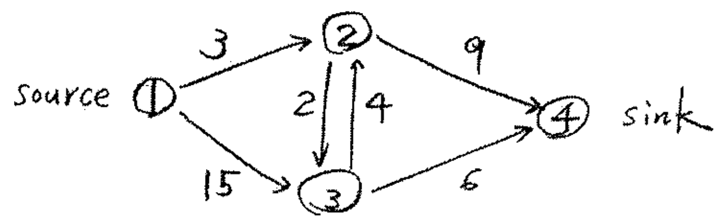
FORMULATING SHORTEST PATHS AS NETWORK FLOWS



$$\min \quad 3x_{1,2} + 15x_{1,3} + 2x_{2,3} + 9x_{2,4} + 4x_{3,2} + 6x_{3,4}$$

s.t.

$$-x_{1,2} - x_{1,3} = -1 \quad \text{(NODE 1)}$$

$$x_{1,2} + x_{3,2} - x_{2,3} - x_{2,4} = 0 \quad \text{(NODE 2)}$$

$$x_{1,3} + x_{2,3} - x_{3,2} - x_{3,4} = 0 \quad \text{(NODE 3)}$$

$$x_{2,4} + x_{3,4} = 1 \quad \text{(NODE 4)}$$

$$x_{1,2},\ x_{1,3},\ x_{2,3},\ x_{2,4},\ x_{3,2},\ x_{3,4} \in \{0, 1\}.$$

$$x_{i,j} = \begin{cases} 1 & \text{arc } i,j \text{ will be on optimal path} \\ 0 & \text{otherwise.} \end{cases}$$

Note: the coefficient of inflow arc +1
the coefficient of outflow arc −1
a supply of 1 at the source (−1)
and a demand of 1 at the sink (+1)

An optimal solution to the model is $x_{1,2} = x_{2,3} = x_{3,4} = 1$

$x_{3,2} = x_{1,3} = x_{2,4} = 0$. The shortest path is

1–2–3–4

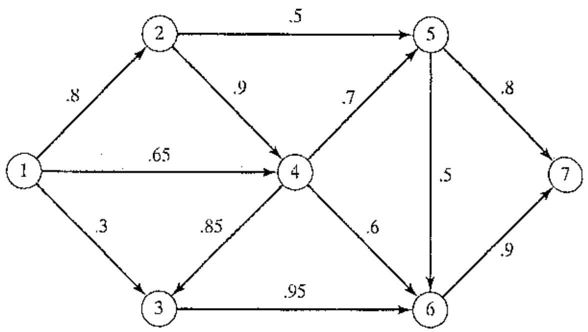


FIGURE 6.13
Network for Problem 2, Set 6.3a

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

4. *Production Planning.* DirectCo sells an item whose demand over the next 4 months is 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.
5. *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 a 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 the decision is made on i.)

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 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. At the point when no better routes 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 6.14 gives the 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

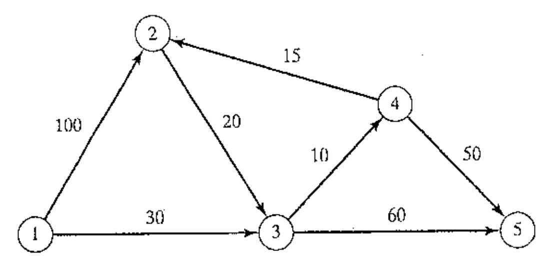


FIGURE 6.14
Network example for Dijkstra's shortest-route algorithm

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] in iteration 2 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 $u_5 = 90$.

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 6.15 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.

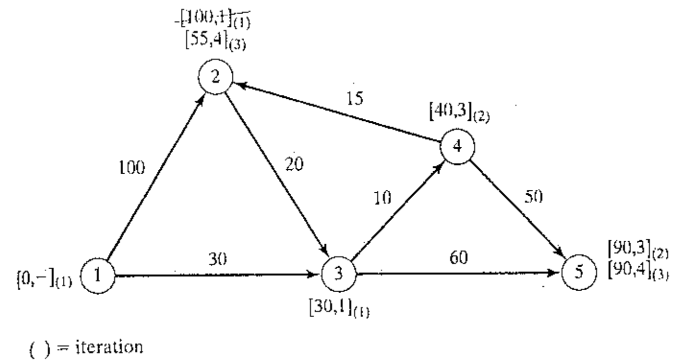


FIGURE 6.15
Dijkstra's labeling procedure

TORA can be used to generate Dijkstra's iterations. From the SOLVE/MODIFY menu, select `Solve problem` ⇒ `Iterations` ⇒ `Dijkstra's algorithm`. Figure 6.16 provides TORA's iterations output for Example 6.3-4 (file ch6ToraDijkstraEx6-3-4.txt).

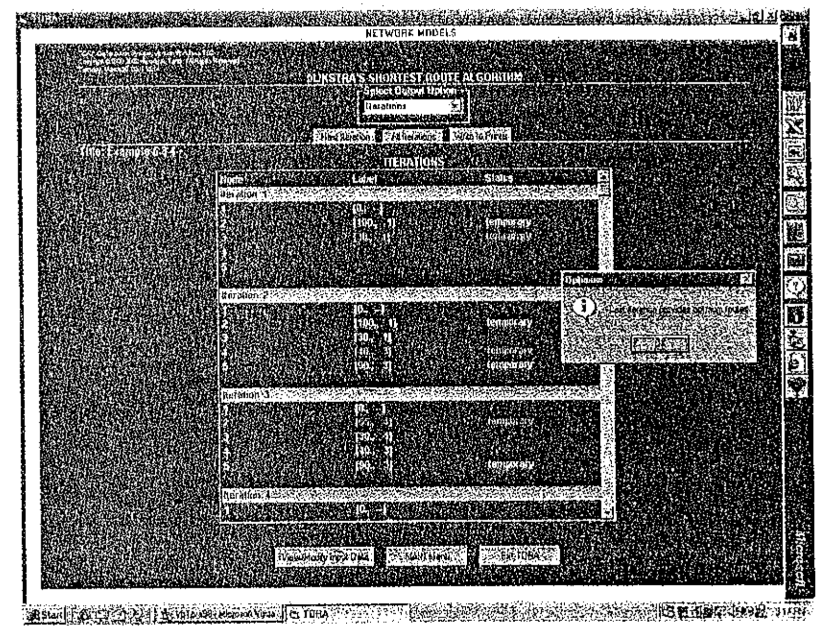
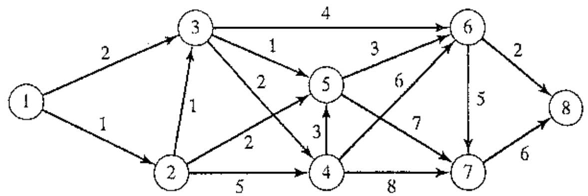


FIGURE 6.16
TORA Dijkstra iterations for Example 6.3-4

PROBLEM SET 6.3B

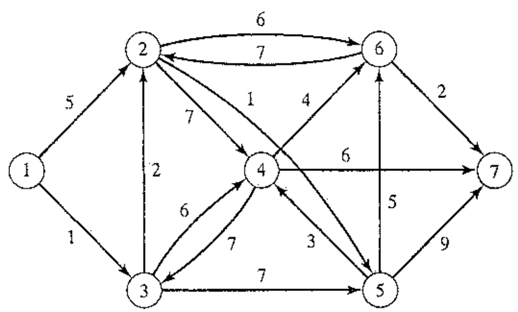
1. The network in Figure 6.17 gives the distances in miles between pairs of cities 1, 2, ..., and 8. Use Dijkstra's algorithm to find the shortest route between the following cities:
   (a) Cities 1 and 8
   (b) Cities 1 and 6
   (c) Cities 4 and 8
   (d) Cities 2 and 6

FIGURE 6.17
Network for Problem 1, Set 6.3b



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

FIGURE 6.18
Network for Problem 2, Set 6.3b



3. Use Dijkstra's 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.

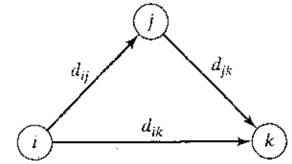


FIGURE 6.19
Floyd's triple operation

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 k from i passing through j if

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

In this case, it is optimal to replace the direct route from $i \to k$ with the indirect route $i \to j \to k$. 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 indicates 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 $k = k + 1$, and repeat step k.

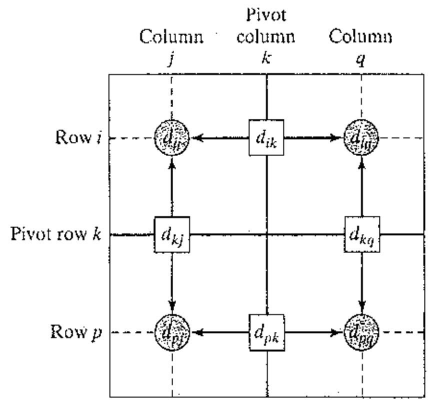


FIGURE 6.20
Implementation of triple operation in matrix form

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 represents any of the rows $1, 2, \ldots$, and $k - 1$, and row p represents any of the rows $k + 1$, $k + 2$, $\ldots$, and n. Similarly, column j represents any of the columns $1, 2, \ldots$, and $k - 1$, and column q represents any of the columns $k + 1$, $k + 2$, $\ldots$, and n. With the *triple operation*, if the sum of the elements on the pivot row and the pivot column (shown by squares) is smaller than the associated intersection element (shown by a circle), then it is optimal to replace the intersection distance by the sum of the pivot distances.

After n steps, we can determine the shortest route between nodes i and j from the matrices D_n and S_n using the following rules:

1. From D_n, d_{ij} gives the shortest distance between nodes i and j.
2. From S_n, determine the intermediate node $k = s_{ij}$ that yields the route $i \to k \to j$. If $s_{ik} = k$ and $s_{kj} = j$, stop; all the intermediate nodes of the route have been found. Otherwise, repeat the procedure between nodes i and k, and between nodes k and j.

Example 6.3-5

For the network in Figure 6.21, find the shortest routes between every two nodes. The distances (in miles) are given on the arcs. Arc (3,5) is directional so that no traffic is allowed from node 5 to node 3. All the other arcs allow traffic in both directions.

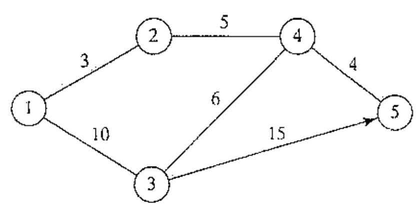


FIGURE 6.21
Network for Example 6.3-5

Iteration 0. The matrices D_0 and S_0 give the initial representation of the network. D_0 is symmetrical except that $d_{53} = \infty$ because no traffic is allowed from node 5 to node 3.

D_0

	1	2	3	4	5
1	—	3	10	∞	∞
2	3	—	∞	5	∞
3	10	∞	—	6	15
4	∞	5	6	—	4
5	∞	∞	∞	4	—

S_0

	1	2	3	4	5
1	—	2	3	4	5
2	1	—	3	4	5
3	1	2	—	4	5
4	1	2	3	—	5
5	1	2	3	4	—

Iteration 1. Set $k = 1$. The pivot row and column are shown by the lightly shaded first row and first column in the D_0-matrix. The darker cells, d_{23} and d_{32}, are the only ones that can be improved by the *triple operation*. Thus, D_1 and S_1 are obtained from D_0 and S_0 in the following manner:

1. Replace d_{23} with $d_{21} + d_{13} = 3 + 10 = 13$ and set $s_{23} = 1$.
2. Replace d_{32} with $d_{31} + d_{12} = 10 + 3 = 13$ and set $s_{32} = 1$.

These changes are shown in bold in matrices D_1 and S_1.

D_1

	1	2	3	4	5
1	—	3	10	∞	∞
2	3	—	**13**	5	∞
3	10	**13**	—	6	15
4	∞	5	6	—	4
5	∞	∞	∞	4	—

S_1

	1	2	3	4	5
1	—	2	3	4	5
2	1	—	**1**	4	5
3	1	**1**	—	4	5
4	1	2	3	—	5
5	1	2	3	4	—

Iteration 2. Set $k = 2$, as shown by the lightly shaded row and column in D_1. The *triple operation* is applied to the darker cells in D_1 and S_1. The resulting changes are shown in bold in D_2 and S_2.

D_2

	1	2	3	4	5
1	—	3	10	**8**	∞
2	3	—	13	5	∞
3	10	13	—	6	15
4	**8**	5	6	—	4
5	∞	∞	∞	4	—

S_2

	1	2	3	4	5
1	—	2	3	**2**	5
2	1	—	1	4	5
3	1	1	—	4	5
4	**2**	2	3	—	5
5	1	2	3	4	—

Iteration 3. Set $k = 3$, as shown by the shaded row and column in D_2. The new matrices are given by D_3 and S_3.

D_3	1	2	3	4	5
1	—	3	10	8	25
2	3	—	13	5	28
3	10	13	—	6	15
4	8	5	6	—	4
5	∞	∞	∞	4	—

S_3	1	2	3	4	5
1	—	2	3	2	3
2	1	—	1	4	3
3	1	1	—	4	5
4	2	2	3	—	5
5	1	2	3	4	—

Iteration 4. Set $k = 4$, as shown by the lightly-shaded row and column in D_3. The new matrices are given by D_4 and S_4.

D_4	1	2	3	4	5
1	—	3	10	8	**12**
2	3	—	**11**	5	**9**
3	10	**11**	—	6	**10**
4	8	5	6	—	4
5	**12**	**9**	**10**	4	—

S_4	1	2	3	4	5
1	—	2	3	2	4
2	1	—	4	4	4
3	1	4	—	4	4
4	2	2	3	—	5
5	4	4	4	4	—

Iteration 5. Set $k = 5$, as shown by the shaded row and column in D_4. No further improvements are possible in this iteration. Hence, D_5 and S_5 are the same as D_4 and S_4.

The final matrices D_5 and S_5 contain all the information needed to determine the shortest route between any two nodes in the network. For example, consider determining the shortest route from node 1 to node 5. First, the associated shortest distance is given by $d_{15} = 12$ miles. To determine the associated route, recall that a segment (i, j) represents a direct link only if $s_{ij} = j$. Otherwise, i and j are linked through at least one other intermediate node. Because $s_{15} = 4$, the route is initially given as $1 \rightarrow 4 \rightarrow 5$. Now, because $s_{14} = 2 \neq 4$, the segment $(1, 4)$ is not a *direct* link, and $1 \rightarrow 4$ must be replaced with $1 \rightarrow 2 \rightarrow 4$, and the route $1 \rightarrow 4 \rightarrow 5$ now becomes $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$. Next, because $s_{12} = 2$, $s_{24} = 4$, and $s_{45} = 5$, the route $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$ needs no further "dissecting" and the process ends.

As in Dijkstra's algorithm, TORA can be used to generate Floyd's iterations. From the SOLVE/MODIFY menu, select `Solve problem` ⇒ `Iterations` ⇒ `Floyd's algorithm`. Figure 6.22 illustrates TORA's output for Floyd's Example 6.3-5 (file ch6ToraFloydEx6-3-5.txt).

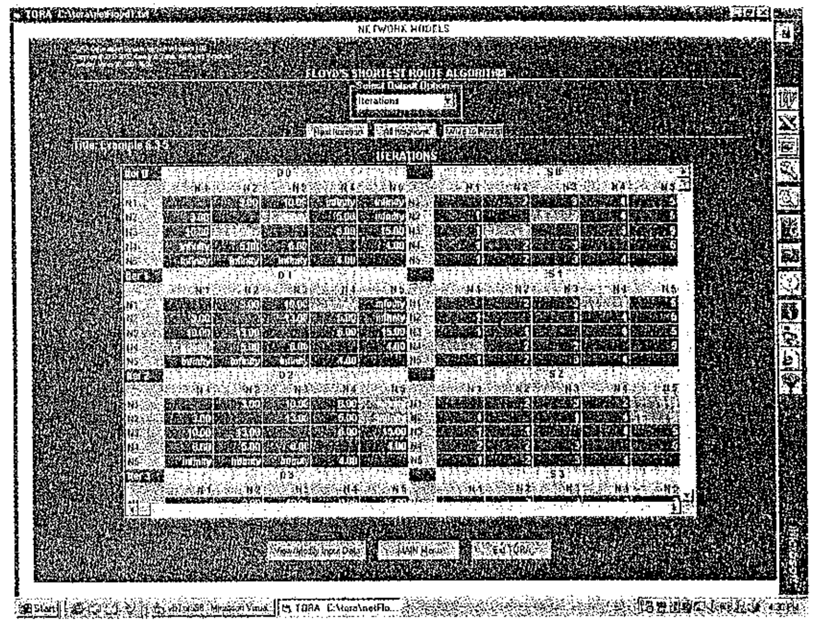


FIGURE 6.22
TORA Floyd iterations for Example 6.3-5

PROBLEM SET 6.3C

1. In Example 6.3-5, use Floyd's algorithm to determine the shortest routes between each of the following pairs of nodes:
   (a) From node 5 to node 1
   (b) From node 3 to node 5
   (c) From node 5 to node 3
   (d) From node 5 to node 2
2. Apply Floyd's algorithm to the network in Figure 6.23. Arcs (7, 6) and (6, 4) are unidirectional, and all the distances are in miles. Determine the shortest route between the following pairs of nodes:
   (a) From node 1 to node 7
   (b) From node 7 to node 1
   (c) From node 6 to node 7
3. The Tell-All mobile phone company services six geographical areas. The satellite distances (in miles) among the six areas are given in Figure 6.24. Tell-All needs to determine the most efficient message routes that should be established between each two areas in the network.
4. Six kids—Joe, Kay, Jim, Bob, Rae, and Kim—play a variation of the game of hide and seek. The hiding place of a child is known only to a select few of the other children. A

運輸問題(Transportation Problem)

1. 範例

當線性規劃問題中的變數有某些特定性質時，可用特殊的方法求解，加速其求解過程。運輸問題即為線性規劃的一特例。典型的運輸問題是有 m 個供給點(工廠)，每一供給點的供給量 S_i(i=1, 2..... m)為已知；另有 n 個需求點(配送中心)，每一需求點的需求量 d_j(j=1, 2...n)為已知。C_{ij} 代表由 i 供給地運送一單位貨物至 j 需求地的運輸成本。運輸問題即是滿足上述條件限制下，如何規劃貨物的運送使得總運輸成本最低。

<範例 1>某石油公司有三營運站與兩座煉油廠。每座煉油廠每天的產能為 60 單位（千桶）的油，三營運站每天的需求分別為 30、40、50 單位的油，其他相關每單位運送成本如下表，試問如何運送，可使得總運輸成本最低。

		營運站			供給量
		1	2	3	
煉油廠	1	2	3	4	60
	2	3	4	5	60
需求量		30	40	50	

上述問題寫成線性規劃問題如下：

$$\min\ Z=2X_{11}+3X_{12}+4X_{13}+3X_{21}+4X_{22}+5X_{23}$$

s. t.

$$X_{11}+X_{12}+X_{13}\leqq 50 \quad \text{（供給限制式）}$$
$$X_{21}+X_{22}+X_{23}\leqq 50 \quad \text{（供給限制式）}$$
$$X_{11}+X_{21}\geqq 30 \quad \text{（需求限制式）}$$
$$X_{12}+X_{22}\geqq 40 \quad \text{（需求限制式）}$$
$$X_{13}+X_{23}\geqq 40 \quad \text{（需求限制式）}$$
$$X_{11}, X_{12}, X_{13}, X_{21}, X_{22}, X_{23} \geqq 0$$

其中 X_{ij} 代表由供給地 i 運送至需求地 j 的貨物數量。

2. The Capacitated Transportation Problem 運輸路線有運量限制問題

<範例 2>接續範例 1，假設基於運輸車輛之調派，或運送路線上會有流量限制。煉油廠 1 可以運送至營運站 2 的油最多為 20 單位，煉油廠 2 可以運送至營運站 3 的油最多為 10 單位，其餘條件相同。

$\min Z=2X_{11}+3X_{12}+4X_{13}+3X_{21}+4X_{22}+5X_{23}$

s.t. $X_{11}+X_{12}+X_{13}\leqq 50$ （供給限制式）

$X_{21}+X_{22}+X_{23}\leqq 50$ （供給限制式）

$X_{11}+X_{21}\geqq 30$ （需求限制式）

$X_{12}+X_{22}\geqq 40$ （需求限制式）

$X_{13}+X_{23}\geqq 40$ （需求限制式）

$X_{12}\leqq 20$（運輸路線有運量限）

$X_{23}\leqq 10$（運輸路線有運量限）

$X_{11}, X_{12}, X_{13}, X_{21}, X_{22}, X_{23}\geqq 0$

3. The Capacitated and bounded Transportation Problem 運輸路線有運量限制與供需有上界與下界問題

<例 3>接續例 2，車輛運送時，基於經濟規模考慮，兩座煉油廠在供給時，必須至少是 40 單位的油，才符合經濟效益，其他條件維持相同。

$\min Z=2X_{11}+3X_{12}+4X_{13}+3X_{21}+4X_{22}+5X_{23}$

s.t. $X_{11}+X_{12}+X_{13}\leqq 50$ （供給限制式）

$X_{21}+X_{22}+X_{23}\leqq 50$ （供給限制式）

$X_{11}+X_{21}\geqq 30$ （需求限制式）

$X_{12}+X_{22}\geqq 40$ （需求限制式）

$X_{13}+X_{23}\geqq 40$ （需求限制式）

$X_{12}\leqq 20$（運輸路線有運量限）

$X_{23}\leqq 10$（運輸路線有運量限）

$X_{11}\geqq 40$ (供給下界)

$X_{12}\geqq 40$ (供給下界)

$X_{13}\geqq 40$ (供給下界)

$X_{21}\geqq 40$ (供給下界)

$X_{22}\geqq 40$ (供給下界)

$X_{23}\geqq 40$ (供給下界)

$X_{11}, X_{12}, X_{13}, X_{21}, X_{22}, X_{23}\geqq 0$

5. The demand for a perishable item over the next four months is 400, 300, 420, and 380 tons, respectively. The supply capacities for the same months are 500, 600, 200, and 300 tons. The purchase price per ton varies from month to month and is estimated at $100, $140, $120, and $150, respectively. Because the item is perishable, a current month's supply must be consumed within 3 months (starting with the current month). The storage cost per ton per month is $3. The nature of the item does not allow back-ordering. Solve the problem as a transportation model by TORA, and determine the optimum delivery schedule for the item over the next 4 months.
6. The demand for a special small engine over the next five quarters is 200, 150, 300, 250, and 400 units. The manufacturer supplying the engine has different production capacities estimated at 180, 230, 430, 300, and 300 for the five quarters. Back-ordering is not allowed, but the manufacturer may use overtime to fill the immediate demand, if necessary. The overtime capacity for each period is half the regular capacity. The production costs per unit for the five periods are $100, $96, $116, $102, and $106, respectively. The overtime production cost per engine is 50% higher than the regular production cost. If an engine is produced now for use in later periods, an additional storage cost of $4 per engine per period is incurred. Formulate the problem as a transportation model. Use TORA to determine the optimum number of engines to be produced during regular time and overtime of each period.
7. Periodic preventive maintenance is carried out on aircraft engines, where an important component must be replaced. The number of aircraft scheduled for such maintenance over the next six months is estimated at 200, 180, 300, 198, 230, and 290, respectively. All maintenance work is done during the first two days of the month, where a used component may be replaced with a new or an overhauled component. The overhauling of used components may be done in a local repair facility, where they will be ready for use at the beginning of the next month, or they may be sent to a central repair shop where a delay of 3 months (including the month in which maintenance occurs) is expected. The repair cost in the local shop is $120 per component. At the central facility, the cost is only $35 per component. An overhauled component used in a later month will incur an additional storage cost of $1.50 per unit per month. New components may be purchased at $200 each in month 1, with a 5% price increase every 2 months. Formulate the problem as a transportation model, and solve by TORA to determine the optimal schedule for satisfying the demand for the component over the next six months.
8. The National Parks Service is receiving four bids for logging at three pine forests in Arkansas. The three locations include 10,000, 20,000, and 30,000 acres. A single bidder can bid for at most 50% of the total acreage available. The bids per acre at the three locations are given in Table 5.15. Bidder 2 does not wish to bid on location 1, and bidder 3 cannot bid on location 2.

TABLE 5.15

		Location 1	Location 2	Location 3
Bidder	1	$520	$210	$570
	2	—	$510	$495
	3	$650	—	$240
	4	$180	$430	$710

(a) In the present situation, we need to *maximize* the total bidding revenue for the Parks Service. Show how the problem can be formulated as a transportation model.
(b) Use TORA to determine the acreage that should be assigned to each of the four bidders.

5.3 THE TRANSPORTATION ALGORITHM

The transportation algorithm follows the exact steps of the simplex method (Chapter 3). However, instead of using the regular simplex tableau, we take advantage of the special structure of the transportation model to organize the computations in a more convenient form.

We must add that the special transportation algorithm was developed early on when hand computations were the norm and the need for "shortcut" solution methods was warranted. Today, we have powerful computer codes that can solve a transportation model of any size as an LP. In fact, TORA uses the transportation model format only as a screen "veneer" but handles all necessary computations in the background using the regular simplex method. Nevertheless, the algorithm, aside from its historical significance, does provide insight into the use of the theoretical primal-dual relationships given in Section 4.2 to achieve a practical result, that of improving hand computations. The exercise is theoretically intriguing.

To facilitate the presentation of the details of the algorithm, we use the following numeric example.

Example 5.3-1 (SunRay Transport)

SunRay Transport Company ships truckloads of grain from three silos to four mills. The supply (in truckloads) and the demand (also in truckloads) together with the unit transportation costs per truckload on the different routes are summarized in the transportation model in Table 5.16. The unit transportation costs, c_{ij} (shown in the northeast corner of each box), are in hundreds of dollars.

TABLE 5.16

	Mill 1	Mill 2	Mill 3	Mill 4	Supply
Silo 1	10 / x_{11}	2 / x_{12}	20 / x_{13}	11 / x_{14}	**15**
Silo 2	12 / x_{21}	7 / x_{22}	9 / x_{23}	20 / x_{24}	**25**
Silo 3	4 / x_{31}	14 / x_{32}	16 / x_{33}	18 / x_{34}	**10**
Demand	5	15	15	15	

The model seeks the minimum-cost shipping schedule between the silos and the mills. This is equivalent to determining the quantity x_{ij} shipped from silo i to mill j ($i = 1, 2, 3$; $j = 1, 2, 3, 4$).

The steps of the transportation algorithm are exact parallels of the simplex algorithm.

Step 1. Determine a *starting* basic feasible solution, and go to step 2.

Step 2. Use the optimality condition of the simplex method to determine the *entering variable* from among all the nonbasic variables. If the optimality condition is satisfied, stop. Otherwise, go to step 3.

Step 3. Use the feasibility condition of the simplex method to determine the *leaving variable* from among all the current basic variables, and find the new basic solution. Return to step 2.

3.1 Determination of the Starting Solution

A general transportation model with m sources and n destinations has $m + n$ constraint equations, one for each source and each destination (see Example 5.1-1 for an illustration). However, because the transportation model is always balanced (sum of the supply = sum of the demand), one of these equations is redundant. Thus, the model has $m + n - 1$ independent constraint equations, which means that the starting basic solution consists of $m + n - 1$ basic variables. In Example 5.3-1, the starting solution has $3 + 4 - 1 = 6$ basic variables.

The special structure of the transportation problem allows securing a nonartificial starting basic solution using one of three methods:[2]

1. Northwest-corner method
2. Least-cost method
3. Vogel approximation method

The three methods differ in the "quality" of the starting basic solution they produce, in the sense that a better starting solution yields a smaller objective value. In general, the Vogel method yields the best starting basic solution, and the northwest-corner method yields the worst. The trade-off is that the northwest-corner method involves the least computations.

Northwest-Corner Method. The method starts at the northwest-corner cell (route) of the tableau (variable x_{11}).

Step 1. Allocate as much as possible to the selected cell, and adjust the associated amounts of supply and demand by subtracting the allocated amount.

[2]All three methods are featured in TORA's user-guided tutorial module. See Section 5.3.3.

Step 2. Cross out the row or column with zero supply or demand to indicate that no further assignments can be made in that row or column. If both a row and a column net to zero simultaneously, *cross out one only*, and leave a zero supply (demand) in the uncrossed-out row (column).

Step 3. If *exactly one* row or column is left uncrossed out, stop. Otherwise, move to the cell to the right if a column has just been crossed out or below if a row has been crossed out. Go to step 1.

Example 5.3-2

The application of the procedure to the model of Example 5.3-1 gives the starting basic solution in Table 5.17. The arrows show the order in which the allocated amounts are generated.

TABLE 5.17

	1	2	3	4	Supply
1	(10) 5 →	(2) 10 ↓	(20)	(11)	**15**
2	(12)	(7) 5 →	(9) 15 →	(20) 5 ↓	**25**
3	(4)	(14)	(16)	(18) 10	**10**
Demand	5	15	15	15	

The starting basic solution is given as

$$x_{11} = 5,\ x_{12} = 10$$
$$x_{22} = 5,\ x_{23} = 15,\ x_{24} = 5$$
$$x_{34} = 10$$

The associated cost of the schedule is

$$z = 5 \times 10 + 10 \times 2 + 5 \times 7 + 15 \times 9 + 5 \times 20 + 10 \times 18 = \$520$$

Least-Cost Method. The least-cost method finds a better starting solution by concentrating on the cheapest routes. The method starts by assigning as much as possible to the cell with the smallest unit cost (ties are broken arbitrarily). Next, the satisfied row or column is crossed out and the amounts of supply and demand are adjusted accordingly. If both a row and a column are satisfied simultaneously, *only one is crossed out*, the same as in the northwest-corner method. Next, look for the uncrossed-out cell with the smallest unit cost and repeat the process until exactly one row or column is left uncrossed out.

Example 5.3-3

The least-cost method is applied to Example 5.3-1 in the following manner:

1. Cell (1, 2) has the least unit cost in the tableau (= \$ 2). The most that can be shipped through (1, 2) is $x_{12} = 15$ truckloads, which happens to satisfy both row 1 and column 2 simultaneously. We arbitrarily cross out column 2 and adjust the supply in row 1 to 0.
2. Cell (3, 1) has the smallest uncrossed-out unit cost (= \$ 4). Assign $x_{31} = 5$, cross out column 1 because it is satisfied, and adjust the demand of row 3 to $10 - 5 = 5$ truckloads.
3. Continuing in the same manner, we successively assign 15 truckloads to cell (2, 3), 0 truckloads to cell (1, 4), 5 truckloads to cell (3, 4), and 10 truckloads to cell (2, 4) (verify!).

The resulting starting solution is summarized in Table 5.18. The arrows show the order in which the allocations are made. The starting solution (consisting of 6 basic variables) is

$$x_{12} = 15,\ x_{14} = 0$$
$$x_{23} = 15,\ x_{24} = 10$$
$$x_{31} = 5,\ x_{34} = 5$$

TABLE 5.18

	1	2	3	4	Supply
1	10	2 / **15**	20	11 / **0**	15
2	12	7	9 / **15**	20 / **10**	25
3	4 / **5**	14	16	18 / **5**	10
Demand	5	15	15	15	

The associated objective value is

$$z = 15 \times 2 + 0 \times 11 + 15 \times 9 + 10 \times 20 + 5 \times 4 + 5 \times 18 = \$475$$

The quality of the least-cost starting solution is better than that of the northwest-corner method (Example 5.3-2) because it yields a smaller value of z (\$475 versus \$520 in the northwest-corner method).

Vogel Approximation Method (VAM). VAM is an improved version of the least-cost method that generally produces better starting solutions.

Step 1. For each row (column), determine a penalty measure by subtracting the *smallest* unit cost element in the row (column) from the *next smallest* unit cost element in the same row (column).

Step 2. Identify the row or column with the largest penalty. Break ties arbitrarily. Allocate as much as possible to the variable with the least unit cost in the selected row or column. Adjust the supply and demand, and cross out the satisfied row *or* column. If a row and a column are satisfied simultaneously, only one of the two is crossed out, and the remaining row (column) is assigned zero supply (demand).

Step 3. (a) If exactly one row or column with zero supply or demand remains uncrossed out, stop.

(b) If one row (column) with *positive* supply (demand) remains uncrossed out, determine the basic variables in the row (column) by the least-cost method. Stop.

(c) If all the uncrossed out rows and columns have (remaining) zero supply and demand, determine the *zero* basic variables by the least-cost method. Stop.

(d) Otherwise, go to step 1.

Example 5.3-4

VAM is applied to Example 5.3-1. Table 5.19 computes the first set of penalties.

TABLE 5.19

	1	2	3	4		Row penalty
1	10	2	20	11	**15**	10 − 2 = 8
2	12	7	9	20	**25**	9 − 7 = 2
3	4 / **5**	14	16	18	**10**	14 − 4 = 10
	5	**15**	**15**	**15**		
Column penalty	10 − 4 = 6	7 − 2 = 5	16 − 9 = 7	18 − 11 = 7		

Because row 3 has the largest penalty (=10) and cell (3, 1) has the smallest unit cost in that row, the amount 5 is assigned to x_{31}. Column 1 is now satisfied and must be crossed out. Next, new penalties are recomputed as in Table 5.20.

Table 5.20 shows that row 1 has the highest penalty (=9). Hence, we assign the maximum amount possible to cell (1, 2), which yields $x_{12} = 15$ and simultaneously satisfies both row 1 and column 2. We arbitrarily cross out column 2 and adjust the supply in row 1 to zero.

Continuing in the same manner, row 2 will produce the highest penalty (=11), and we assign $x_{23} = 15$, which crosses out column 3 and leaves 10 units in row 2. Only

TABLE 5.20

	1	2	3	4		Row penalty
1	10 (shaded)	2	20	11	15	9
2	12 (shaded)	7	9	20	25	2
3	4 / 5 (shaded)	14	16	18	10	2
	5	15	15	15		
Column penalty	—	5	7	7		

column 4 is left, and it has a positive supply of 15 units. Applying the least-cost method to that column, we successively assign $x_{14} = 0$, $x_{34} = 5$, and $x_{24} = 10$ (verify!). Other solutions are possible depending on how ties are broken. The associated objective value for this solution is

$$z = 15 \times 2 + 0 \times 11 + 15 \times 9 + 10 \times 20 + 5 \times 4 + 5 \times 18 = \$475$$

This solution happens to have the same objective value as in the least-cost method. Usually, VAM produces a better starting solution.

PROBLEM SET 5.3A

1. Compare the starting solutions obtained by the northwest-corner, least-cost, and Vogel methods for each of the following models:

(a)

0	2	1	6
2	1	5	7
2	4	3	7
5	5	10	

(b)

1	2	6	7
0	4	2	12
3	1	5	11
10	10	10	

(c)

5	1	8	12
2	4	0	14
3	6	7	4
9	10	11	

2 Iterative Computations of the Transportation Algorithm

After determining the starting solution (using any of the three methods in Section 5.3.1), we use the following algorithm to determine the optimum solution:

Step 1. Use the simplex *optimality condition* to determine the *entering variable* as the current nonbasic variable that can improve the solution. If the optimality condition is satisfied, stop. Otherwise, go to step 2.

Step 2. Determine the *leaving variable* using the simplex *feasibility condition.* Change the basis, and return to step 1.

The change of basis computations do not involve the familiar row operations used in the simplex method. Instead, the special structure of the transportation model allows simpler computations.

Example 5.3-5

Solve the transportation model of Example 5.3-1, starting with the northwest-corner solution.

Table 5.21 gives the northwest-corner starting solution as determined in Table 5.17, Example 5.3-2.

TABLE 5.21

	1	2	3	4	Supply
1	10 / 5	2 / 10	20	11	15
2	12	7 / 5	9 / 15	20 / 5	25
3	4	14	16	18 / 10	10
Demand	5	15	15	15	

The determination of the entering variable from among the current nonbasic variables (those that are not part of the starting basic solution) is done by computing the nonbasic coefficients in the z-row, using the **method of multipliers** (which, as we show in Section 5.3.4, is rooted in LP duality theory).

In the method of multipliers, we associate the multipliers u_i and v_j with row i and column j of the transportation tableau. For each current *basic* variable x_{ij}, these multipliers are shown in Section 5.3.4 to satisfy the following equations:

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

In Example 5.3-1, 7 variables and 6 equations correspond to the six basic variables. To solve these equations, the method of multipliers calls for arbitrarily setting $u_i = 0$, and then solving for the remaining variables as shown below.

Basic variable	(u, v) equation	Solution
x_{11}	$u_1 + v_1 = 10$	$u_1 = 0 \rightarrow v_1 = 10$
x_{12}	$u_1 + v_2 = 2$	$u_1 = 0 \rightarrow v_2 = 2$
x_{22}	$u_2 + v_2 = 7$	$v_2 = 2 \rightarrow u_2 = 5$
x_{23}	$u_2 + v_3 = 9$	$u_2 = 5 \rightarrow v_3 = 4$
x_{24}	$u_2 + v_4 = 20$	$u_2 = 5 \rightarrow v_4 = 15$
x_{34}	$u_3 + v_4 = 18$	$v_4 = 15 \rightarrow u_3 = 3$

To summarize, we have

$$u_1 = 0,\ u_2 = 5,\ u_3 = 3$$

$$v_1 = 10,\ v_2 = 2,\ v_3 = 4,\ v_4 = 15$$

Next, we use u_i and v_j to evaluate the nonbasic variables by computing

$$u_i + v_j - c_{ij}, \text{ for each } nonbasic\ x_{ij}$$

The results of these evaluations are shown in the following table:

Nonbasic variable	$u_i + v_j - c_{ij}$
x_{13}	$u_1 + v_3 - c_{13} = 0 + 4 - 20 = -16$
x_{14}	$u_1 + v_4 - c_{14} = 0 + 15 - 11 = 4$
x_{21}	$u_2 + v_1 - c_{21} = 5 + 10 - 12 = 3$
x_{31}	$u_3 + v_1 - c_{31} = 3 + 10 - 4 = 9$
x_{32}	$u_3 + v_2 - c_{32} = 3 + 2 - 14 = -9$
x_{33}	$u_3 + v_3 - c_{33} = 3 + 4 - 16 = -9$

The preceding information, together with the fact that $u_i + v_j - c_{ij} = 0$ for each basic x_{ij}, is actually equivalent to computing the z-row of the simplex tableau as the following summary shows.

Basic	x_{11}	x_{12}	x_{13}	x_{14}	x_{21}	x_{22}	x_{23}	x_{24}	x_{31} ↓	x_{32}	x_{33}	x_{34}
z	0	0	−16	4	3	0	0	0	9	−9	−9	0

Because the transportation model seeks to *minimize* cost, the entering variable is the one having the *most positive* coefficient in the z-row. Thus, x_{31} is the entering variable.

The preceding computations are usually done directly on the transportation tableau as shown in Table 5.22, meaning that it is not necessary to write the (u, v)-equations explicitly. Instead, we start by setting $u_1 = 0$.[3] Then we can compute the v-values of all the columns that have *basic* variables in row 1, namely, v_1 and v_2. Next, we compute u_2 based on the (u, v)-equation of basic x_{22}. Now, given u_2, we can compute v_3 and v_4. Finally, we determine u_3 using the basic equation of x_3. Once all the u's and v's have been determined, we can evaluate the nonbasic variables by computing $u_i + v_j - c_{ij}$ for each nonbasic x_{ij}. These evaluations are shown in Table 5.22 in the boxed southeast corner of each cell.

TABLE 5.22

	$v_1 = 10$	$v_2 = 2$	$v_3 = 4$	$v_4 = 15$	Supply
$u_1 = 0$	(10) 5	(2) 10	(20) [−16]	(11) [4]	15
$u_2 = 5$	(12) [3]	(7) 5	(9) 15	(20) 5	25
$u_3 = 3$	(4) [9]	(14) [−9]	(16) [−9]	(18) 10	10
Demand	5	15	15	15	

[3]The tutorial module of TORA is designed to demonstrate that assigning a zero initial value to any u or v does not affect the optimization results. See Section 5.3.3.

Having determined x_{31} as the entering variable, we need to determine the leaving variable. Remember that if x_{31} enters the solution to become basic, one of the current basic variables must leave as nonbasic (at zero level).

The selection of x_{31} as the entering variable means that we want to ship through this route because it reduces the total shipping cost. What is the most that we can ship through the new route? Observe in Table 5.22 that if route (3, 1) ships θ (i.e., $x_{31} = \theta$), then the maximum value of θ is determined based on two conditions.

1. Supply limits and demand requirements remain satisfied.
2. Shipments through all routes must be nonnegative.

These two conditions determine the maximum value of θ and the leaving variable in the following manner. First, construct a *closed loop* that starts and ends at the entering variable cell (3, 1). The loop consists of *connected horizontal* and *vertical* segments only (no diagonals are allowed).[4] Except for the entering variable cell, each corner of the closed loop must coincide with a basic variable. Table 5.23 shows the loop for x_{31}. Exactly one loop exists for a given entering variable.

TABLE 5.23

	$v_1 = 10$	$v_2 = 2$	$v_3 = 4$	$v_4 = 15$	Supply
$u_1 = 0$	(10) 5 − θ (−)	(2) 10 + θ (+)	(20) [−16]	(11) [4]	15
$u_2 = 5$	(12) [3]	(7) 5 − θ (−)	(9) 15	(20) 5 + θ (+)	25
$u_3 = 3$	(4) θ (+) [9]	(14) [−9]	(16) [−9]	(18) 10 − θ (−)	10
Demand	5	15	15	15	

Next, we assign the amount θ to the entering variable cell (3, 1). For the supply and demand limits to remain satisfied, we must alternate between subtracting and adding the amount θ at the successive *corners* of the loop as shown in Table 5.23 (it is immaterial if the loop is traced in a clockwise or counterclockwise direction). The new values of the variables then remain nonnegative if

$$x_{11} = 5 - \theta \geq 0$$

$$x_{22} = 5 - \theta \geq 0$$

$$x_{34} = 10 - \theta \geq 0$$

[4]TORA's tutorial module allows you to determine the cells of the *closed loop* interactively with immediate feedback regarding the validity of your selections. See Section 5.3.3.

The maximum value of θ is 5, which occurs when both x_{11} and x_{22} reach zero level. Because only one current basic variable must leave the basic solution, we can choose either x_{11} or x_{22} as the leaving variable. We arbitrarily choose x_{11} to leave the solution.

The selection of x_{31} ($=5$) as the entering variable and x_{11} as the leaving variable requires adjusting the values of the basic variables at the corners of the closed loop as Table 5.24 shows. Because each unit shipped through route (3, 1) reduces the shipping cost by \$9 ($= u_3 + v_1 - c_{31}$), the total cost associated with the new schedule is \$9 × 5 = \$45 less than in the previous schedule. Thus, the new cost is \$520 − \$45 = \$475.

TABLE 5.24

	$v_1 = 1$	$v_2 = 2$	$v_3 = 4$	$v_4 = 15$	Supply
$u_1 = 0$	(10) −9	(2) $15 - \theta$	(20) −16	(11) θ; 4	15
$u_2 = 5$	(12) −6	(7) $0 + \theta$	(9) 15	(20) $10 - \theta$	25
$u_3 = 3$	(4) 5	(14) −9	(16) −9	(18) 5	10
Demand	5	15	15	15	

Given the new basic solution, we repeat the computation of the multipliers u and v as Table 5.24 shows. The entering variable is x_{14}. The closed loop shows that $x_{14} = 10$ and that the leaving variable is x_{24}.

The new solution, shown in Table 5.25, costs \$4 × 10 = \$40 less than the preceding one, thus yielding the new cost \$475 − \$40 = \$435. The new $u_i + v_j - c_{ij}$ are now negative for all nonbasic x_{ij}. Thus, the solution in Table 5.25 is optimal.

TABLE 5.25

	$v_1 = -3$	$v_2 = 2$	$v_3 = 4$	$v_4 = 11$	Supply
$u_1 = 0$	(10) −13	(2) 5	(20) −16	(11) 10	15
$u_2 = 5$	(12) −10	(7) 10	(9) 15	(20) −4	25
$u_3 = 7$	(4) 5	(14) −5	(16) −5	(18) 5	10
Demand	5	15	15	15	

The following table summarizes the optimum solution.

From silo	To mill	Number of truckloads
1	2	5
1	4	10
2	2	10
2	3	15
3	1	5
3	4	5
	Optimal cost = \$435	

5.3.3 Solution of the Transportation Model with TORA

We have already used TORA in an automated mode to solve the transportation model. This section introduces TORA's tutorial/iterative module. It also shows how the same model is solved by Excel Solver and LINGO.

TORA's Tutorial/Iterative Module. From `Solve/Modify Menu`, select `Solve ⇒ Iterations`, and then choose one of the three methods (northwest corner, least cost, or Vogel) to start the transportation model iterations. The iterations module offers two useful interactive features:

1. You can set any u or v to zero before generating Iteration 2 (the default is $u_1 = 0$). Observe then that although the values of u_i and v_j change, the evaluation of the nonbasic cells ($= u_i + v_j - c_{ij}$) remains the same. This means that, initially, any u or v can be set to zero (in fact, any value) without affecting the optimality calculations.
2. You can test your understanding of the selection of the *closed loop* by clicking (in any order) the cells that constitute the path. If your selection is correct, the cell will change color (green for entering variable, red for leaving variable, and gray otherwise).

Figure 5.4 provides TORA's iterations of Example 5.3-1 starting with the northwest-corner method.

Excel Solver Solution. Entering the transportation model into an Excel spreadsheet is straightforward. Figure 5.5 solves Example 5.3-1 (file ch5SolverTransportation.xls). The template can be used to solve models of up to 10 sources and 10 destinations. It divides the spreadsheet into input and output sections. In the input section, *mandatory* data include the number of sources (cell B3), number of destinations (cell B4), unit cost matrix (cells B6:K15), source names (cells A6:A15), destination names (cells B5:K5), supply amounts (cells L6:L15), and demand (cells B16:K16). The output section (cells B20:K29) provides the optimal solution in matrix form automatically. The associated total cost is given in cell A19. We have arbitrarily limited the model size