5	8	10	20	18	11	0

The objective is to determine the least distance needed to make the daily deliveries to all five customers. Though the solution may result in a customer being served by more than one route, the implementation phase will use only one such route. Formulate the problem as an ILP and find the optimum solution.

*2. The U of A is in the process of forming a committee to handle students' grievances. The administration wants the committee to include at least one female, one male, one student, one administrator, and one faculty member. Ten individuals (identified, for simplicity, by the letters *a* to *j*) have been nominated. The mix of these individuals in the different categories is given as follows:

Category	Individuals
Females	*a, b, c, d, e*
Males	*f, g, h, i, j*
Students	*a, b, c, j*
Administrators	*e, f*
Faculty	*d, g, h, i*

The U of A wants to form the smallest committee with representation from each of the five categories. Formulate the problem as an ILP and find the optimum solution.

3. Washington County includes six towns that need emergency ambulance service. Because of the proximity of some of the towns, a single station may serve more than one community. The stipulation is that the station must be within 15 minutes of driving time from the towns it serves. The table below gives the driving times in minutes among the six towns.

	Time in minutes from i to j					
i \ j	1	2	3	4	5	6
1	0	23	14	18	10	32
2	23	0	24	13	22	11
3	14	24	0	60	19	20
4	18	13	60	0	55	17
5	10	22	19	55	0	12
6	32	11	20	17	12	0

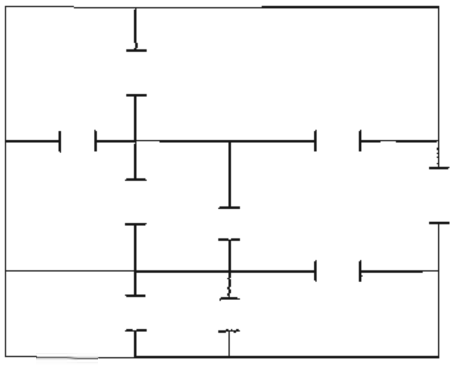


FIGURE 9.3
Museum Layout for Problem 4, Set 9.1c

Formulate an ILP whose solution will produce the smallest number of stations and their locations. Find the optimum solution.

4. The treasures of King Tut are on display in a museum in New Orleans. The layout of the museum is shown in Figure 9.3, with the different rooms joined by open doors. A guard standing at a door can watch two adjoining rooms. The museum wants to ensure guard presence in every room, using the minimum number possible. Formulate the problem as an ILP and find the optimum solution.

5. Bill has just completed his exams for the academic year and wants to celebrate by seeing every movie showing in theaters in his town and in six other neighboring cities. If he travels to another town, he will stay there until he has seen all the movies he wants. The following table provides the information about the movie offerings and the round-trip distance to the neighboring town.

Theater location	Movie offerings	Round-trip miles	Cost per show ($)
In-town	1, 3	0	7.95
City A	1, 6, 8	25	5.50
City B	2, 5, 7	30	5.00
City C	1, 8, 9	28	7.00
City D	2, 4, 7	40	4.95
City E	1, 3, 5, 10	35	5.25
City F	4, 5, 6, 9	32	6.75

The cost of driving is 75 cents per mile. Bill wishes to determine the towns he needs to visit to see all the movies while minimizing his total cost.

6. Walmark Stores is in the process of expansion in the western United States. During the next year, Walmark is planning to construct new stores that will serve 10 geographically dispersed communities. Past experience indicates that a community must be within 25 miles of a store to attract customers. In addition, the population of a community plays an important role in where a store is located, in the sense that bigger communities generate more participating customers. The following tables provide the populations as well as the distances (in miles) between the communities:

	Miles from community i to community j										
j / i	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*Population*
1		20	40	35	17	24	50	58	33	12	10,000
2	20		23	68	40	30	20	19	70	40	15,000
3	40	23		36	70	22	45	30	21	80	28,000
4	35	68	36		70	80	24	20	40	10	30,000
5	17	40	70	70		23	70	40	13	40	40,000
6	24	30	22	80	23		12	14	50	50	30,000
7	50	20	45	24	70	12		26	40	30	20,000
8	58	19	30	20	40	14	26		20	50	15,000
9	33	70	21	40	13	50	40	20		22	60,000
10	12	40	80	10	40	50	30	50	22		12,000

The idea is to construct the least number of stores, taking into account the distance restriction and the concentration of populations.

Specify the communities where the stores should be located.

***7.** (Guéret and Associates, 2002, Section 12.6) MobileCo is budgeting 15 million dollars to construct as many as 7 transmitters to cover as much population as possible in 15 contiguous geographical communities. The communities covered by each transmitter and the budgeted construction costs are given below.

Transmitter	Covered communities	Cost (million \$)
1	1, 2	3.60
2	2, 3, 5	2.30
3	1, 7, 9, 10	4.10
4	4, 6, 8, 9	3.15
5	6, 7, 9, 11	2.80
6	5, 7, 10, 12, 14	2.65
7	12, 13, 14, 15	3.10

The following table provides the populations of the different communities:

Community	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Population (in 1000s)	4	3	10	14	6	7	9	10	13	11	6	12	7	5	16

Which of the proposed transmitters should be constructed?

8. (Gavernini and Associates, 2004) In modern electric networks, automated electric utility meter reading replaces the costly labor-intensive system of manual meter reading. In the automated system, meters from several customers are linked wirelessly to a single receiver. The meter sends monthly signals to a designated receiver to report the customer's consumption of electricity. The receiver then sends the data to a central computer to generate the electricity bills. The problem reduces to determining the least number of receivers needed to serve a number of customers. In real life, the problem encompasses thousands of meters and receivers. However, for the purpose of this problem, consider the case of 10 meters and 8 receivers, using the following configurations:

Receiver	1	2	3	4	5	6	7	8
Meters	1, 2, 3	2, 3, 9	5, 6, 7	7, 9, 10	3, 6, 8	1, 4, 7, 9	4, 5, 9	1, 4, 8

Determine the minimum number of receivers.

9. Solve Problem 8 if, additionally, each receiver can handle at most 3 meters.

9.1.3 Fixed-Charge Problem

The fixed-charge problem deals with situations in which the economic activity incurs two types of costs: an initial "flat" fee that must be incurred to start the activity and a variable cost that is directly proportional to the level of the activity. For example, the initial tooling of a machine prior to starting production incurs a fixed setup cost regardless of how many units are manufactured. Once the setup is done, the cost of labor and material is proportional to the amount produced. Given that F is the fixed charge, c is the variable unit cost, and x is the level of production, the cost function is expressed as

$$C(x) = \begin{cases} F + cx, \text{if } x > 0 \\ 0, \text{otherwise} \end{cases}$$

The function $C(x)$ is intractable analytically because it involves a discontinuity at $x = 0$. The next example shows how binary variables are used to remove this intractability.

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

$$x_1 = \text{MaBell long-distance minutes per month}$$
$$x_2 = \text{PaBell long-distance minutes per month}$$
$$x_3 = \text{BabyBell long-distance minutes per month}$$
$$y_1 = 1 \text{ if } x_1 > 0 \text{ and } 0 \text{ if } x_1 = 0$$
$$y_2 = 1 \text{ if } x_2 > 0 \text{ and } 0 \text{ if } x_2 = 0$$
$$y_3 = 1 \text{ if } x_3 > 0 \text{ and } 0 \text{ if } x_3 = 0$$

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

$$x_j \leq My_j, j = 1, 2, 3$$

The value of M should be selected sufficiently large so as not restrict to 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

$$\text{Minimize } z = .25x_1 + .21x_2 + .22x_3 + 16y_1 + 25y_2 + 18y_3$$

subject to

$$\begin{aligned} x_1 + x_2 + x_3 &= 200 \\ x_1 \qquad\qquad &\leq 200y_1 \\ x_2 \qquad\qquad &\leq 200y_2 \\ x_3 \qquad\qquad &\leq 200y_3 \\ x_1, x_2, x_3 \quad &\geq 0 \\ y_1, y_2, y_3 \quad &= (0, 1) \end{aligned}$$

The formulation shows that the jth 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 $x_3 = 200$, $y_3 = 1$, 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 $y_3 = 1$ 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.

PROBLEM SET 9.1C

1. Leatherco is contracted to manufacture batches of pants, vests, and jackets. Each product requires a special setup of the machines needed in the manufacturing processes. The following table provides the pertinent data regarding the use of raw material (leather) and labor time together with cost and revenue estimates. Current supply of leather is estimated at 3000 ft^2 and available labor time is limited to 2500 hours.

	Pants	Vests	Jackets
Leather material per unit (ft^2)	5	3	8
Labor time per unit (hrs)	4	3	5
Production cost per unit ($)	30	20	80
Equipment setup cost per batch ($)	100	80	150
Price per unit ($)	60	40	120
Minimum number of units needed	100	150	200

Determine the optimum number of units that Leatherco must manufacture of each product.

*2. Jobco is planning to produce at least 2000 widgets on three machines. The minimum lot size on any machine is 500 widgets. The following table gives the pertinent data of the situation.

Machine	Cost	Production cost/unit	Capacity (units)
1	300	2	600
2	100	10	800
3	200	5	1200

Formulate the problem as an ILP, and find the optimum solution.

*3. Oilco is considering two potential drilling sites for reaching four targets (possible oil wells). The following table provides the preparation costs at each of the two sites and the cost of drilling from site i to target j ($i = 1, 2; j = 1, 2, 3, 4$).

	Drilling cost (million $) to target				
Site	*1*	*2*	*3*	*4*	Preparation cost (million $)
1	2	1	8	5	5
2	4	6	3	1	6

Formulate the problem as an ILP, and find the optimum solution.

4. Three industrial sites are considered for locating manufacturing plants. The plants send their supplies to three customers. The supply at the plants, the demand at the customers, and the unit transportation cost from the plants to the customers are given in the following table.

	Unit transportations cost ($)			
Plant \ *Customer*	*1*	*2*	*3*	*Supply*
1	10	15	12	1800
2	17	14	20	1400
3	15	10	11	1300
Demand	1200	1700	1600	

In addition to the transportation costs, fixed costs are incurred at the rate of $12,000, $11,000, and $12,000 for plants 1, 2, and 3, respectively. Formulate the problem as an ILP and find the optimum solution.

5. Repeat Problem 4 assuming that the demands at each of customers 2 and 3 are changed to 800.

6. (Liberatore and Miller, 1985) A manufacturing facility uses two production lines to produce three products over the next 6 months. Backlogged demand is not allowed. However, a product may be overstocked to meet demand in later months. The following table provides the data associated with the demand, production, and storage of the three products.

	Demand in period							
Product	*1*	*2*	*3*	*4*	*5*	*6*	Unit holding cost ($)/month	Initial inventory
1	50	30	40	60	20	45	.50	55
2	40	60	50	30	30	55	.35	75
3	30	40	20	70	40	30	.45	60

There is a fixed cost for switching a line from one product to another. The following tables give the switching cost, the production rates, and the unit production cost for each line:

	Line switching cost ($)		
	Product 1	*Product 2*	*Product 3*
Line 1	200	180	300
Line 2	250	200	174

	Production rate (units/month)			Unit production cost ($)		
	Product 1	*Product 2*	*Product 3*	*Product 1*	*Product 2*	*Product 3*
Line 1	40	60	80	10	8	15
Line 2	90	70	60	12	6	10

Develop a model for determining the optimal production schedule.

7. (Jarvis and Associates, 1978) Seven cities are being considered as potential locations for the construction of at most four wastewater treatment plants. The table below provides the data for the situation. Missing links indicate that a pipeline cannot be constructed.

	Cost ($) of pipeline construction between cities per 1000 gal/hr capacity						
To / From	*1*	*2*	*3*	*4*	*5*	*6*	*7*
1		100		200		50	
2				120		150	
3	400				120		90
4			120		120		
5		200				100	200
6			110	180			70
7	200			150			
Cost ($million) of plant construction	1.00	1.20	2.00	1.60	1.80	.90	1.40
Population (1000s)	50	100	45	90	75	60	30

The capacity of a pipeline (in gallons per hour) is a direct function of the amount of wastewater generated, which is a function of the populations. Approximately 500 gallons per 1000 residents are discharged in the sewer system per hour. The maximum plant capacity is 100,000 gal/hr. Determine the optimal location and capacity of the plants.

8. (Brown and Associates, 1987) A company uses four special tank trucks to deliver four different gasoline products to customers. Each tank has five compartments with different capacities: 500, 750, 1200, 1500, and 1750 gallons. The daily demands for the four products are estimated at 10, 15, 12, and 8 thousand gallons. Any quantities that cannot be delivered by the company's four trucks must be subcontracted at the additional costs of 5, 12, 8, and 10 cents per gallon for products 1, 2, 3, and 4, respectively. Develop the optimal daily loading schedule for the four trucks that will minimize the additional cost of subcontracting.
9. A household uses at least 3000 minutes of long-distance telephone calls monthly and can choose to use the services of any of three companies: A, B, and C. Company A charges a fixed monthly fee of $10 and 5 cents per minute for the first 1000 minutes and 4 cents per minute for all additional minutes. Company B's monthly fee is $20 with a flat 4 cents per minute. Company C's monthly charge is $25 with 5 cents per minute for the first 1000 minutes and 3.5 cents per minute beyond that limit. Which company should be selected to minimize the total monthly charge?

*10. (Barnett, 1987) Professor Yataha needs to schedule six round-trips between Boston and Washington, D.C. The route is served by three airlines: Eastern, US Air, and Continental and there is no penalty for the purchase of one-way tickets. Each airline offers bonus miles for frequent fliers. Eastern gives 1000 miles per (one-way) ticket plus 5000 extra miles if the number of tickets in a month reaches 2 and another 5000 miles if the number exceeds 5. US Air gives 1500 miles per trip plus 10,000 extra for each 6 tickets. Continental gives 1800 miles plus 7000 extra for each 5 tickets. Professor Yataha wishes to allocate the 12 one-way tickets among the three airlines to maximize the total number of bonus miles earned.

9.1.4 Either-Or and If-Then Constraints

In the fixed-charge problem (Section 9.1.3), we used binary variables to handle the discontinuity in the objective cost function. In this section, we deal with models in which constraints are not satisfied simultaneously (either-or) or are dependent (if-then), again using binary variables. The transformation does not change the "or" or "dependence" nature of the constraints. It simply uses a mathematical trick to present them in the desired format of "and" constraints.

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.

Job	Processing time (days)	Due date (days)	Late penalty $/day
1	5	25	19
2	20	22	12
3	15	35	34

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

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

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

$$\text{Minimize } z = 19s_1^+ + 12s_2^+ + 34s_3^+$$

subject to

$$\begin{array}{lllllllll}
x_1 - x_2 & + My_{12} & & & & & & \geq 20 \\
-x_1 + x_2 & - My_{12} & & & & & & \geq 5 - M \\
x_1 \quad - x_3 & & + My_{13} & & & & & \geq 15 \\
-x_1 \quad + x_3 & & - My_{13} & & & & & \geq 5 - M \\
\quad x_2 - x_3 & & & + My_{23} & & & & \geq 15 \\
\quad - x_2 + x_3 & & & - My_{23} & & & & \geq 20 - M \\
x_1 & & & & + s_1^- - s_1^+ & & & = 25 - 5 \\
\quad x_2 & & & & & + s_2^- - s_2^+ & & = 22 - 20 \\
\quad\quad x_3 & & & & & & + s_3^- - s_3^+ & = 35 - 15
\end{array}$$

$$x_1, x_2, x_3, s_1^-, s_1^+, s_2^-, s_2^+, s_3^-, s_3^+ \geq 0$$

$$y_{12}, y_{13}, y_{23} = (0, 1)$$

The integer variables, y_{12}, y_{13}, and y_{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 $2 \rightarrow 1 \rightarrow 3$. 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 $5 \times \$34 = \$170$.

AMPL Moment

File amplEx9.1-4.txt provides the AMPL model for the problem of Example 9.1-4. The model is self-explanatory because it is a direct translation of the general mathematical model given above. It can handle any number of jobs by changing the input data. Note that the model is a direct function of the raw data: processing time `p`, due date `d`, and delay penalty `perDayPenalty`.

FIGURE 9.4

AMPL model of the job sequencing problem (file amplEx9.1-4.txt)

```
#-----------------Example 9.1-4-------------------
param n;
set I={1..n};
set J={1..n};  #I is the same as J
param p{I};
param d{I};
param perDayPenalty{I};
param M=1000;
var x{J}>=0;            #continuous
var y{I,J} binary;      #0-1
var sMinus{J}>=0;       # s=sMinus-sPlus
var sPlus{J}>=0;
minimize penalty: sum {j in J}
           perDayPenalty[j]*sPlus[j];
subject to
eitherOr1{i in I,j in J:i<>j}:
          M*y[i,j]+x[i]-x[j]>=p[j];
eitherOr2{i in I,j in J:i<>j}:
          M*(1-y[i,j])+x[j]-x[i]>=p[i];
dueDate{j in J}:x[j]+sMinus[j]-sPlus[j]=d[j]-p[j];
data;
param n:=3;
param p:= 1 5  2 20  3 15;
param d:= 1 25  2 22  3 35;
param perDayPenalty := 1 19  2 12 3 34;
option solver cplex; solve;
display penalty,x;
```

Example 9.1-5 (Job Sequencing Model Revisited)

In Example 9.1-4, suppose that we have the following additional condition: If job i precedes job j then job k must precede job m. Mathematically, this **if-then condition** is translated as

$$\text{if } x_i + p_i \leq x_j \text{ then } x_k + p_k \leq x_m$$

Given $\varepsilon > 0$ and infinitesimally small and M sufficiently large, this condition is equivalent to the following two simultaneous constraints:

$$x_j - (x_i + p_i) \leq M(1 - w) - \varepsilon$$

$$(x_k + p_k) - x_m \leq Mw$$

$$w = (0, 1)$$

If $x_i + p_i \leq x_j$, then $x_j - (x_i + p_i) \geq 0$, which requires $w = 0$, and the second constraint becomes $x_k + p_k \leq x_m$, as desired. Else, w may assume the value 0 or 1, in which case the second constraint may or may not be satisfied, depending on other conditions in the model.

PROBLEM SET 9.1D

*1. A game board consists of nine equal squares. You are required to fill each square with a number between 1 and 9 such that the sum of the numbers in each row, each column, and each diagonal equals 15. Additionally, the numbers in all the squares must be distinct. Use ILP to determine the assignment of numbers to squares.

2. A machine is used to produce two interchangeable products. The daily capacity of the machine can produce at most 20 units of product 1 and 10 units of product 2. Alternatively, the machine can be adjusted to produce at most 12 units of product 1 and 25 units of product 2 daily. Market analysis shows that the maximum daily demand for the two products combined is 35 units. Given that the unit profits for the two respective products are \$10 and \$12, which of the two machine settings should be selected? Formulate the problem as an ILP and find the optimum. [*Note:* This two-dimensional problem can be solved by inspecting the graphical solution space. This is not the case for the n-dimensional problem.]

*3. Gapco manufactures three products, whose daily labor and raw material requirements are given in the following table.

Product	Required daily labor (hr/unit)	Required daily raw material (lb/unit)
1	3	4
2	4	3
3	5	6

The profits per unit of the three products are \$25, \$30, and \$22, respectively. Gapco has two options for locating its plant. The two locations differ primarily in the availability of labor and raw material, as shown in the following table:

Location	Available daily labor (hr)	Available daily raw material (lb)
1	100	100
2	90	120

Formulate the problem as an ILP, and determine the optimum location of the plant.

4. Jobco Shop has 10 outstanding jobs to be processed on a single machine. The following table provides processing times and due dates. All times are in days and due time is measured from time 0:

Job	Processing time	Due time
1	10	20
2	3	98
3	13	100
4	15	34
5	9	50
6	22	44
7	17	32
8	30	60
9	12	80
10	16	150

If job 4 precedes job 3, then job 9 must precede job 7. The objective is to process all 10 jobs in the shortest possible time. Formulate the model as an ILP and determine the optimum solution by modifying AMPL file amplEx9.1-4.txt.

5. In Problem 4, suppose that job 4 cannot be processed until job 3 has been completed. Also, machine settings for jobs 7 and 8 necessitate processing them one right after the other (i.e., job 7 immediately succeeds or immediately precedes 8). Jobco's objective is to process all ten jobs with the smallest sum of due-time violations. Formulate the model mathematically and determine the optimum solution.

6. Jaco owns a plant in which three products are manufactured. The labor and raw material requirements for the three products are given in the following table.

Product	Required daily labor (hr/unit)	Required daily raw material (lb/unit)
1	3	4
2	4	3
3	5	6
Daily availability	100	100

The profits per unit for the three products are \$25, \$30, and \$45, respectively. If product 3 is to be manufactured at all, then its production level must be at least 5 units daily. Formulate the problem as a mixed ILP, and find the optimal mix.

9.2

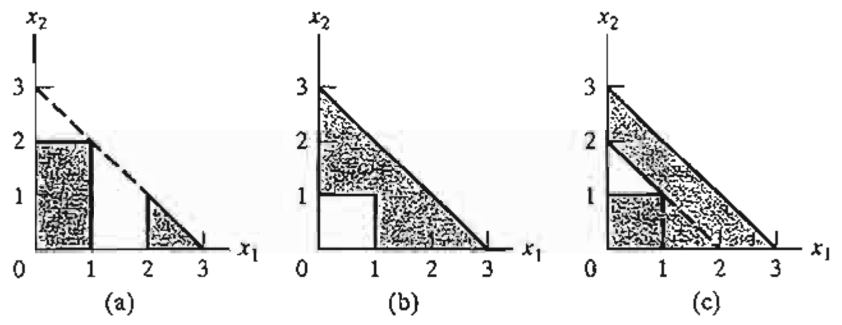


FIGURE 9.5

Solution spaces for Problem 9, Set 9.1d

7. UPak is a subsidiary of an LTL (less-than-truck-load) transportation company. Customers bring their shipments to the UPak terminal to be loaded on the trailer and can rent space up to 36 ft. The customer pays for the exact linear space (in foot increments) the shipment occupies. No partial shipment is allowed, in the sense that the entire shipment per customer must be on the same trailer. A movable barrier, called bulkhead, is installed to separate different shipments. The per-foot fee UPak collects depends on the destination of the shipment: The longer the trip, the higher the fee. The following table provides the outstanding orders UPak needs to process.

Order	1	2	3	4	5	6	7	8	9	10
Size (ft)	5	11	22	15	7	9	18	14	10	12
Rate ($)	120	93	70	85	125	104	98	130	140	65

The terminal currently has two trailers ready to be loaded. Determine the priority orders that will maximize the total income from the two trailers. (*Hint:* A formulation using binary x_{ij} to represent load i on trailer j is straightforward. However, you are challenged to define x_{ij} as *feet* assigned to load i in trailer j. The use *if-then* constraint to prevent partial load shipping.)

8. Show how the nonconvex shaded solution spaces in Figure 9.5 can be represented by a set of simultaneous constraints. Find the optimum solution that maximizes $z = 2x_1 + 3x_2$ subject to the solution space given in (a).

9. Suppose that it is required that *any k* out of the following m constraints must be active:

$$g_i(x_1, x_2, \ldots, x_n) \leq b_i, i = 1, 2, \ldots, m$$

Show how this condition may be represented.

10. In the following constraint, the right-hand side may assume one of values, $b_1, b_2, \ldots,$ and b_m.

$$g(x_1, x_2, \ldots, x_n) \leq (b_1, b_2, \ldots, \text{ or } b_m)$$

Show how this condition is represented.

9.2 INTEGER PROGRAMMING ALGORITHMS

The ILP algorithms are based on exploiting the tremendous computational success of LP. The strategy of these algorithms involves three steps.

Step 1. Relax the solution space of the ILP by deleting the integer restriction on all integer variables and replacing any binary variable y with the continuous range $0 \leq y \leq 1$. The result of the relaxation is a regular LP.

Step 2. Solve the LP, and identify its continuous optimum.

Step 3. Starting from the continuous optimum point, add special constraints that iteratively modify the LP solution space in a manner that will eventually render an optimum extreme point satisfying the integer requirements.

Two general methods have been developed for generating the special constraints in step 3.

1. Branch-and-bound (B&B) method
2. Cutting-plane method

Although neither method is consistently effective computationally, experience shows that the B&B method is far more successful than the cutting-plane method. This point is discussed further in this chapter.

9.2.1 Branch-and-Bound (B&B) Algorithm[3]

The first B&B algorithm was developed in 1960 by A. Land and G. Doig for the general mixed and pure ILP problem. Later, in 1965, E. Balas developed the **additive algorithm** for solving ILP problems with pure binary (zero or one) variables[4]. The additive algorithm's computations were so simple (mainly addition and subtraction) that it was hailed as a possible breakthrough in the solution of general ILP. Unfortunately, it failed to produce the desired computational advantages. Moreover, the algorithm, which initially appeared unrelated to the B&B technique, was shown to be but a special case of the general Land and Doig algorithm.

This section will present the general Land-Doig B&B algorithm only. A numeric example is used to explain the details.

Example 9.2-1

$$\text{Maxmize } z = 5x_1 + 4x_2$$

subject to

$$x_1 + x_2 \leq 5$$
$$10x_1 + 6x_2 \leq 45$$
$$x_1, x_2 \text{ nonnegative integer}$$

[3]TORA integer programming module is equipped with a facility for generating the B&B tree interactively. To use this facility, select User-guided B&B in the output screen of the integer programming module. The resulting screen provides all the information needed to create the B&B tree.

[4]A general ILP can be expressed in terms of binary (0–1) variables as follows. Given an integer variable x with a finite upper bound u (i.e., $0 \leq x \leq u$), then

$$x = 2^0 y_0 + 2^1 y_1 + 2^2 y_2 + \cdots + 2^k y_k$$

The variables $y_0, y_1, \ldots,$ and y_k are binary and the index k is the smallest integer satisfying $2^{k+1} - 1 \geq u$.

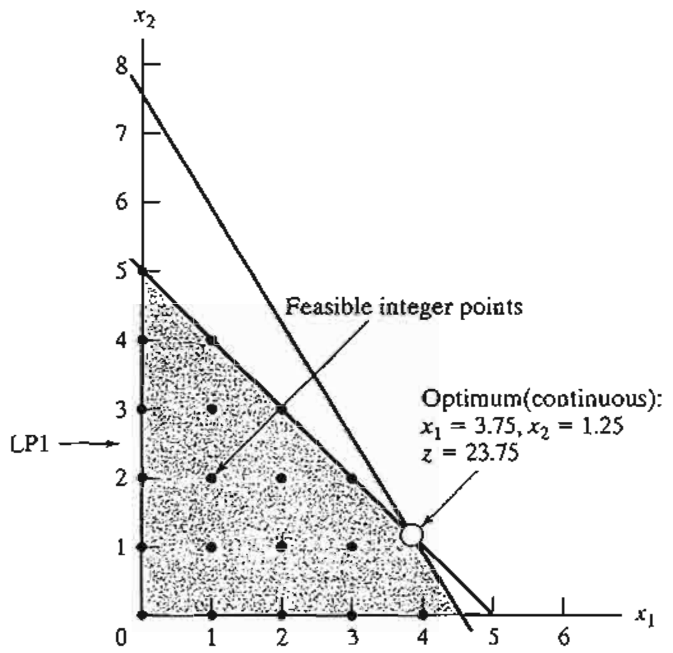


FIGURE 9.6
Solution spaces for ILP (lattice points) and LP1 (shaded area) of Example 9.2-1

The lattice points (dots) in Figure 9.6 define the ILP solution space. The associated continuous LP1 problem at node 1 (shaded area) is defined from ILP by removing the integer restrictions. The optimum solution of LP1 is $x_1 = 3.75$, $x_2 = 1.25$, and $z = 23.75$.

Because the optimum LP1 solution does not satisfy the integer requirements, the B&B algorithm modifies the solution space in a manner that eventually identifies the ILP optimum. First, we select one of the integer variables whose optimum value at LP1 is not integer. Selecting x_1 ($= 3.75$) arbitrarily, the region $3 < x_1 < 4$ of the LP1 solution space contains no integer values of x_1, and thus can be eliminated as nonpromising. This is equivalent to replacing the original LP1 with two new LPs:

$$\text{LP2 space} = \text{LP1 space} + (x_1 \leq 3)$$
$$\text{LP3 space} = \text{LP1 space} + (x_1 \geq 4)$$

Figure 9.7 depicts the LP2 and LP3 spaces. The two spaces combined contain the same feasible integer points as the original ILP, which means that, from the standpoint of the integer solution, dealing with LP2 and LP3 is the same as dealing with the original LP1; no information is lost.

If we *intelligently* continue to remove the regions that do not include integer solutions (e.g., $3 < x_1 < 4$ at LP1) by imposing the appropriate constraints, we will eventually produce LPs whose optimum extreme points satisfy the integer restrictions. In effect, we will be solving the ILP by dealing with a sequence of (continuous) LPs.

The new restrictions, $x_1 \leq 3$ and $x_1 \geq 4$, are mutually exclusive, so that LP2 and LP3 at nodes 2 and 3 must be dealt with as separate LPs, as Figure 9.8 shows. This dichotomization gives rise to the concept of **branching** in the B&B algorithm. In this case, x_1 is called the **branching variable**.

The optimum ILP lies in *either* LP2 *or* LP3. Hence, both subproblems must be examined. We arbitrarily examine LP2 (associated with $x_1 \leq 3$) first:

$$\text{Maxmize } z = 5x_1 + 4x_2$$

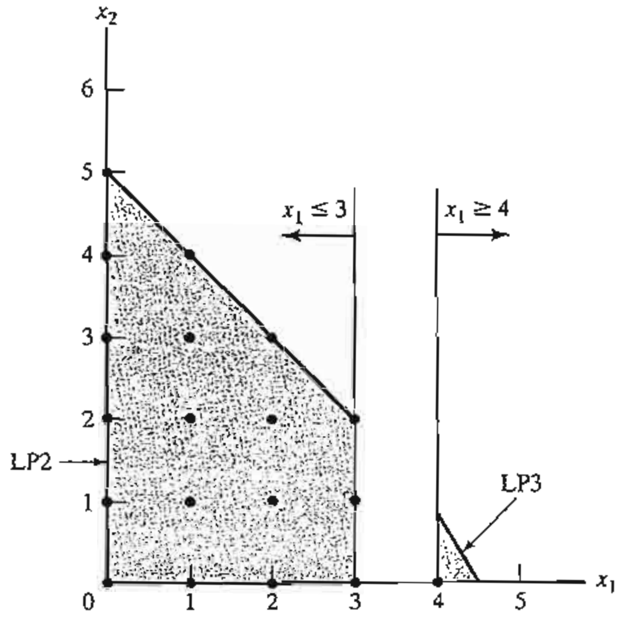
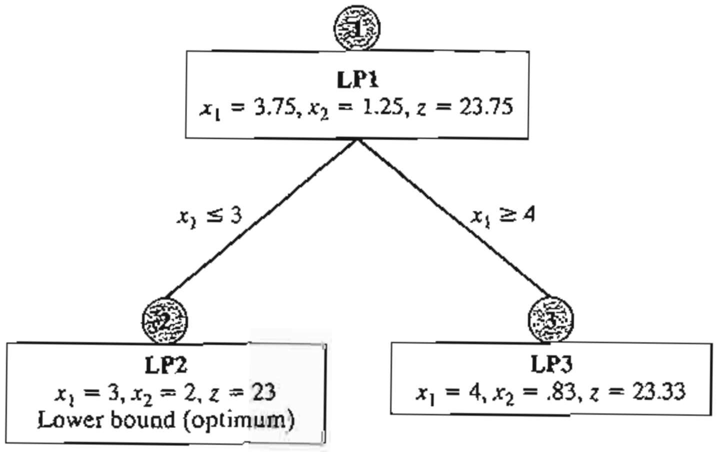


FIGURE 9.7
Solution Spaces of LP2 and LP3 for Example 9.2-1



LP1
$x_1 = 3.75, x_2 = 1.25, z = 23.75$
$x_1 \le 3$
$x_1 \ge 4$
LP2
$x_1 = 3, x_2 = 2, z = 23$
Lower bound (optimum)
LP3
$x_1 = 4, x_2 = .83, z = 23.33$

FIGURE 9.8
Using branching variable x_1 to create LP2 and LP3 for Example 9.2-1

subject to

$$\begin{aligned} x_1 + x_2 &\le 5 \\ 10x_1 + 6x_2 &\le 45 \\ x_1 &\le 3 \\ x_1, x_2 &\ge 0 \end{aligned}$$

The solution of LP2 (which can be solved efficiently by the upper-bounded algorithm of Section 7.3) yields the solution

$$x_1 = 3, x_2 = 2, \text{ and } z = 23$$

The LP2 solution satisfies the integer requirements for x_1 and x_2. Hence, LP2 is said to be **fathomed**, meaning that it need not be investigated any further because it cannot yield any *better* ILP solution.

We cannot at this point say that the integer solution obtained from LP2 is optimum for the original problem, because LP3 may yield a better integer solution with a higher value of z. All we can say is that $z = 23$ is **a lower bound** on the optimum (maximum) objective value of the original ILP. This means that any unexamined subproblem that cannot yield a better objective value than the lower bound must be discarded as nonpromising. If an unexamined subproblem produces a better integer solution, then the lower bound must be updated accordingly.

Given the lower bound $z = 23$, we examine LP3 (the only remaining unexamined subproblem at this point). Because optimum $z = 23.75$ at LP1 *and all the coefficients of the objective function happen to be integers,* it is impossible that LP3 (which is more restrictive than LP1) will produce a better integer solution with $z > 23$. As a result, we discard LP3 and conclude that it has been *fathomed.*

The B&B algorithm is now complete because both LP2 and LP3 have been examined and fathomed (the first for producing an integer solution and the second for failing to produce a *better* integer solution). We thus conclude that the optimum ILP solution is the one associated with the lower bound—namely, $x_1 = 3$, x_2, and $z = 23$.

Two questions remain unanswered regarding the procedure.

1. At LP1, could we have selected x_2 as the *branching variable* in place of x_1?
2. When selecting the next subproblem to be examined, could we have solved LP3 first instead of LP2?

The answer to both questions is "yes," but ensuing computations could differ dramatically. Figure 9.9 demonstrates this point. Suppose that we examine LP3 first (instead of LP2 as we did in Figure 9.8). The solution is $x_1 = 4$, $x_2 = .83$, and $z = 23.33$ (verify!). Because x_2 ($= .83$) is noninteger, LP3 is examined further by creating subproblems LP4 and LP5 using the branches $x_2 \leq 0$ and $x_2 \geq 1$, respectively. This means that

$$\begin{aligned}\text{LP4 space} &= \text{LP3 space} + (x_2 \leq 0)\\ &= \text{LP1 space} + (x_1 \geq 4) + (x_2 \leq 0)\\ \text{LP5 space} &= \text{LP3 space} + (x_2 \geq 1)\\ &= \text{LP1 space} + (x_1 \geq 4) + (x_2 \geq 1)\end{aligned}$$

We now have three "dangling" subproblems to be examined: LP2, LP4, and LP5. Suppose that we arbitrarily examine LP5 first. LP5 has no solution, and hence it is fathomed. Next, let us examine LP4. The optimum solution is $x_1 = 4.5$, $x_2 = 0$, and $z = 22.5$. The noninteger value of x_1 leads to the two branches $x_1 \leq 4$ and $x_1 \geq 5$, and the creation of subproblems LP6 and LP7 from LP4.

$$\begin{aligned}\text{LP6 space} &= \text{LP1 space} + (x_1 \geq 4) + (x_2 \leq 0) + (x_1 \leq 4)\\ \text{LP7 space} &= \text{LP1 space} + (x_1 \geq 4) + (x_2 \leq 0) + (x_1 \geq 5)\end{aligned}$$

Now, subproblems LP2, LP6, and LP7 remain unexamined. Selecting LP7 for examination, the problem has no feasible solution, and thus is fathomed. Next, we select LP6. The problem yields the first integer solution ($x_1 = 4$, $x_2 = 0$, $z = 20$), and thus provides the first lower bound

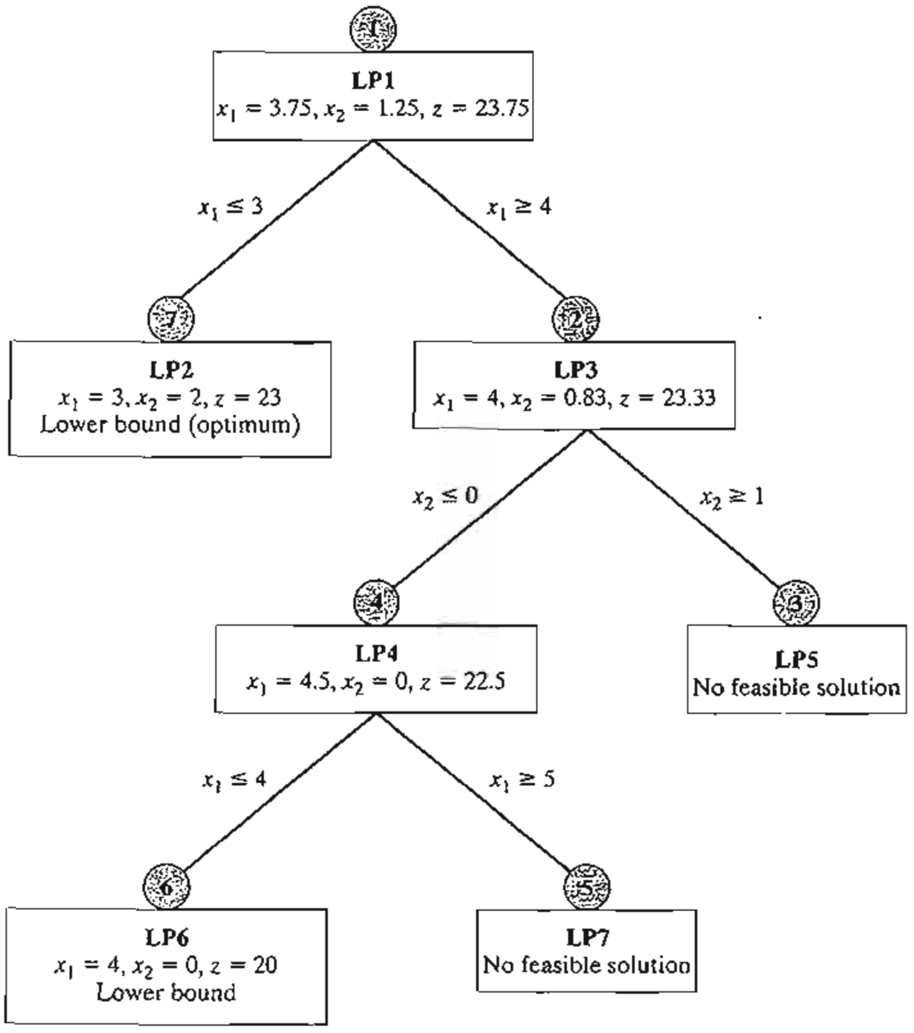


FIGURE 9.9

Alternative B&B tree for Example 9.2-1

(= 20) on the optimum ILP objective value. We are now left with subproblem LP2, which yields a better integer solution ($x_1 = 3, x_2 = 2, z = 23$). Thus, the lower bound is updated from $z = 20$ to $z = 23$. At this point, *all* the subproblems have been fathomed (examined) and the optimum solution is the one associated with the most up-to-date lower bound—namely, $x_1 = 3, x_2 = 2$, and $z = 23$.

The solution sequence in Figure 9.9 (LP1 $\rightarrow$ LP3 $\rightarrow$ LP5 $\rightarrow$ LP4 $\rightarrow$ LP7 $\rightarrow$ LP6 $\rightarrow$ LP2) is a worst-case scenario that, nevertheless, may well occur in practice. In Figure 9.8, we were lucky to "stumble" upon a good lower bound at the very first subproblem we examined (LP2), thus allowing us to fathom LP3 without investigating it. In essence, we completed the procedure by solving a total of two LPs. In Figure 9.9, the story is different: We needed to solve seven LPs before the B&B algorithm could be terminated.

Remarks. The example points to a principal weakness in the B&B algorithm: Given multiple choices, how do we select the next subproblem and its branching variable? Although there are heuristics for enhancing the ability of B&B to "foresee" which branch can lead to an improved ILP solution (see Taha, 1975, pp. 154–171), solid theory with consistent results does not exist, and herein lies the difficulty that plagues computations in ILP. Indeed, Problem 7, Set 9.2a, demonstrates the bizarre behavior of the B&B algorithm in investigating over 25,000 LPs before optimality is verified, even though the problem is quite small (16 binary variables and 1 constraint). Unfortunately, to date, and after more than four decades of research coupled with tremendous advances in computing power, available ILP codes (commercial and academic alike) are not totally reliable, in the sense that they may not find the optimum ILP solution regardless of how long they execute on the computer. What is even more frustrating is that this behavior can apply just the same to some relatively small problems.

AMPL Moment

AMPL can be used interactively to generate the B&B search tree. The following table shows the sequence of commands needed to generate the tree of Example 9.2-1 (Figure 9.9) starting with the continuous LP0. AMPL model (file amplEx9.2-1.txt) has two variables `x1` and `x2` and two constraints `c0` and `c1`. You will find it helpful to synchronize the AMPL commands with the branches in Figure 9.9.

AMPL command	Result
ampl: `model amplEx9.2-1.txt;solve;display x1,x2;`	LP1 ($x_1 = 3.75, x_2 = 1.25$)
ampl: `c2:x1>=4;solve;display x1,x2;`	LP3 ($x_1 = 4, x_2 = .83$)
ampl: `c3:x2>=1;solve;display x1,x2;`	LP5 (no solution)
ampl: `drop c3;c4:x2<=0;solve;display x1,x2;`	LP4 ($x_1 = 4.5, x_2 = 0$)
ampl: `c5:x1>=5;solve;display x1,x2;`	LP7 (no solution)
ampl: `drop c5;c6:x1<=4;solve;display x1,x2;`	LP6 ($x_1 = 4, x_2 = 0$)
ampl: `drop c2;drop c4;drop c6;c7:x1<=3;` `solve;display x1,x2;`	LP2 ($x_1 = 3, x_2 = 2$)

Solver Moment

Solver can be used to obtain the solution of the different subproblems by using the add/change/delete options in the **Solver Parameters** dialogue box.

Summary of the B&B Algorithm. We now summarize the B&B algorithm. Assuming a maximization problem, set an initial lower bound $\underline{z} = -\infty$ on the optimum objective value of ILP. Set $i = 0$.

Step 1. (*Fathoming/bounding*). Select LP*i*, the next subproblem to be examined. Solve LP*i*, and attempt to fathom it using one of three conditions:

(a) The optimal z-value of LP*i* cannot yield a better objective value than the current lower bound.

(b) LP*i* yields a better feasible integer solution than the current lower bound.

(c) LP*i* has no feasible solution.

Two cases will arise.

(a) If LPi is fathomed and a better solution is found, update the lower bound. If all subproblems have been fathomed, stop; the optimum ILP is associated with the current finite lower bound. If no finite lower bound exists, the problem has no feasible solution. Else, set $i = i + 1$, and repeat step 1.

(b) If LPi is not fathomed, go to step 2 for branching.

Step 2. (*Branching*). Select one of the integer variables x_j, whose optimum value x_j^* in the LPi solution is not integer. Eliminate the region

$$[x_j^*] < x_j < [x_j^*] + 1$$

(where $[v]$ defines the largest integer $\leq v$) by creating two LP subproblems that correspond to

$$x_j \leq [x_j^*] \text{ and } x_j \geq [x_j^*] + 1$$

Set $i = i + 1$, and go to step 1.

The given steps apply to maximization problems. For minimization, we replace the lower bound with an upper bound (whose initial value is $z = +\infty$).

The B&B algorithm can be extended directly to mixed problems (in which only some of the variables are integer). If a variable is continuous, we simply never select it as a branching variable. A feasible subproblem provides a new bound on the objective value if the values of the discrete variables are integer and the objective value is improved relative to the current bound.

PROBLEM SET 9.2A[5]

1. Solve the ILP of Example 9.2-1 by the B&B algorithm starting with x_2 as the branching variable. Start the procedure by solving the subproblem associated with $x_2 \leq [x_2^*]$.
2. Develop the B&B tree for each of the following problems. For convenience, always select x_1 as the branching variable at node 0.

   *(a) Maximize $z = 3x_1 + 2x_2$
   subject to

$$2x_1 + 5x_2 \leq 9$$
$$4x_1 + 2x_2 \leq 9$$
$$x_1, x_2 \geq 0 \text{ and integer}$$

[5]In this set, you may solve the subproblems interactively with AMPL or Solver or using TORA's MODIFY option for the upper and lower bounds.

(b) Maximize $z = 2x_1 + 3x_2$
subject to

$$5x_1 + 7x_2 \leq 35$$
$$4x_1 + 9x_2 \leq 36$$
$$x_1, x_2 \geq 0 \text{ and integer}$$

(c) Maximize $z = x_1 + x_2$
subject to

$$2x_1 + 5x_2 \leq 16$$
$$6x_1 + 5x_2 \leq 27$$
$$x_1, x_2 \geq 0 \text{ and integer}$$

***(d)** Minimize $z = 5x_1 + 4x_2$
subject to

$$3x_1 + 2x_2 \geq 5$$
$$2x_1 + 3x_2 \geq 7$$
$$x_1, x_2 \geq 0 \text{ and integer}$$

(e) Maximize $z = 5x_1 + 7x_2$
subject to

$$2x_1 + x_2 \leq 13$$
$$5x_1 + 9x_2 \leq 41$$
$$x_1, x_2 \geq 0 \text{ and integer}$$

***3.** Repeat Problem 2, assuming that x_1 is continuous.

4. Show graphically that the following ILP has no feasible solution, and then verify the result using B&B.

$$\text{Maximize } z = 2x_1 + x_2$$

subject to

$$10x_1 + 10x_2 \leq 9$$
$$10x_1 + 5x_2 \geq 1$$
$$x_1, x_2 \geq 0 \text{ and integer}$$

5. Solve the following problems by B&B.

$$\text{Maximize } z = 18x_1 + 14x_2 + 8x_3 + 4x_4$$

subject to

$$15x_1 + 12x_2 + 7x_3 + 4x_4 + x_5 \leq 37$$
$$x_1, x_2, x_3, x_4, x_5 = (0, 1)$$

6. Convert the following problem into a mixed ILP and find the optimum solution.

$$\text{Maximize } z = x_1 + 2x_2 + 5x_3$$

subject to

$$|-x_1 + 10x_2 - 3x_3| \geq 15$$
$$2x_1 + x_2 + x_3 \leq 10$$
$$x_1, x_2, x_3 \geq 0$$

7. *TORA/Solver/AMPL Experiment.* The following problem is designed to demonstrate the bizarre behavior of the B&B algorithm even for small problems. In particular, note how many subproblems are examined before the optimum is found and how many are needed to verify optimality.

$$\text{Minimize } y$$

subject to

$$2(x_1 + x_2 + \cdots + x_{15}) + y = 15$$
$$\text{All variables are } (0, 1)$$

(a) Use TORA's automated option to show that although the optimum is found after only 9 subproblems, over 25,000 subproblems are examined before optimality is confirmed.
(b) Show that Solver exhibits an experience similar to TORA's. [*Note*: In Solver, you can watch the change in the number of generated branches (subproblems) at the bottom of the spreadsheet.]
(c) Solve the problem with AMPL and show that the solution is obtained instantly with 0 MIP simplex iterations and 0 B&B nodes. The reason for this superior performance can only be attributed to preparatory steps performed by AMPL and/or the CPLEX solver prior to solving the problem.

8. *TORA Experiment.* Consider the following ILP:

$$\text{Maximize } z = 18x_1 + 14x_2 + 8x_3$$

subject to

$$15x_1 + 12x_2 + 7x_3 \leq 43$$
$$x_1, x_2, x_3 \text{ nonnegative integers}$$

Use TORA's B&B user-guided option to generate the search tree with and without activating the objective-value bound. What is the impact of activating the objective-value bound on the number of generated subproblems? For consistency, always select the branching variable as the one with the lowest index and investigate all the subproblems in a current row from left to right before moving to the next row.

*9. *TORA Experiment.* Reconsider Problem 8 above. Convert the problem into an equivalent 0-1 ILP, then solve it with TORA's automated option. Compare the size of the search trees in the two problems.

10. *AMPL Experiment.* In the following 0-1 ILP use interactive AMPL to generate the associated search tree. In each case, show how the z-bound is used to fathom subproblems.

$$\text{Maximize } z = 3x_1 + 2x_2 - 5x_3 - 2x_4 + 3x_5$$

subject to

$$\begin{aligned} x_1 + x_2 + x_3 + 2x_4 + x_5 &\leq 4 \\ 7x_1 + 3x_3 - 4x_4 + 3x_5 &\leq 8 \\ 11x_1 - 6x_2 + 3x_4 - 3x_5 &\geq 3 \\ x_1, x_2, x_3, x_4, x_5 &= (0, 1) \end{aligned}$$

9.2.2 Cutting-Plane Algorithm

As in the B&B algorithm, the cutting-plane algorithm also starts at the continuous optimum LP solution. Special constraints (called **cuts**) are added to the solution space in a manner that renders an integer optimum extreme point. In Example 9.2-2, we first demonstrate graphically how cuts are used to produce an integer solution and then implement the idea algebraically.

Example 9.2-2

Consider the following ILP.

$$\text{Maximize } z = 7x_1 + 10x_2$$

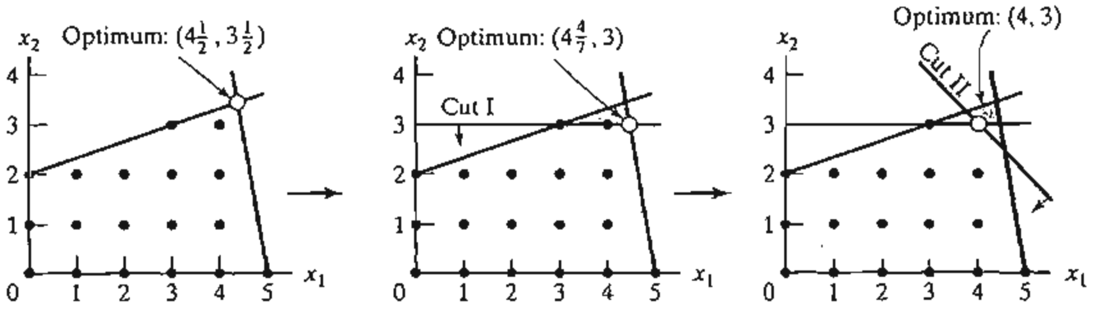
subject to

$$\begin{aligned} -x_1 + 3x_2 &\leq 6 \\ 7x_1 + x_2 &\leq 35 \\ x_1, x_2 &\geq 0 \text{ and integer} \end{aligned}$$

The cutting-plane algorithm modifies the solution space by adding *cuts* that produce an optimum integer extreme point. Figure 9.10 gives an example of two such cuts.
Initially, we start with the continuous LP optimum $z = 66\frac{1}{2}$, $x_1 = 4\frac{1}{2}$, $x_2 = 3\frac{1}{2}$. Next, we add cut I, which produces the (continuous) LP optimum solution $z = 62$, $x_1 = 4\frac{4}{7}$, $x_2 = 3$. Then, we add cut II, which, together with cut I and the original constraints, produces the LP optimum $z = 58$, $x_1 = 4$, $x_2 = 3$. The last solution is all integer, as desired.

The added cuts do not eliminate any of the original feasible integer points, but must pass through at least one feasible or infeasible integer point. These are basic requirements of any cut.

FIGURE 9.10
Illustration of the use of cuts in ILP


x_2 Optimum: $(4\frac{1}{2}, 3\frac{1}{2})$
x_2 Optimum: $(4\frac{4}{7}, 3)$
Cut I
Optimum: (4, 3)
Cut II
x_1
0 1 2 3 4 5
1 2 3 4

It is purely accidental that a 2-variable problem used exactly 2 cuts to reach the optimum integer solution. In general, the number of cuts, though finite, is independent of the size of the problem, in the sense that a problem with a small number of variables and constraints may require more cuts than a larger problem.

Next, we use the same example to show how the cuts are constructed and implemented algebraically.

Given the slacks x_3 and x_4 for constraints 1 and 2, the optimum LP tableau is given as

Basic	x_1	x_2	x_3	x_4	Solution
z	0	0	$\frac{63}{22}$	$\frac{31}{22}$	$66\frac{1}{2}$
x_2	0	1	$\frac{7}{22}$	$\frac{1}{22}$	$3\frac{1}{2}$
x_1	1	0	$-\frac{1}{22}$	$\frac{3}{22}$	$4\frac{1}{2}$

The optimum continuous solution is $z = 66\frac{1}{2}$, $x_1 = 4\frac{1}{2}$, $x_2 = 3\frac{1}{2}$, $x_3 = 0$, $x_4 = 0$. The cut is developed under the assumption that *all* the variables (including the slacks x_3 and x_4) are integer. Note also that because all the original objective coefficients are integer in this example, the value of z is integer as well.

The information in the optimum tableau can be written explicitly as

$$z + \frac{63}{22}x_3 + \frac{31}{22}x_4 = 66\tfrac{1}{2} \qquad (z\text{-equation})$$

$$x_2 + \frac{7}{22}x_3 + \frac{1}{22}x_4 = 3\tfrac{1}{2} \qquad (x_2\text{-equation})$$

$$x_1 - \frac{1}{22}x_3 + \frac{3}{22}x_4 = 4\tfrac{1}{2} \qquad (x_1\text{-equation})$$

A constraint equation can be used as a **source row** for generating a cut, provided its right-hand side is fractional. We also note that the z-equation can be used as a source row because z happens to be integer in this example. We will demonstrate how a cut is generated from each of these source rows, starting with the z-equation.

First, we factor out all the noninteger coefficients of the equation into an integer value and a fractional component, *provided that the resulting fractional component is strictly positive.* For example,

$$\frac{5}{2} = \left(2 + \frac{1}{2}\right)$$

$$-\frac{7}{3} = \left(-3 + \frac{2}{3}\right)$$

The factoring of the z-equation yields

$$z + \left(2 + \frac{19}{22}\right)x_3 + \left(1 + \frac{9}{22}\right)x_4 = \left(66 + \frac{1}{2}\right)$$

Moving all the integer components to the left-hand side and all the fractional components to the right-hand side, we get

$$z + 2x_3 + 1x_4 - 66 = -\frac{19}{22}x_3 - \frac{9}{22}x_4 + \frac{1}{2} \qquad (1)$$

Because x_3 and x_4 are nonnegative and all fractions are originally strictly positive, the right-hand side must satisfy the following inequality:

$$-\tfrac{19}{22}x_3 - \tfrac{9}{22}x_4 + \tfrac{1}{2} \leq \tfrac{1}{2} \tag{2}$$

Next, because the left-hand side in Equation (1), $z + 2x_3 + 1x_4 - 66$, is an integer value by construction, the right-hand side, $-\frac{19}{22}x_3 - \frac{19}{22}x_4 + \frac{1}{2}$, must also be integer. It then follows that (2) can be replaced with the inequality:

$$-\tfrac{19}{22}x_3 - \tfrac{9}{22}x_4 + \tfrac{1}{2} \leq 0$$

This result is justified because an integer value $\leq \frac{1}{2}$ must necessarily be ≤ 0.

The last inequality is the desired cut and it represents a *necessary* (but not sufficient) condition for obtaining an integer solution. It is also referred to as the **fractional cut** because all its coefficients are fractions.

Because $x_3 = x_4 = 0$ in the optimum continuous LP tableau given above, the current continuous solution violates the cut (because it yields $\frac{1}{2} \leq 0$). Thus, if we add this cut to the optimum tableau, the resulting optimum extreme point moves the solution toward satisfying the integer requirements.

Before showing how a cut is implemented in the optimal tableau, we will demonstrate how cuts can also be constructed from the constraint equations. Consider the x_1-row:

$$x_1 - \tfrac{1}{22}x_3 + \tfrac{3}{22}x_4 = 4\tfrac{1}{2}$$

Factoring the equation yields

$$x_1 + \left(-1 + \tfrac{21}{22}\right)x_3 + \left(0 + \tfrac{3}{22}\right)x_4 = \left(4 + \tfrac{1}{2}\right)$$

The associated cut is

$$-\tfrac{21}{22}x_2 - \tfrac{3}{22}x_4 + \tfrac{1}{2} \leq 0$$

Similarly, the x_2-equation

$$x_2 + \tfrac{7}{22}x_3 + \tfrac{1}{22}x_4 = 3\tfrac{1}{2}$$

is factored as

$$x_2 + \left(0 + \tfrac{7}{22}\right)x_3 + \left(0 + \tfrac{1}{22}\right)x_4 = 3 + \tfrac{1}{2}$$

Hence, the associated cut is given as

$$-\tfrac{7}{22}x_3 - \tfrac{1}{22}x_4 + \tfrac{1}{2} \leq 0$$

Any one of three cuts given above can be used in the first iteration of the cutting-plane algorithm. It is not necessary to generate all three cuts before selecting one.

Arbitrarily selecting the cut generated from the x_2-row, we can write it in equation form as

$$-\tfrac{7}{22}x_3 - \tfrac{1}{22}x_4 + s_1 = -\tfrac{1}{2}, \quad s_1 \geq 0 \qquad \text{(Cut I)}$$

This constraint is added to the LP optimum tableau as follows:

Basic	x_1	x_2	x_3	x_4	s_1	Solution
z	0	0	$\frac{63}{22}$	$\frac{31}{22}$	0	$66\frac{1}{2}$
x_2	0	1	$\frac{7}{22}$	$\frac{1}{22}$	0	$3\frac{1}{2}$
x_1	1	0	$-\frac{1}{22}$	$\frac{3}{22}$	0	$4\frac{1}{2}$
s_1	0	0	$-\frac{7}{22}$	$-\frac{1}{22}$	1	$-\frac{1}{2}$

The tableau is optimal but infeasible. We apply the dual simplex method (Section 4.4.1) to recover feasibility, which yields

Basic	x_1	x_2	x_3	x_4	s_1	Solution
z	0	0	0	1	9	62
x_2	0	1	0	0	1	3
x_1	1	0	0	$\frac{1}{7}$	$-\frac{1}{7}$	$4\frac{4}{7}$
x_3	0	0	1	$\frac{1}{7}$	$-\frac{22}{7}$	$1\frac{4}{7}$

The last solution is still noninteger in x_1 and x_3. Let us arbitrarily select x_1 as the next source row—that is,

$$x_1 + \left(0 + \tfrac{1}{7}\right)x_4 + \left(-1 + \tfrac{6}{7}\right)s_1 = 4 + \tfrac{4}{7}$$

The associated cut is

$$-\tfrac{1}{7}x_4 - \tfrac{6}{7}s_1 + s_2 = -\tfrac{4}{7}, \quad s_2 \geq 0 \qquad \text{(Cut II)}$$

Basic	x_1	x_2	x_3	x_4	s_1	s_2	Solution
z	0	0	0	1	9	0	62
x_2	0	1	0	0	1	0	3
x_1	1	0	0	$\frac{1}{7}$	$-\frac{1}{7}$	0	$4\frac{4}{7}$
x_3	0	0	1	$\frac{1}{7}$	$-\frac{22}{7}$	0	$1\frac{4}{7}$
s_2	0	0	0	$-\frac{1}{7}$	$-\frac{6}{7}$	1	$-\frac{4}{7}$

The dual simplex method yields the following tableau:

Basic	x_1	x_2	x_3	x_4	s_1	s_2	Solution
z	0	0	0	0	3	7	58
x_2	0	1	0	0	1	0	3
x_1	1	0	0	0	-1	1	4
x_3	0	0	1	0	-4	1	1
x_4	0	0	0	1	6	-7	4

The optimum solution ($x_1 = 4, x_2 = 3, z = 58$) is all integer. It is not accidental that all the coefficients of the last tableau are integers, a property of the implementation of the fractional cut.

Remarks. It is important to point out that the fractional cut assumes that *all* the variables, *including slack and surplus,* are integer. This means that the cut deals with pure integer problems only. The importance of this assumption is illustrated by an example.

Consider the constraint

$$x_1 + \tfrac{1}{3}x_2 \leq \tfrac{13}{2}$$
$$x_1, x_2 \geq 0 \text{ and integer}$$

From the standpoint of solving the associated ILP, the constraint is treated as an equation by using the nonnegative slack s_1—that is,

$$x_1 + \tfrac{1}{3}x_2 + s_1 = \tfrac{13}{2}$$

The application of the fractional cut assumes that the constraint has a feasible integer solution in all x_1, x_2, and s_1. However, the equation above will have a feasible integer solution in x_1 and x_2 *only if s_1 is noninteger.* This means that the cutting-plane algorithm will show that the problem has no feasible integer solution, even though the variables of concern, x_1 and x_2, can assume feasible integer values.

There are two ways to remedy this situation.

1. Multiply the entire constraint by a proper constant to remove all the fractions. For example, multiplying the constraint above by 6, we get

$$6x_1 + 2x_2 \leq 39$$

Any integer solution of x_1 and x_2 automatically yields integer slack. However, this type of conversion is appropriate for only simple constraints, because the magnitudes of the integer coefficients may become excessively large in some cases.

2. Use a special cut, called the **mixed cut**, which allows only a subset of variables to assume integer values, with all the other variables (including slack and surplus) remaining continuous. The details of this cut will not be presented in this chapter (see Taha, 1975, pp. 198–202).

PROBLEM SET 9.2B

1. In Example 9.2-2, show graphically whether or not each of the following constraints can form a legitimate cut:
   *(a) $x_1 + 2x_2 \leq 10$
   (b) $2x_1 + x_2 \leq 10$
   (c) $3x_2 \leq 10$
   (d) $3x_1 + x_2 \leq 15$

2. In Example 9.2-2, show graphically how the following two (legitimate) cuts can lead to the optimum integer solution:

$$x_1 + 2x_2 \leq 10 \qquad \text{(cut I)}$$
$$3x_1 + x_2 \leq 15 \qquad \text{(cut II)}$$

3. Express cuts I and II of Example 9.2-2 in terms of x_1 and x_2 and show that they are the same ones used graphically in Figure 9.10.
4. In Example 9.2-2, derive cut II from the x_3-row. Use the new cut to complete the solution of the example.
5. Show that, even though the following problem has a feasible integer solution in x_1 and x_2, the fractional cut would not yield a feasible solution unless all the fractions in the constraint were eliminated.

$$\text{Maximize } z = x_1 + 2x_2$$

subject to

$$x_1 + \tfrac{1}{2}x_2 \leq \tfrac{13}{4}$$
$$x_1, x_2 \geq 0 \text{ and integer}$$

6. Solve the following problems by the fractional cut, and compare the true optimum integer solution with the solution obtained by rounding the continuous optimum.

***(a)** Maximize $z = 4x_1 + 6x_2 + 2x_3$
subject to

$$\begin{aligned} 4x_1 - 4x_2 \quad &\leq 5 \\ -x_1 + 6x_2 \quad &\leq 5 \\ -x_1 + x_2 + x_3 &\leq 5 \\ x_1, x_2, x_3 &\geq 0 \text{ and integer} \end{aligned}$$

(b) Maximize $z = 3x_1 + x_2 + 3x_3$
subject to

$$\begin{aligned} -x_1 + 2x_2 + x_3 &\leq 4 \\ 4x_2 - 3x_3 &\leq 2 \\ x_1 - 3x_2 + 2x_3 &\leq 3 \\ x_1, x_2, x_3 &\geq 0 \text{ and integer} \end{aligned}$$

9.2.3 Computational Considerations in ILP

To date, and despite over 40 years of research, there does not exist a computer code that can solve ILP consistently. Nevertheless, of the two solution algorithms presented in this chapter, B&B is more reliable. Indeed, practically all commercial ILP codes are B&B based. Cutting-plane methods are generally difficult and uncertain, and the roundoff error presents a serious problem. This is true because the "accuracy" of the cut depends on the accuracy of a true representation of its fractions on the computer. For instance, in Example 9.2-2, the fraction $\frac{1}{7}$ cannot be represented exactly as a floating

9.3

point regardless of the level of precision that may be used. Though attempts have been made to improve the cutting-plane computational efficacy, the end results are not encouraging. In most cases, the cutting-plane method is used in a secondary capacity to improve B&B performance at each subproblem by eliminating a portion of the solution space associated with a subproblem.

The most important factor affecting computations in integer programming is the number of integer variables and the feasible range in which they apply. Because available algorithms are not consistent in producing a numeric ILP solution, it may be advantageous computationally to reduce the number of integer variables in the ILP model as much as possible. The following suggestions may prove helpful:

1. Approximate integer variables by continuous ones wherever possible.
2. For the integer variables, restrict their feasible ranges as much as possible.
3. Avoid the use of nonlinearity in the model.

The importance of the integer problem in practice is not yet matched by the development of reliable solution algorithms. The nature of discrete mathematics and the fact that the integer solution space is a nonconvex set make it unlikely that new theoretical breakthroughs will be achieved in the area of integer programming. Instead, new technological advances in computers (software and hardware) remain the best hope for improving the efficiency of ILP codes.

9.3 TRAVELING SALESPERSON (TSP) PROBLEM

Historically, the TSP problem deals with finding the shortest (closed) tour in an n-city situation where each city is visited exactly once. The problem, in essence, is an assignment model that excludes subtours. Specifically, in an n-city situation, define

$$x_{ij} = \begin{cases} 1, \text{ if city } j \text{ is reached from city } i \\ 0, \text{ otherwise} \end{cases}$$

Given that d_{ij} is the distance from city i to city j, the TSP model is given as

$$\text{Minimize } z = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij},\ d_{ij} = \infty \text{ for all } i = j$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1, i = 1, 2, \ldots, n \tag{1}$$

$$\sum_{i=1}^{n} x_{ij} = 1, j = 1, 2, \ldots, n \tag{2}$$

$$x_{ij} = (0, 1) \tag{3}$$

$$\text{Solution forms an } n\text{-city tour} \tag{4}$$

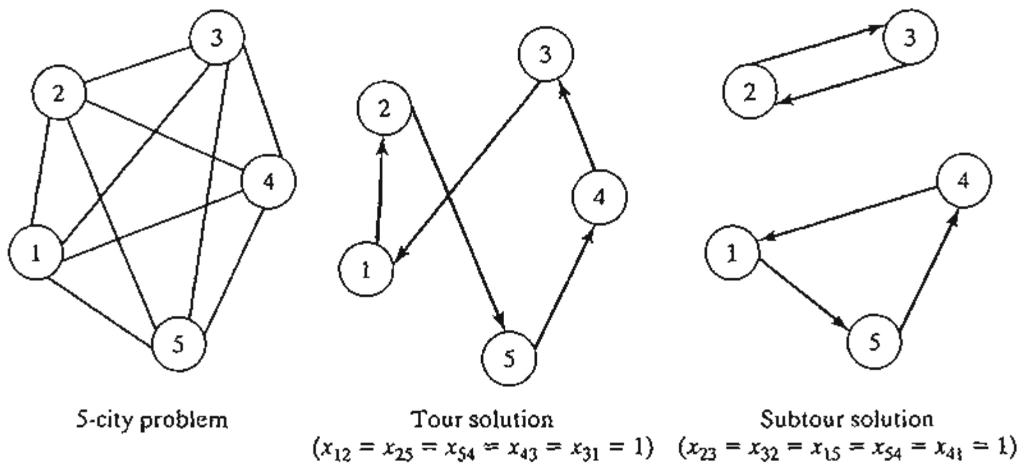

5-city problem
Tour solution
$(x_{12} = x_{25} = x_{54} = x_{43} = x_{31} = 1)$
Subtour solution
$(x_{23} = x_{32} = x_{15} = x_{54} = x_{41} = 1)$

FIGURE 9.11
A 5-city TSP example with a tour and subtour solutions of the associated assignment model

Constraints (1), (2), and (3) define a regular assignment model (Section 5.4). Figure 9.11 demonstrates a 5-city problem. The arcs represent two-way routes. The figure also illustrates a tour and a subtour solution of the associated assignment model. If the optimum solution of the assignment model (i.e., excluding constraint 4) happens to produce a tour, then it is also optimum for the TSP. Otherwise, restriction (4) must be accounted for to ensure a tour solution.

Exact solutions of the TSP problem include branch-and-bound and cutting-plane algorithms. Both are rooted in the ideas of the general B&B and cutting plane algorithms presented in Section 9.2. Nevertheless, the problem is typically difficult computationally, in the sense that either the size or the computational time needed to obtain a solution may become inordinately large. For this reason, heuristics are sometimes used to provide a "good" solution for the problem.

Before presenting the heuristic and exact solution algorithms, we present an example that demonstrates the versatility of the TSP model in representing other practical situations (see also Problem Set 9.3a).

Example 9.3-1

The daily production schedule at the Rainbow Company includes batches of white (W), yellow (Y), red (R), and black (B) paints. Because Rainbow uses the same facilities for all four types of paint, proper cleaning between batches is necessary. The table below summarizes the clean-up time in minutes. Because each color is produced in a single batch, diagonal entries in the table are assigned infinite setup time. The objective is to determine the optimal sequencing for the daily production of the four colors that will minimize the associated total clean-up time.

	Cleanup min given next paint is			
Current paint	*White*	*Yellow*	*Black*	*Red*
White	∞	10	17	15
Yellow	20	∞	19	18
Black	50	44	∞	25
Red	45	40	20	∞

Each paint is thought of as a "city" and the "distances" are represented by the clean-up time needed to switch from one paint batch to the next. The situation reduces to determining the *shortest loop* that starts with one paint batch and passes through each of the remaining three paint batches exactly once before returning back to the starting paint.

We can solve this problem by exhaustively enumerating the six $[(4-1)! = 3! = 6]$ possible loops of the network. The following table shows that $W \rightarrow Y \rightarrow R \rightarrow B \rightarrow W$ is the optimum loop.

Production loop	Total clean-up time
$W \rightarrow Y \rightarrow B \rightarrow R \rightarrow W$	$10 + 19 + 25 + 45 = 99$
$W \rightarrow Y \rightarrow R \rightarrow B \rightarrow W$	$10 + 18 + 20 + 50 = 98$
$W \rightarrow B \rightarrow Y \rightarrow R \rightarrow W$	$17 + 44 + 18 + 45 = 124$
$W \rightarrow B \rightarrow R \rightarrow Y \rightarrow W$	$17 + 25 + 40 + 20 = 102$
$W \rightarrow R \rightarrow B \rightarrow Y \rightarrow W$	$15 + 20 + 44 + 20 = 99$
$W \rightarrow R \rightarrow Y \rightarrow B \rightarrow W$	$15 + 40 + 19 + 50 = 124$

Exhaustive enumeration of the loops is not practical in general. Even a modest size 11-city problem will require enumerating 10! = 3,628,800 tours, a daunting task indeed. For this reason, the problem must be formulated and solved in a different manner, as we will show later in this section.

To develop the assignment-based formulation for the paint problem, define

$$x_{ij} = 1 \text{ if paint } j \text{ follows paint } i \text{ and zero otherwise}$$

Letting M be a sufficiently large positive value, we can formulate the Rainbow problem as

$$\begin{aligned}\text{Minimize } z = {} & Mx_{WW} + 10x_{WY} + 17x_{WB} + 15x_{WR} + 20x_{YW} + Mx_{YY} + 19x_{YB} + 18x_{YR} \\ & + 50x_{BW} + 44x_{BY} + Mx_{BB} + 25x_{BR} + 45x_{RW} + 40x_{RY} + 20x_{RB} + Mx_{RR}\end{aligned}$$

subject to

$$\begin{aligned} x_{WW} + x_{WY} + x_{WB} + x_{WR} &= 1 \\ x_{YW} + x_{YY} + x_{YB} + x_{YR} &= 1 \\ x_{BW} + x_{BY} + x_{BB} + x_{BR} &= 1 \\ x_{RW} + x_{RY} + x_{RB} + x_{RR} &= 1 \\ x_{WW} + x_{YW} + x_{BW} + x_{RW} &= 1 \\ x_{WY} + x_{YY} + x_{BY} + x_{RY} &= 1 \\ x_{WB} + x_{YB} + x_{BB} + x_{RB} &= 1 \\ x_{WR} + x_{YR} + x_{BR} + x_{RR} &= 1 \\ x_{ij} = (0, 1) \quad & \text{for all } i \text{ and } j \end{aligned}$$

Solution is a tour (loop)

The use of M in the objective function guarantees that a paint job cannot follow itself. The same result can be realized by deleting x_{WW}, x_{YY}, x_{BB}, and x_{RR} from the entire model.

PROBLEM SET 9.3A

*1. A manager has a total of 10 employees working on six projects. There are overlaps among the assignments as the following table shows:

		Project					
		1	2	3	4	5	6
	1		x		x	x	
	2	x		x		x	
	3		x	x	x		x
	4			x	x	x	
Employee	5	x	x	x			
	6	x	x	x	x		x
	7	x	x			x	x
	8	x		x	x		
	9					x	x
	10	x	x		x	x	x

The manager meets with each employee individually once a week for a progress report. Each meeting lasts about 20 minutes for a total of 3 hours and 20 minutes for all 10 employees. To reduce the total time, the manager wants to hold group meetings depending on shared projects. The objective is to schedule the meetings in a way that will reduce the traffic (number of employees) in and out of the meeting room. Formulate the problem as a mathematical model.

2. A book salesperson who lives in Basin must call once a month on four customers located in Wald, Bon, Mena, and Kiln. The following table gives the distances in miles among the different cities.

	Miles between cities				
	Basin	*Wald*	*Bon*	*Mena*	*Kiln*
Basin	0	120	220	150	210
Wald	120	0	80	110	130
Bon	220	80	0	160	185
Mena	150	110	160	0	190
Kiln	210	130	185	190	0

The objective is to minimize the total distance traveled by the salesperson. Formulate the problem as an assignment-based ILP.

3. Circuit boards (such as those used with PCs) are fitted with holes for mounting different electronic components. The holes are drilled with a movable drill. The following table provides the distances (in centimeters) between pairs of 6 holes of a specific circuit board.

$$\|d_{ij}\| = \begin{pmatrix} - & 1.2 & .5 & 2.6 & 4.1 & 3.2 \\ 1.2 & - & 3.4 & 4.6 & 2.9 & 5.2 \\ .5 & 3.4 & - & 3.5 & 4.6 & 6.2 \\ 2.6 & 4.6 & 3.5 & - & 3.8 & .9 \\ 4.1 & 2.9 & 4.6 & 3.8 & - & 1.9 \\ 3.2 & 5.2 & 6.2 & .9 & 1.9 & - \end{pmatrix}$$

Formulate the assignment portion of an ILP representing this problem.

9.3.1 Heuristic Algorithms

This section presents two heuristics: the *nearest-neighbor* and the *subtour-reversal* algorithms. The first is easy to implement and the second requires more computations. The tradeoff is that the second algorithm generally yields better results. Ultimately, the two heuristics are combined into one heuristic, in which the output of the nearest-neighbor algorithm is used as input to the reversal algorithm.

The Nearest-Neighbor Heuristic. As the name of the heuristic suggests, a "good" solution of the TSP problem can be found by starting with any city (node) and then connecting it with the closest one. The just-added city is then linked to its nearest unlinked city (with ties broken arbitrarily). The process continues until a tour is formed.

Example 9.3-2

The matrix below summarizes the distances in miles in a 5-city TSP problem.

$$\|d_{ij}\| = \begin{pmatrix} \infty & 120 & 220 & 150 & 210 \\ 120 & \infty & 100 & 110 & 130 \\ 220 & 80 & \infty & 160 & 185 \\ 150 & \infty & 160 & \infty & 190 \\ 210 & 130 & 185 & \infty & \infty \end{pmatrix}$$

The heuristic can start from any of the five cities. Each starting city may lead to a different tour. The following table provides the steps of the heuristic starting at city 3.

Step	Action	(Partial) tour
1	Start with city 3	3
2	Link to city 2 because it is closest to city 3 ($d_{32} = \min\{220, \mathbf{80}, \infty, 160, 185\}$)	3-2
3	Link to node 4 because it is closest to node 2 ($d_{24} = \min\{120, \infty, —, \mathbf{110}, 130\}$)	3-2-4
4	Link to node 1 because it is closest to node 4 ($d_{41} = \min\{\mathbf{150}, \infty, —, —, 190\}$)	3-2-4-1
5	Link to node 5 by default and connect back to node 3 to complete the tour	3-2-4-1-5-3

Notice the progression of the steps: Comparisons exclude distances to nodes that are part of a constructed partial tour. These are indicated by (—) in the *Action* column of the table.

The resulting tour 3-2-4-1-5-3 has a total length of 80 + 110 + 150 + 210 + 185 = 735 miles. Observe that the quality of the heuristic solution is starting-node dependent. For example, starting from node 1, the constructed tour is 1-2-3-4-5-1 with a total length of 780 miles (try it!).

Subtour Reversal Heuristic. In an n-city situation, the subtour reversal heuristic starts with a feasible tour and then tries to improve on it by reversing 2-city subtours, followed by 3-city subtours, and continuing until reaching subtours of size $n - 1$.

Example 9.3-3

Consider the problem of Example 9.3-2. The reversal steps are carried out in the following table using the feasible tour 1-4-3-5-2-1 of length 745 miles:

Type	Reversal	Tour	Length
Start	—	(1-4-3-5-2-1)	**745**
Two-at-a-time reversal	4-3	1-3-4-5-2-1	820
	3-5	(1-4-5-3-2-1)	**725**
	5-2	1-4-3-2-5-1	730
Three-at-a-time reversal	4-5-3	1-3-5-4-2-1	∞
	5-3-2	1-4-2-3-5-1	∞
Four-at-a-time reversal	4-5-3-2	1-2-3-5-4-1	∞

The two-at-a-time reversals of the initial tour 1-4-3-5-2-1 are 4-3, 3-5, and 5-2, which leads to the given tours with their associated lengths of 820, 725, and 730. Since 1-4-5-3-2-1 yields a smaller length (= 725), it is used as the starting tour for making the three-at-a-time reversals. As shown in the table, these reversals produce no better results. The same result applies to the four-at-a-time reversal. Thus, 1-4-5-3-2-1 (with length 725 miles) provides the best solution of heuristic.

Notice that the three-at-a-time reversals did not produce a better tour, and, for this reason, we continued to use the best two-at-a-time tour with the four-at-a-time reversal. Notice also that the reversals do not include the starting city of the tour (= 1 in this example) because the process does not yield a tour. For example, the reversal 1-4 leads to 4-1-3-5-2-1, which is not a tour.

The solution determined by the reversal heuristic is a function of the initial feasible tour used to start the algorithm. For example, if we start with 2-3-4-1-5-2 with length 750 miles, the heuristic produces the tour 2-1-4-3-5-2 with length 745 miles (verify!), which is inferior to the solution we have in the table above. For this reason, it may be advantageous to first utilize the nearest-neighbor heuristic to determine *all* the tours that result from using each city as a starting node and then select the best as the starting tour for the reversal heuristic. This combined heuristic should, in general, lead to superior solutions than if either heuristic is applied separately. The following table shows the application of the composite heuristic to the present example.

Heuristic	Starting city	Tour	Length
Nearest neighbor	1	1-2-3-4-5-1	780
	2	2-3-4-1-5-2	750
	3	**(3-2-4-1-5-3)**	**735**
	4	4-1-2-3-5-4	∞
	5	5-2-3-4-1-5	750
Reversals	2-4	3-<u>4-2</u>-1-5-3	∞
	4-1	(3-2-<u>1-4</u>-5-3)	**725**
	1-5	3-2-4-<u>5-1</u>-3	810
	2-1-4	3-<u>4-1-2</u>-5-3	745
	1-4-5	3-2-<u>5-4-1</u>-3	∞
	2-1-4-5	3-<u>5-4-1-2</u>-3	∞

Excel Moment.

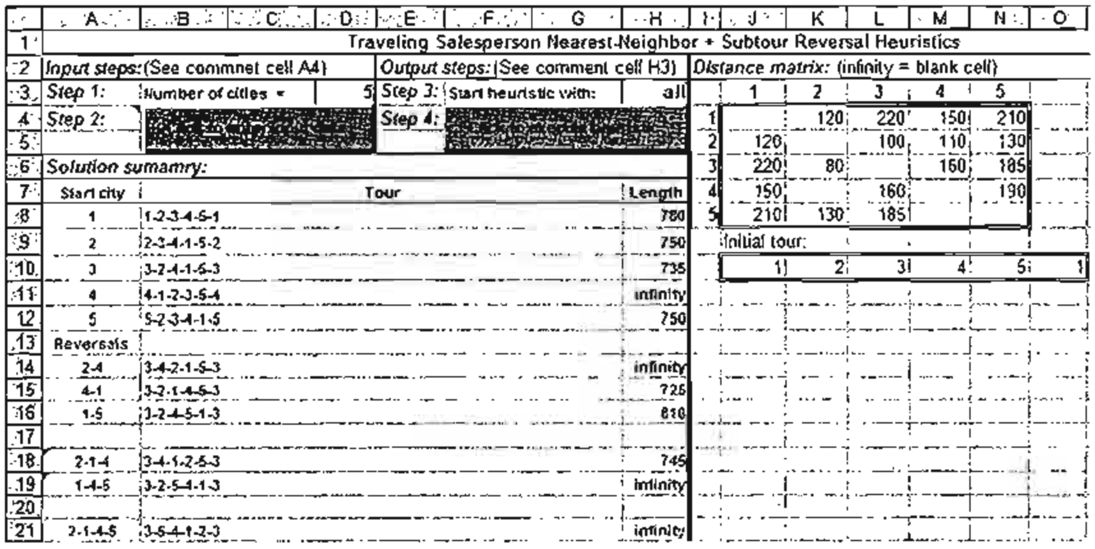
Figure 9.12 provides a general Excel template (file excelTSP.xls) for the heuristics. It uses three execution options depending on the entry in cell H3:

1. If you enter a city number, the nearest-neighbor heuristic is used to find a tour starting with the designated city.
2. If you enter the word "tour" (without the quotes), you must simultaneously provide an initial feasible tour in the designated space. In this case, only the reversal heuristic is applied to the tour you provided.
3. If you enter the word "all," the nearest-neighbor heuristic is used first, and its best tour is then used to execute the reversal heuristic.

File excelTSP.v2.xls automates the operations of Step 3.

FIGURE 9.12

Execution of the TSP heuristic using Excel spreadsheet (file excelTSP.xls)



Traveling Salesperson Nearest-Neighbor + Subtour Reversal Heuristics

Input steps: (See commnet cell A4)
Step 1: Number of cities = 5
Step 2:

Output steps: (See comment cell H3)
Step 3: Start heuristic with: all
Step 4:

Solution sumamry:

Start city	Tour	Length
1	1-2-3-4-5-1	780
2	2-3-4-1-5-2	750
3	3-2-4-1-5-3	735
4	4-1-2-3-5-4	infinity
5	5-2-3-4-1-5	750
Reversals		
2-4	3-4-2-1-5-3	infinity
4-1	3-2-1-4-5-3	725
1-5	3-2-4-5-1-3	810
2-1-4	3-4-1-2-5-3	745
1-4-5	3-2-5-4-1-3	infinity
2-1-4-5	3-5-4-1-2-3	infinity

Distance matrix: (infinity = blank cell)

	1	2	3	4	5
1		120	220	150	210
2	120		100	110	130
3	220	80		160	185
4	150		160		190
5	210	130	185		

Initial tour:

1	2	3	4	5	1

PROBLEM SET 9.3B

1. Apply the heuristic to the following problems:
   (a) The paint sequencing problem of Example 9.3-1.
   (b) Problem 1 of Set 9.3a.
   (c) Problem 2 of Set 9.3a.
   (d) Problem 3 of Set 9.3a.

9.3.2 B&B Solution Algorithm

The idea of the B&B algorithm is to start with the optimum solution of the associated assignment problem. If the solution is a tour, the process ends. Otherwise, restrictions are imposed to remove the subtours. This can be achieved by creating as many branches as the number of x_{ij}-variables associated with one of the subtours. Each branch will correspond to setting one of the variables of the subtour equal to zero (recall that all the variables associated with a subtour equal 1). The solution of the resulting assignment problem may or may not produce a tour. If it does, we use its objective value as an upper bound on the true minimum tour length. If it does not, further branching is necessary, again creating as many branches as the number of variables in one of the subtours. The process continues until all unexplored subproblems have been fathomed, either by producing a better (smaller) *upper bound* or because there is evidence that the subproblem cannot produce a better solution. The optimum tour is the one associated with the best upper bound.

The following example provides the details of the TSP B&B algorithm.

Example 9.3-4

Consider the following 5-city TSP problem:

$$\|d_{ij}\| = \begin{pmatrix} \infty & 10 & 3 & 6 & 9 \\ 5 & \infty & 5 & 4 & 2 \\ 4 & 9 & \infty & 7 & 8 \\ 7 & 1 & 3 & \infty & 4 \\ 3 & 2 & 6 & 5 & \infty \end{pmatrix}$$

We start by solving the associated assignment, which yields the following solution:

$$z = 15, (x_{13} = x_{31} = 1), (x_{25} = x_{54} = x_{42} = 1), \text{all others} = 0$$

This solution yields two subtours: (1-3-1) and (2-5-4-2), as shown at node 1 in Figure 9.13. The associated total distance is $z = 15$, which provides a lower bound on the optimal length of the 5-city tour.

A straightforward way to determine an upper bound is to select any tour and use its length as an upper bound estimate. For example, the tour 1-2-3-4-5-1 (selected totally arbitrarily) has a total length of $10 + 5 + 7 + 4 + 3 = 29$. Alternatively, a better upper bound can be found by applying the heuristic of Section 9.3.1. For the moment, we will use the upper bound of length 29 to apply the B&B algorithm. Later, we use the "improved" upper bound obtained by the heuristic to demonstrate its impact on the search tree.

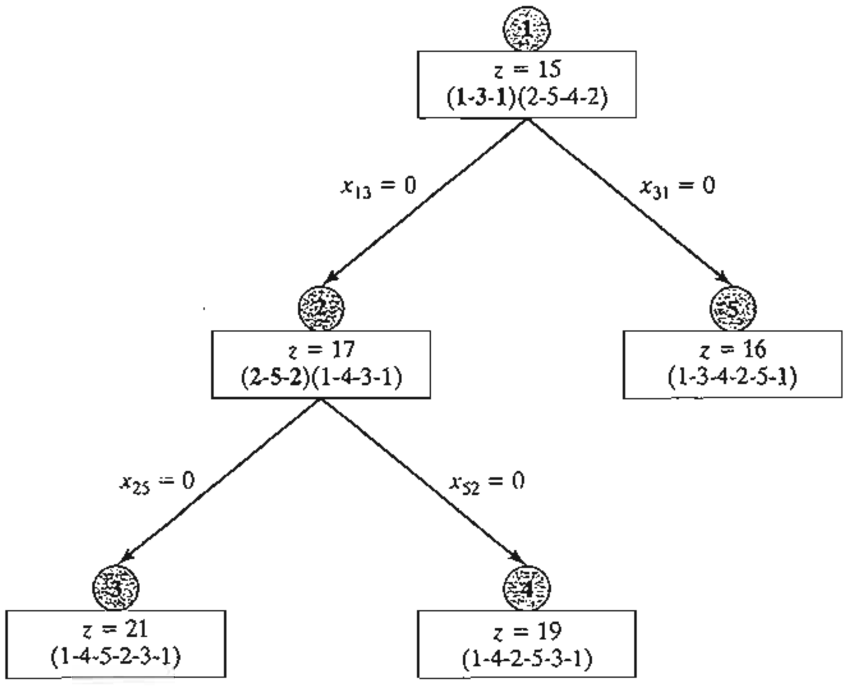


FIGURE 9.13
B&B solution of the TSP problem of Example 9.3-4

The computed lower and upper bounds indicate that the optimum tour length lies in range (15, 29). A solution that yields a tour length larger than (or equal to) 29 is discarded as nonpromising.

To eliminate the subtours at node 1, we need to "disrupt" its loop by forcing its member variables, x_{ij}, to be zero. Subtour 1-3-1 is disrupted if we impose the restriction $x_{13} = 0$ or $x_{31} = 0$ (i.e., one at a time) on the assignment problem at node 1. Similarly, subtour 2-5-4-2 is eliminated by imposing one of the restrictions $x_{25} = 0$, $x_{54} = 0$, or $x_{42} = 0$. In terms of the B&B tree, each of these restrictions gives rise to a branch and hence a new subproblem. It is important to notice that branching *both* subtours at node 1 is *not* necessary. Instead, only *one* subtour needs to be disrupted at any one node. The idea is that a breakup of one subtour automatically alters the member variables of the other subtour and hence produces conditions that are favorable to creating a tour. Under this argument, it is more efficient to select the subtour with the smallest number of cities because it creates the smallest number of branches.

Targeting subtour (1-3-1), two branches $x_{13} = 0$ and $x_{31} = 0$ are created at node 1. The associated assignment problems are constructed by removing the row and column associated with the zero variable, which makes the assignment problem smaller. Another way to achieve the same result is to leave the size of the assignment problem unchanged and simply assign an infinite distance to the branching variable. For example, the assignment problem associated with $x_{13} = 0$ requires substituting $d_{13} = \infty$ in the assignment model at node 1. Similarly, for $x_{31} = 0$, we substitute $d_{31} = \infty$.

In Figure 9.13, we arbitrarily start by solving the subproblem associated with $x_{13} = 0$ by setting $d_{13} = \infty$. Node 2 gives the solution $z = 17$ but continues to produce the subtours (2-5-2) and (1-4-3-1). Repeating the procedure we applied at node 1 gives rise to two branches: $x_{25} = 0$ and $x_{52} = 0$.

We now have three unexplored subproblems, one from node 1 and two from node 2, and we are free to investigate any of them at this point. Arbitrarily exploring the subproblem associated with $x_{25} = 0$ from node 2, we set $d_{13} = \infty$ and $d_{25} = \infty$ in the *original* assignment problem, which yields the solution $z = 21$ and the tour solution 1-4-5-2-3-1 at node 3. The tour solution at node 3 lowers the upper bound from $z = 29$ to $z = 21$. This means that any unexplored subproblem that can be shown to yield a tour length larger than 21 is discarded as nonpromising.

We now have two unexplored subproblems. Selecting the subproblem 4 for exploration, we set $d_{13} = \infty$ and $d_{52} = \infty$ in the *original* assignment, which yields the tour solution 1-4-2-5-3-1 with $z = 19$. The new solution provides a better tour than the one associated with the current upper bound of 21. Thus, the new upper bound is updated to $z = 19$ and its associated tour, 1-4-2-5-3-1, is the best available so far.

Only subproblem 5 remains unexplored. Substituting $d_{31} = \infty$ in the *original* assignment problem at node 1, we get the tour solution 1-3-4-2-5-1 with $z = 16$, at node 5. Once again, this is a better solution than the one associated with node 4 and thus requires updating the upper bound to $z = 16$.

There are no remaining unfathomed nodes, which completes the search tree. The optimal tour is the one associated with the current upper bound: 1-3-4-2-5-1 with length 16 miles.

Remarks. The solution of the example reveals two points:

1. Although the search sequence $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ was selected deliberately to demonstrate the mechanics of the B&B algorithm and the updating of its upper bound, we generally have no way of predicting which sequence should be adopted to improve the efficiency of the search. Some rules of thumb can be of help. For example, at a given node we can start with the branch associated with the *largest* d_{ij} among all the created branches. By canceling the tour leg with the largest d_{ij}, the hope is that a "good" tour with a smaller total length will be found. In the present example, this rule calls for exploring branch $x_{31} = 0$ to node 5 before branch x_{13} to node 2 because $(d_{31} = 4) > (d_{13} = 3)$, and this would have produced the upper bound $z = 16$, which automatically fathoms node 2 and, hence, eliminates the need to create nodes 3 and 4. Another rule calls for sequencing the exploration of the nodes in a horizontal tier (rather than vertically). The idea is that nodes closer to the starting node are *more likely* to produce a tighter upper bound because the number of additional constraints (of the type $x_{ij} = 0$) is smaller. This rule would have also discovered the solution at node 5 sooner.

2. The B&B should be applied in conjunction with the heuristic in Section 9.3.1. The heuristic provides a "good" upper bound which can be used to fathom nodes in the search tree. In the present example, the heuristic yields the tour 1-3-4-2-5-1 with a length of 16 distance units.

AMPL Moment

Interactive AMPL commands are ideal for the implementation of the TSP B&B algorithm using the general assignment model (file amplAssignment.txt). The following table summarizes the AMPL commands needed to create the B&B tree in Figure 9.13 (Example 9.3-4):

AMPL command	Result
ampl: `model amplAssignment.txt;display x;`	Node 1 solution
ampl: `fix x[1,3]:=0;solve;display x;`	Node 2 solution
ampl: `fix x[2,5]:=0;solve;display x;`	Node 3 solution
ampl: `unfix x[2,5];fix x[5,2]:=0;solve;display x;`	Node 4 solution
ampl: `unfix x[5,2];unfix x[1,3];fix x[3,1]:=0;` `solve;display x;`	Node 5 solution

PROBLEM SET 9.3C

1. Solve Example 9.3-3 using subtour 2-5-4-2 to start the branching process at node 1, using the following sequences for exploring the nodes.
   (a) Explore all the subproblems horizontally from left to right in each tier before preceeding to the next tier.
   (b) Follow each path vertically from node 1 until it ends with a fathomed node.

*2. Solve Problem 1, Set 9.3a using B&B.

3. Solve Problem 2, Set 9.3a using B&B.

4. Solve Problem 3, Set 9.3a using B&B.

9.3.3 Cutting-Plane Algorithm

The idea of the cutting plane algorithm is to add a set of constraints to the assignment problem that prevent the formation of a subtour. The additional constraints are defined as follows. In an n-city situation, associate a continuous variable $u_j\ (\geq 0)$ with cities $2, 3, \ldots,$ and n. Next, define the required set of additional constraints as

$$u_i - u_j + nx_{ij} \leq n - 1, i = 2, 3, \ldots, n; j = 2, 3, \ldots, n; i \neq j$$

These constraints, when added to the assignment model, will automatically remove all subtour solutions.

Example 9.3-5

Consider the following distance matrix of a 4-city TSP problem.

$$\|d_{ij}\| = \begin{pmatrix} - & 13 & 21 & 26 \\ 10 & - & 29 & 20 \\ 30 & 20 & - & 5 \\ 12 & 30 & 7 & - \end{pmatrix}$$

The associated LP consists of the assignment model constraints plus the additional constraints in the table below. All $x_{ij} = (0, 1)$ and all $u_j \geq 0$.

No.	x_{11}	x_{12}	x_{13}	x_{14}	x_{21}	x_{22}	x_{23}	x_{24}	x_{31}	x_{32}	x_{33}	x_{34}	x_{41}	x_{42}	x_{43}	x_{44}	u_2	u_3	u_4	
1							4										1	−1		≤3
2								4									1		−1	≤3
3										4							−1	1		≤3
4												4						1	−1	≤3
5														4			−1		1	≤3
6															4			−1	1	≤3

The optimum solution is

$$u_2 = 0, u_3 = 2, u_4 = 3, x_{12} = x_{23} = x_{34} = x_{41} = 1, \text{tour length} = 59.$$

This corresponds to the tour solution 1-2-3-4-1. The solution satisfies all the additional constraints in u_j (verify!).

To demonstrate that subtour solutions do not satisfy the additional constraints, consider (1-2-1, 3-4-3), which corresponds to $x_{12} = x_{21} = 1, x_{34} = x_{43} = 1$. Now, consider constraint 6 in the tableau above:

$$4x_{43} + u_4 - u_3 \leq 3$$

Substituting $x_{43} = 1, u_3 = 2, u_4 = 3$ yields $5 \leq 3$, which is impossible, thus disallowing $x_{43} = 1$ and subtour 3-4-3.

The disadvantage of the cutting-plane model is that the number of variables grows exponentially with the number of cities, making it difficult to obtain a numeric solution for practical situations. For this reason, the B&B algorithm (coupled with the heuristic) may be a more feasible alternative for solving the problem.

AMPL Moment

Figure 9.14 provides the AMPL model of the cutting-plane algorithm (file amplEx9.3-5.txt). The data of the 4-city TSP of Example 9.3-5 are used to drive the model. The formulation is straightforward: The first two sets of constraints define the assignment model associated with the problem, and the third set represents the cuts needed to remove subtour solutions. Notice that the assignment-model variables must be binary and that `option solver cplex;` must precede `solve;` to ensure that the obtained solution is integer.

The `for` and `if-then` statements at the bottom of the model are used to present the output in the following readable format:

```
Optimal tour length = 59.00
Optimal tour:  1- 2- 3- 4- 1
```

PROBLEM SET 9.3D

1. An automatic guided vehicle (AGV) is used to deliver mail to 5 departments located on a factory floor. The trip starts at the mail sorting room and makes the delivery round to the different departments before returning to the mailroom. Using the mailroom as the origin (0, 0), the (x, y) locations of the delivery spots are (10, 30), (10, 50), (30, 10), (40, 40), and (50, 60) for departments 1 through 5, respectively. All distances are in meters. The AGV can move along horizontal and vertical aisles only. The objective is to minimize the length of the round trip.

   Formulate the problem as a TSP, including the cuts.

2. Write down the cuts associated with the following TSP:

$$\|d_{ij}\| = \begin{pmatrix} \infty & 43 & 21 & 20 & 10 \\ 12 & \infty & 9 & 22 & 30 \\ 20 & 10 & \infty & 5 & 13 \\ 14 & 30 & 42 & \infty & 20 \\ 44 & 7 & 9 & 10 & \infty \end{pmatrix}$$

REFI

```
param k;
param n;
param c{1..n,1..n} default 10000;
var x{i in 1..n,j in 1..n} binary;
var u{i in 1..n:i>1}>=0;

minimize tourLength:sum{i in 1..n,j in 1..n}c[i,j]*x[i,j];
subject to
 fromCity {i in 1..n}:sum {j in 1..n} x[i,j] = 1;
 toCity {j in 1..n}:sum {i in 1..n} x[i,j] = 1;
 cut{i in 1..n,j in 1..n:i>1 and j>1 and i<>j}:
                      u[i]-u[j]+n*x[i,j] <= n-1;
data;
param n:=4;
param c:
       1    2    3    4:=
  1    .    13   21   26
  2    10   .    29   20
  3    30   20   .    5
  4    12   30   7    .;

option solver cplex; solve;
display u;
#---------------------------------print formatted output
printf "\n\nOptimal tour length = %7.2f\n",tourLength;
printf "Optimal tour:";
let k:=1;                 #tour starts at city k=1
for {i in 1..n}
    {
    printf "%3i", k;
    for {j in 1..n}  #search for next city following k
        {
        if x[k,j]=1 then
           {
           let k:=j;  #next city found, set k=j
           break;
           }
        }
    printf "-";          #insert last hyphen
    }
printf "  1\n\n";
```

FIGURE 9.14

AMPL cutting-plane model of the TSP problem (file amplEx9.3-5.txt)

3. *AMPL experiment.* Use AMPL to solve the following TSP problem by the cutting plane algorithm.
   - **(a)** Problem 2, Set 9.3a.
   - **(b)** Problem 3, Set 9.3a.

REFERENCES

Barnett, A., "Misapplication Review: High Road to Glory," *Interfaces*, Vol. 17, No. 5, pp. 51–54, 1987.

Graves, R., L. Schrage, and J. Sankaran, "An Auction Method for Course Registration," *Interfaces*, Vol. 23, No. 5, pp. 81–97, 1993.

Guéret, C., C. Prins, and M. Sevaux, *Applications of Optimization with Xpress-MP*, Dash Optimization, London, 2002.

Jarvis, J., R. Rardin, V. Unger, R. Moore, and C. Schimpeler, "Optimal Design of Regional Wastewater System: A Fixed Charge Network Flow Model," *Operations Research*, Vol. 26, No. 4, pp. 538–550, 1978.

Lee, J., *A First Course in Combinatorial Optimization*, Cambridge University Press, 2004.

Liberatore, M., and T. Miller, "A Hierarchial Production Planning System," *Interfaces*, Vol. 15, No. 4, pp. 1–11, 1985.

Nemhauser, G., and L. Wolsey, *Integer and Combinatorial Optimization*, Wiley, New York, 1988.

Salkin, H., and K. Mathur, *Foundations of Integer Programming*, North-Holland, New York, 1989.

Schrijver, A. *Theory of Linear and Integer Programming*, Wiley, New York, 1998.

Taha, H., *Integer Programming: Theory, Applications, and Computations*, Academic Press, Orlando, FL, 1975.

Weber, G., "Puzzle Contests in MS/OR Education," *Interfaces*, Vol. 20, No. 2, pp. 72–76, 1990.

Wolsey, L., *Integer Programming*, Wiley, New York, 1998.