TABLE 5.30 Data for Problem 7

\$1	\$1	\$2	5
\$6	\$5	\$1	6
2	7	1	

subject to

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

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

$$x_{ij} \geq 0, \text{ all } i \text{ and } j$$

It may appear logical to assume that the optimum solution will require the first (second) set of inequalities to be replaced with equations if $\sum a_i \geq \sum b_j$ ($\sum a_i \leq \sum b_j$). The counterexample in Table 5.30 shows that this assumption is not correct.

Show that the application of the suggested procedure yields the solution $x_{11} = 2$, $x_{12} = 3$, $x_{22} = 4$, and $x_{23} = 2$, with $z = \$27$, which is worse than the feasible solution $x_{11} = 2$, $x_{12} = 7$, and $x_{23} = 6$, with $z = \$15$.

5.3.3 Simplex Method Explanation of the Method of Multipliers

The relationship between the method of multipliers and the simplex method can be explained based on the primal-dual relationships (Section 4.2). From the special structure of the LP representing the transportation model (see Example 5.1-1 for an illustration), the associated dual problem can be written as

$$\text{Maximize } z = \sum_{i=1}^{m} a_i u_i + \sum_{j=1}^{n} b_j v_j$$

subject to

$$u_i + v_j \leq c_{ij}, \text{ all } i \text{ and } j$$

$$u_i \text{ and } v_j \text{ unrestricted}$$

where

a_i = Supply amount at source i

b_j = Demand amount at destination j

c_{ij} = Unit transportation cost from source i to destination j

u_i = Dual variable of the constraint associated with source i

v_j = Dual variable of the constraint associated with destination j

From Formula 2, Section 4.2.4, the objective-function coefficients (reduced costs) of the variable x_{ij} equal the difference between the left- and right-hand sides of the corresponding dual constraint—that is, $u_i + v_j - c_{ij}$. However, we know that this quantity must equal zero for each *basic variable*, which then produces the following result:

$$u_i + v_j = c_{ij}, \text{ for each } basic \text{ variable } x_{ij}.$$

There are $m + n - 1$ such equations whose solution (after assuming an arbitrary value $u_1 = 0$) yields the multipliers u_i and v_j. Once these multipliers are computed, the entering variable is determined from all the *nonbasic* variables as the one having the largest positive $u_i + v_j - c_{ij}$.

The assignment of an arbitrary value to one of the dual variables (i.e., $u_1 = 0$) may appear inconsistent with the way the dual variables are computed using Method 2 in Section 4.2.3. Namely, for a given basic solution (and, hence, inverse), the dual values must be unique. Problem 2, Set 5.3c, addresses this point.

PROBLEM SET 5.3C

1. Write the dual problem for the LP of the transportation problem in Example 5.3-5 (Table 5.21). Compute the associated optimum *dual* objective value using the optimal dual values given in Table 5.25, and show that it equals the optimal cost given in the example.
2. In the transportation model, one of the dual variables assumes an arbitrary value. This means that for the same basic solution, the values of the associated dual variables are not unique. The result appears to contradict the theory of linear programming, where the dual values are determined as the product of the vector of the objective coefficients for the basic variables and the associated inverse basic matrix (see Method 2, Section 4.2.3). Show that for the transportation model, although the inverse basis is unique, the vector of *basic* objective coefficients need not be so. Specifically, show that if c_{ij} is changed to $c_{ij} + k$ for all i and j, where k is a constant, then the optimal values of x_{ij} will remain the same. Hence, the use of an arbitrary value for a dual variable is implicitly equivalent to assuming that a specific constant k is added to all c_{ij}.

5.4 THE ASSIGNMENT MODEL

"The best person for the job" is an apt description of the assignment model. The situation can be illustrated by the assignment of workers with varying degrees of skill to jobs. A job that happens to match a worker's skill costs less than one in which the operator is not as skillful. The objective of the model is to determine the minimum-cost assignment of workers to jobs.

The general assignment model with n workers and n jobs is represented in Table 5.31.

The element c_{ij} represents the cost of assigning worker i to job j $(i, j = 1, 2, \ldots, n)$. There is no loss of generality in assuming that the number of workers always

TABLE 5.31 Assignment Model

		Jobs			
	1	2	...	n	
Worker 1	c_{11}	c_{12}	...	c_{1n}	1
2	c_{21}	c_{22}	...	c_{2n}	1
⋮	⋮	⋮	⋮	⋮	⋮
n	c_{n1}	c_{n2}	...	c_{nn}	1
	1	1	...	1	

equals the number of jobs, because we can always add fictitious workers or fictitious jobs to satisfy this assumption.

The assignment model is actually a special case of the transportation model in which the workers represent the sources, and the jobs represent the destinations. The supply (demand) amount at each source (destination) exactly equals 1. The cost of "transporting" worker i to job j is c_{ij}. In effect, the assignment model can be solved directly as a regular transportation model. Nevertheless, the fact that all the supply and demand amounts equal 1 has led to the development of a simple solution algorithm called the **Hungarian method.** Although the new solution method appears totally unrelated to the transportation model, the algorithm is actually rooted in the simplex method, just as the transportation model is.

5.4.1 The Hungarian Method[8]

We will use two examples to present the mechanics of the new algorithm. The next section provides a simplex-based explanation of the procedure.

Example 5.4-1

Joe Klyne's three children, John, Karen, and Terri, want to earn some money to take care of personal expenses during a school trip to the local zoo. Mr. Klyne has chosen three chores for his children: mowing the lawn, painting the garage door, and washing the family cars. To avoid anticipated sibling competition, he asks them to submit (secret) bids for what they feel is fair pay for each of the three chores. The understanding is that all three children will abide by their father's decision as to who gets which chore. Table 5.32 summarizes the bids received. Based on this information, how should Mr. Klyne assign the chores?

The assignment problem will be solved by the Hungarian method.

Step 1. For the original cost matrix, identify each row's minimum, and subtract it from all the entries of the row.

[8]As with the transportation model, the classical Hungarian method, designed primarily for *hand* computations, is something of the past and is presented here purely for historical reasons. Today, the need for such computational shortcuts is not warranted as the problem can be solved as a regular LP using highly efficient computer codes.

TABLE 5.32 Klyne's Assignment Problem

	Mow	Paint	Wash
John	$15	$10	$9
Karen	$9	$15	$10
Terri	$10	$12	$8

Step 2. For the matrix resulting from step 1, identify each column's minimum, and subtract it from all the entries of the column.

Step 3. Identify the optimal solution as the feasible assignment associated with the zero elements of the matrix obtained in step 2.

Let p_i and q_j be the minimum costs associated with row i and column j as defined in steps 1 and 2, respectively. The row minimums of step 1 are computed from the original cost matrix as shown in Table 5.33.

Next, subtract the row minimum from each respective row to obtain the reduced matrix in Table 5.34.

The application of step 2 yields the column minimums in Table 5.34. Subtracting these values from the respective columns, we get the reduced matrix in Table 5.35.

TABLE 5.33 Step 1 of the Hungarian Method

	Mow	Paint	Wash	Row minimum
John	15	10	9	$p_1 = 9$
Karen	9	15	10	$p_2 = 9$
Terri	10	12	8	$p_3 = 8$

TABLE 5.34 Step 2 of the Hungarian Method

	Mow	Paint	Wash
John	6	1	0
Karen	0	6	1
Terri	2	4	0
Column minimum	$q_1 = 0$	$q_2 = 1$	$q_3 = 0$

TABLE 5.35 Step 3 of the Hungarian Method

	Mow	Paint	Wash
John	6	<u>0</u>	0
Karen	<u>0</u>	5	1
Terri	2	3	<u>0</u>

The cells with underscored zero entries provide the optimum solution. This means that John gets to paint the garage door, Karen gets to mow the lawn, and Terri gets to wash the family cars. The total cost to Mr. Klyne is $9 + 10 + 8 = \$27$. This amount also will always equal $(p_1 + p_2 + p_3) + (q_1 + q_2 + q_3) = (9 + 9 + 8)+(0 + 1 + 0) = \$27$. (A justification of this result is given in the next section.)

The given steps of the Hungarian method work well in the preceding example because the zero entries in the final matrix happen to produce a *feasible* assignment (in the sense that each child is assigned a distinct chore). In some cases, the zeros created by steps 1 and 2 may not yield a feasible solution directly, and further steps are needed to find the optimal (feasible) assignment. The following example demonstrates this situation.

Example 5.4-2

Suppose that the situation discussed in Example 5.4-1 is extended to four children and four chores. Table 5.36 summarizes the cost elements of the problem.

The application of steps 1 and 2 to the matrix in Table 5.36 (using $p_1 = 1, p_2 = 7, p_3 = 4, p_4 = 5, q_1 = 0, q_2 = 0, q_3 = 3$, and $q_4 = 0$) yields the reduced matrix in Table 5.37 (verify!).

The locations of the zero entries do not allow assigning unique chores to all the children. For example, if we assign child 1 to chore 1, then column 1 will be eliminated, and child 3 will not have a zero entry in the remaining three columns. This obstacle can be accounted for by adding the following step to the procedure outlined in Example 5.4-1:

Step 2a. If no feasible assignment (with all zero entries) can be secured from steps 1 and 2,

(i) Draw the *minimum* number of horizontal and vertical lines in the last reduced matrix that will cover *all* the zero entries.

TABLE 5.36 Assignment Model

		Chore 1	2	3	4
Child	1	\$1	\$4	\$6	\$3
	2	\$9	\$7	\$10	\$9
	3	\$4	\$5	\$11	\$7
	4	\$8	\$7	\$8	\$5

TABLE 5.37 Reduced Assignment Matrix

		Chore 1	2	3	4
Child	1	**0**	3	2	2
	2	2	**0**	**0**	2
	3	**0**	1	4	3
	4	3	2	**0**	**0**

TABLE 5.38 Application of Step 2a

		Chore 1	Chore 2	Chore 3	Chore 4
Child	1	0	3	2	2
	2	**2**	0	0	2
	3	0	*1*	4	3
	4	**3**	2	0	0

TABLE 5.39 Optimal Assignment

		Chore 1	Chore 2	Chore 3	Chore 4
Child	1	<u>0</u>	2	1	1
	2	3	0	<u>0</u>	2
	3	0	<u>0</u>	3	2
	4	4	2	0	<u>0</u>

(ii) Select the *smallest uncovered* entry, subtract it from every uncovered entry, then add it to every entry at the intersection of two lines.

(iii) If no feasible assignment can be found among the resulting zero entries, repeat step 2a. Otherwise, go to step 3 to determine the optimal assignment.

The application of step 2a to the last matrix produces the shaded cells in Table 5.38. The smallest unshaded entry (shown in italics) equals 1. This entry is added to the bold intersection cells and subtracted from the remaining shaded cells to produce the matrix in Table 5.39.

The optimum solution (shown by the underscored zeros) calls for assigning child 1 to chore 1, child 2 to chore 3, child 3 to chore 2, and child 4 to chore 4. The associated optimal cost is $1 + 10 + 5 + 5 = \$21$. The same cost is also determined by summing the p_i's, the q_j's, and the entry that was subtracted after the shaded cells were determined—that is, $(1 + 7 + 4 + 5) + (0 + 0 + 3 + 0) + (1) = \$21$.

AMPL Moment.

File amplEx5.4-2.txt provides the AMPL model for the assignment model. The model is very similar to that of the transportation model.

PROBLEM SET 5.4A

1. Solve the assignment models in Table 5.40.
   - **(a)** Solve by the Hungarian method.
   - **(b)** *TORA Experiment.* Express the problem as an LP and solve it with TORA.
   - **(c)** *TORA Experiment.* Use TORA to solve the problem as a transportation model.

TABLE 5.40 Data for Problem 1

(i)

$3	$8	$2	$10	$3
$8	$7	$2	$9	$7
$6	$4	$2	$7	$5
$8	$4	$2	$3	$5
$9	$10	$6	$9	$10

(ii)

$3	$9	$2	$3	$7
$6	$1	$5	$6	$6
$9	$4	$7	$10	$3
$2	$5	$4	$2	$1
$9	$6	$2	$4	$5

(d) *Solver Experiment.* Modify Excel file solverEx5.3-1.xls to solve the problem.

(e) *AMPL Experiment.* Modify amplEx5.3-1b.txt to solve the problem.

2. JoShop needs to assign 4 jobs to 4 workers. The cost of performing a job is a function of the skills of the workers. Table 5.41 summarizes the cost of the assignments. Worker 1 cannot do job 3 and worker 3 cannot do job 4. Determine the optimal assignment using the Hungarian method.

3. In the JoShop model of Problem 2, suppose that an additional (fifth) worker becomes available for performing the four jobs at the respective costs of $60, $45, $30, and $80. Is it economical to replace one of the current four workers with the new one?

4. In the model of Problem 2, suppose that JoShop has just received a fifth job and that the respective costs of performing it by the four current workers are $20, $10, $20, and $80. Should the new job take priority over any of the four jobs JoShop already has?

***5.** A business executive must make the four round trips listed in Table 5.42 between the head office in Dallas and a branch office in Atlanta.

The price of a round-trip ticket from Dallas is $400. A discount of 25% is granted if the dates of arrival and departure of a ticket span a weekend (Saturday and Sunday). If the stay in Atlanta lasts more than 21 days, the discount is increased to 30%. A one-way

TABLE 5.41 Data for Problem 2

		Job 1	Job 2	Job 3	Job 4
Worker	1	$50	$50	—	$20
	2	$70	$40	$20	$30
	3	$90	$30	$50	—
	4	$70	$20	$60	$70

TABLE 5.42 Data for Problem 5

Departure date from Dallas	Return date to Dallas
Monday, June 3	Friday, June 7
Monday, June 10	Wednesday, June 12
Monday, June 17	Friday, June 21
Tuesday, June 25	Friday, June 28

ticket between Dallas and Atlanta (either direction) costs $250. How should the executive purchase the tickets?

***6.** Figure 5.6 gives a schematic layout of a machine shop with its existing work centers designated by squares 1, 2, 3, and 4. Four new work centers, I, II, III, and IV, are to be added to the shop at the locations designated by circles *a*, *b*, *c*, and *d*. The objective is to assign the new centers to the proposed locations to minimize the total materials handling traffic between the existing centers and the proposed ones. Table 5.43 summarizes the frequency of trips between the new centers and the old ones. Materials handling equipment travels along the rectangular aisles intersecting at the locations of the centers. For example, the one-way travel distance (in meters) between center 1 and location *b* is 30 + 20 = 50 m.

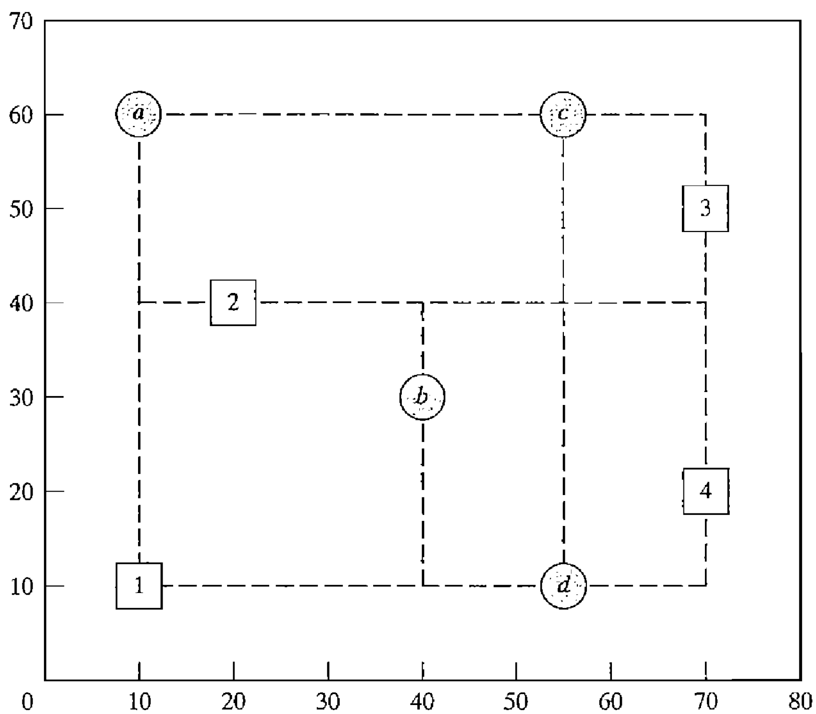


FIGURE 5.6

Machine shop layout for Problem 6, Set 5.4a

TABLE 5.43 Data for Problem 6

		New center		
	I	II	III	IV
1	10	2	4	3
Existing 2	7	1	9	5
center 3	0	8	6	2
4	11	4	0	7

7. In the Industrial Engineering Department at the University of Arkansas, INEG 4904 is a capstone design course intended to allow teams of students to apply the knowledge and skills learned in the undergraduate curriculum to a practical problem. The members of each team select a project manager, identify an appropriate scope for their project, write and present a proposal, perform necessary tasks for meeting the project objectives, and write and present a final report. The course instructor identifies potential projects and provides appropriate information sheets for each, including contact at the sponsoring organization, project summary, and potential skills needed to complete the project. Each design team is required to submit a report justifying the selection of team members and the team manager. The report also provides a ranking for each project in order of preference, including justification regarding proper matching of the team's skills with the project objectives. In a specific semester, the following projects were identified: Boeing F-15, Boeing F-18, Boeing Simulation, Cargil, Cobb-Vantress, ConAgra, Cooper, DaySpring (layout), DaySpring (material handling), J.B. Hunt, Raytheon, Tyson South, Tyson East, Wal-Mart, and Yellow Transportation. The projects for Boeing and Raytheon require U.S. citizenship of all team members. Of the eleven design teams available for this semester, four do not meet this requirement.

   Devise a procedure for assigning projects to teams and justify the arguments you use to reach a decision.

5.4.2 Simplex Explanation of the Hungarian Method

The assignment problem in which n workers are assigned to n jobs can be represented as an LP model in the following manner: Let c_{ij} be the cost of assigning worker i to job j, and define

$$x_{ij} = \begin{cases} 1, & \text{if worker } i \text{ is assigned to job } j \\ 0, & \text{otherwise} \end{cases}$$

Then the LP model is given as

$$\text{Minimize } z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

5.5

subject to

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

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

$$x_{ij} = 0 \text{ or } 1$$

The optimal solution of the preceding LP model remains unchanged if a constant is added to or subtracted from any row or column of the cost matrix (c_{ij}). To prove this point, let p_i and q_j be constants subtracted from row i and column j. Thus, the cost element c_{ij} is changed to

$$c'_{ij} = c_{ij} - p_i - q_j$$

Now

$$\sum_i \sum_j c'_{ij}x_{ij} = \sum_i \sum_j (c_{ij} - p_i - q_j)x_{ij} = \sum_i \sum_j c_{ij}x_{ij} - \sum_i p_i\Big(\sum_j x_{ij}\Big) - \sum_j q_j\Big(\sum_i x_{ij}\Big)$$

$$= \sum_i \sum_j c_{ij}x_{ij} - \sum_i p_i(1) - \sum_j q_j(1)$$

$$= \sum_i \sum_j c_{ij}x_{ij} - \text{constant}$$

Because the new objective function differs from the original one by a constant, the optimum values of x_{ij} must be the same in both cases. The development thus shows that steps 1 and 2 of the Hungarian method, which call for subtracting p_i from row i and then subtracting q_j from column j, produce an equivalent assignment model. In this regard, if a feasible solution can be found among the zero entries of the cost matrix created by steps 1 and 2, then it must be optimum because the cost in the modified matrix cannot be less than zero.

If the created zero entries cannot yield a feasible solution (as Example 5.4-2 demonstrates), then step 2a (dealing with the covering of the zero entries) must be applied. The validity of this procedure is again rooted in the simplex method of linear programming and can be explained by duality theory (Chapter 4) and the complementary slackness theorem (Chapter 7). We will not present the details of the proof here because they are somewhat involved.

The reason $(p_1 + p_2 + \cdots + p_n) + (q_1 + q_2 + \cdots + q_n)$ gives the optimal objective value is that it represents the dual objective function of the assignment model. This result can be seen through comparison with the dual objective function of the transportation model given in Section 5.3.4. [See Bazaraa and Associates (1990, pp. 499–508) for the details.]

5.5 THE TRANSSHIPMENT MODEL

The transshipment model recognizes that it may be cheaper to ship through intermediate or *transient* nodes before reaching the final destination. This concept is more general than that of the regular transportation model, where direct shipments only are allowed between a source and a destination.

This section shows how a transshipment model can be converted to (and solved as) a regular transportation model using the idea of a **buffer**.

Example 5.5-1

Two automobile plants, *P*1 and *P*2, are linked to three dealers, *D*1, *D*2, and *D*3, by way of two transit centers, *T*1 and *T*2, according to the network shown in Figure 5.7. The supply amounts at plants *P*1 and *P*2 are 1000 and 1200 cars, and the demand amounts at dealers *D*1, *D*2, and *D*3, are 800, 900, and 500 cars. The shipping costs per car (in hundreds of dollars) between pairs of nodes are shown on the connecting links (or arcs) of the network.

Transshipment occurs in the network in Figure 5.7 because the entire supply amount of 2200 (= 1000 + 1200) cars at nodes *P*1 and *P*2 could conceivably pass through any node of the

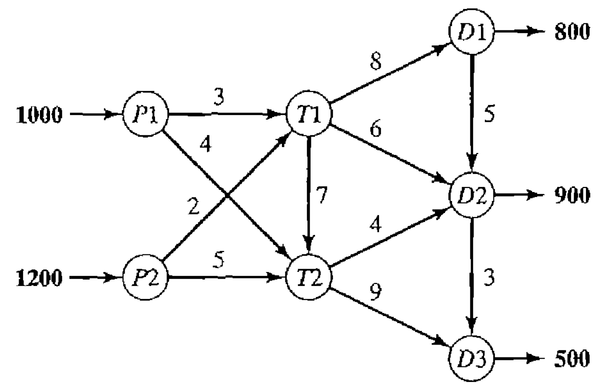


FIGURE 5.7

Transshipment network between plants and dealers

network before ultimately reaching their destinations at nodes $D1$, $D2$, and $D3$. In this regard, each node of the network with both input and output arcs ($T1$, $T2$, $D1$, and $D2$) acts as both a source and a destination and is referred to as a **transshipment node**. The remaining nodes are either **pure supply nodes** ($P1$ and $P2$) or **pure demand nodes** ($D3$).

The transshipment model can be converted into a regular transportation model with six sources ($P1, P2, T1, T2, D1,$ and $D2$) and five destinations ($T1, T2, D1, D2,$ and $D3$). The amounts of supply and demand at the different nodes are computed as

$$\text{Supply at a } \textit{pure supply node} = \text{Original supply}$$
$$\text{Demand at a } \textit{pure demand node} = \text{Original demand}$$
$$\text{Supply at a } \textit{transshipment node} = \text{Original supply} + \text{Buffer amount}$$
$$\text{Demand at a } \textit{transshipment node} = \text{Original demand} + \text{Buffer amount}$$

The buffer amount should be sufficiently large to allow all of the *original* supply (or demand) units to pass through any of the *transshipment* nodes. Let B be the desired buffer amount; then

$$\begin{aligned} B &= \text{Total supply (or demand)} \\ &= 1000 + 1200 \text{ (or } 800 + 900 + 500) \\ &= 2200 \text{ cars} \end{aligned}$$

Using the buffer B and the unit shipping costs given in the network, we construct the equivalent regular transportation model as in Table 5.44.

The solution of the resulting transportation model (determined by TORA) is shown in Figure 5.8. Note the effect of transshipment: Dealer $D2$ receives 1400 cars, keeps 900 cars to satisfy its demand, and sends the remaining 500 cars to dealer $D3$.

PROBLEM SET 5.5A[9]

1. The network in Figure 5.9 gives the shipping routes from nodes 1 and 2 to nodes 5 and 6 by way of nodes 3 and 4. The unit shipping costs are shown on the respective arcs.
   (a) Develop the corresponding transshipment model.
   (b) Solve the problem, and show how the shipments are routed from the sources to the destinations.

[9]You are encouraged to use TORA, Excel Solver, or AMPL to solve the problems in this set.

TABLE 5.44 Transshipment Model

	T1	*T2*	*D1*	*D2*	*D3*	
P1	3	4	*M*	*M*	*M*	**1000**
P2	2	5	*M*	*M*	*M*	**1200**
T1	0	7	8	6	*M*	***B***
T2	*M*	0	*M*	4	9	***B***
D1	*M*	*M*	0	5	*M*	***B***
D2	*M*	*M*	*M*	0	3	***B***
	B	***B***	**800 + *B***	**900 + *B***	**500**	

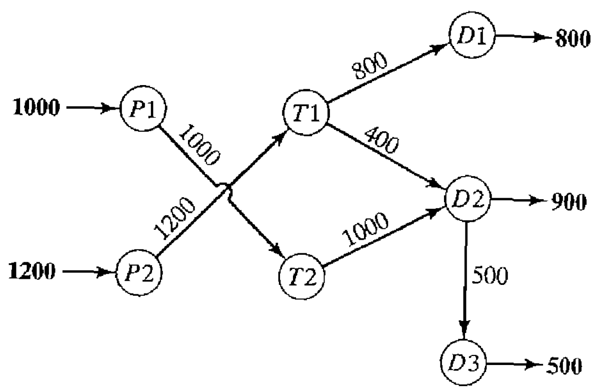


FIGURE 5.8
Solution of the transshipment model

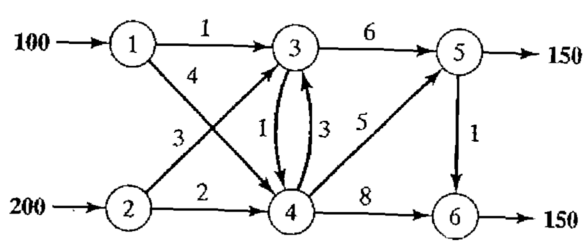


FIGURE 5.9
Network for Problem 1, Set 5.5a

2. In Problem 1, suppose that source node 1 can be linked to source node 2 with a unit shipping cost of $1. The unit shipping cost from node 1 to node 3 is increased to $5. Formulate the problem as a transshipment model, and find the optimum shipping schedule.
3. The network in Figure 5.10 shows the routes for shipping cars from three plants (nodes 1, 2, and 3) to three dealers (nodes 6 to 8) by way of two distribution centers (nodes 4 and 5). The shipping costs per car (in $100) are shown on the arcs.
   - **(a)** Solve the problem as a transshipment model.
   - **(b)** Find the new optimum solution assuming that distribution center 4 can sell 240 cars directly to customers.
*4. Consider the transportation problem in which two factories supply three stores with a commodity. The numbers of supply units available at sources 1 and 2 are 200 and 300; those demanded at stores 1, 2, and 3 are 100, 200, and 50, respectively. Units may be transshipped among the factories and the stores before reaching their final destination. Find the optimal shipping schedule based on the unit costs in Table 5.45.
5. Consider the oil pipeline network shown in Figure 5.11. The different nodes represent pumping and receiving stations. Distances in miles between the stations are shown on the network. The transportation cost per gallon between two nodes is directly proportional to the length of the pipeline. Develop the associated transshipment model, and find the optimum solution.
6. *Shortest-Route Problem.* Find the shortest route between nodes 1 and 7 of the network in Figure 5.12 by formulating the problem as a transshipment model. The distances between the different nodes are shown on the network. (*Hint:* Assume that node 1 has a net supply of 1 unit, and node 7 has a net demand also of 1 unit.)

FIGURE 5.10
Network for Problem 3, Set 5.5a

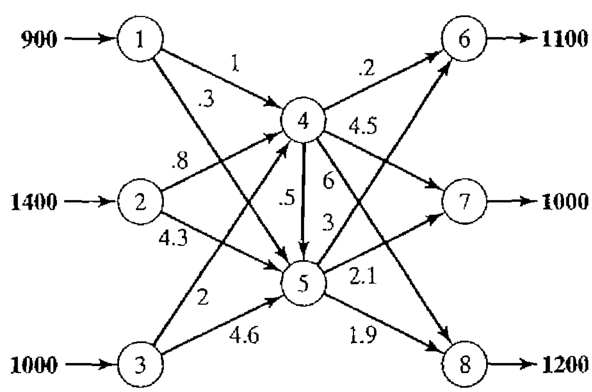


TABLE 5.45 Data for Problem 4

		Factory 1	Factory 2	Store 1	Store 2	Store 3
Factory	1	$0	$6	$7	$8	$9
	2	$6	$0	$5	$4	$3
Store	1	$7	$2	$0	$5	$1
	2	$1	$5	$1	$0	$4
	3	$8	$9	$7	$6	$0

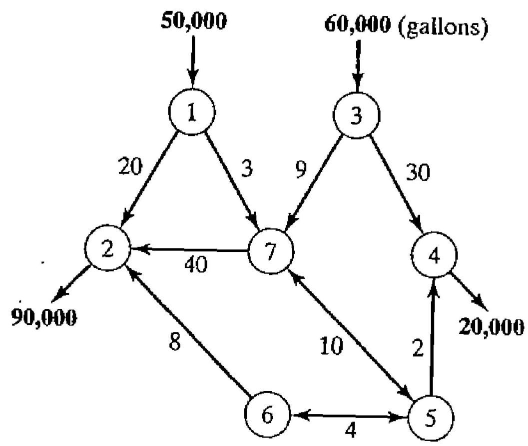


FIGURE 5.11
Network for Problem 5, Set 5.5a

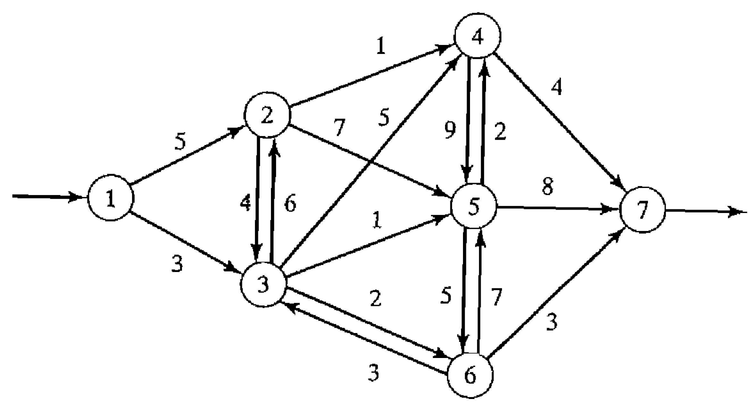


FIGURE 5.12
Network for Problem 6, Set 5.5a

7. In the transshipment model of Example 5.5-1, define x_{ij} as the amount shipped from node i to node j. The problem can be formulated as a linear program in which each node produces a constraint equation. Develop the linear program, and show that the resulting formulation has the characteristic that the constraint coefficients, a_{ij}, of the variable x_{ij} are

$$a_{ij} = \begin{cases} 1, \text{in constraint } i \\ -1, \text{in constraint } j \\ 0, \text{otherwise} \end{cases}$$

8. An employment agency must provide the following laborers over the next 5 months:

Month	1	2	3	4	5
No. of laborers	100	120	80	170	50

Because the cost of labor depends on the length of employment, it may be more economical to keep more laborers than needed during some months of the 5-month planning horizon. The following table estimates the labor cost as a function of the length of employment:

Months of employment	1	2	3	4	5
Cost per laborer ($)	100	130	180	220	250

Formulate the problem as a linear program. Then, using proper algebraic manipulations of the constraint equations, show that the model can be converted to a transshipment model, and find the optimum solution. (*Hint*: Use the transshipment characteristic in Problem 7 to convert the constraints of the scheduling problem into those of the transshipment model.)

REFERENCES

Bazaraa, M., J. Jarvis, and H. Sherali, *Linear Programming and Network Flows*, 2nd ed., Wiley, New York, 1990.

Dantzig, G., *Linear Programming and Extensions*, Princeton University Press, Princeton, N.J., 1963.

Hansen, P. and R. Wendell, "A Note on Airline Commuting," *Interfaces*, Vol. 12, No. 1, pp. 85–87, 1982.

Murty, K., *Network Programming*, Prentice Hall, Upper Saddle River, NJ, 1992.

CHAPTER 6

Network Models

Chapter Guide. The network models in this chapter include the traditional applications of finding the most efficient way to link a number of locations directly or indirectly, finding the shortest route between two cities, determining the maximum flow in a pipeline network, determining the minimum-cost flow in a network that satisfies supply and demand requirements at different locations, and scheduling the activities of a project.

The minimum-cost capacitated algorithm is a generalized network that subsumes the shortest-route and the maximal-flow models presented in this chapter. Its details can be found in Section 20.1 on the CD.

As you study the material in this chapter, you should pay special attention to the nontraditional applications of these models. For example, the shortest-route model can be used to determine the optimal equipment replacement policy and the maximum-flow model can be used to determine the optimum number of ships that meet a specific shipping schedule. These situations are included in the chapter as solved examples, problems, or cases.

Throughout the chapter, the formulation and solution of a network model as a linear program is emphasized. It is recommended that you study these relationships, because most commercial codes solve network problems as mere linear programs. Additionally, some formulations require imposing side constraints, which can be implemented only if the problem is solved as an LP.

To understand the computational details, you are encouraged to use TORA's interactive modules that create the steps of the solution in the exact manner presented in the book. For large-scale problems, the chapter offers both Excel Solver and AMPL models for the different algorithms.

This chapter includes a summary of 1 real-life application, 17 solved examples, 2 Solver models, 3 AMPL models, 69 end-of-section problems, and 5 cases. The cases are in Appendix E on the CD. The AMPL/Excel/Solver/TORA programs are in folder ch6Files.

Real-Life Application—Saving Federal Travel Dollars

U.S. Federal Government employees are required to attend development conferences and training courses in different locations around the country. Because the federal employees are located in offices scattered around the United States, the selection of the host city impacts travel cost. Currently, the selection of the city hosting conferences /training events is done without consideration of incurred travel cost. The problem seeks the determination of the optimal location of the host city. For Fiscal Year 1997, the developed model was estimated to save at least $400,000. Case 4 in Chapter 24 on the CD provides the details of the study.

6.1 SCOPE AND DEFINITION OF NETWORK MODELS

A multitude of operations research situations can be modeled and solved as networks (nodes connected by branches):

1. Design of an offshore natural-gas pipeline network connecting well heads in the Gulf of Mexico to an inshore delivery point. The objective of the model is to minimize the cost of constructing the pipeline.

2. Determination of the shortest route between two cities in an existing network of roads.

3. Determination of the maximum capacity (in tons per year) of a coal slurry pipeline network joining coal mines in Wyoming with power plants in Houston. (Slurry pipelines transport coal by pumping water through specially designed pipes.)

4. Determination of the time schedule (start and completion dates) for the activities of a construction project.

5. Determination of the minimum-cost flow schedule from oil fields to refineries through a pipeline network.

The solution of these situations, and others like it, is accomplished through a variety of network optimization algorithms. This chapter presents four of these algorithms.

1. Minimal spanning tree (situation 1)
2. Shortest-route algorithm (situation 2)
3. Maximal-flow algorithm (situation 3)
4. Critical path (CPM) algorithm (situation 4)

For the fifth situation, the minimum-cost capacitated network algorithm is presented in Section 20.1 on the CD.

Network Definitions. A network consists of a set of **nodes** linked by **arcs** (or **branches**). The notation for describing a network is (N, A), where N is the set of nodes and A is the set of arcs. As an illustration, the network in Figure 6.1 is described as

$$N = \{1, 2, 3, 4, 5\}$$
$$A = \{(1,2), (1,3), (2,3), (2,5), (3,4), (3,5), (4,2), (4,5)\}$$

Associated with each network is a **flow** (e.g., oil products flow in a pipeline and automobile traffic flows in highways). In general, the flow in a network is limited by the capacity of its arcs, which may be finite or infinite.

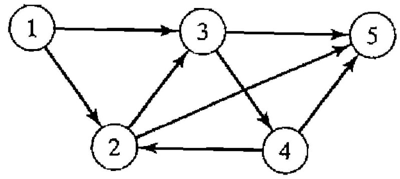
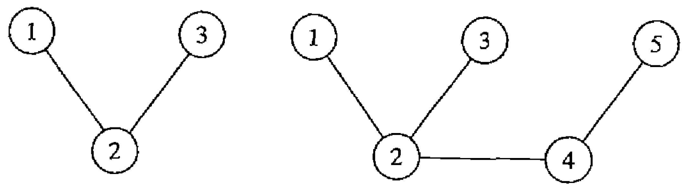


FIGURE 6.1
Example of (N, A) Network



Tree

Spanning tree

FIGURE 6.2
Examples of a tree and a spanning tree

An arc is said to be **directed** or **oriented** if it allows positive flow in one direction and zero flow in the opposite direction. A **directed network** has all directed arcs.

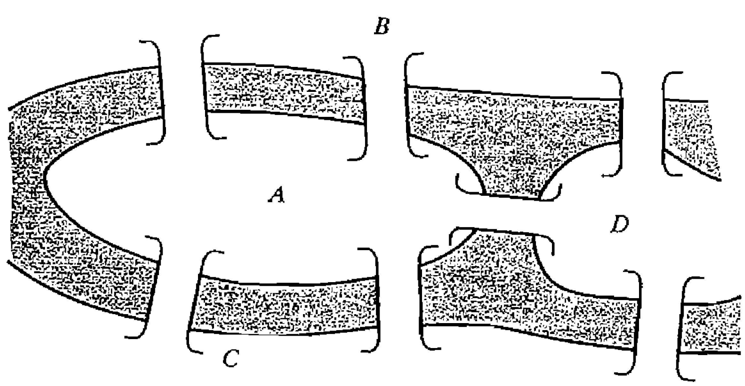
A **path** is a sequence of distinct arcs that join two nodes through other nodes regardless of the direction of flow in each arc. A path forms a **cycle** or a **loop** if it connects a node to itself through other nodes. For example, in Figure 6.1, the arcs (2, 3), (3, 4), and (4, 2) form a cycle.

A **connected network** is such that every two distinct nodes are linked by at least one path. The network in Figure 6.1 demonstrates this type of network. A **tree** is a *cycle-free* connected network comprised of a *subset* of all the nodes, and a **spanning tree** is a tree that links *all* the nodes of the network. Figure 6.2 provides examples of a tree and a spanning tree from the network in Figure 6.1.

Example 6.1-1 (Bridges of Königsberg)

The Prussian city of Königsberg (now Kalingrad in Russia) was founded in 1254 on the banks of river Pergel with seven bridges connecting its four sections (labeled *A*, *B*, *C*, and *D*) as shown in Figure 6.3. A problem circulating among the inhabitants of the city was to find out if a *round trip*

FIGURE 6.3
Bridges of Königsberg



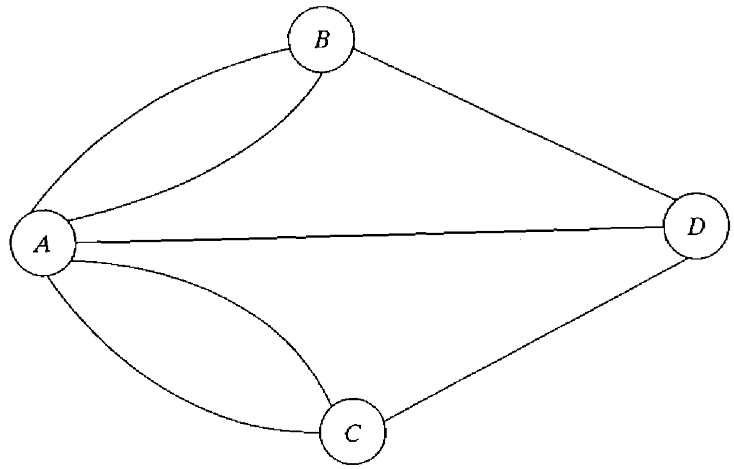


FIGURE 6.4
Network representation of Königsberg problem

of the four sections could be made with each bridge being crossed exactly once. No limits were set on the number of times any of the four sections could be visited.

In the mid-eighteenth century, the famed mathematician Leonhard Euler developed a special "path construction" argument to prove that it was impossible to make such a trip. Later, in the early nineteenth century the same problem was solved by representing the situation as a network in which each of the four sections (A, B, C, and D) is a node and each bridge is an arc joining applicable nodes, as shown in Figure 6.4.

The network-based solution is that the desired round trip (starting and ending in one section of the city) is impossible, because there are four nodes and each is associated with an *odd* number of arcs, which does not allow distinct entrance and exit (and hence distinct use of the bridges) to each section of the city.[1] The example demonstrates how the solution of the problem is facilitated by using network representation.

PROBLEM SET 6.1A

*1. For each network in Figure 6.5 determine (a) a path, (b) a cycle, (c) a tree, and (d) a spanning tree.

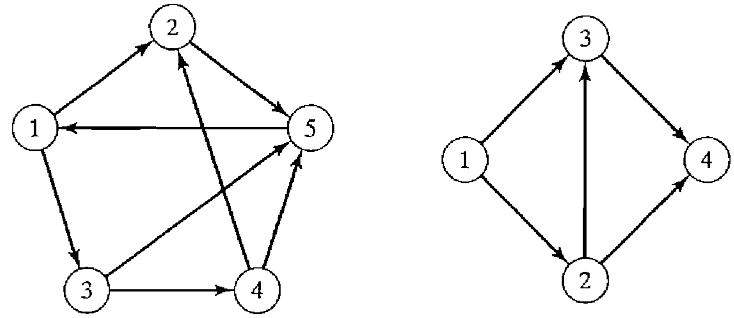
2. Determine the sets N and A for the networks in Figure 6.5.

3. Draw the network defined by

$$N = \{1, 2, 3, 4, 5, 6\}$$
$$A = \{(1, 2), (1, 5), (2, 3), (2, 4), (3, 4), (3, 5), (4, 3), (4, 6), (5, 2), (5, 6)\}$$

*4. Consider eight equal squares arranged in three rows, with two squares in the first row, four in the second, and two in the third. The squares of each row are arranged symmetrically about the vertical axis. It is desired to fill the squares with distinct numbers in the range 1 through 8

[1]General solution: A tour exists if all nodes have an even number of branches or if exactly two nodes have an odd number of branches. Else no tour exists. See B. Hopkins and R. Wilson, "The Truth about Königsberg," *College Math Journal*, Vol. 35, No. 3, pp. 198–207, 2004.



(i) (ii)

FIGURE 6.5
Networks for Problem 1, Set 6.1a

so that no two *adjacent* vertical, horizontal, or diagonal squares hold consecutive numbers. Use some form of a network representation to find the solution in a systematic way.

5. Three inmates escorted by 3 guards must be transported by boat from the mainland to a penitentiary island to serve their sentences. The boat cannot transfer more than two persons in either direction. The inmates are certain to overpower the guards if they outnumber them at any time. Develop a network model that designs the boat trips in a manner that ensures a smooth transfer of the inmates.

6.2 MINIMAL SPANNING TREE ALGORITHM

The minimal spanning tree algorithm deals with linking the nodes of a network, directly or indirectly, using the shortest total length of connecting branches. A typical application occurs in the construction of paved roads that link several rural towns. The road between two towns may pass through one or more other towns. The most economical design of the road system calls for minimizing the total miles of paved roads, a result that is achieved by implementing the minimal spanning tree algorithm.

The steps of the procedure are given as follows. Let $N = \{1, 2, \ldots, n\}$ be the set of nodes of the network and define

C_k = Set of nodes that have been permanently connected at iteration k
$\overline{C}_k$ = Set of nodes as yet to be connected permanently after iteration k

Step 0. Set $C_0 = \emptyset$ and $\overline{C}_0 = N$.

Step 1. Start with *any* node i in the unconnected set $\overline{C}_0$ and set $C_1 = \{i\}$, which renders $\overline{C}_1 = N - \{i\}$. Set $k = 2$.

General step k. Select a node, j^*, in the unconnected set $\overline{C}_{k-1}$ that yields the shortest arc to a node in the connected set C_{k-1}. Link j^* permanently to C_{k-1} and remove it from $\overline{C}_{k-1}$; that is,

$$C_k = C_{k-1} + \{j^*\}, \overline{C}_k = \overline{C}_{k-1} - \{j^*\}$$

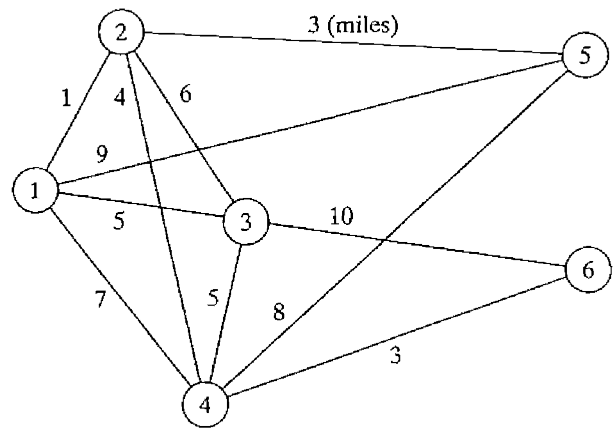


FIGURE 6.6
Cable connections for Midwest TV Company

If the set of unconnected nodes, $\overline{C}_k$, is empty, stop. Otherwise, set $k = k + 1$ and repeat the step.

Example 6.2-1

Midwest TV Cable Company is in the process of providing cable service to five new housing development areas. Figure 6.6 depicts possible TV linkages among the five areas. The cable miles are shown on each arc. Determine the most economical cable network.

The algorithm starts at node 1 (any other node will do as well), which gives

$$C_1 = \{1\}, \overline{C}_1 = \{2, 3, 4, 5, 6\}$$

The iterations of the algorithm are summarized in Figure 6.7. The thin arcs provide all the candidate links between C and $\overline{C}$. The thick branches represent the permanent links between the nodes of the connected set C, and the dashed branch represents the new (permanent) link added at each iteration. For example, in iteration 1, branch (1, 2) is the shortest link (= 1 mile) among all the candidate branches from node 1 to nodes 2, 3, 4, 5, and 6 of the unconnected set $\overline{C}_1$. Hence, link (1, 2) is made permanent and $j^* = 2$, which yields

$$C_2 = \{1, 2\}, \overline{C}_2 = \{3, 4, 5, 6\}$$

The solution is given by the minimal spanning tree shown in iteration 6 of Figure 6.7. The resulting minimum cable miles needed to provide the desired cable service are $1 + 3 + 4 + 3 + 5 = 16$ miles.

TORA Moment

You can use TORA to generate the iterations of the minimal spanning tree. From Main menu, select Network models ⇒ Minimal spanning tree. Next, from SOLVE/MODIFY menu, select Solve problem ⇒ Go to output screen. In the output screen, select a Starting node then use Next iteration or All iterations to generate the successive iterations. You can restart the iterations by selecting a new Starting Node. File toraEx6.2-1.txt gives TORA's data for Example 6.2-1.

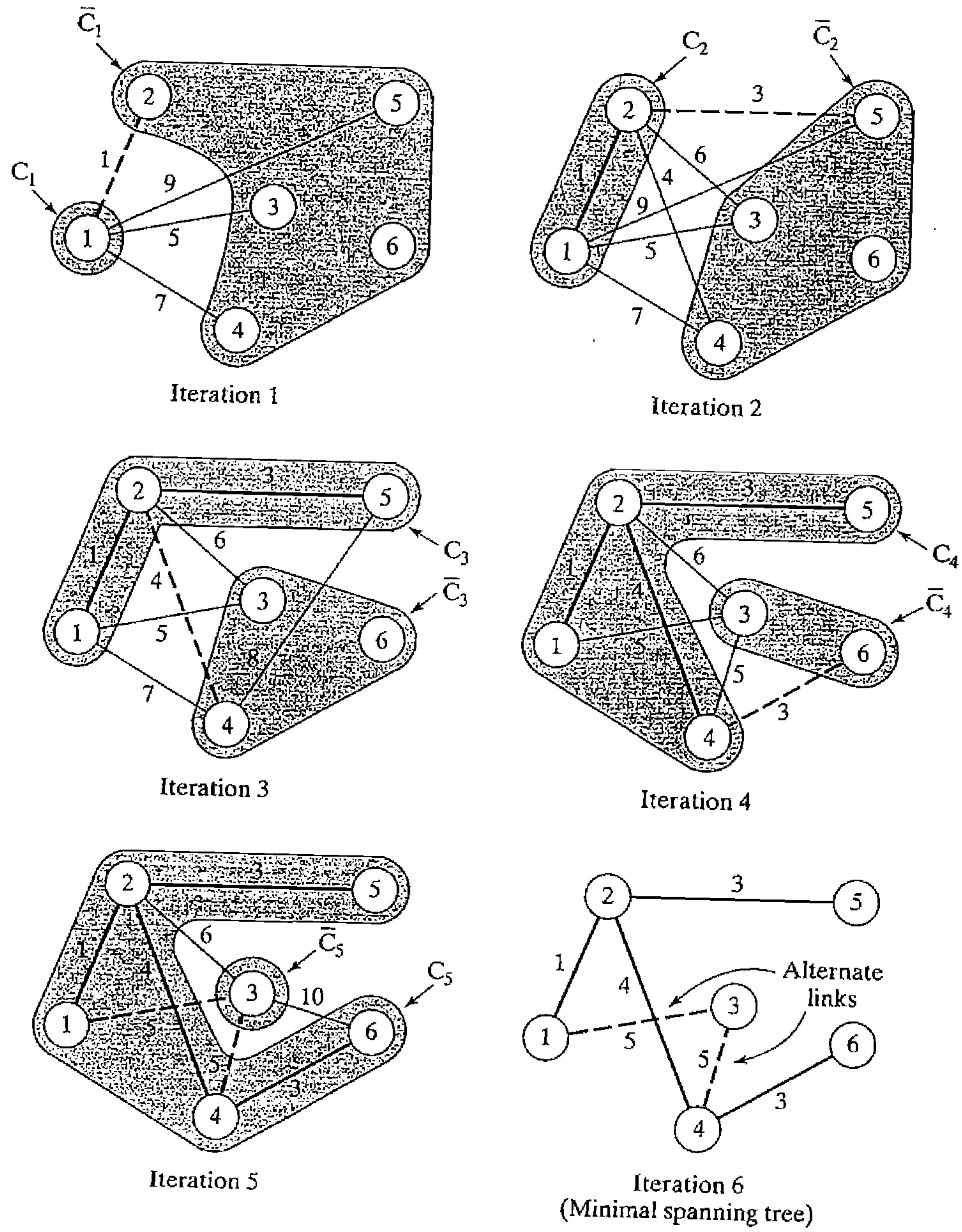


FIGURE 6.7
Solution iterations for Midwest TV Company

PROBLEM SET 6.2A

1. Solve Example 6.2-1 starting at node 5 (instead of node 1), and show that the algorithm produces the same solution.
2. Determine the minimal spanning tree of the network of Example 6.2-1 under each of the following separate conditions:
   *(a) Nodes 5 and 6 are linked by a 2-mile cable.
   (b) Nodes 2 and 5 cannot be linked.
   (c) Nodes 2 and 6 are linked by a 4-mile cable.

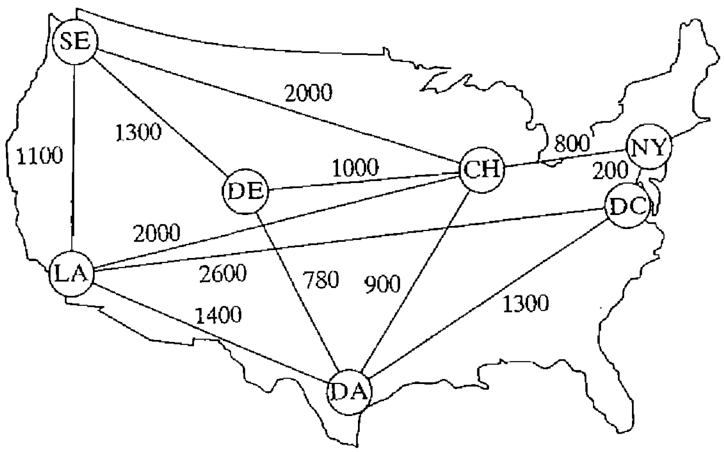


FIGURE 6.8
Network for Problem 3, Set 6.2a

(d) The cable between nodes 1 and 2 is 8 miles long.

(e) Nodes 3 and 5 are linked by a 2-mile cable.

(f) Node 2 cannot be linked directly to nodes 3 and 5.

3. In intermodal transportation, loaded truck trailers are shipped between railroad terminals on special flatbed carts. Figure 6.8 shows the location of the main railroad terminals in the United States and the existing railroad tracks. The objective is to decide which tracks should be "revitalized" to handle the intermodal traffic. In particular, the Los Angeles (LA) terminal must be linked directly to Chicago (CH) to accommodate expected heavy traffic. Other than that, all the remaining terminals can be linked, directly or indirectly, such that the total length (in miles) of the selected tracks is minimized. Determine the segments of the railroad tracks that must be included in the revitalization program.

4. Figure 6.9 gives the mileage of the feasible links connecting nine offshore natural gas wellheads with an inshore delivery point. Because wellhead 1 is the closest to shore, it is equipped with sufficient pumping and storage capacity to pump the output of the remaining eight wells to the delivery point. Determine the minimum pipeline network that links the wellheads to the delivery point.

***5.** In Figure 6.9 of Problem 4, suppose that the wellheads can be divided into two groups depending on gas pressure: a high-pressure group that includes wells 2, 3, 4, and 6, and a low-pressure group that includes wells 5, 7, 8, and 9. Because of pressure difference, it is not possible to link the wellheads from the two groups. At the same time, both groups must be connected to the delivery point through wellhead 1. Determine the minimum pipeline network for this situation. 6.

6. Electro produces 15 electronic parts on 10 machines. The company wants to group the machines into cells designed to minimize the "dissimilarities" among the parts processed in each cell. A measure of "dissimilarity," d_{ij}, among the parts processed on machines i and j can be expressed as 6.

$$d_{ij} = 1 - \frac{n_{ij}}{n_{ij} + m_{ij}}$$

where n_{ij} is the number of parts shared between machines i and j, and m_{ij} is the number of parts that are used by either machine i or machine j only.

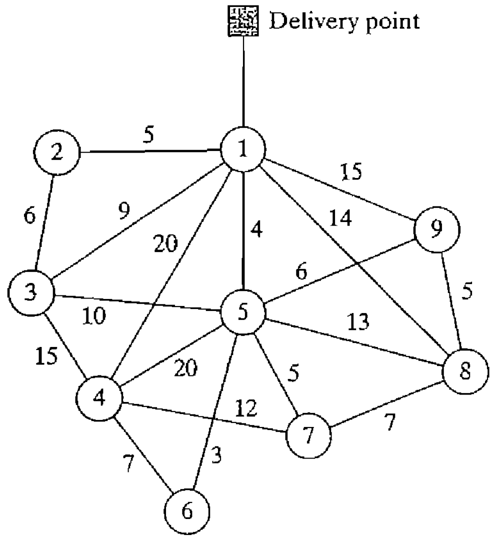


FIGURE 6.9
Network for Problem 4, Set 6.2a

The following table assigns the parts to machines:

Machine	Assigned parts
1	1, 6
2	2, 3, 7, 8, 9, 12, 13, 15
3	3, 5, 10, 14
4	2, 7, 8, 11, 12, 13
5	3, 5, 10, 11, 14
6	1, 4, 5, 9, 10
7	2, 5, 7, 8, 9, 10
8	3, 4, 15
9	4, 10
10	3, 8, 10, 14, 15

(a) Express the problem as a network model.

(b) Show that the determination of the cells can be based on the minimal spanning tree solution.

(c) For the data given in the preceding table, construct the two- and three-cell solutions.

6.3 SHORTEST-ROUTE PROBLEM

The shortest-route problem determines the shortest route between a source and destination in a transportation network. Other situations can be represented by the same model, as illustrated by the following examples.

6.3.1 Examples of the Shortest-Route Applications

Example 6.3-1 (Equipment Replacement)

RentCar is developing a replacement policy for its car fleet for a 4-year planning horizon. At the start of each year, a decision is made as to whether a car should be kept in operation or replaced.

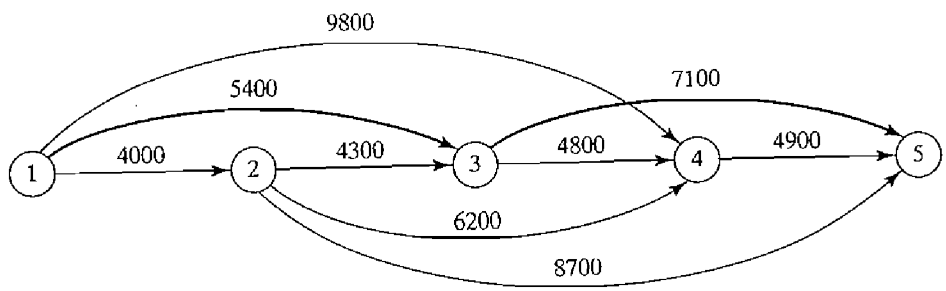


FIGURE 6.10
Equipment replacement problem as a shortest route model

A car must be in service a minimum of 1 year and a maximum of 3 years. The following table provides the replacement cost as a function of the year a car is acquired and the number of years in operation.

Equipment acquired at start of year	Replacement cost ($) for given years in operation		
	1	2	*3*
1	4000	5400	9800
2	4300	6200	8700
3	4800	7100	—
4	4900	—	—

The problem can be formulated as a network in which nodes 1 to 5 represent the start of years 1 to 5. Arcs from node 1 (year 1) can reach only nodes 2, 3, and 4 because a car must be in operation between 1 and 3 years. The arcs from the other nodes can be interpreted similarly. The length of each arc equals the replacement cost. The solution of the problem is equivalent to finding the shortest route between nodes 1 and 5.

Figure 6.10 shows the resulting network. Using TORA,[2] the shortest route (shown by the thick path) is $1 \rightarrow 3 \rightarrow 5$. The solution means that a car acquired at the start of year 1 (node 1) must be replaced after 2 years at the start of year 3 (node 3). The replacement car will then be kept in service until the end of year 4. The total cost of this replacement policy is \$12,500 (= \$5400 + \$7100).

Example 6.3-2 (Most Reliable Route)

I. Q. Smart drives daily to work. Having just completed a course in network analysis, Smart is able to determine the shortest route to work. Unfortunately, the selected route is heavily patrolled by police, and with all the fines paid for speeding, the shortest route may not be the best choice. Smart has thus decided to choose a route that maximizes the probability of *not* being stopped by police.

The network in Figure 6.11 shows the possible routes between home and work, and the associated probabilities of not being stopped on each segment. The probability of not being

[2]From Main menu, select Network models ⇒ Shortest route. From SOLVE/MODIFY menu, select Solve problem ⇒ Shortest routes.

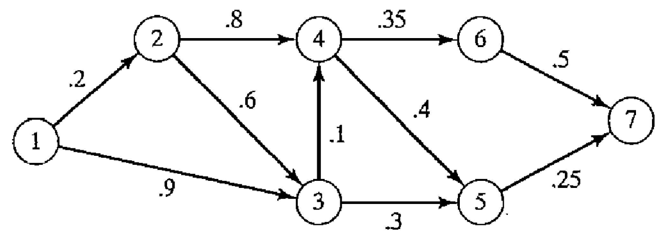


FIGURE 6.11
Most-reliable-route network model

stopped on a route is the product of the probabilities associated with its segments. For example, the probability of not receiving a fine on the route $1 \rightarrow 3 \rightarrow 5 \rightarrow 7$ is $.9 \times .3 \times .25 = .0675$. Smart's objective is to select the route that *maximizes* the probability of not being fined.

The problem can be formulated as a shortest-route model by using a logarithmic transformation that converts the product probability into the sum of the logarithms of probabilities—that is, if $p_{1k} = p_1 \times p_2 \times \cdots \times p_k$ is the probability of not being stopped, then $\log p_{1k} = \log p_1 + \log p_2 + \cdots + \log p_k$.

Mathematically, the maximization of $\log p_{1k}$ is equivalent to the maximization of $\log p_{1k}$. Because $\log p_{1k} \leq 0$, the maximization of $\log p_{1k}$ is equivalent to the minimization of $-\log p_{1k}$. Using this transformation, the individual probabilities p_j in Figure 6.11 are replaced with $-\log p_j$ for all j in the network, thus yielding the shortest-route network in Figure 6.12.

Using TORA, the shortest route in Figure 6.12 is defined by the nodes 1,3,5, and 7 with a corresponding "length" of 1.1707 ($= -\log p_{17}$). Thus, the maximum probability of not being stopped is $p_{17} = .0675$ only, not very encouraging news for Smart!

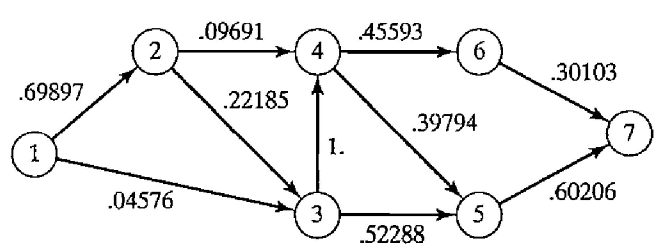
Example 6.3-3 (Three-Jug Puzzle)

An 8-gallon jug is filled with fluid. Given two empty 5- and 3-gallon jugs, we want to divide the 8 gallons of fluid into two equal parts using the three jugs. No other measuring devices are allowed. What is the smallest number of transfers (decantations) needed to achieve this result?

You probably can guess the solution to this puzzle. Nevertheless, the solution process can be systematized by representing the problem as a shortest-route problem.

A node is defined to represent the amount of fluid in the 8-, 5-, and 3-gallon jugs, respectively. This means that the network starts with node (8, 0, 0) and terminates with the desired

FIGURE 6.12
Most-reliable-route representation as a shortest-route model



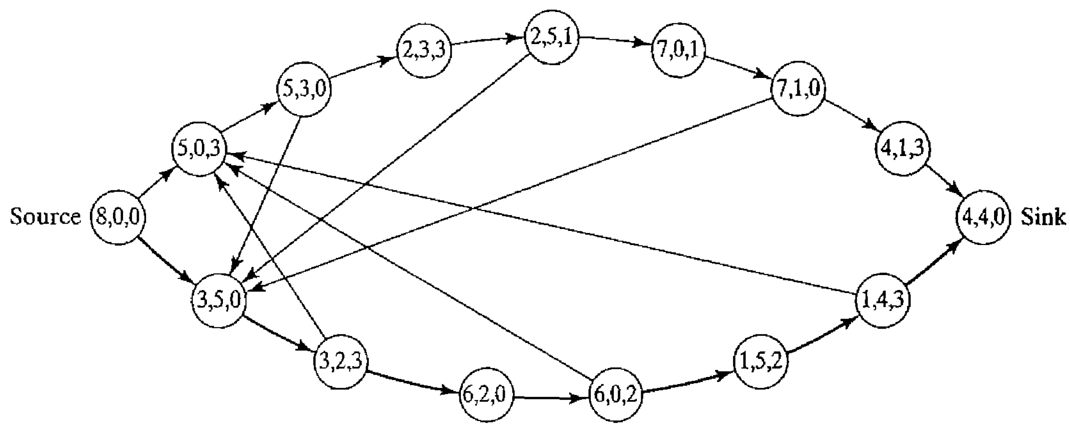


FIGURE 6.13
Three-jug puzzle representation as a shortest-route model

solution node (4, 4, 0). A new node is generated from the current node by decanting fluid from one jug into another.

Figure 6.13 shows different routes that lead from the start node (8, 0, 0) to the end node (4, 4, 0). The arc between two successive nodes represents a single transfer, and hence can be assumed to have a length of 1 unit. The problem thus reduces to determining the shortest route between node (8, 0, 0) and node (4, 4, 0).

The optimal solution, given by the bottom path in Figure 6.13, requires 7 transfers.

PROBLEM SET 6.3A

*1. Reconstruct the equipment replacement model of Example 6.3-1, assuming that a car must be kept in service at least 2 years, with a maximum service life of 4 years. The planning horizon is from the start of year 1 to the end of year 5. The following table provides the necessary data.

	Replacement cost ($) for given years in operation		
Year acquired	2	3	4
1	3800	4100	6800
2	4000	4800	7000
3	4200	5300	7200
4	4800	5700	—
5	5300	—	—

2. Figure 6.14 provides the communication network between two stations, 1 and 7. The probability that a link in the network will operate without failure is shown on each arc. Messages are sent from station 1 to station 7, and the objective is to determine the route that will maximize the probability of a successful transmission. Formulate the situation as a shortest-route model and determine the optimum solution.

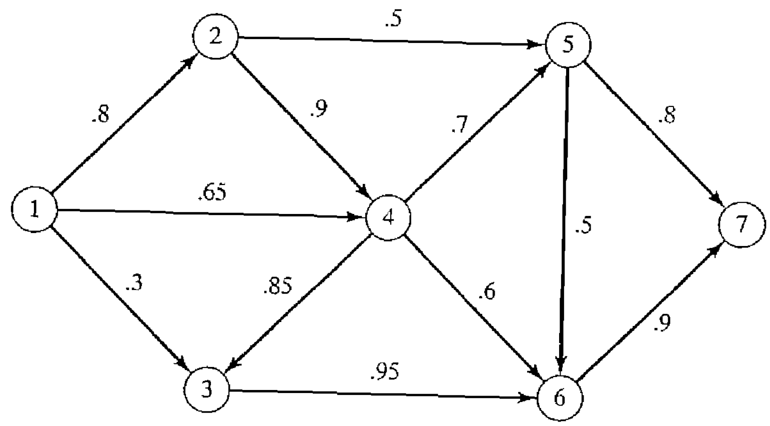


FIGURE 6.14
Network for Problem 2, Set 6.3a

3. *Production Planning.* DirectCo sells an item whose demands over the next 4 months are 100, 140, 210, and 180 units, respectively. The company can stock just enough supply to meet each month's demand, or it can overstock to meet the demand for two or more successive and consecutive months. In the latter case, a holding cost of $1.20 is charged per overstocked unit per month. DirectCo estimates the unit purchase prices for the next 4 months to be $15, $12, $10, and $14, respectively. A setup cost of $200 is incurred each time a purchase order is placed. The company wants to develop a purchasing plan that will minimize the total costs of ordering, purchasing, and holding the item in stock. Formulate the problem as a shortest-route model, and use TORA to find the optimum solution.

*4. *Knapsack Problem.* A hiker has a 5-ft^3 backpack and needs to decide on the most valuable items to take on the hiking trip. There are three items from which to choose. Their volumes are 2, 3, and 4 ft^3, and the hiker estimates their associated values on a scale from 0 to 100 as 30, 50, and 70, respectively. Express the problem as longest-route network, and find the optimal solution. (*Hint:* A node in the network may be defined as $[i, v]$, where i is the item number considered for packing, and v is the volume remaining immediately before a decision is made on i.)

5. An old-fashioned electric toaster has two spring-loaded base-hinged doors. The two doors open outward in opposite directions away from the heating element. A slice of bread is toasted one side at a time by pushing open one of the doors with one hand and placing the slice with the other hand. After one side is toasted, the slice is turned over to get the other side toasted. The goal is to determine the sequence of operations (placing, toasting, turning, and removing) needed to toast three slices of bread in the shortest possible time. Formulate the problem as a shortest-route model, using the following elemental times for the different operations:

Operation	Time (seconds)
Place one slice in either side	3
Toast one side	30
Turn slice already in toaster	1
Remove slice from either side	3

6.3.2 Shortest-Route Algorithms

This section presents two algorithms for solving both cyclic (i.e., containing loops) and acyclic networks:

1. Dijkstra's algorithm
2. Floyd's algorithm

Dijkstra's algorithm is designed to determine the shortest routes between the source node and every other node in the network. Floyd's algorithm is more general because it allows the determination of the shortest route between *any* two nodes in the network.

Dijkstra's Algorithm. Let u_i be the shortest distance from source node 1 to node i, and define d_{ij} (≥ 0) as the length of arc (i, j). Then the algorithm defines the label for an immediately succeeding node j as

$$[u_j, i] = [u_i + d_{ij}, i], d_{ij} \geq 0$$

The label for the starting node is [0,—], indicating that the node has no predecessor.

Node labels in Dijkstra's algorithm are of two types: *temporary* and *permanent*. A temporary label is modified if a shorter route to a node can be found. If no better route can be found, the status of the temporary label is changed to permanent.

Step 0. Label the source node (node 1) with the *permanent* label [0,—]. Set $i = 1$.

Step i. **(a)** Compute the *temporary* labels $[u_i + d_{ij}, i]$ for each node j that can be reached from node i, *provided j is not permanently labeled.* If node j is already labeled with $[u_j, k]$ through another node k and if $u_i + d_{ij} < u_j$, replace $[u_j, k]$ with $[u_i + d_{ij}, i]$.

(b) If *all* the nodes have *permanent* labels, stop. Otherwise, select the label $[u_r, s]$ having the shortest distance ($= u_r$) among all the *temporary* labels (break ties arbitrarily). Set $i = r$ and repeat step i.

Example 6.3-4

The network in Figure 6.15 gives the permissible routes and their lengths in miles between city 1 (node 1) and four other cities (nodes 2 to 5). Determine the shortest routes between city 1 and each of the remaining four cities.

Iteration 0. Assign the *permanent* label [0,—] to node 1.

Iteration 1. Nodes 2 and 3 can be reached from (the last permanently labeled) node 1. Thus, the list of labeled nodes (temporary and permanent) becomes

Node	Label	Status
1	**[0,—]**	**Permanent**
2	[0 + 100, 1] = [100, 1]	Temporary
3	[0 + 30, 1] = [30, 1]	Temporary

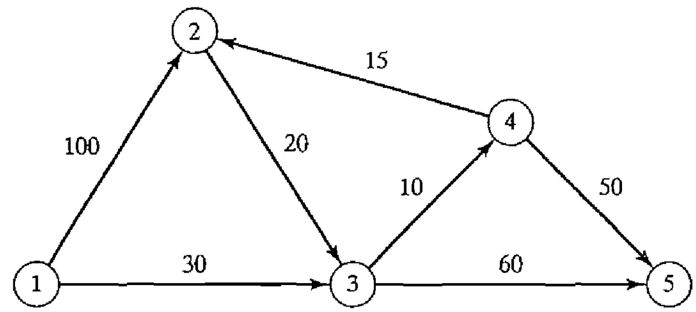


FIGURE 6.15

Network example for Dijkstra's shortest-route algorithm

For the two temporary labels [100, 1] and [30, 1], node 3 yields the smaller distance ($u_3 = 30$). Thus, the status of node 3 is changed to permanent.

Iteration 2. Nodes 4 and 5 can be reached from node 3, and the list of labeled nodes becomes

Node	Label	Status
1	[0, —]	Permanent
2	[100, 1]	Temporary
3	**[30, 1]**	**Permanent**
4	[30 + 10, 3] = [40, 3]	Temporary
5	[30 + 60, 3] = [90, 3]	Temporary

The status of the temporary label [40, 3] at node 4 is changed to permanent ($u_4 = 40$).

Iteration 3. Nodes 2 and 5 can be reached from node 4. Thus, the list of labeled nodes is updated as

Node	Label	Status
1	[0, —]	Permanent
2	[40 + 15, 4] = [55, 4]	Temporary
3	[30, 1]	Permanent
4	**[40, 3]**	**Permanent**
5	[90, 3] or [40 + 50, 4] = [90, 4]	Temporary

Node 2's temporary label [100, 1] obtained in iteration 1 is changed to [55, 4] in iteration 3 to indicate that a shorter route has been found through node 4. Also, in iteration 3, node 5 has two alternative labels with the same distance $u_5 = 90$.

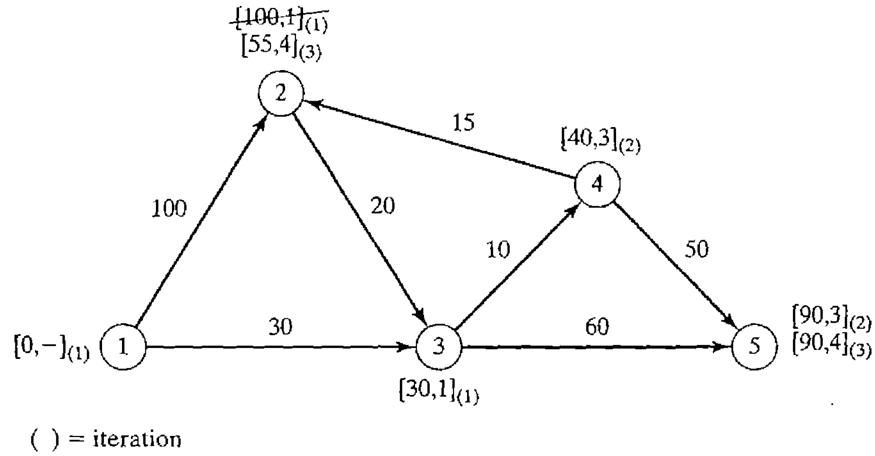


FIGURE 6.16
Dijkstra's labeling procedure

The list for iteration 3 shows that the label for node 2 is now permanent.

Iteration 4. Only node 3 can be reached from node 2. However, node 3 has a permanent label and cannot be relabeled. The new list of labels remains the same as in iteration 3 except that the label at node 2 is now permanent. This leaves node 5 as the only temporary label. Because node 5 does not lead to other nodes, its status is converted to permanent, and the process ends.

The computations of the algorithm can be carried out more easily on the network, as Figure 6.16 demonstrates.

The shortest route between nodes 1 and any other node in the network is determined by starting at the desired destination node and backtracking through the nodes using the information given by the permanent labels. For example, the following sequence determines the shortest route from node 1 to node 2:

$$(2) \rightarrow [55, 4] \rightarrow (4) \rightarrow [40, 3] \rightarrow (3) \rightarrow [30, 1] \rightarrow (1)$$

Thus, the desired route is $1 \rightarrow 3 \rightarrow 4 \rightarrow 2$ with a total length of 55 miles.

TORA Moment

TORA can be used to generate Dijkstra's iterations. From SOLVE/MODIFY menu, select Solve problem ⇒ Iterations ⇒ Dijkstra's algorithm. File toraEx6.3-4.txt provides TORA's data for Example 6.3-4.

PROBLEM SET 6.3B

1. The network in Figure 6.17 gives the distances in miles between pairs of cities 1, 2, ..., and 8. Use Dijkstra's algorithm to find the shortest route between the following cities:
   (a) Cities 1 and 8
   (b) Cities 1 and 6

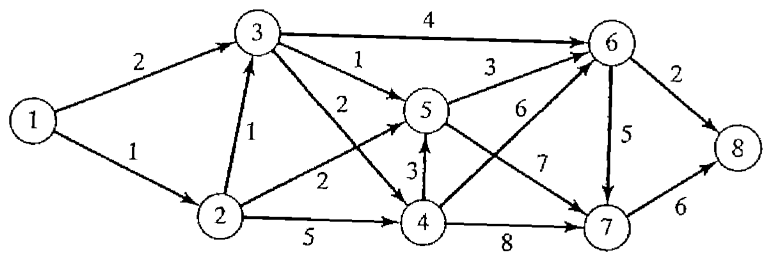


FIGURE 6.17

Network for Problem 1, Set 6.3b

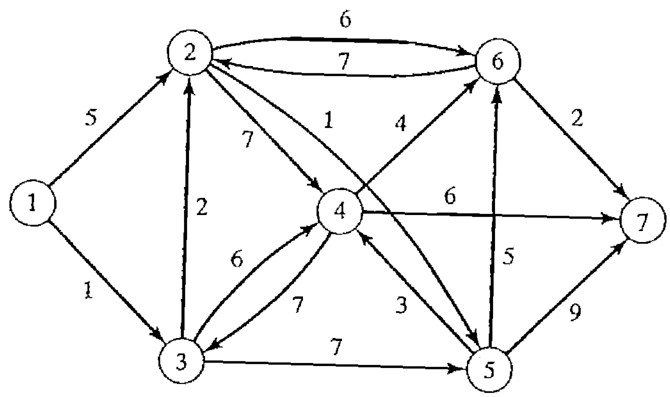


FIGURE 6.18

Network for Problem 2, Set 6.3b

*(c) Cities 4 and 8

(d) Cities 2 and 6

2. Use Dijkstra's algorithm to find the shortest route between node 1 and every other node in the network of Figure 6.18.

3. Use Dijkstr'a algorithm to determine the optimal solution of each of the following problems:

   (a) Problem 1, Set 6.3a.

   (b) Problem 2, Set 6.3a.

   (c) Problem 4, Set 6.3a.

Floyd's Algorithm. Floyd's algorithm is more general than Dijkstra's because it determines the shortest route between *any* two nodes in the network. The algorithm represents an n-node network as a square matrix with n rows and n columns. Entry (i, j) of the matrix gives the distance d_{ij} from node i to node j, which is finite if i is linked directly to j, and infinite otherwise.

The idea of Floyd's algorithm is straightforward. Given three nodes i, j, and k in Figure 6.19 with the connecting distances shown on the three arcs, it is shorter to reach j from i passing through k if

$$d_{ik} + d_{kj} < d_{ij}$$

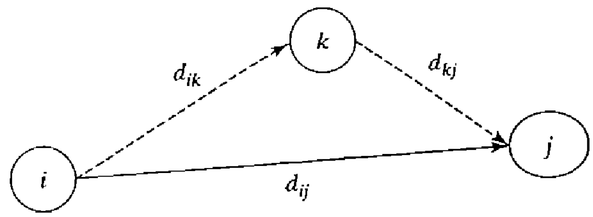


FIGURE 6.19
Floyd's triple operation

In this case, it is optimal to replace the direct route from $i \to j$ with the indirect route $i \to k \to j$. This **triple operation** exchange is applied systematically to the network using the following steps:

Step 0. Define the starting distance matrix D_0 and node sequence matrix S_0 as given below. The diagonal elements are marked with (—) to indicate that they are blocked. Set $k = 1$.

$D_0 =$

	1	2	...	j	...	n
1	—	d_{12}	...	d_{1j}	...	d_{1n}
2	d_{21}	—	...	d_{2j}	...	d_{2n}
⋮	⋮	⋮	⋮	⋮	⋮	⋮
i	d_{i1}	d_{i2}	...	d_{ij}	...	d_{in}
⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	D_{n1}	d_{n2}	...	d_{nj}	...	—

$S_0 =$

	1	2	...	j	...	n
1	—	2	...	j	...	n
2	1	—	...	j	...	n
⋮	⋮	⋮	⋮	⋮	⋮	⋮
i	1	2	...	j	...	n
⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	1	2	...	j	...	—

General step k. Define row k and column k as *pivot row* and *pivot column*. Apply the *triple operation* to each element d_{ij} in D_{k-1}, for all i and j. If the condition

$$d_{ik} + d_{kj} < d_{ij}, (i \neq k, j \neq k, \text{and } i \neq j)$$

is satisfied, make the following changes:

(a) Create D_k by replacing d_{ij} in D_{k-1} with $d_{ik} + d_{kj}$

(b) Create S_k by replacing s_{ij} in S_{k-1} with k. Set $k = k + 1$. If $k = n + 1$, stop; else repeat step k.

Step k of the algorithm can be explained by representing D_{k-1} as shown in Figure 6.20. Here, row k and column k define the current pivot row and column. Row i