3.98 (*a*) It is known that $\langle 1, 1, 2, 3, 3, t \rangle$ is the score sequence of a tournament. Is it a score sequence of a strong tournament? (*b*) It is known that $\langle 1, 2, 2, 3, 3, t \rangle$ is the score sequence of a tournament. Is it a score sequence of a strong tournament?
Ans. (*a*) No. (Here $t = 5$.) (*b*) Yes. (Here $t = 4$.)

3.99 Show that the sum of the squares of the terms in the score sequence of a tournament is equal to the sum of the squares of the indegrees of the vertices. [*Hint:* If n is the order of the graph, the sum of the outdegree and indegree at each vertex is $(n - 1)$, and the sum of all the indegrees is equal to the sum of all the outdegrees.]

3.100 Show that a regular tournament is strongly connected. [*Hint:* If the outdegree of each vertex is d in a tournament with n vertices, nd is $n(n - 1)/2$. Use the condition established in Problem 3.86.]

3.101 Show that if $\langle s_1, s_2, \ldots, s_n \rangle$ is the score sequence of a tournament, $\langle t_1, t_2, \ldots, t_n \rangle$ is also a score sequence of a tournament, where $t_i = (n - 1) - s_i$ for $i = 1, 2, \ldots, n$. [*Hint:* Use the condition established in Problem 3.85.]

3.102 Examine the case when the inequalities in the statement of Problem 3.85 are all equalities.
Ans. In this case, the tournament is transitive.

3.103 Show that in a tournament involving n players, the number of players with the score $(n - 1)$ is at most 1. [*Hint:* Otherwise, the inequality condition established in Problem (85) will not hold.]

3.104 Show that if n is an odd number other than 1, there is a tournament of order n in which every vertex is a winner. [*Hint:* Each term in the score sequence is $(n - 1)/2$.]

3.105 Show that the k-cube (see Solved Problem 1.56) is a Hamiltonian graph. [*Hint:* The k-cube is a k-regular bipartite graph $G = (X, Y, E)$, where both X and Y have 2^{k-1} vertices. If $k = 3$, the path 000 010 110 100 101 111 011 001 is a Hamiltonian path from 000 to 001 that can be completed into a Hamiltonian cycle. Notice that the first four vertices have 0 as the last digit and that the last four have 1 as the last digit. To obtain a Hamiltonian path (when $k = 4$) from 0000 to 0001, we take these eight vertices and attach a 0 to them as a last digit, then take these new vertices in reverse order and replace the 0 in the last digit by 1. Finally, join 0001 and 0000 to get the Hamiltonian cycle. The proof is completed by this induction argument.]

3.106 For what value of k is the k-cube graph an Eulerian graph?
Ans. The k-cube is Eulerian if and only if k is even.

3.107 Show that the de Bruijn digraph is a Hamiltonian graph. [*Hint:* See the definition of the de Bruijn graph.]

Chapter 4

Optimization Involving Trees

4.1 MINIMUM WEIGHT SPANNING TREES

If each edge e of a graph G is assigned a real number $w(e)$ as its weight, the graph equipped with this allocation of weights is known as a **weighted graph.** The definition of a weighted digraph is analogous. A **network** is a weighted graph or a weighted digraph. In some situations with weights defined on the edges and arcs of a mixed graph, we have a network in which there are weighted edges as well as weighted arcs. In a figure representing a network, the weight of an edge or arc is usually written as a number on the edge or arc, as the case may be. If H is a subgraph of a weighted graph G, the weight of H is the sum of the weights of all the edges in H; this is denoted as $w(H)$. A spanning tree T in a weighted digraph G is a **minimum weight spanning tree** in G (or **minimum spanning tree** or **minimum connector**) if $w(T) \leq w(T')$ for every spanning tree T' in G.

In this section, we discuss two algorithms to obtain a minimum spanning tree (M.S.T.) in a connected weighted graph. Both the procedures are "greedy" in the sense that at every stage, a decision is to be made to select the best possible edge from the graph for inclusion as an edge in the M.S.T. after making sure that the selection of an edge does not create a cycle in the subgraph already constructed.

Kruskal's Algorithm

The input is the set of edges in a weighted graph with n vertices. The output is either a report that the graph is not connected or the set of edges in a minimum spanning tree T in the network. There are four steps in this algorithm.

Step 1. Arrange the edges of the weighted graph in nondecreasing order of their weights as a list L, and set T to be the empty set.

Step 2. Select the first edge from L, and include that in T.

Step 3. If every edge in L is examined for possible inclusion in T, stop and report that G is not connected. Otherwise, take the first unexamined edge in L. If it does not form a cycle in T, select it for inclusion in T and go to step 4. Otherwise, discard that edge as an examined (but unselected) edge and repeat step 3.

Step 4. Stop if T has $(n - 1)$ edges. Otherwise, go to step 3.

Theorem 4.1. Kruskal's algorithm solves the M.S.T. problem in a network. (See Solved Problem 4.14.)

Example 1. Show by using Kruskal's algorithm that the network in Fig. 4-1 is a disconnected weighted graph.

Here $n = 7$. The list L of edges with weights in nondecreasing order is $\{1, 2\}$, $\{1, 3\}$, $\{5, 6\}$, $\{6, 7\}$, $\{2, 3\}$, $\{5, 7\}$, $\{1, 4\}$, and $\{3, 4\}$. As specified in the algorithm, we select edges $\{1, 2\}$, $\{1, 3\}$, $\{5, 6\}$, and $\{6, 7\}$. Then edge $\{2, 3\}$ is examined but not selected. Subsequently, edge $\{5, 7\}$ is examined but not selected. Then edge $\{1, 4\}$ is examined and selected. Finally, edge $\{3, 4\}$ is examined but not selected. At this stage all the edges in L are examined. So the graph is not connected.

Example 2. Obtain an M.S.T. in the network shown in Fig. 4-2 using Kruskal's algorithm.

Here $n = 8$. We take edges $\{1, 2\}$, $\{3, 4\}$, $\{1, 8\}$, $\{4, 5\}$, and $\{7, 8\}$, in that order, using the greedy procedure. We disregard the next entry, $\{2, 7\}$, and select $\{5, 6\}$. Then we ignore $\{3, 6\}$ and selected $\{6, 7\}$. At this stage, the number of

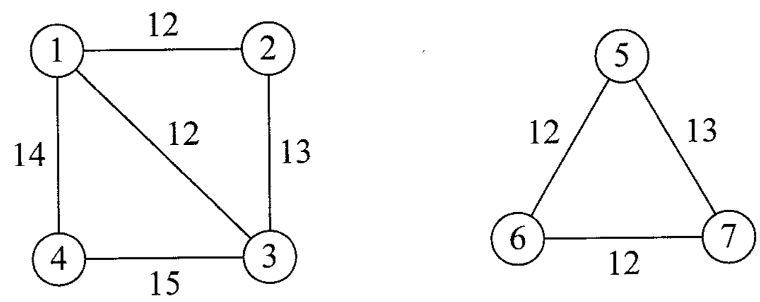


Fig. 4-1

edges selected for inclusion in the tree is seven, which is the number of trees in a spanning tree. The seven edges thus selected constitute the edges of a minimum spanning tree T in the given network with $w(T) = 93$, as shown in Fig. 4-3. The positive integer displayed on an edge in this figure indicates the order in which the edge was chosen for inclusion in T, and it should not be confused with the weight of the edge.

Prim's Algorithm

The input and the output are the same as in Kruskal's algorithm. There are three steps in this algorithm.

Step 1. Select an arbitrary vertex of the weighted graph, and include it as a vertex in T.

Step 2. Let W be the set of vertices in T. If the disconnecting set $(W, V - W)$ is empty, report that G is not connected. Otherwise, find an edge of minimum weight in the disconnecting set $(W, V - W)$. Tiebreaking is arbitrary. If this edge does not create a cycle in T, select it for inclusion in T and go to step 3. Otherwise, discard that edge and repeat step 2.

Step 3. Stop if T has $(n - 1)$ edges. Otherwise, go to step 2.

Theorem 4.2. Prim's algorithm solves the M.S.T. problem. (See Solved Problem 4.17.)

Example 3. Use Prim's algorithm to show that the network shown in Fig. 4-1 is not a connected graph.

Since $n = 7$, there should be six edges in the tree. Initially, $W = \{1\}$. The algorithm selects edges $\{1, 2\}$, $\{1, 3\}$, and $\{1, 4\}$. At this stage, the number of selected edges is less than six, and $W = \{1, 2, 3, 4\}$. The disconnecting set $(W, V - W)$ is empty. So the graph is not connected.

Example 4. Obtain an M.S.T. in the network shown in Fig. 4-2 using Prim's algorithm.

We start from vertex 1 and select edge $\{1, 2\}$. At this stage, W consists of two vertices 1 and 2. An edge of minimum weight in the disconnecting set $(W, V - W)$ is edge $\{1, 8\}$. Currently, edges $\{1, 2\}$ and $\{1, 8\}$ are in T, and $W = \{1, 2, 8\}$. An edge of smallest weight in $(W, V - W)$ is either $\{2, 7\}$ or $\{7, 8\}$. The tie is broken by selecting edge $\{2, 7\}$.

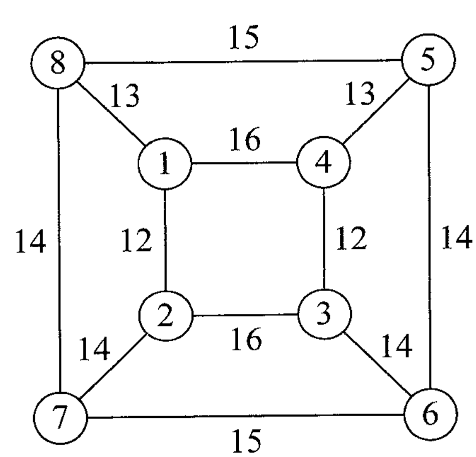


Fig. 4-2

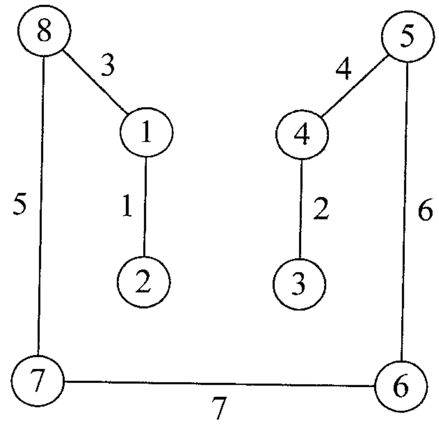


Fig. 4-3

At this stage, $W = \{1, 2, 7, 8\}$, and the edges in T are $\{1, 2\}$, $\{1, 8\}$, and $\{2, 7\}$. An edge of minimum weight in $(W, V - W)$ is $\{7, 8\}$, which forms a cycle in T. So this edge is discarded. Then *an* edge of minimum weight in $(W, V - W)$ is $\{5, 8\}$. We continue like this and select $\{8, 5\}$, $\{5, 4\}$, $\{4, 3\}$, and $\{3, 6\}$, in that order. At this stage, T has seven edges and we stop. In Fig. 4-3, the graph has seven edges and we stop. Figure 4-4 shows the graph of T in which the integers marked on the edges indicate the order in which they are selected.

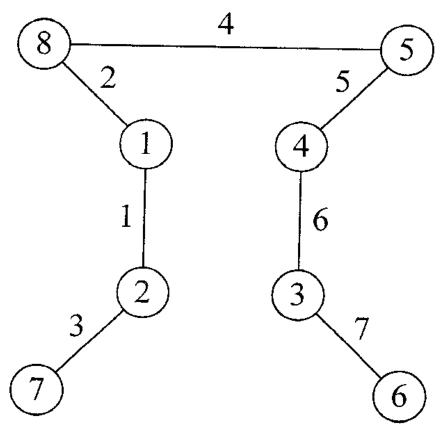


Fig. 4-4

Prim's Algorithm (Matrix Version)

For the sake of simplicity, we assume that the network is connected. Let $V = \{1, 2, \ldots, n\}$ be the vertices of a connected network. If there is an edge joining vertices i and j, let $w(i, j)$ be the weight of this edge, and let $D = [w(i, j)]$ be the matrix in which the (i, j) entry is blank if there is no edge joining i and j. Delete all elements of column 1, and mark row 1 with an asterisk. This means that we are starting with vertex 1. So $W = \{1\}$. Initially, all numerical entries in the matrix are uncircled. Each iteration has two steps, as follows.

Step 1. Select a smallest element from the uncircled numerical entries in the rows that are marked with an asterisk. Stop if no such element exists. The edges that correspond to the circled entries constitute an M.S.T.

Step 2. If $w(i, j)$ is selected, circle the (i, j) entry in D and mark row j with an asterisk. Delete the remaining uncircled elements in column j. Go to step 1.

Example 5. Obtain an M.S.T. in the network of Fig. 4-2 using the matrix form of Prim's algorithm.

The initial matrix is

$$\left[\begin{array}{cccccccc} \text{—} & 12 & \text{—} & 16 & \text{—} & \text{—} & \text{—} & 13 \\ \text{—} & \text{—} & 16 & \text{—} & \text{—} & \text{—} & 14 & \text{—} \\ \text{—} & 16 & \text{—} & 12 & \text{—} & 14 & \text{—} & \text{—} \\ \text{—} & \text{—} & 12 & \text{—} & 13 & \text{—} & \text{—} & \text{—} \\ \text{—} & \text{—} & \text{—} & 13 & \text{—} & 14 & \text{—} & 15 \\ \text{—} & \text{—} & 14 & \text{—} & 14 & \text{—} & 15 & \text{—} \\ \text{—} & 14 & \text{—} & \text{—} & \text{—} & 15 & \text{—} & 14 \\ \text{—} & \text{—} & \text{—} & \text{—} & 15 & \text{—} & 14 & \text{—} \end{array}\right]\begin{array}{l} * \\ \\ \\ \\ \\ \\ \\ \\ \end{array}$$

All the entries in column 1 are deleted and row 1 is marked with an asterisk.

Iteration 1: The smallest uncircled entry in the marked row is $w(1, 2) = 12$. This entry is circled, the remaining entries in column 2 are deleted, and row 2 is marked with an asterisk:

$$\left[\begin{array}{cccccccc} \text{—} & 12 & \text{—} & 16 & \text{—} & \text{—} & \text{—} & 13 \\ \text{—} & \text{—} & 16 & \text{—} & \text{—} & \text{—} & 14 & \text{—} \\ \text{—} & \text{—} & \text{—} & 12 & \text{—} & 14 & \text{—} & \text{—} \\ \text{—} & \text{—} & 12 & \text{—} & 13 & \text{—} & \text{—} & \text{—} \\ \text{—} & \text{—} & \text{—} & 13 & \text{—} & 14 & \text{—} & 15 \\ \text{—} & \text{—} & 14 & \text{—} & 14 & \text{—} & 15 & \text{—} \\ \text{—} & \text{—} & \text{—} & \text{—} & \text{—} & 15 & \text{—} & 14 \\ \text{—} & \text{—} & \text{—} & \text{—} & 15 & \text{—} & 14 & \text{—} \end{array}\right]\begin{array}{l} * \\ * \\ \\ \\ \\ \\ \\ \\ \end{array}$$

Iteration 2: The smallest uncircled entry in the marked rows is $w(1, 8) = 13$. This element is circled, the remaining entries in column 8 are deleted, and row 8 is marked with an asterisk:

$$\left[\begin{array}{cccccccc} \text{—} & 12 & \text{—} & 16 & \text{—} & \text{—} & \text{—} & 13 \\ \text{—} & \text{—} & 16 & \text{—} & \text{—} & \text{—} & 14 & \text{—} \\ \text{—} & \text{—} & \text{—} & 12 & \text{—} & 14 & \text{—} & \text{—} \\ \text{—} & \text{—} & 12 & \text{—} & 13 & \text{—} & \text{—} & \text{—} \\ \text{—} & \text{—} & \text{—} & 13 & \text{—} & 14 & \text{—} & \text{—} \\ \text{—} & \text{—} & 14 & \text{—} & 14 & \text{—} & 15 & \text{—} \\ \text{—} & \text{—} & \text{—} & \text{—} & \text{—} & 15 & \text{—} & \text{—} \\ \text{—} & \text{—} & \text{—} & \text{—} & 15 & \text{—} & 14 & \text{—} \end{array}\right]\begin{array}{l} * \\ * \\ \\ \\ \\ \\ \\ * \end{array}$$

Iteration 3: The smallest uncircled entry in the marked rows is $w(2, 7) = 14$. This element is circled, the remaining entries in column 7 are deleted, and row 7 is marked with an asterisk:

$$\left[\begin{array}{cccccccc} \text{—} & 12 & \text{—} & 16 & \text{—} & \text{—} & \text{—} & 13 \\ \text{—} & \text{—} & 16 & \text{—} & \text{—} & \text{—} & 14 & \text{—} \\ \text{—} & \text{—} & \text{—} & 12 & \text{—} & 14 & \text{—} & \text{—} \\ \text{—} & \text{—} & 12 & \text{—} & 13 & \text{—} & \text{—} & \text{—} \\ \text{—} & \text{—} & \text{—} & 13 & \text{—} & 14 & \text{—} & \text{—} \\ \text{—} & \text{—} & 14 & \text{—} & 14 & \text{—} & \text{—} & \text{—} \\ \text{—} & \text{—} & \text{—} & \text{—} & \text{—} & 15 & \text{—} & \text{—} \\ \text{—} & \text{—} & \text{—} & \text{—} & 15 & \text{—} & \text{—} & \text{—} \end{array}\right]\begin{array}{l} * \\ * \\ \\ \\ \\ \\ * \\ * \end{array}$$

Iteration 4: The smallest uncircled entry in the marked rows is $w(7, 6) = 15$. This element is circled, the remaining entries in column 6 are deleted, and row 6 is marked with an asterisk:

$$\begin{array}{cc}
\begin{bmatrix}
- & 12 & - & 16 & - & - & - & 13 \\
- & - & 16 & - & - & - & 14 & - \\
- & - & - & 12 & - & - & - & - \\
- & - & 12 & - & 13 & - & - & - \\
- & - & - & 13 & - & - & - & - \\
- & - & 14 & - & 14 & - & - & - \\
- & - & - & - & - & 15 & - & - \\
- & - & - & - & 15 & - & - & -
\end{bmatrix} &
\begin{array}{c} * \\ * \\ \\ \\ \\ * \\ * \\ * \end{array}
\end{array}$$

Iteration 5: The smallest uncircled entry in the marked rows is $w(6, 5) = 14$. This entry is circled, the remaining entries in column 5 are deleted, and row 5 is marked with an asterisk:

$$\begin{array}{cc}
\begin{bmatrix}
- & 12 & - & 16 & - & - & - & 13 \\
- & - & 16 & - & - & - & 14 & - \\
- & - & - & 12 & - & - & - & - \\
- & - & 12 & - & - & - & - & - \\
- & - & - & 13 & - & - & - & - \\
- & - & 14 & - & 14 & - & - & - \\
- & - & - & - & - & 15 & - & - \\
- & - & - & - & - & - & - & -
\end{bmatrix} &
\begin{array}{c} * \\ * \\ \\ \\ * \\ * \\ * \\ * \end{array}
\end{array}$$

Iteration 6: The smallest uncircled entry in the marked rows is $w(5, 4) = 13$. This entry is circled, the remaining entries in column 4 are deleted, and row 4 is marked with an asterisk:

$$\begin{array}{cc}
\begin{bmatrix}
- & 12 & - & - & - & - & - & 13 \\
- & - & 16 & - & - & - & 14 & - \\
- & - & - & - & - & - & - & - \\
- & - & 12 & - & - & - & - & - \\
- & - & - & 13 & - & - & - & - \\
- & - & 14 & - & 14 & - & - & - \\
- & - & - & - & - & 15 & - & - \\
- & - & - & - & - & - & - & -
\end{bmatrix} &
\begin{array}{c} * \\ * \\ \\ * \\ * \\ * \\ * \\ * \end{array}
\end{array}$$

Iteration 7: The smallest uncircled entry in the marked rows is $w(4, 3) = 12$. This entry is circled, the remaining vertices in column 3 are deleted, and row 3 is marked with an asterisk:

$$\begin{array}{cc}
\begin{bmatrix}
- & 12 & - & - & - & - & - & 13 \\
- & - & - & - & - & - & 14 & - \\
- & - & - & - & - & - & - & - \\
- & - & 12 & - & - & - & - & - \\
- & - & - & 13 & - & - & - & - \\
- & - & - & - & 14 & - & - & - \\
- & - & - & - & - & 15 & - & - \\
- & - & - & - & - & - & - & -
\end{bmatrix} &
\begin{array}{c} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{array}
\end{array}$$

At this stage, all the rows of the matrix are marked, giving $(n - 1)$ circled entries corresponding to the edges of an M.S.T. The weight of the M.S.T. is the sum of the circled entries in this last matrix.

4.2 MAXIMUM WEIGHT BRANCHINGS

A **directed forest** is a digraph whose underlying graph is a forest. A **branching B in a digraph G** is a subgraph of G such that B is a directed forest and that the indegree of each vertex in B is either 0 or 1. The problem of finding a branching of maximum weight in a weighted digraph is known as the **maximum weight branching problem.**

It is easy to see that the greedy approach may not produce a maximum weight branching in a weighted digraph in general. For example, consider the digraph $G = (V, E)$ and $V = \{1, 2, 3, 4\}$ and E the set of arcs $\{(1, 2,)\ (2, 3), (3, 4), (4, 3)\}$ with weights 9, 8, 5, and 10, respectively. The greedy method will select arcs (4, 3) and (1, 2) with a total weight of 19, whereas the optimal solution consists of arcs (1, 2), (2, 3), and (3, 4) with a total weight of 22.

If e is an arc from i to j, the **source** $s(e)$ of e is i and the **terminal** $t(e)$ of e is j. Arc e from i to j is a **critical arc** if its weight is not less than the weight of any other arc whose terminal is also vertex j. If there is more than one critical arc directed to a vertex, we select any one of them. The subgraph consisting of the set of all critical arcs thus chosen in a weighted digraph is a **critical graph** in the digraph. If a critical graph H in a digraph G is acyclic, H is obviously a maximum weight branching in G. In this case, the optimization problem is readily solvable. What is now needed is a procedure to obtain a maximum weight branching in a digraph when it does not have an acyclic critical graph.

Condensing a Weighted Digraph

Let H be a critical graph in G with weight function w, and let the cycles in H be $C_i (i = 1, 2, \ldots, k)$. Also let W be the set of those vertices in G that do not belong to any of the cycles in H. Replace each cycle C_i in H by a single vertex X_i. Let $V_1 = \{X_1, X_2, \ldots, X_k\} \cup W$. If e is an arc in G that is not an arc of C_i and if $t(e)$ is a vertex of C_i, define $w_1(e) = w(e) - w(f) + w(e_i)$, where f is the unique arc in C_i such that $t(e) = t(f)$ and e_i is an arc of minimum weight among all the arcs in that cycle. If $t(e)$ is not a vertex of any of these k cycles, $w_1(e) = w(e)$. The weighted multigraph G_1 thus constructed with V_1 as the set of vertices with the revised weight function w_1 is the **condensed graph** of G. If H_1 is a critical graph of G_1, we have moved from the pair (G, H) to the (G_1, H_1) after condensing G by using H. We continue this condensation process until we reach the pair (G_m, H_m), where H_m is acyclic.

Example 6. Carry out the condensation process in the weighted digraph shown in Fig. 4-5.

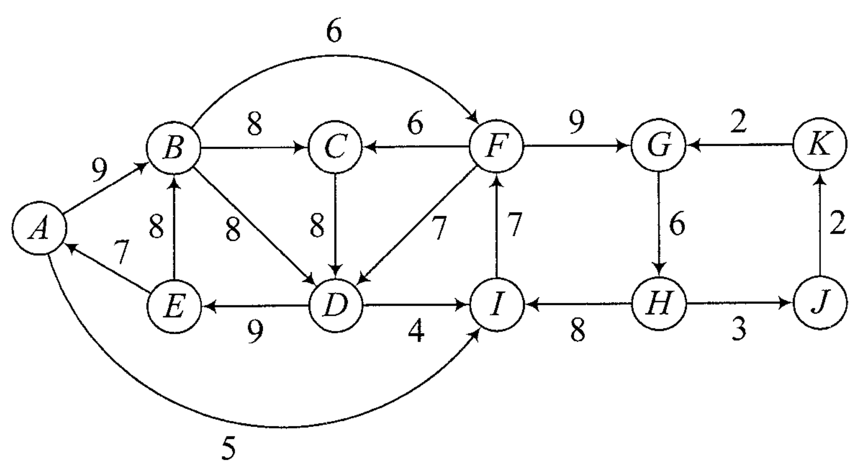


Fig. 4-5

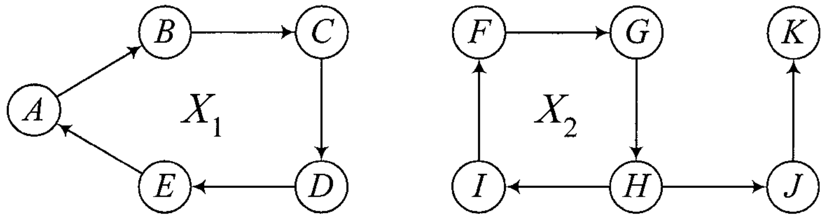


Fig. 4-6

A critical graph H in G is shown in Fig. 4-6 with two disjoint cycles marked X_1 and X_2. If the two cycles are shrunk into two vertices and the weight function is updated, we get the condensed graph G_1 shown in Fig. 4-7.

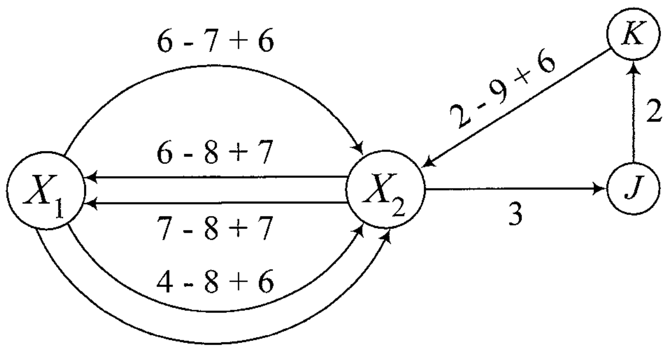


Fig. 4-7

A critical graph H_1 in G_1 is shown in Fig. 4-8. The cycle in H_1 is condensed into a single vertex denoted by X_{12}, as shown in Fig. 4-9, defining the acyclic digraph G_2 in which the critical graph H_2 is acyclic and is the same as G_2.

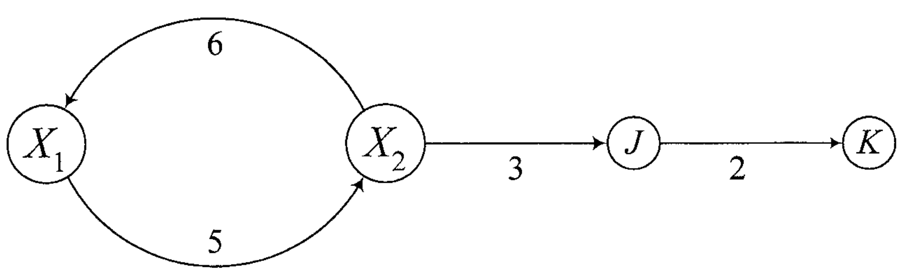


Fig. 4-8

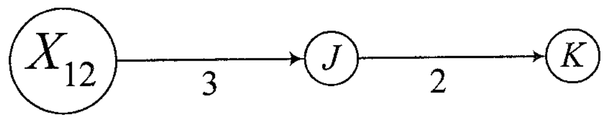


Fig. 4-9

The Maximum Weight Branching Algorithm

Step 1. (The condensation process) The input is the weighted digraph $G = G_0$. Construct G_i from G_{i-1} by condensing the cycles in its critical digraph. G_k is the first digraph in the sequence for which the critical graph H_k is acyclic.

Step 2. (The unraveling process) The graph H_k is a maximum weight branching in G_k. Let $B_k = H_k$. Construct B_{i-1} from B_i by expanding the condensed cycles. B_i is a maximum weight branching in G_i for $i = k, k-1, k-2, \ldots, 1, 0$. The output is $B = B_0$. (See Solved Problem 4.25.)

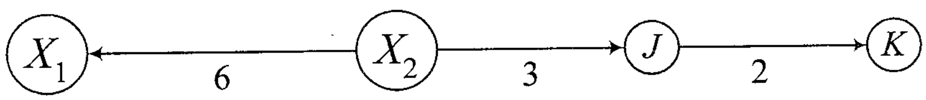


Fig. 4-10

Example 7. Obtain a maximum weight branching for the weighted digraph in Fig. 4-5.

For this example, the condensation process terminates with the digraph G_2, and its critical acyclic graph is as shown in Fig. 4-9. When we open the vertex X_{12}, we obtain the maximum weight branching B_1 in G_1, as shown in Fig. 4-10. At the next stage, we open the vertices X_1 and X_2 to obtain a maximum weight branching B_0 as shown in Fig. 4-11, which is a maximum weight branching in G with a weight of 69.

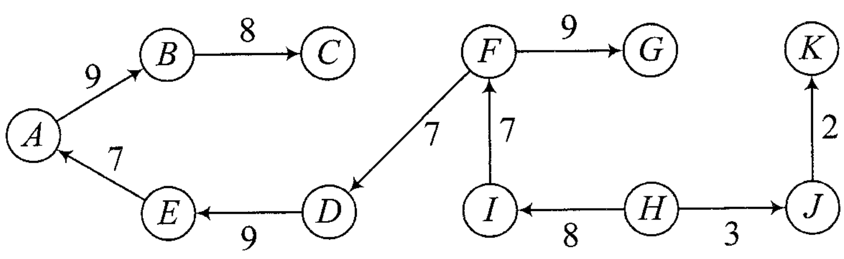


Fig. 4-11

4.3 MINIMUM WEIGHT ARBORESCENCES

A branching $B = (V, F)$ in a digraph $G = (V, E)$ is an **arborescence** if there is exactly one vertex (the **root** of the arborescence) with indegree equal to zero. If a graph is strongly connected, it has an arborescence rooted at every vertex. But strong connectivity is not a necessary condition for the existence of an arborescence in a digraph. For example, consider the digraph G with $V = \{1, 2, 3\}$ and $E = \{(1, 2), (2, 3), (2, 1)\}$. This is not a strongly connected digraph, but is has an arborescence rooted at vertex 1.

A digraph is **quasi-strongly connected** if for every pair of vertices u and v in the digraph there exists a vertex w such that there are directed paths from w to u and from w to v. Strong connectivity implies quasi-strong connectivity. But the converse is not true, as was seen in Example 7. Obviously, any graph that has an arborescence is quasi-strongly connected. The converse also is true. Thus we have the following theorem.

Theorem 4.3. A digraph has an arborescence if and only if it is quasi-strongly connected. (See Solved Problem 4.29.)

Suppose it is known that a weighted digraph has an arborescence with root at vertex r. If we adopt a greedy procedure and choose an arc of minimum weight directed to each vertex other than the root and if we can obtain a subgraph H that is acyclic by this method, H is indeed a minimum weight arborescence in the digraph. Observe that H is acyclic if and only if it is connected. For example, if the weighted digraph is $G = (V, E)$, where $V = \{1, 2, 3, 4\}$ and $E = \{(1, 2), (1, 3), (2, 4), (3, 2), (4, 3)\}$ with weights 3, 5, 4, 6, and 7, respectively, the greedy algorithm will select (1, 2), (2, 4), and (1, 3), giving a minimum weight arborescence of weight 12. If, on the other hand, the weights are 6, 7, 4, 3, and 5, respectively, the resulting digraph is not acyclic and not connected. So we need a procedure to solve the minimum weight arborescence problem when the digraph chosen by the greedy procedure is not acyclic.

Theorem 4.4. Let $T^* = (V, E^*)$ be a minimum weight arborescence rooted at vertex r in a weighted digraph $G = (V, E)$ with distinct arc weights, and let $H = (V, E')$ be the subgraph obtained from G by choosing the arc of minimum weight directed to each vertex other than r. Then $|(C - E^*)| = 1$ for every cycle C in H. (See Solved Problem 4.30.)

Condensing a Weighted Digraph

Let $G = (V, E)$, T^*, and $H = (V, E')$ be as in Theorem 4.4. Suppose H is not acyclic and W is the set of vertices in a cycle C of H. Consider the digraph G_1 (probably with multiple arcs) obtained from G by condensing the vertices in W to a single vertex X. The vertex set of this digraph is $V_1 = (V - W) \cup X$, and E_1 is the set of arcs. Each arc in E_1 corresponds to a unique arc in E; therefore, E_1 can be considered a subset of E. Then $T_1^* = (V_1, E^* \cap E_1)$ is an arborescence in G_1, and $w(T^*) = w(T_1^*) - w(C) + w(e')$. We now define a weight function w_1 on set E_1 as follows. If e is an arc that is not directed to a vertex in the cycle C, $w_1(e) = w(e)$. Otherwise, $w_1(e) = w(e) - w(e')$. If there is more than one arc from a vertex to another vertex in this condensed graph, replace each by a single arc of minimum weight where tiebreaking is arbitrary. Then $w_1(T_1^*) = w(T_1^*) - w(e')$. So $w(T^*) = w(T_1^*) + w(C) - w(e') = w_1(T_1^*) + w(C)$. Thus T^* is a minimum weight arborescence in G if and only if T_1^* is a minimum weight arborescence in the condensed graph G/C.

The Minimum Weight Arborescence Algorithm

(*a*) Step $i = 0$. $G^0 = G$, $w^0(e) = w(e)$ for each i in G.

(*b*) At step i, using the weight function w^i, construct the subgraph H^i of G^i by selecting the arc of smallest weight directed to every vertex other than the root of G^i.

(*c*) If there is no cycle in H^i, it is a minimum weight arborescence in G^i from which a minimum weight arborescence in G^0 can be derived. Otherwise, go to (*d*).

(*d*) If H^i has a cycle C, define $G^{i+1} = G^i/C$ and $w^{i+1}(e) = w^i(e)$ when e is not directed to a vertex in the cycle and $w^{i+1}(e) = w^i(e) - w^i(e')$ when $e = (i, j)$ is directed to a vertex j in C and $e' = (k, j)$ is an arc of the cycle. If H^i contains cycles $C_1, C_2, \ldots$, first condense G^i with respect to C_1, then condense G^i/C_1 with respect to C_2, and so on. Set $i = i + 1$ and return to (*b*).

[If we assume that at each step of the algorithm, all arcs of the current digraph have different weights, the minimum weight arborescence is unique. When there are arcs of equal weight, the algorithm remains valid since the difference $w(T^*) - w(T_1^*)$ is always equal to the weight of the cycle C. If C has two arcs $e' = (p, q)$ and $f' = (r, s)$ that are not in E^*, arcs e and f are directed to q and s, respectively. Then $w_1(e) = w(e) - w(e')$ and $w_1(f) = w(f) - w(f')$.]

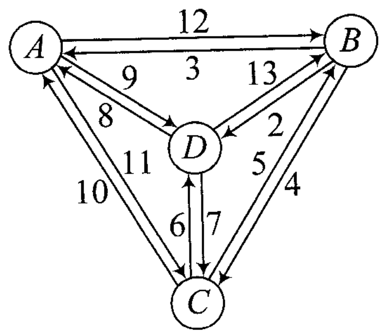


Fig. 4-12

Example 8. Obtain a minimum weight arborescence with root at vertex A in the weighted digraph of Fig. 4-12.

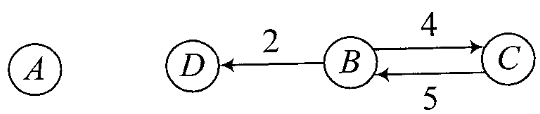


Fig. 4-13

Step 0. The arcs of H^0 are (B, D), (B, C), and (C, B), as shown in Fig. 4-13. Since (B, C) and (C, B) form a cycle, the two vertices are condensed into a single vertex BC. At this stage, we construct the condensed graph G^1 with vertices as shown in Fig. 4-14.

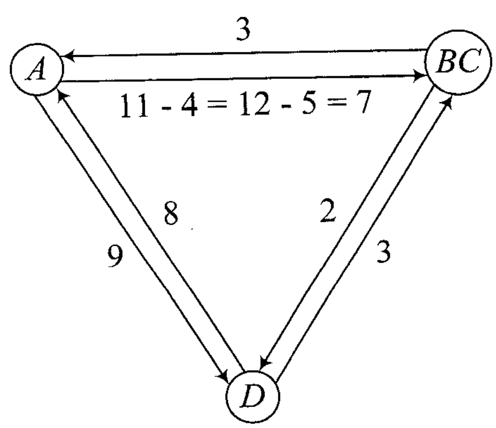


Fig. 4-14

Step 1. The digraph H^1 is shown in Fig. 4-15. Vertices D and BC constitute a cycle, and this cycle is shrunk into a single vertex BCD, forming the digraph G^2 as shown in Fig. 4-16.

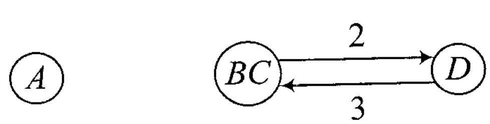


Fig. 4-15

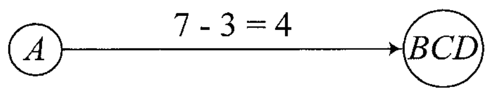


Fig. 4-16

Step 2. At this stage, we have a minimum weight arborescence in the condensed graph. We now derive a minimum weight arborescence in G by working backward from the current graph G^2. Since the weight of arc (A, BCD) is $7 - 3 = 4$, we locate the arc from A with weight 7 in G^1. Thus the minimum weight arborescence in G^1 is $A \to BC \to D$. The weight of arc (A, BC) is 7, which is either $11 - 4$ or $12 - 5$, indicating a tie. In the former case, we take (A, C) and (C, B). In the latter case, we take (A, B) and (B, C). In G^1, the weight of the arc from BC to D is 2, which is the weight of the arc from B to D in G. So the arc (B, D) is in the minimum weight arborescence.

Thus there are two minimum weight arborescences with root at A, as shown in Fig. 4-17.

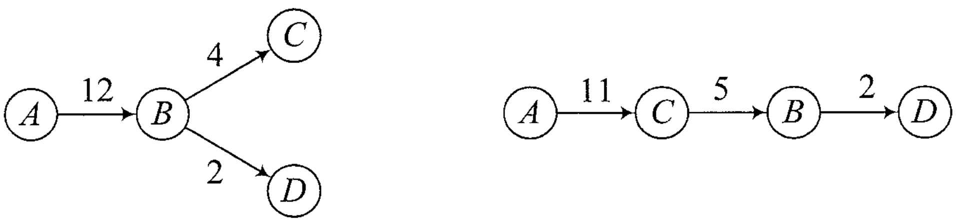


Fig. 4-17

Arborescence in a Quasi-connected Graph

Let $G = (V, E)$ be a quasi-connected weighted digraph. Construct a new vertex r, and draw arcs from r to each vertex in G. Let each of these new arcs be assigned a weight W that is some number larger than the weight of the maximum weight arc in G. The enlarged digraph G' is quasi-strongly connected and has an arborescence rooted at the vertex r. We can obtain a minimum weight arborescence T' in G' with root at r using the algorithm. By deleting the new vertex from T', we obtain a minimum weight arborescence T in G.

4.4 MATROIDS AND THE GREEDY ALGORITHM

If E is a finite set and $\mathscr{A}$ is a class of subsets of E, the pair $(E, \mathscr{A})$ is called an **independent system** if $A \in \mathscr{A}$ whenever $A \subset B$ and $B \in \mathscr{A}$. The set E is the **ground set** of the independent system. The sets in $\mathscr{A}$ are the **independent sets,** and a subset of E that is not independent is called a **dependent set** of the system. A dependent set C is called a **circuit** if every proper subset of C is independent. An independent set I that is a subset of $A \subset E$ is **maximal in** A if $I \subset J \subset A$ implies that $I = J$ whenever J is an independent set. An independent system is a **matroid** if any two sets maximal in a set have the same cardinality. An independent set in a matroid $(E, \mathscr{A})$ that is maximal in E is a **base** of the matroid. The matroid in which the only independent sets are the ground set and the empty set is the **trivial matroid,** whereas the matroid in which every subset of the ground set is an independent set is the **discrete matroid.** If E is a set with n elements, the k**-uniform matroid** with E as the ground set is the matroid in which a subset I of E is independent if and only if it has at most k elements, where $k \leq n$.

If H is a subgraph of an undirected graph $G = (V, E)$, any two spanning forests in H will have the same number of edges. So if $\mathscr{A} = \{A : A \subset E, A \text{ is the set of edges of an acyclic subgraph of } G\}$, the pair $(E, \mathscr{A})$ is a matroid and is the **circuit matroid** defined by G. Thus every undirected graph can be considered a matroid. The set of edges belonging to any spanning forest of graph G is a base of this matroid. Furthermore, a circuit in this matroid is precisely the set of edges belonging to a cycle in the graph.

Theorem 4.5. An independent system is a matroid if and only if whenever I and J are two independent sets with the property that J has more elements than I, there exists an element $e \in (J - I)$ such that $I \cup \{e\}$ is an independent set. (See Solved Problem 4.33.)

As a consequence of this theorem, an **alternate definition of a matroid** can be given as follows. If $\mathscr{A}$ is a class of subsets of a finite set E, the pair $(E, \mathscr{A})$ is called a matroid if (*i*) $I \in \mathscr{A}$ whenever $I \subset J$ and $J \in \mathscr{A}$ and (*ii*) I and J are any two sets in $\mathscr{A}$ such that J has more elements than I. Then there exists an element $e \in (J - I)$ such that $I \cup \{e\}$ is an independent set.

The cardinality of an independent set in a matroid that is maximal in a set A is the **rank** of A, denoted by $r(A)$. Consequently, the **rank function r of a matroid** is a function whose domain is the power set of E and whose range is a subset of the set of nonnegative integers. The **rank of the matroid** is the rank of its ground set.

Theorem 4.6. The rank function r of a matroid satisfies the following four properties: (*i*) the rank of the empty set is 0, (*ii*) $r(A) \leq r(B)$ if $A \subset B$, (*iii*) the function r is **submodular:** $r(A \cup B) + r(A \cap B) \leq r(A) + r(B)$, and (*iv*) the rank of a singleton set is either 0 or 1. (See Solved Problem 4.34.)

The converse of Theorem 4.6 is also true, which enables us to define a matroid in terms of a rank function.

Theorem 4.7. If r is an integer-valued function defined on the power set of a finite set E satisfying the four properties listed in Theorem 4.6, and if $\mathscr{A} = \{A : A \subseteq E, r(A) = |A|\}$, the pair $(E, \mathscr{A})$ is a matroid. (See Solved Problem 4.35.)

An Optimization Problem in a Weighted Matroid

Let w be a nonnegative weight function defined on the ground set E of an independent system. If A is a subset of E, the **weight of A,** denoted by $w(A)$, is the sum of the weights of all the elements in A. An optimization problem associated with the independent system is the problem of finding an independent set with maximum weight. A greedy approach to solve this optimization problem (reminiscent of the algorithms of Kruskal and Prim) will have the following two steps at iteration k:

Step 1. Choose $x(k)$ distinct from $x(1), x(2), \ldots, x(k-1)$ such that (*i*) the set $\{x(1), x(2), \ldots, x(k-1), x(k)\}$ is an independent set; and (*ii*) if $\{x(1), x(2), \ldots, x(k-1), x\}$ is an independent set, the weight of x does not exceed the weight of $x(k)$.

Step 2. Stop if no such $x(k)$ exists.

In the case of an arbitrary independent system, this greedy approach may not produce an optimal solution, as in the following simple example. Let $E = \{a, b, c, d\}$ with weights 2, 6, 7, and 5, respectively. Suppose the independent sets are $\{a\}, \{b\}, \{c\}, \{a, b\}$, and the empty set. The greedy algorithm will choose the independent set $\{c\}$, whereas the optimal solution is the set $\{a, b\}$. On the other hand, if the independent sets are $\{a), \{c\}$, $\{a, c\}$, and the empty set, the greedy algorithm will pick the optimal solution. Observe that in the former case (when the greedy method failed), the independent system is not a matroid, whereas in the latter case, the independent system is a matroid.

We are now ready to establish the connection between the greedy algorithm and the structure of matroids.

Theorem 4.8. A solution of the problem of finding a maximum weight independent set in an independent system can be obtained by using the greedy algorithm for every nonnegative weight function defined on its ground set if and only if the independent system is a matroid. (See Solved Problem 4.36.)

As a consequence of Theorem 4.8, we once again conclude that the greedy methods of Kruskal and Prim to obtain a minimum spanning tree in a connected graph will correctly solve the optimization problem because (1) the underlying structure is a matroid, and (2) the problem of finding a minimum weight spanning tree is equivalent to that of finding a maximum weight spanning tree.

Solved Problems

MINIMUM WEIGHT SPANNING TREES

4.1 Show that if the vertex set of a connected graph $G = (V, E)$ is partitioned into two nonempty sets X and Y, the disconnecting set $F = (X, Y)$ consisting of all edges of G joining vertices in X and vertices in Y is a cut set if the subgraph $G' = (V, E - F)$ has exactly two components.

Solution. Suppose $F = (X, Y)$ is not a cut set. Then there is a proper subset F' of F that is a cut set. Let $e = \{x, y\}$ in an edge in $F - F'$ joining vertex x in X and vertex y in Y. If v is any vertex in X, there is a path joining v and x consisting of vertices from X only. Likewise, if w is any vertex in Y, there is a path joining w and y consisting of vertices from Y only. Thus the deletion of all the edges belonging to F' from the graph will not make it disconnected. This is a contradiction since F' is a disconnecting set.

4.2 The deletion of an edge belonging to a spanning tree $T = (V, F)$ of a graph $G = (V, E)$ from F defines a partition of V into two subsets X and Y, creating a disconnecting set $D = \{X, Y\}$ of the graph. Show that this disconnecting set is a cut set of G.

Solution. Since every edge in a tree is a bridge, the deletion of an edge from a spanning tree gives two sets X and Y that constitute a partition of V. So $D = \{X, Y\}$ is a disconnecting set in G. It is a cut set since $G' = (V, E - D)$ has exactly two components.

4.3 If T is a spanning tree in a graph G, the cut set of G formed by deleting an edge e of T, denoted by $D_T(e)$, is called the **fundamental cut set** of G with respect to T relative to edge e. Find the number of cut sets of a connected graph with n vertices with respect to a spanning tree.

Solution. Since the graph has n vertices, any spanning tree in the graph will have $(n - 1)$ edges. The deletion of each edge defines a fundamental cut set. So there are $(n - 1)$ fundamental cut sets of the graph with respect to a spanning tree.

4.4 If T is a spanning tree in G, any edge $e = \{x, y\}$ in G that is not an edge in T is called a **chord** of T. The unique cycle in G formed by the path in T joining x and y and edge e, denoted by $C_T(e)$, is called the **fundamental cycle** of G with respect to T relative to edge e. Find the number of fundamental cycles of a connected graph with respect to a spanning tree.

Solution. If G is a connected graph of order n and size m, it has $r = m - (n - 1)$ chords with respect to any spanning tree T, so it has r fundamental cycles with respect to T.

4.5 Let G be a connected graph in which C is a cycle, D is a cut set, and T is a spanning tree. Show that (*a*) the number of edges common to C and D is even, (*b*) at least one edge of C is a chord of T, and (*c*) at least one edge of D is an edge of T.

Solution.

(*a*) Let $D = \{X, Y\}$. If all the vertices of C are exclusively in X (or in Y), C and D have no edges in common. Suppose C has two vertices x and y, where x is in X and y is in Y. Then the cycle C that starts from x and ends in x will contain the edges from D an even number of times.

(*b*) If no edge of C is a chord of T, then every edge of C is an edge of T, which is a contradiction since T is acyclic.

(*c*) If no edge of D is an edge of T, the deletion of all the edges belonging to D will not affect T, implying that G continues to be a connected even after all the edges belonging to D are deleted. This is a contradiction since D is a cut set.

4.6 Let e be an edge of a spanning tree T in a graph G. Show that (a) if f is any edge (other than e) in the fundamental cut set $D_T(e)$, f is a chord of T and e is an edge of the fundamental cycle $C_T(f)$; and (b) e is not an edge of the fundamental cycle $C_T(e')$ relative to any chord e' that is not an edge in $D_T(e)$.

Solution.

(a) An edge f in the cut set cannot be an edge of the spanning tree T since T is acyclic. So f is necessarily a chord of T. Now $D_T(e) = \{e, f\} \cup A$ and $C_T(f) = \{f\} \cup B$, where A is a set of chords of T and B is a set of edges of T. Since the intersection of A and B is empty and since the intersection of $D_T(e)$ and $C_T(f)$ has an even number of elements, edge e should be an element of B. Hence, e is an edge of $C_T(f)$.

(b) The set of edges of $C_T(e')$ is the union of $\{e'\}$ and a set X of edges of T. If e is an edge of $C_T(e')$, e is an element of X. The set of edges of $D_T(e)$ is the union of $\{e\}$ and a set Y of chords of T. Since the intersection of $D_T(e)$ and $C_T(e')$ has an even number of edges and since X and Y have no elements in common, edge e' is an element of $D_T(e)$, which contradicts the hypothesis.

4.7 Let e be a chord of a spanning tree T in a graph G. Show that (a) if f is any edge (other than e) in the fundamental cycle $C_T(e)$, f is an edge of T and e is an edge of the fundamental cut set $D_T(f)$; and (b) e is not an edge of the fundamental cut set $D_T(e')$ relative to any edge e' of T that is not an edge in $C_T(e)$.

Solution.

(a) By definition, any edge f of $C_T(e)$ other than e is an edge of T. Now $D_T(f)$ is the union of the set $\{f\}$ and a set A of chords of T. At the same time, $C_T(e)$ is the union of $\{e, f\}$ and a set B of edges of T. So e is an element of A; therefore, it is an element of $D_T(f)$.

(b) $C_T(e)$ is the union of $\{e\}$ and a set X of edges of T. If e' is in $D_T(e')$, $D_T(e')$ is the union of $\{e, e'\}$ and a set Y of chords of T. This implies that e' is an element of $C_T(e)$, which contradicts the hypothesis.

4.8 Show that if T and T' are two spanning trees in G and if e is an edge of T and a chord of T', there exists an edge e' in T' such that e' is a chord of T and $T - e + e'$ is a spanning tree in G.

Solution. If $e = \{x, y\}$ is deleted from T, the set of vertices of G is partitioned into two sets X and Y such that x is in X and y is in Y. Since e is a chord of T', there will be a path in T' between x and y that will have edge $e' = \{p, q\}$ joining vertex p in X and vertex q in Y. This edge e' is necessarily a chord of T. Then $T - e + e'$ is obviously a spanning tree in G. At the same time, $T' - e' + e$ is also a spanning tree.

4.9 If no two edge weights of a connected graph G are equal, show that G has a unique minimum spanning tree.

Solution. Suppose there are two minimum spanning trees, T and T'. Then $w(T) = w(T') = s$. As shown in Problem 4.8, there are two edges e and f such that $T_1 = T - e + f$ and $T_2 = T' - f + e$ are both spanning trees in G. Now $w(T_1) = s - w(e) + w(f)$, and $w(T_2) = s - w(f) + w(f)$. If $w(e) > w(f)$, $w(T_1) < s$, which is a contradiction since T_1 is an M.S.T. If $w(e) < w(f)$, $w(T_2) < s$, which is also a contradiction. So there is only one M.S.T.

4.10 Show that if a connected weighted graph G contains a unique edge e of minimum weight, e is an edge of every M.S.T. of G.

Solution. Suppose T is an M.S.T. of G and e is not an edge of T. Let f be any edge of the fundamental cycle $C_T(e)$ other than e. Then $T' = T - f + e$ is spanning tree of G, and $w(T') = w(T) - w(f) + w(e)$. Since $w(e) < w(f)$, $w(T') < w(T)$, and this contradicts the assumption that T is an M.S.T.

4.11 Show that a spanning tree T in a weighted graph is a minimum spanning tree if and only if every edge of T is a minimum weight edge in the fundamental cut set relative to that edge.

Solution. Let e be an edge of an M.S.T. in a weighted graph G. Suppose $w(f) < w(e)$ for some chord of T belonging to the fundamental cut set $D_T(e)$. Then $T' = T - e + f$ with $w(T') < w(T)$, which is a contradiction. So the condition is necessary. On the other hand, suppose T is a spanning tree that satisfies the given condition. Let T' be an M.S.T. of G. If T and T' are not the same, let $e_1 = \{i, j\}$ be an edge of T that is not an edge of T'. Since e_1 is a chord of T', we have the fundamental cycle $C_{T'}(e_1)$ that contains edge f_1 (other than e_1), an edge belonging to $D_T(e_1)$. Let $T'' = T' - f_1 + e_1$. By assumption, $w(e_1) \leq w(f_1)$. So $w(T'') \leq w(T)$. If $w(e_1) < w(f_1)$, $w(T'') < w(T')$, which is a contradiction. So $w(e_1) = w(f_1)$. Thus $T_1 = T - e_1 + f_1$ is a spanning tree of the graph such that $w(T) = w(T_1)$. If $T_1 = T'$, T is an M.S.T. Otherwise, we consider edge e_2 of T_1, which is not an edge of T'. This edge e_2 is an edge of T and a chord of T'. We proceed as before. Obtain edge f_2 in $D_T(e_2)$, and construct the spanning tree $T_2 = T_1 - e_2 + f_2$ such that $w(T_2) = w(T_1) = w(T)$. If $T_2 = T'$, T is an M.S.T. Otherwise, we continue this process until we get a spanning tree T_k such that $T_k = T'$.

4.12 Show that if in a weighted connected graph G there is an edge e that is a maximum weight edge in any cycle that contains e, there is an M.S.T. in G that does not contain e. In particular, if the weight of e exceeds the weight of every edge in any cycle that contains e, then no M.S.T. in G contains e as an edge.

Solution. Let e be an edge in a weighted connected graph such that $w(e) \geq w(f)$ for any cycle in the graph in which both e and f are edges. Let T be an M.S.T. in G. If e is not an edge of T, we are done. Suppose e is an edge of T. Then $w(e) \leq w(e')$ for any e' in the fundamental cut set $D_T(e)$, as proved in Problem 4.11. Since e' is a chord, the fundamental cycle $C_T(e')$ contains e as an edge, as proved in Problem 4.6. So by hypothesis, $w(e) \geq w(e')$. Thus $w(e) = w(e')$. Consequently, $T' = T - e + e'$ is an M.S.T. that does not contain e. Next we show that if $w(e) > w(f)$ for any cycle in which both e and f are edges, no M.S.T. contains e. Suppose this is not the case. Then there is a minimum spanning tree T that contains e. Proceeding as before, we get a spanning tree $T' = T - e + e'$ such that $w(T') < w(T)$, which is a contradiction.

4.13 A spanning tree T in a graph G is a minimum spanning tree if and only if every edge not in T is a maximum weight edge in the fundamental cycle defined by that edge.

Solution. Suppose there is a chord of the minimum spanning tree T such that in the fundamental cycle $C_T(e)$ there exists an edge f violating the maximality condition. In other words, $w(e) < w(f)$. Then $T' = T - f + e$ is an M.S.T. whose weight is less than $w(T)$, which is a contradiction. So the condition is necessary. To prove that the condition is sufficient, suppose T is a spanning tree satisfying the given condition. If f is an edge in T, any chord e of T belonging to the fundamental cut set $D_T(f)$ will define a fundamental cycle $C_T(e)$ in which f is an edge. By assumption, $w(e) \geq w(f)$. In other words, whenever f is an edge of T, we have the inequality $w(e) \leq w(f)$, where e is any edge in $D_T(f)$. So T is an M.S.T. because of the condition established in Problem 4.11.

4.14 Prove Theorem 4.1: Kruskal's algorithm solves the M.S.T. problem in a network.

Solution. In this algorithm, an edge e is discarded in favor of an edge of larger weight only because the inclusion of e would have created a cycle C. At the same time, the weight of the discarded edge is greater than or equal to the weight of any other edge in C. So by the optimality condition established in Problem 4.13, if a spanning tree is obtained by this method, it has to be a minimum spanning tree.

4.15 If e is an edge incident at vertex x of a connected weighted graph G and if $w(e) \leq w(f)$ for every edge f incident at x, there exists an M.S.T. of G that contains e as an edge. In particular, if $w(e) < w(f)$, every M.S.T. of G contains e.

Solution. Let T be an M.S.T. If $e = \{x, y\}$ is an edge of T, we are done. Let e be a chord of T. According to the optimality condition in Problem 4.11, $w(e) \geq w(f)$, where f is any edge in the fundamental cycle $C_T(e)$. Suppose f is an edge in $C_T(e)$ that is adjacent to x or y. Then, by hypothesis, $w(e) \leq w(f)$. Hence, $T' = T - f + e$ is also an M.S.T. In particular, if $w(e) < w(f)$, every M.S.T. of G contains e.

4.16 If W is the set of vertices of any subgraph H of a minimal spanning tree T of a graph $G = (V, E)$ and if e is any edge of minimum weight in the disconnecting set $D = (W, V - W)$, there exists an M.S.T. that contains e as an edge and that has H as a subgraph.

Solution. If e is an edge of T, there is nothing to prove. So assume that T is a chord of T defining the fundamental cycle $C_T(e)$. Then $e \in C_T(e) \cap D$. So there should be an edge f of T (other than e) in $C_T(e) \cap D$. By hypothesis, $w(e) \leq w(f)$. The edge f of T defines the fundamental cut set $D_T(f)$. Thus $f \in C_T(e) \cap D_T(f)$. There should be at least one more edge in $C_T(e) \cap D_T(f)$. The only choice is for e. Since e is in $D_T(f)$ and since T is an M.S.T., $w(f) \geq w(e)$. Thus $w(f) = w(e)$. Let $T' = T - f + e$. Then T' is an M.S.T. that contains e as an edge and that has H as a subgraph.

4.17 Prove Theorem 4.2: Prim's algorithm solves the M.S.T. problem.

Solution. In this algorithm, we start from an arbitrary vertex and add edges one at a time. We maintain a spanning tree T on a set W of vertices of a graph $G = (V, E)$ such that the edge adjoining to T is a minimum weight edge in the disconnecting set $(W, V - W)$. The correctness of the procedure follows immediately from the result established in Problem 4.16.

4.18 Obtain a minimum spanning tree in the graph shown in Fig. 4-18.

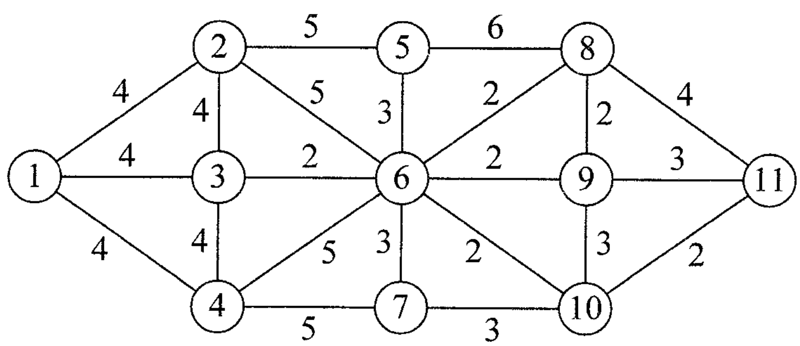


Fig. 4-18

Solution. A minimum spanning tree of weight 28 is displayed in Fig. 4-19.

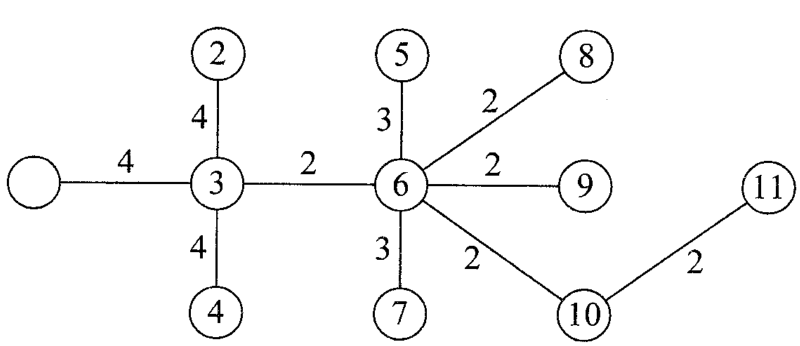


Fig. 4-19

4.19 Suppose we are using Prim's algorithm to obtain an M.S.T. in the graph of Fig. 4-18. At the current stage, $W = \{3, 6, 8, 9\}$ is the set of vertices of the tree H, which is a subtree of a minimum spanning tree T to be obtained by this method. Select the next edge for inclusion, and list the edges after this selection is made.

Solution. The edges of H are $\{3, 6\}$, $\{6, 8\}$, and either $\{6, 9\}$ or $\{8, 9\}$. The edge to be selected is $\{6, 10\}$ in either case.

4.20 Obtain a maximum weight spanning tree in the graph of Fig. 4-18.

Solution. A maximum weight spanning tree of weight 44 is the tree shown in Fig. 4-20.

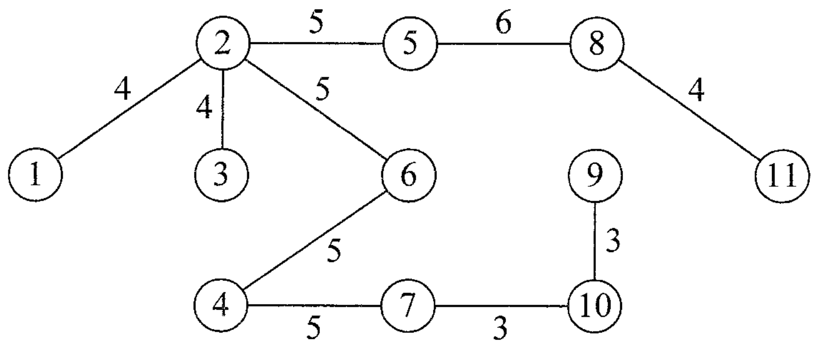


Fig. 4-20

MAXIMUM WEIGHT BRANCHINGS

4.21 Let $G = (V, E)$ be a weighted digraph, and let $B = (V', E')$ be a branching in G. An arc $e \in (E - E')$ is called a B-eligible arc if the arcs in $E'' = E' + e - \{f \in E': t(e) = t(f)\}$ constitute a branching. Show that e is B-eligible if and only if there is no directed path in B from $t(e)$ to $s(e)$.

Solution. The set E'' will not constitute a branching if and only if it contains a cycle. Any such cycle C contains arc e, and the arcs in $(C - e)$ form a directed path from $t(e)$ to $s(e)$. Thus e is B-eligible if and only if there is no directed path in B from $t(e)$ to $s(e)$. (Notice that eligibility is on an individual basis. If e is B-eligible and if f is B-eligible, it is not necessary that they both be B-eligible at the same time.)

4.22 Let C be a directed cycle in $G = (V, E)$, and let $B = (V', E')$ be a branching in G. Show that no arc in $(C - E')$ is B-eligible if and only if $(C - E')$ contains exactly one arc.

Solution. The set $(C - E')$ is obviously nonempty. If it contains the unique arc e, there is a directed path in B from $t(e)$ and $s(e)$. So, by Problem 4.21, the arc e is not B-eligible. To prove the converse, suppose $(C - E') = \{e_i: i = 1, 2, \ldots, k\}$. Assume that these arcs appear clockwise in the cycle such that e_{i+1} follows e_i with no other arc from the set $(C - E')$ in this cycle between these two arcs. Hence, $t(e_{i-1}) = s(e_i)$, or there is a path in $C \cap E'$ from $t(e_{i-1})$ to $s(e_i)$. Also, $t(e_k) = s(e_1)$, or there is a path in $C \cap E'$ from $t(e_k)$ to $s(e_1)$. Suppose no arc in $(C - E')$ is B-eligible. Then there is a directed path in E' from $t(e_i)$ to $s(e_i)$ for each i. Since there is a directed path in $C \cap E'$ from $t(e_{i-1})$ to $s(e_i)$ and since E' is a branching, either there is a directed path P: $t(e_{i-1}) \to t(e_i) \to s(e_i)$ in E' or the directed path Q: $t(e_i) \to t(e_{i-1}) \to s(e_i)$ in E'. If path P exists, it has to be the same as the unique path in the branching from $t(e_{i-1})$ to $s(e_i)$, which is completely in C. So the subpath $t(e_{i-1}) \to t(e_i)$ should also be in cycle C, implying that there should be an arc in $C \cap E'$ other than arc e_i directed to vertex $t(e_i)$. But there cannot be two arcs directed to a vertex in a cycle, so path P does not exist. Suppose path Q exists. This implies that there is a path $t(e_i) \to t(e_{i-1})$ in the branching for $i = 1, 2, \ldots, k$. In that case, we can obtain the cycle $t(e_k) \to t(e_{k-1}) \to \cdots \to t(e_1) \to t(e_k)$ in the branching, which is a contradiction. So Q does not exist, implying that $(C - E')$ cannot have more than one B-eligible arc.

4.23 Show that if H is a critical subgraph of a weighted digraph $G = (V, E)$, there is a maximum weight branching $B = (V, E')$ in G such that for every cycle C in H, the set $(C - E')$ has exactly one arc.

Solution. Let $H = (V, E_1)$ be a critical graph. From the collection of all maximum weight branchings, select the branching $B = (V, E')$ that has the maximum number of arcs in common with E_1. Let e be an arc in $(E_1 - E')$. If e is B-eligible, the arcs in the set $E'' = E' + e - \{f: f \in E' \text{ and } t(e) = t(f)\}$ will form a maximum weight branching that has more arcs in common with H than B. So no arc in H is B-eligible. In particular, no arc in cycle C in H is B-eligible. So the cardinality of $(C - E')$ is one as established in Problem 4.22.

4.24 Show that the cycles in a critical graph H of a weighted digraph $G = (V, E)$ are C_i $(i = 1, 2, \ldots, k)$, there exists a maximum weight branching $B = (V, E')$ of G such that (i) $C_i - E'$ has exactly one arc for each i; and (ii) if no arc in $(E' - C_i)$ is directed to a vertex in the cycle C_i, the unique arc in $(C_i - E')$ is an arc of minimum weight in C_i for each i.

Solution. Since no two cycles have a vertex in common, we see that (i) is a consequence of the result established in Problem 4.23. Suppose e_i is an arc of minimum weight in C_i for each i, and let $S = \{e_i: i = 1, 2, \ldots, k\}$. Choose a maximum weight spanning branching $B = (V, E')$ satisfying property (i) such that E' contains the minimum number of arcs from S. Suppose B does not satisfy (ii). So there is a cycle C_j where (ii) does not hold. No arc of $E' - C_j$ is directed to a vertex in C_j, and e_j is not the arc $C_j - E'$. If the unique arc in $C_j - E'$ is e, $w(e) \geq w(e_j)$. Then $E' - e_j + e$ is a maximum weight branching that has fewer edges from S than E'. This is a contradiction.

4.25 Prove that the maximum weight branching algorithm solves the maximum weight branching problem.

Solution. It is enough if we establish a one-to-one correspondence between the set of all maximum weight branchings satisfying properties (i) and (ii) stated in Problem 4.24 and the set of maximum weight branchings in a condensed graph. Let $G = (V, E)$ be a digraph with a weight function w defined on its arcs, H be a critical graph in G with cycles C_i $(i = 1, 2, \ldots, k)$, and $G_1 = (V_1, E_1)$ be the corresponding condensed graph. A new weight function w_1 is defined on the E_1 as follows.

If e is an arc that is not directed to any of the cycles in H, $w_1(e) = w(e)$. If e is directed to a vertex in C_i, $w_1(e) = w(e) - w(f) + w(e_i)$, where f is the unique arc in C_i that is directed to $t(e)$ and e_i is an arc of minimum weight in the cycle C_i. Let $B = (V, E')$ be any branching in G that satisfies the two properties listed in Problem 4.24 using these k cycles. Arc e in E' such that both $s(e)$ and $t(e)$ are not in the same cycle in H defines a unique arc in A_1. Let D_1 be the set of arcs thus defined. Then $B_1 = (V_1, D_1)$ is a branching in G_1. Thus once a critical graph in G is defined, a branching in G defines a branching in the condensed graph G_1. Now consider the condensed graph defined by a critical graph H. Let the cycles in H be C_i $(i = 1, 2, \ldots, k)$, and let $B_1 = (V_1, A_1)$ be a branching in G_1. If the indegree in B_1 of the condensed vertex corresponding to C_i is 0, let $C_i' = C_i - e_i$. If the indegree is 1, there is a unique arc f (belonging to E) in C_i directed to that condensed vertex. In this case, let $C_i' = C_i - f$. Thus in either case, from each cycle, we take all arcs except one. Let X be the set of arcs thus obtained from these k cycles. Now consider arcs in B_1 that are not directed to condensed vertices. Each such arc corresponds to a unique arc in E. Let Y be the set of arcs thus obtained. The union of X and Y forms a branching in G. Next, we turn our attention to optimality. If P is the sum of the weights of the k cycles and Q is the sum of the weights of the minimum arc weights in these cycles, $w(B) - w_1(B_1) = P - Q$. Thus the one-to-one correspondence is established.

4.26 Show that a maximum weight branching in the network shown in Fig. 4-21 cannot be obtained by the greedy algorithm.

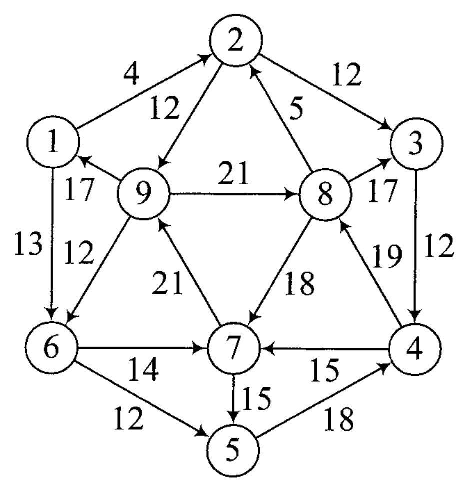


Fig. 4-21

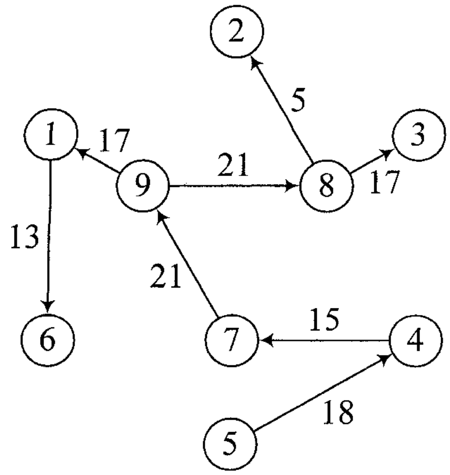


Fig. 4-22

Solution. The branching (Fig. 4-22) obtained by the greedy method has a weight of 127, whereas the maximum weight branching (Fig. 4-23) has a weight of 128.

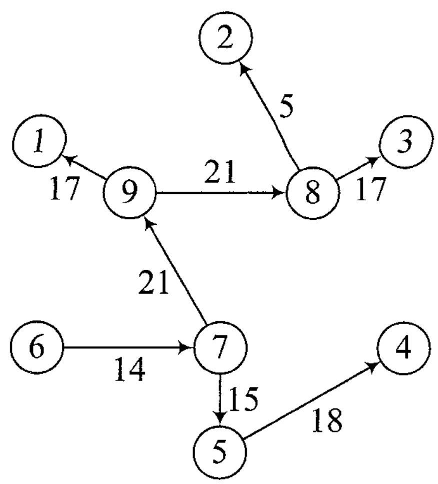


Fig. 4-23

4.27 Find a maximum weight branching in the network shown in Fig. 4-24.

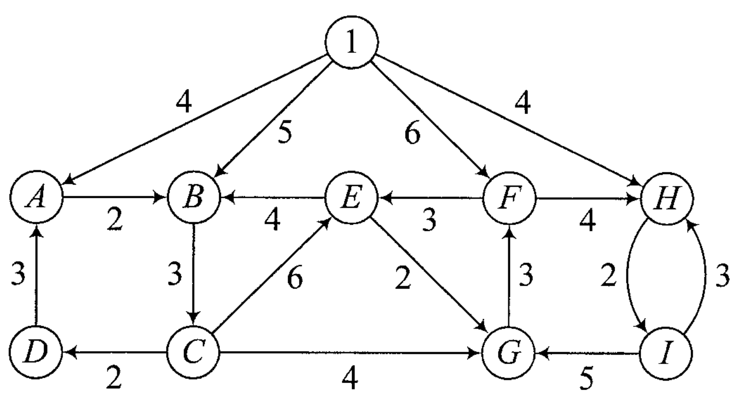


Fig. 4-24

Solution. A maximum weight branching is shown in Fig. 4-25.

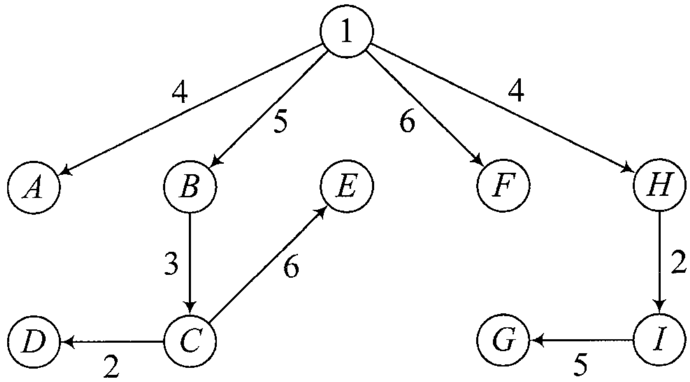


Fig. 4-25

4.28 Find a maximum weight branching on the network shown in Fig. 4-26.

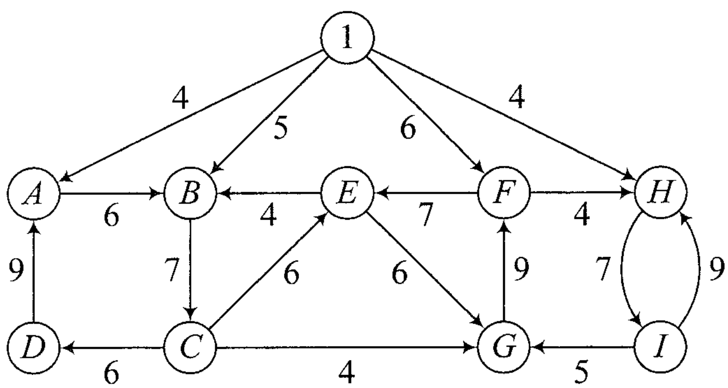


Fig. 4-26

Solution. A maximum weight branching is shown in Fig. 4-27.

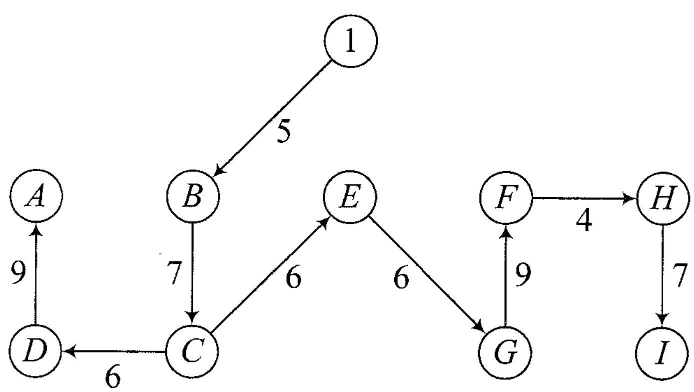


Fig. 4-27

MINIMUM WEIGHT ARBORESCENCES

4.29 Prove Theorem 4.3: A digraph has an arborescence if and only if it is quasi-strongly connected.

Solution. If the digraph G has an arborescence, it is definitely quasi-strongly connected. To prove the converse, consider a quasi-strongly connected digraph $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$. Let $H = (V, E')$ be a maximal quasi-strongly connected subgraph such that the deletion of one more arc from E' will destroy the quasi-strong connectivity of H. There exists a vertex x_1 such that there are paths (in H) from x_1 to vertices 1 and 2. There exists a vertex x_2 such that there are paths (in H) from x_2 to vertices 3 and x_1. Proceeding like this, one can establish that there exists a vertex v from which there is a path (in H) to every other vertex. Let w be any vertex other than v. The indegree (in H) of w is at least 1. Suppose the indegree is more than 1. So there are (at least) two vertices p and q such that (p, w) and (q, w) are arcs in H, which implies that H will continue to be quasi-strongly connected if one of these two arcs is deleted. This violates the maximality assumption. So the indegree in H of every vertex is at most 1. Finally, if there is an arc in H directed to v, the deletion of that arc will not affect the quasi-strong connectivity of H. So the indegree of v is 0. So there is an arborescence with v as a root.

4.30 Prove Theorem 4.4: Let $T^* = (V, E^*)$ be a minimum weight arborescence rooted at vertex r in a weighted digraph $G = (V, E)$ with distinct arc weights, and let $H = (V, E')$ be the subgraph obtained from G by choosing the arc of minimum weight directed to each vertex other than r. Then the set $(C - E^*)$ has exactly one arc for every cycle C in H.

Solution. The set $(C - E^*)$ should have at least one arc. Suppose e' and f' are two arcs in this set. Since T^* is an arborescence, every vertex is reachable from the root using arcs from E^*. So there should be arcs e and f in E^* such that $t(e) = t(e')$ and $t(f) = t(f')$ with $w(e') < w(e)$ and $w(f') < w(f)$. In that case, the arcs in the set $E^* - e + e'$ (or in the set $E^* - f + f'$) will form an arborescence T', where $w(T') < w(T)$, violating the minimality of T.

4.31 Obtain a minimum weight arborescence in the digraph shown in Fig. 4-24.

Solution. A minimum weight arborescence is shown in Fig. 4-28.

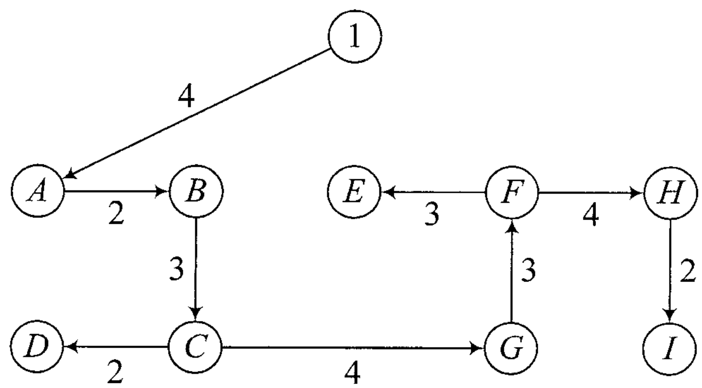


Fig. 4-28

4.32 Obtain a minimum weight arborescence in the digraph shown in Fig. 4-26.

Solution. A minimum weight arborescence is shown in Fig. 4-29.

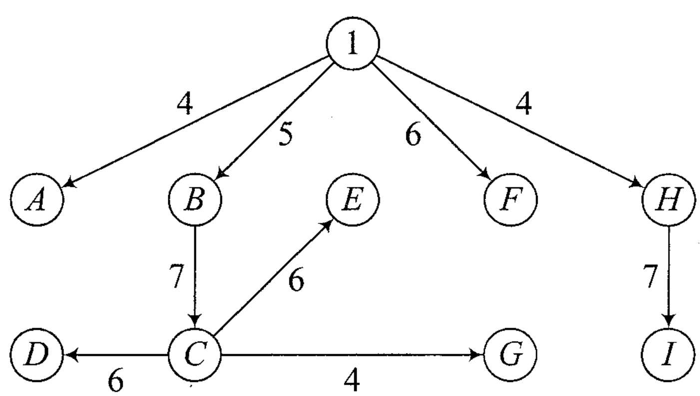


Fig. 4-29

MATROIDS AND THE GREEDY ALGORITHM

4.33 Prove Theorem 4.5: An independent system is a matroid if and only whenever I and J are two independent sets with the property that J has more elements than I, there exists an element $e \in (J - I)$ such that $I \cup \{e\}$ is an independent set.

Solution. Suppose I and J are two independent sets (in a matroid), and suppose the latter has more elements than the former. Obviously, I is not maximal in $I \cup J$. So there exists an independent set $K \subseteq (I \cup J)$ such that I is a proper subset of K. In other words, there exists e in $(J - I)$ such that $I + e$ is an independent set. On the other hand, suppose A and B are two sets that are maximal in a set D of an independent system that satisfies the given hypothesis. If the two sets do not have the same number of elements, suppose there are more elements in B. In that case, there exists e in $(B - A)$ such that $A' = A + e$ is an independent set. This implies that there exists an independent set A' that properly contains A and that is contained in D, violating the maximality of A in D. Thus both A and B have the same cardinality. Hence, the independent system is a matroid.

4.34 Prove Theorem 4.6: The rank function r of a matroid satisfies the following four properties: (*i*) the rank of the empty set is 0, (*ii*) $r(A) \leq r(B)$ if $A \subset B$, (*iii*) the function r is submodular: $r(A \cup B) + r(A \cap B) \leq r(A) + r(B)$, and (*iv*) the rank of a singleton set is either 0 or 1.

Solution. Properties (*i*), (*ii*), and (*iv*) are obvious. Suppose P is a maximal independent subset of $(A \cap B)$, where A and B are arbitrary subsets of the ground set E of a matroid. Since P is an independent subset of A, we can enlarge P by adjoining elements from $(A - B)$ to an independent set Q that is maximal in A. See Fig. 4-30. Thus $Q = (Q - P) \cup P$ and $(Q - P) \subset (A - B)$. Since A is a subset of $(A \cup B)$, set Q can be enlarged by adjoining elements from $(B - A)$ to an independent set R that is maximal in $(A \cup B)$. Hence, $R = Q \cup (R - Q)$ and $(R - Q) \subset (B - A)$. Thus R is the so disjoint union of three independent sets P, $(Q - P)$ and $(R - Q)$. Thus $r(R) = r(P) + r(Q - P) + r(R - Q)$. But $r(Q) = r(P) + r(Q - P)$. So $r(R) = r(P) + r(Q) - r(P) + r(R - Q) = r(Q) + r(R - Q)$. Hence, $r(R) - r(Q) = r(R - Q)$. Now $P \cup (R - Q)$ is a subset of B, which implies that $r(P) + r(R - Q) \leq r(B)$. Thus $r(P) + r(R) \leq r(Q) + r(B)$. But $r(P) = r(A \cap B)$, $r(R) = r(A \cup B)$, and $r(Q) = r(A)$.

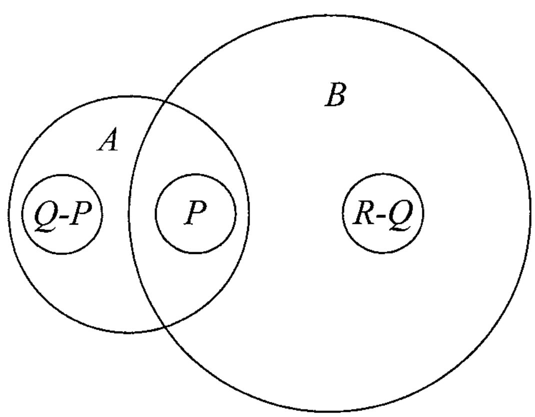


Fig. 4-30

4.35 Prove Theorem 4.7: If r is an integer-valued function defined on the power set of a finite set E satisfying the four properties listed in Theorem 4.6, and if $\mathcal{A} = \{A: A \subseteq E, r(A) = |A|\}$, the pair $(E, \mathcal{A})$ is a matroid.

Solution. Let A be any set in $\mathcal{A}$, and let B be any subset of A. Then $r(A) = |A|$, $r(B) \leq |B|$, and $r(B) \leq r(A)$. Suppose $r(B) < |B|$. In that case, let $A = B \cup C$, where B and $C = \{e_1, e_2, \ldots, e_k\}$ have no elements in common. Then $r(A) = |A| = |B| + k$. Now $r(B \cup C) + r(B \cap C) \leq r(B) + r(C)$. But B and C are disjoint sets, and A is the union of B and C. So $r(A) \leq r(B) + r(C) = r(B) + k < |B| + k$. This contradiction establishes that $r(B) = |B|$. Let I and J be two sets in $\mathcal{A}$ with p and $p + 1$ elements, respectively. Let $I =$

$\{e_1, e_2, \ldots, e_q, e_{q+1}, e_{q+2}, \ldots, e_p\}$ and $J = \{e_1, e_2, \ldots, e_q, f_{q+1}, f_{q+2}, \ldots, f_p, f_{p+1}\}$ (where e_i and f_j are unequal for any i and j) be two sets in $\mathscr{A}$. Suppose $r(I + f_i) = r(I)$ for every f_i.

Now $r(I \cup \{f_i, f_j\}) + r(I) \leq r(I + f_i) + r(I + f_j)$. So $r(I \cup \{f_i, f_j\}) + p \leq p + p$ and hence $r(I \cup \{f_i, f_j\}) \leq p$. But $r(I + f_i) = p$. So, $r(I \cup \{f_i, f_j\}) = p$. Proceeding like this, we establish that $r(I \cup \{f_{q+1}, f_{q+2}, \ldots, f_{p+1}\}) = p$. In other words, $r(J) = p$, which is a contradiction. Thus the assumption that $r(I + f_i) = p$ for every i is false. So there exists an element f in J such that $r(I + f) = p + 1$.

4.36 Prove Theorem 4.8: A solution of the problem of finding a maximum weight independent set in an independent system can be obtained by using the greedy algorithm for every nonnegative weight function defined on its ground set if and only if the independent system is a matroid.

Solution.

(*a*) If I and J are two independent sets in an independent system with p and $(p + 1)$ elements, respectively, let $w(e) = (p + 2)$ for all e in I, $w(e) = (p + 1)$ for all e in $(J - I)$, and $w(e) = 0$ for all other e in the ground set. Then $w(J) \geq (p + 1)(p + 1) > p(p + 2) = w(I)$; hence, I is not a solution. By the greedy procedure, we take I and then take an element from $(J - I)$. In other words, there exists an element e in the set $(J - I)$ such that $I + e$ is an independent set. So the independent system is a matroid.

(*b*) Suppose that by applying the greedy algorithm, an independent set $I = \{e_1, e_2, \ldots, e_r\}$ is obtained (in a matroid) in which the elements are arranged in nondecreasing order by weight. If $J = \{f_1, f_2, \ldots, f_r\}$ is any independent set in the matroid, it is enough if we prove by induction that $w(f_i) \leq w(e_i)$ for every i. It is true when $i = 1$. Suppose it holds for $i = 1, 2, \ldots, (m - 1)$. We have to prove it holds when $i = m$. Suppose $w(f_m) > w(e_m)$. Let $D = \{e_1, e_2, \ldots, e_{m-1}\}$ and $A = \{e: w(e) \geq w(f_m)\}$. Then D is an independent set and, by the induction hypothesis, is a subset of A. If D is not maximal in A, there exists e in A such that $D + e$ is independent. But if e is in A, $w(e) \geq w(f_m) > w(e_m)$, which implies that after picking e_{m-1}, the greedy algorithm would have selected e and not e_m. So D is maximal in A. Since D has $(m - 1)$ elements, any independent subset of A cannot have more than $(m - 1)$ elements. But $\{f_1, f_2, \ldots, f_m\}$ is an independent subset of A. This contradiction shows that $w(f_m) \leq w(e_m)$.

4.37 If $\{E_1, E_2, \ldots, E_k\}$ is a partition of a finite set E and if $\mathscr{A} = \{I \subset E: |I \cap E_i| \leq 1 \text{ for each } i\}$, show that the pair $(E, \mathscr{A})$ is a matroid (known as a **partition matroid** on E), and find its rank function.

Solution. If J is a subset of I that belongs to class $\mathscr{A}$, the intersection of J with every set in the class is at most 1. In other words, the pair is an independent system. Suppose J has p elements and I has $(p + 1)$ elements such that they have r elements in common. Specifically, let $J \cap E_i = e_i$ for $i = 1, 2, \ldots, p$; $I \cap E_i = e_i$ for $i = 1, 2, \ldots, r$, and $I \cap E_i = f_i$ for $i = r + 1, \ldots, p, p + 1$. Then the set J' obtained by adjoining f_{p+1} to J is in class $\mathscr{A}$, and the pair is a matroid, as proved in Problem 4.33. If A is a subset of E, define $r(A)$ to be the number of sets in the partition that have nonempty intersection with the set A. Then it is easily verified that r is a rank function on the ground set E.

4.38 If $G = (X, Y, E)$ is a bipartite graph, show that we can obtain two partition matroids on E: one using the set X and the other using Y.

Solution. Let $X = \{v_1, v_2, \ldots, v_x\}$ and $Y = \{w_1, w_2, \ldots, w_y\}$ be such that every edge in G is between some vertex in X and some vertex in Y. Let $A_i = \{e \in E: e \text{ and } v_i \text{ are adjacent}\}$ and $B_j = \{e \in E: e \text{ and } w_j \text{ are adjacent}\}$. Then the partition matroid on E defined by the partition $\mathscr{A}_1 = \{A_1, A_2, \ldots, A_x\}$ is the **left partition matroid,** and the partition matroid on E defined by the partition $\mathscr{A}_2 = \{B_1, B_2, \ldots, B_y\}$ is the **right partition matroid.**

4.39 If $G = (V, E)$ is a directed graph, show that it is possible to obtain two partition matroids on the set E of the arcs on the digraph.

Solution. For each vertex i, let H_i be the set of arcs that are directed to that vertex, and let T_i be the set of arcs that are directed from that vertex. Then the matroid on E using the partition $\{H_i\}$ is called the **head partition matroid,** and the matroid using the partition $\{T_i\}$ is called the **tail partition matroid.**

4.40 Let E be a finite set with a weight function w defined on E, and let $(E, \mathcal{A}_i)$ be a collection of k matroids defined on the same ground set E. The ***k*-matroid cardinality intersection problem** is the problem of finding a subset of maximum cardinality that is independent in each of the k matroids. The ***k*-matroid weighted intersection problem** is the problem of finding a subset of maximum weight that is independent in each of the k matroids. Show that (a) the former is a special case of the latter and (b) the maximum weight branching problem and the minimum weight arborescence problem are 2-matroid weighted intersection problems.

Solution.

(a) If the weight of each element is 1, a set of maximum cardinality is a set of maximum weight and vice versa.

(b) A set I of arcs will constitute a branching in the digraph if and only if I is an independent set both in the head partition matroid and in the circuit matroid defined on E by treating each arc as an edge. If G is a quasi-strongly connected digraph of order n, a set I of $(n - 1)$ arcs is an arborescence if and only if I is independent both in the head partition matroid and the circuit matroid.

4.41 The **span** of any subset A of the ground set of a matroid is a maximal superset of A having the same rank as A. Show that the span of a set is unique.

Solution. Suppose A_1 and A_2 are two spans of A. Let $r(A) = r(A_1) = r(A_2) = p$. By definition, the rank of any set that properly contains A_1 (or, for that matter, A_2) is more than p. Hence, if e_2 is in $(A_2 - A_1)$, $r(A_1 + e_2) > p$. Let I be a subset of A with p elements, and let J be a subset of $(A_1 + e_2)$ with $(p + 1)$ elements. So there is an element in $(J - I)$ that can be adjoined to I to obtain a larger independent set. The only such element is e_2. So $(I + e_2)$ is an independent set with $(p + 1)$ elements. But $(I + e_2)$ is a subset of A_2, and the rank of A_2 is only p. This is a contradiction.

Supplementary Problems

4.42 Find the weight of a minimum spanning tree in the network whose weight matrix is the following matrix:

$$\begin{bmatrix} 0 & 4 & — & 5 & 2 & — & — & — & — \\ 4 & 0 & 4 & — & 7 & — & — & — & — \\ — & 4 & 0 & — & 2 & 5 & — & — & — \\ 5 & — & — & 0 & 1 & — & 3 & — & — \\ 2 & 7 & 2 & 1 & 0 & 1 & 6 & 7 & 6 \\ — & — & 5 & — & 1 & 0 & — & — & 3 \\ — & — & — & 3 & 6 & — & 0 & 4 & — \\ — & — & — & — & 7 & — & 4 & 0 & 4 \\ — & — & — & — & 6 & 3 & — & 4 & 0 \end{bmatrix}$$

Ans. 20

4.43 In Problem 4.42, suppose we have to find a spanning tree T of minimum weight that contains the two edges of maximum weight in the network. Find the weight of T. *Ans.* 26

4.44 Find the weight of a maximum weight spanning tree in the network of Problem 4.42. *Ans.* 44

4.45 Find the weight of a maximum weight branching in the directed network $G = (V, E)$, where $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$ and $E = \{(1, 2), (1, 7), (2, 3), (2, 7), (3, 4), (3, 5), (4, 5), (6, 3), (6, 5), (6, 7), (7, 8), (8, 1)\}$ with weights 5, 8, 8, 8, 4, 5, 1, 6, 8, 6, 8, and 2, respectively. *Ans.* 34

4.46 Find the weight of a maximum weight branching in the directed network $G = (V, E)$, where $V = \{1, 2, 3, 4, 5, 6\}$ and $E = \{(1, 2), (1, 3), (1, 4), (2, 3), (3, 5), (5, 6), (6, 4), (4, 3)\}$ with weights 1, 0, 1, 2, 3, 2, 2, and 3, respectively. *Ans.* 10

4.47 Find the weight of a maximum weight branching in the directed network $G = (V, E)$, where $V = \{1, 2, 3, 4, 5, 6\}$ and $E = \{(1, 4), (1, 5), (2, 1), (2, 3), (3, 1), (4, 2), (4, 3), (4, 5), (4, 6), (5, 4), (6, 5)\}$ with weights 7, 2, 4, 1, 3, 4, 1, -5, -1, 6, and 1, respectively. *Ans.* 16

4.48 Find the weight of a minimum weight arborescence rooted at vertex 1 in the directed network $G = (V, E)$, where $V = \{1, 2, 3, 4, 5, 6\}$ and $E = \{(1, 2), (1, 4), (1, 6), (2, 3), (2, 6), (3, 2), (4, 2), (4, 5), (5, 6), (6, 3), (6, 4)\}$ with weights 16, 15, 19, 1, 10, 4, 11, 5, 3, 8, and 2, respectively. *Ans.* 33

4.49 Find the weight of a minimum weight arborescence rooted at vertex 1 in the directed network $G = (V, E)$, where $V = \{1, 2, 3, 4, 5, 6\}$ and $E = \{(1, 2), (1, 4), (1, 6), (2, 3), (3, 5), (4, 3), (4, 5), (5, 2), (5, 6), (6, 2), (6, 4)\}$ with weights 1, 4, 6, 12, 9, 9, 5, 11, 7, 10, and 6, respectively. *Ans.* 25

4.50 Give an example of a matroid and two subsets of the ground set such that the inequality in the submodularity relation involving these two subsets is a strict inequality. [*Hint:* In the ground set $E = \{a, b, c\}$ of the three edges in the complete graph with three vertices, take $A = \{a, b\}$ and $B = \{b, c\}$.]

4.51 Find the circuits and the bases of a k-uniform matroid.
Ans. Any subset of the ground set with $k + 1$ elements is a circuit. Any subset with k elements is a base.

Chapter 5

Shortest Path Problems

5.1 TWO SHORTEST PATH ALGORITHMS

If there is a path from vertex u to vertex v in a network G, any path of minimum length from u to v is a **shortest path (SP)** from u to v, and its weight is the **shortest distance (SD)** from u to v. The problem of finding shortest paths in networks is called the **shortest path problem.**

In this section, we investigate two well-known SP algorithms. In the first case, we have a network in which no arc weight is negative. Using the algorithm, we can obtain the SP and SD from a fixed vertex v (the source vertex) to every other vertex u provided that there is a path in the network from v to u. In the second case, the weight function need not be nonnegative. Furthermore, using the algorithm, we can obtain the SD and an SP from any vertex u to any vertex v if there is a path from u to v. The existence of negative cycles in the network can also be detected. The algorithm terminates as soon as a negative cycle is located because the presence of a negative cycle in a network makes the SP problem "unbounded" in the sense that if we allow repeated vertices in a path, the shortest distance between two vertices may become unbounded.

Dijkstra's Algorithm

Let $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$ is a directed network in which the weight of every arc is nonnegative. This algorithm can be used to find the SP and SD from any fixed vertex (say vertex 1) to any vertex i if there is a directed path from 1 to i. Let $a(i, j)$ be the weight of the arc from i to j. If there is no arc from i to j, $a(i, j)$ is $+\infty$. Each vertex i is assigned a label that is either permanent or tentative. The permanent label $L(i)$ is the SD from 1 to i. The tentative label $L'(i)$ is an upper bound of $L(i)$. At each stage of the algorithm, P is the set of vertices with permanent labels and T is the set of vertices with tentative labels. Initially, P is the set $\{1\}$ with $L(1) = 0$ and $L'(j) = a(1, j)$ for all j. The procedure terminates when $P = V$. Each iteration consists of two steps:

Step 1. Find a vertex k in T for which $L'(k)$ is finite and minimum. If there is no k, stop; there is no path from 1 to any unlabeled vertex. Otherwise, declare k to be permanently labeled, and adjoin k to P. Stop if $P = V$. Label the arc (i, k), where i is a labeled vertex that determines the minimum value of $L'(k)$.

Step 2. Replace $L'(j)$ by the smaller value of $L'(j)$ and $L(k) + a(k, j)$ for every j in T. Go to step 1.

Notice that if G has n vertices and it is possible to obtain the SD from the starting vertex v to every other vertex, the set of $(n - 1)$ arcs obtained by this method will form an arborescence rooted at v that will give both the SD and SP from v to every other vertex.

Theorem 5.1. Dijkstra's algorithm finds the SD from a fixed vertex v to any vertex i in the network if there is a path from v to i. (See Solved Problem 5.1)

Example 1. Obtain the SD and SP from vertex 1 to every other vertex in the network shown in Fig. 5-1.

Iteration 1:

Step 1. $P = \{1\}$, and $L(1) = 0$. $L'(2) = 4$, $L'(3) = 6$, and $L'(4) = 8$. Adjoin vertex 2 to P. The arc (1, 2) is labeled.

Step 2. $P = \{1, 2\}$, and $L(2) = 4$. $L'(3) = \min\{6, L(2) + a(2, 3)\}$, $L'(4) = \min\{8, L(2) + a(2, 4)\}$, $L'(5) = \min\{\infty, L(2) + a(2, 5)\}$, $L'(6) = \min\{\infty, L(2) + a(2, 6)\}$, and $L'(7) = \min\{\infty, L(2) + a(2, 7)\}$.

Iteration 2:

Step 1. $P = \{1, 2\}$, $L(1) = 0$, and $L(2) = 4$. $L'(3) = 5$, $L'(4) = 8$, and $L'(5) = 11$. Adjoin vertex 3 to P. The arc (2, 3) is labeled.

Step 2. $P = \{1, 2, 3\}$, and $L(3) = 5$. $L'(4) = \min\{8, L(3) + a(3, 4)\}$, $L'(5) = \min\{11, L(3) + a(3, 5)\}$, $L'(6) = \min\{\infty, L(3) + a(3, 6)\}$, and $L'(7) = \min\{\infty, L(3) + a(3, 7)\}$.

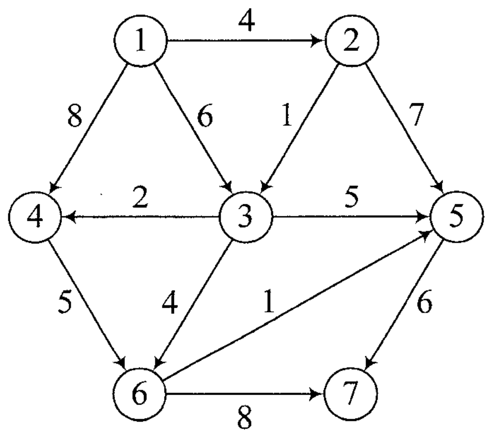


Fig. 5-1

Iteration 3:

Step 1. $P = \{1, 2, 3\}$, $L(1) = 0$, $L(2) = 4$, and $L(3) = 5$. $L'(4) = 7$, $L'(5) = 10$, and $L'(6) = 9$. Adjoin vertex 4 to P. The arc (3, 4) is labeled.

Step 2. $P = \{1, 2, 3, 4\}$, and $L(4) = 7$. $L'(5) = \min\{10, L(4) + a(4, 5)\}$, $L'(6) = \min\{9, L(4) + a(4, 6)\}$, and $L'(7) = \min\{\infty, L(4) + a(4, 7)\}$.

Iteration 4:

Step 1. $P = \{1, 2, 3, 4\}$, $L(1) = 0$, $L(2) = 4$, $L(3) = 5$, and $L(4) = 7$. $L'(5) = 10$, and $L'(6) = 9$. Adjoin vertex 6 to P. The arc (3, 6) is labeled.

Step 2. $P = \{1, 2, 3, 4, 6\}$, and $L(6) = 9$. $L'(5) = \min\{10, L(6) + a(6, 5)\}$, and $L'(7) = \min\{\infty, L(6) + a(6, 7)\}$.

Iteration 5:

Step 1. $P = \{1, 2, 3, 4, 6\}$, $L(1) = 0$, $L(2) = 4$, $L(3) = 5$, $L(4) = 7$, and $L(6) = 9$. $L'(5) = 10$, and $L'(7) = 17$. Adjoin vertex 5 to P. The arc (3, 5) is labeled.

Step 2. $P = \{1, 2, 3, 4, 6, 5\}$, and $L(5) = 10$. $L'(7) = \min\{17, L(5) + a(5, 7)\}$.

Iteration 6:

Step 1. $P = \{1, 2, 3, 4, 6, 5\}$, $L(1) = 0$, $L(2) = 4$, $L(3) = 5$, $L(4) = 7$, $L(6) = 9$, and $L(5) = 10$. $L'(7) = 16$. Adjoin vertex 7 to P. The arc (5, 7) is labeled.

Step 2. $P = \{1, 2, 3, 4, 6, 5, 7\}$, and $L(7) = 16$. At this stage, $P = V$.

The labeled arcs (1, 2), (2, 3), (3, 4), (3, 5), (3, 6), and (5, 7) constitute a shortest path arborescence rooted at vertex 1, as shown in Fig. 5-2.

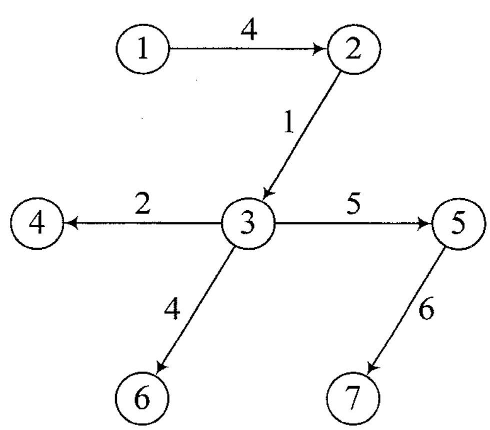


Fig. 5-2

This algorithm need not solve the SD problem if we allow the arc weights to be negative. For example, consider the digraph with $V = \{1, 2, 3\}$ and arcs (1, 2), (1, 3), and (2, 3) with weights 5, 4, and -3, respectively. The starting vertex is vertex 1. In the second iteration, vertex 3 gets the permanent label with $L(3) = 4$, but the SD from vertex 1 to vertex 3 is only 2.

The Floyd–Warshall Algorithm

The weight matrix A of a network $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$, is defined as in the case of Dijkstra's algorithm. In this case, the weight function need not be nonnegative. The initial path matrix P is the $n \times n$ matrix $[p(i, j)]$, where $p(i, j) = j$. There are n iterations in the execution of the algorithm. Iteration j (based at vertex 1) begins with two matrices A_{j-1} and P_{j-1} and ends with A_j and P_j. Initially, $A_0 = A$ and $P_0 = P$. The (u, v) entries in A_j and P_j are denoted by $a_j(u, v)$ and $p_j(u, v)$, respectively. For a fixed j, the matrices A_j and P_j are obtained from A_{j-1} and P_{j-1} by applying the following rules, known as the triangle (triple) operation: If $a_{j-1}(u, v) \le a_{j-1}(u, j) + a_{j-1}(j, v)$, $a_j(u, v) = a_{j-1}(u, v)$ and $p_j(u, v) = p_{j-1}(u, v)$. Otherwise, $a_j(u, v) = a_{j-1}(u, j) + a_{j-1}(j, v)$ and $p_j(u, v) = p_{j-1}(u, j)$.

When the algorithm terminates, we are left with the **SD matrix A_n and the SP matrix P_n.** The (u, v) entry in the first matrix is the shortest distance between these two vertices, and the (u, v) entry in the second matrix denotes the first vertex (after u) in a shortest path from u to v.

Theorem 5.2. The Floyd–Warshall algorithm using the triangle operation correctly solves the SD and SP problem. (See Solved Problem 5.4.)

Example 2. Obtain the SD matrix and the SP matrix of the network shown in Fig. 5-3.

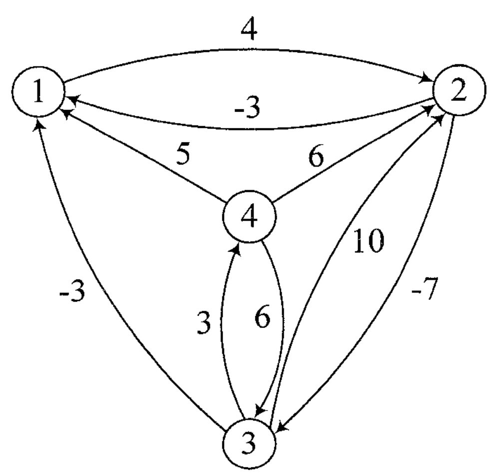


Fig. 5-3

The matrices are

$$A = \begin{bmatrix} 0 & 4 & -3 & \infty \\ -3 & 0 & -7 & \infty \\ \infty & 10 & 0 & 3 \\ 5 & 6 & 6 & 0 \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{bmatrix}$$

Iteration 1: We begin with $A_1 = A$ and $P_0 = P$. In performing the triangle operation based at vertex 1, the only change is at the (4, 3) entry: $a_1(4, 3) = \min\{a_0(4, 3), a_0(4, 1) + a_0(1, 3)\} = \min\{6, 5 - 3\} = 2$. Then $p_1(4, 3) = p_0(4, 1) = 1$. Thus at the end of this iteration, we have A_1 and P_1:

$$A_1 = \begin{bmatrix} 0 & 4 & -3 & \infty \\ -3 & 0 & -7 & \infty \\ \infty & 10 & 0 & 3 \\ 5 & 6 & 2 & 0 \end{bmatrix} \quad \text{and} \quad P_1 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 1 & 4 \end{bmatrix}$$

Iteration 2: We begin with A_1 and P_1 and obtain the matrices A_2 and P_2 after applying the triangle operation:

$$A_2 = \begin{bmatrix} 0 & 4 & -3 & \infty \\ -3 & 0 & -7 & \infty \\ 7 & 10 & 0 & 3 \\ 3 & 6 & -1 & 0 \end{bmatrix} \quad \text{and} \quad P_2 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 2 & 2 & 3 & 4 \\ 2 & 2 & 2 & 4 \end{bmatrix}$$

Iteration 3: At the end of this iteration,

$$A_3 = \begin{bmatrix} 0 & 4 & -3 & 0 \\ -3 & 0 & -7 & -4 \\ 7 & 10 & 0 & 3 \\ 3 & 6 & -1 & 0 \end{bmatrix} \quad \text{and} \quad P_3 = \begin{bmatrix} 1 & 2 & 3 & 3 \\ 1 & 2 & 3 & 3 \\ 2 & 2 & 3 & 4 \\ 2 & 2 & 2 & 4 \end{bmatrix}$$

Iteration 4: At the end of this iteration,

$$A_4 = \begin{bmatrix} 0 & 4 & -3 & 0 \\ -3 & 0 & -7 & -4 \\ 6 & 9 & 0 & 3 \\ 3 & 6 & -1 & 0 \end{bmatrix} \quad \text{and} \quad P_4 = \begin{bmatrix} 1 & 2 & 3 & 3 \\ 1 & 2 & 3 & 3 \\ 4 & 4 & 3 & 4 \\ 2 & 2 & 2 & 4 \end{bmatrix}$$

At this stage, we have the SD between every pair of vertices in the network as the appropriate entries in SD matrix A_4. An SP from one vertex to another can be obtained from the entries in SP matrix P_4. For example, the SD from 1 to 2 is 4, as can be seen from the (1, 2) entry in the SD matrix. In the SP matrix, the (1, 2) entry is 2. So an SP from 1 to 2 is the arc from 1 to 2. The SD from 1 to 4 is 0. The (1, 4) entry in the SP matrix is 3. So in an SP from 1 to 4, the first vertex after 1 is 3. The (3, 4) entry in the SP matrix is 4. So an SP from 1 to 4 is $1 \to 3 \to 4$.

Locating Negative Cycles

Consider a network with the weight matrix

$$A = \begin{bmatrix} 0 & \infty & \infty & 1 \\ 2 & 0 & 1 & 3 \\ \infty & \infty & 0 & \infty \\ 2 & -4 & 3 & 0 \end{bmatrix}$$

At the end of iteration 2,

$$A_2 = \begin{bmatrix} 0 & \infty & \infty & 1 \\ 2 & 0 & 1 & 3 \\ \infty & \infty & 0 & \infty \\ 2 & -4 & 3 & -1 \end{bmatrix} \quad \text{and} \quad P_2 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 1 \\ 1 & 2 & 3 & 4 \\ 2 & 2 & 2 & 2 \end{bmatrix}$$

The (4, 4) diagonal entry is negative (instead of 0), indicating the presence of a negative cycle. From the second matrix, we see that this cycle (seen as a path from 4 to 4) is $4 \to 2 \to 1 \to 4$ with weight $-4 + 2 + 1 = -1$.

5.2 THE STEINER NETWORK PROBLEM

The Steiner network problem is the optimization problem of finding a tree $T = (W', F)$ in G such that $W \subset W'$ and the weight $w(F)$ is as small as possible, given a set W of vertices in an undirected graph $G = (V, E)$ with a nonnegative weight function w defined on E. The tree T is a **Steiner tree** of the set W. The vertices in $(W' - W)$ are the **Steiner points** of W with respect to T. A Steiner tree with respect to a pair of vertices in an undirected network is any shortest path P between them, and the Steiner points of this pair with respect to P are the intermediate vertices of this path. At the other extreme, any minimum spanning tree in the network is a Steiner tree of the set V. So both the SP problem and the M.S.T. problem can be considered special cases of the Steiner network problem.

Notice that if W is a proper subset of V, a minimum spanning tree of the subgraph H induced by the set W need not be a Steiner tree with respect to W. This can be seen in the following example.

Example 3. In the network shown in Fig. 5-4, the weight of a minimum spanning tree of the subgraph induced by the set $W = \{1, 2, 3, 4\}$ is easily seen to be 9. But it is possible to obtain a tree T that spans these four vertices and vertex 5 such that the weight of the edge set of T is only 8.

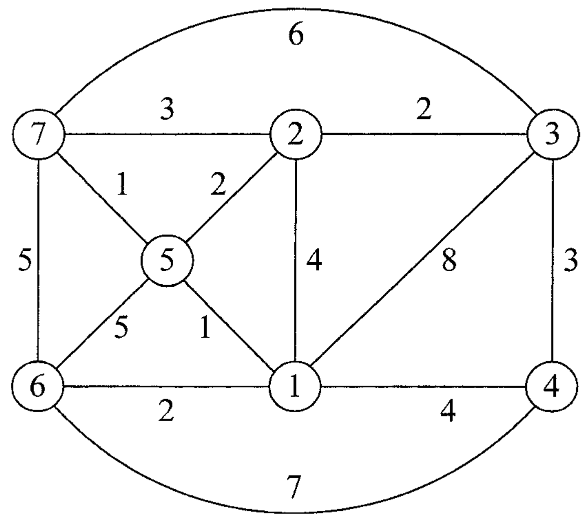


Fig. 5-4

Theorem 5.3. Let $\{1, 2, 3, \ldots, n\}$ be the vertex set of the complete graph G with a nonnegative weight function w satisfying the triangle property $w(\{i, j\}) \leq w(\{i, k\}) + w(\{k, j\})$ for all vertices i, j, and k. If W is a set of m vertices in network G, there exists a Steiner tree for W in the network that contains at most $(n - 2)$ Steiner points. (See Solved Problem 5.11.)

Spanning Tree Enumeration Algorithm

The input is a set W of m vertices in a connected network $G = (V, E)$ of order n in which a nonnegative weight is defined on each edge. Construct the complete network $G' = (V, F)$ in which the weight of an edge joining two vertices is the SD between them. Let $\mathcal{S} = \{S \subset (V - W) : |S| \leq (m - 2)\}$. For each subset S in $\mathcal{S}$, find an M.S.T. of the subgraph of G' induced by $W \cup S$ on G'. Among the trees thus obtained, select a tree T' of minimum weight. Construct a tree T from T' by replacing each edge joining two vertices by the set of edges in a shortest path between them. Those vertices of T that are not in W will form a set of Steiner points for W.

Example 4. Find the Steiner trees with respect to $W = \{3, 6, 7\}$ in the network shown in Fig. 5-4.
The SD matrix for this network is

$$D = \begin{bmatrix} 0 & 3 & 5 & 4 & 1 & 2 & 2 \\ 3 & 0 & 2 & 5 & 2 & 5 & 3 \\ 5 & 2 & 0 & 3 & 4 & 7 & 5 \\ 4 & 5 & 3 & 0 & 5 & 6 & 6 \\ 1 & 2 & 4 & 5 & 0 & 3 & 1 \\ 2 & 5 & 7 & 6 & 3 & 0 & 4 \\ 2 & 3 & 5 & 6 & 1 & 4 & 0 \end{bmatrix}$$

Since $m = 3$, S is any subset of $\{1, 2, 4, 5\}$ with at most one element. There are five choices for $(W \cup S)$: $W_1 = \{3, 6, 7\}$, $W_2 = \{3, 6, 7, 1\}$, $W_3 = \{3, 6, 7, 2\}$, $W_4 = \{3, 6, 7, 4\}$, and $W_5 = \{3, 6, 7, 5\}$. The minimum weight spanning trees of the subgraphs of G' induced by these sets have weights 9, 9, 9, 12, and 8, respectively. So we take the subgraph induced by W_5 on the complete network G'. In this subgraph, a minimum spanning tree T' is as shown in Fig. 5-5.

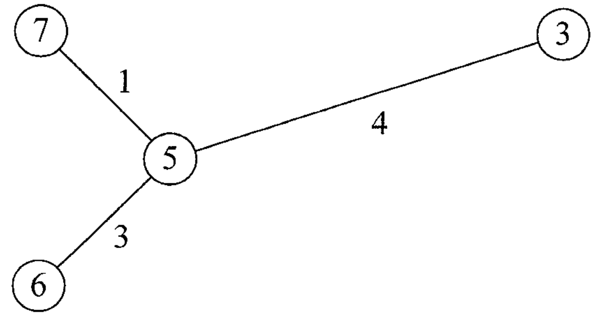


Fig. 5-5

The weight of the edge in T' between 3 and 5 is 4. We replace this edge by a shortest path between 3 and 5 that is also necessarily of weight 4. This SP is 5 — 2 — 3. Likewise the edge between 6 and 5 is replaced by the path 5 — 1 — 6. Thus the Steiner tree T with respect to $\{3, 6, 7\}$ is as shown in Fig. 5-6. The Steiner points are 1, 2, and 5.

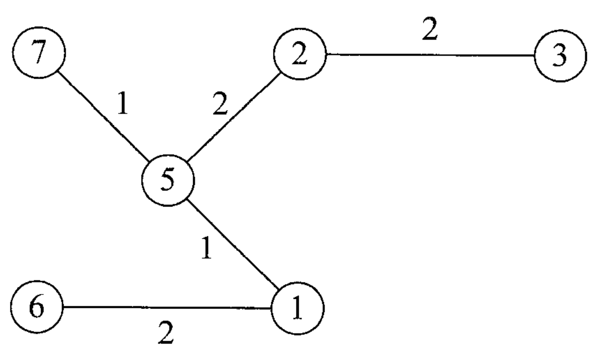


Fig. 5-6

5.3 FACILITY LOCATION PROBLEMS

The problem of deciding the exact place in a community where a facility (such as a school, a post office, or a fire station) should be located to serve the needs of the community as economically and efficiently as possible is known as the facility location problem. Such location problems can often be modeled as networks in which the facility could be located at one or more vertices.

If the facility is an institution like a post office or a school, it is desirable to locate it such that the sum of the distances from the facility to several parts of the community is as small as possible. This category, in which the aim is to minimize the sum of several weights, is aptly called a **minsum problem.** On the other hand, if the facility is an institution like a fire station, it is always desirable to locate it such that the distance from the fire station to the farthest point in the community is minimized. This category, in which the aim is to minimize the maximum shortest distance, is known as a **minmax problem.**

Median (Minsum) Problems

Let $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$ is a weighted network with a nonnegative weight function defined on E, and let $D = [d(i, j)]$ be its SD matrix, where $d(i, j)$ is the SD from i to j in the network. For each i, let $s(i)$ be the sum of all the elements in row i of D. Vertex j is called a **median vertex** if $s(j) \leq s(i)$ for every vertex i of the network. The **median** is the set of all median vertices. In a more general setting, suppose that a nonnegative weight function is defined on the set V in addition to the weight function w on the set E. In that case, we define $s'(i) = w(1)d(i, 1) + w(2)d(i, 2) + \cdots + w(n)d(i, n)$. The vertex j is a **weighted median**

vertex if $s'(j) \leq s'(i)$ for every vertex i of the network, and the **weighted median** is the set of all weighted median vertices.

Center (Minmax) Problems

For each vertex i, define the **eccentricity** $e(i)$ as the largest entry in row i of the SD matrix. Vertex j is called a **center vertex** if $e(j) \leq e(i)$ for every vertex i of the network. The set of all center vertices is called the **center** of the network. As in the case of medians, there are analogous definitions for weighted center vertices and weighted centers.

Example 5. Find (*a*) the median and (*b*) the center of the network shown in Fig. 5-7 in which the weight of each edge is 1.

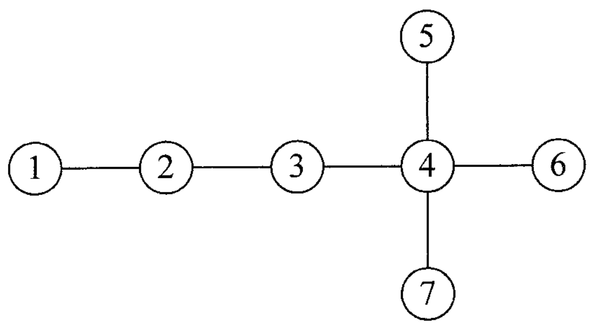


Fig. 5-7

The SD matrix is

$$D = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 4 & 4 \\ 1 & 0 & 1 & 2 & 3 & 3 & 3 \\ 2 & 1 & 0 & 1 & 2 & 2 & 2 \\ 3 & 2 & 1 & 0 & 1 & 1 & 1 \\ 4 & 3 & 2 & 1 & 0 & 2 & 2 \\ 4 & 3 & 2 & 1 & 2 & 0 & 2 \\ 4 & 3 & 2 & 1 & 2 & 2 & 0 \end{bmatrix}$$

(*a*) The row sums are 18, 13, 10, 9, 14, 14, and 14. Row 4 has the lowest sum, so 4 is the only median vertex. The median is the set $\{4\}$.

(*b*) The eccentricities of the seven vertices are 4, 3, 2, 3, 4, 4, and 4, respectively, so 3 is the only center vertex. The center is the set $\{3\}$.

Solved Problems

TWO SHORTEST PATH ALGORITHMS

5.1 Prove Theorem 1: Dijkstra's algorithm finds the SD from a fixed vertex (v) to any vertex i in the network if there is a path from v to i.

Solution. The set of vertices is $V = \{1, 2, \ldots, n\}$. Vertex 1 is taken as the root. At each stage, vertex i has either the permanent label $L(i)$, which is the SD from the root to i, or a tentative label $L'(i)$, which is an upper bound of the SD from the root to i. The set of vertices with permanent labels is denoted by P, and the set of tentative labels is denoted by T. The proof is by induction on the cardinality of P. The induction hypothesis is that (*a*) $L(i)$ is the SD from 1 to each i in P and (*b*) for any vertex j in T, $L'(j)$ is the length of a shortest path from 1 to j, of which every vertex other than j is in P. Statement (*a*) is true when P is the set $\{1\}$. Suppose it is true when

P has $(k-1)$ elements. Specifically, let $P = \{1, 2, \ldots, k-1\}$. Suppose the next vertex to be assigned to P is k. By the induction hypothesis, $L'(k)$ is the length of a SP from 1 to k in which every vertex other than k is in P. By our definition, $L(k) = L'(k)$. We claim that $L(k)$ is the SD from 1 to k. If this is not the case, $L(k) > d$, where d is the SD. Hence, any shortest path from 1 to k has at least one intermediate vertex from the set T. Let v be the first vertex (from T) in this path. If the SD from 1 to v is d', $d' \leq d < L(k)$. But because k is in P and v is in T, it is implied that $L(k) \leq d'$. So (a) is true when P has k vertices. We now prove (b), as follows. Vertex k is used to revise the labels of the remaining vertices in T. The label $L'(j)$ remains the same or gets the new label $L(k) + a(k, j)$. Thus (b) holds when P has k vertices.

5.2 Using Dijksra's algorithm, find a shortest distance arborescence rooted at vertex 1 of the directed network shown in Fig. 5-8.

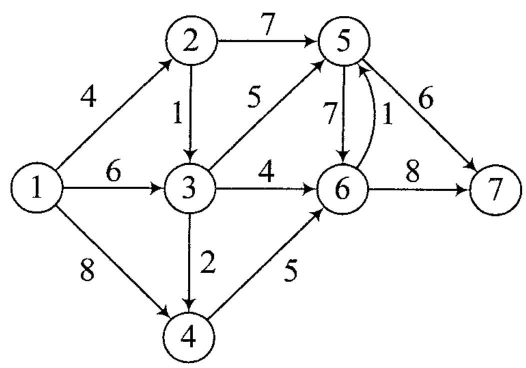


Fig. 5-8

Solution. The shortest distance arborescence is the network shown in Fig. 5-9.

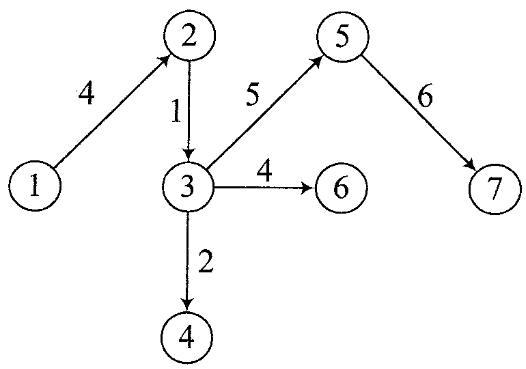


Fig. 5-9

5.3 Using Dijkstra's algorithm, find a shortest distance arborescence for the network whose weight matrix is

$$A = \begin{bmatrix} 0 & - & 4 & 10 & 3 & - & - \\ - & 0 & 1 & 1 & 2 & 11 & 0 \\ - & 9 & 0 & 8 & 3 & 2 & 1 \\ - & 4 & 0 & 0 & 8 & 6 & 3 \\ - & 0 & 1 & 2 & 0 & 3 & 1 \\ - & 1 & 1 & 3 & 2 & 0 & 0 \\ - & 4 & 3 & - & - & 2 & 0 \end{bmatrix}$$

Solution. The shortest distance arborescence rooted at vertex 1 is the network shown in Fig. 5-10.

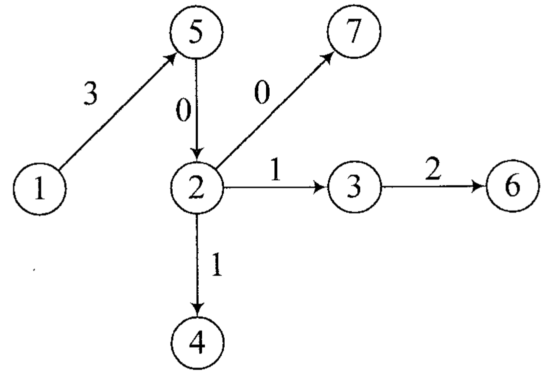


Fig. 5-10

5.4 Prove Theorem 5.2: The Floyd–Warshall algorithm using the triangle operation correctly solves the SD and SP problem.

Solution. Assume that the weight function is nonnegative. First, we show that if we perform the triple operation successively for the vertices $j = 1, 2, \ldots, n$, the (u, v) entry in A_n at the end is the SD from u to v. The proof is by induction on j. The induction hypothesis is that when the triple operation based at vertex j is completed, the (u, v) entry in the matrix A_j is the SD from u to v using a path with intermediate vertices $w \le j$. Suppose this is true for $j = (k - 1)$. Consider the triple operation that determines the (u, v) entry in A_k. If the SP from u to v at this stage does not pass through k, the (u, v) entry in A_{k-1} is the same as the (u, v) entry in A_k. So the hypothesis holds for $j = k$ as well. On the other hand, if the SP contains k as an intermediate vertex, the (u, v) entry will be the sum of the (u, k) entry and the (k, v) entry of A_{k-1}. But, by the induction hypothesis, each of these entries corresponds to an optimal path with intermediate vertices $\le k$. In this case, the hypothesis also holds for k. This completes the inductive argument when the weight function is nonnegative. This procedure works well as long as there are no negative cycles. If there is a negative cycle, a diagonal entry will, at some stage, become negative and the algorithm terminates. If the (u, v) entry in P_n is w, an SP from u to v is obtained by joining (concatenating) the SP from u to w and the SP from w to v. This is also the consequence of the triple operation.

5.5 Find the SD matrix and the SP matrix of the network in which the weight matrix A is

$$A = \begin{bmatrix} 0 & 1 & \text{—} & \text{—} & \text{—} & 1 & 4 \\ 1 & 0 & 2 & \text{—} & \text{—} & \text{—} & 1 \\ \text{—} & 2 & 0 & 2 & \text{—} & \text{—} & 4 \\ \text{—} & \text{—} & 2 & 0 & 3 & \text{—} & \text{—} \\ \text{—} & \text{—} & \text{—} & 3 & 0 & 9 & 3 \\ 1 & \text{—} & \text{—} & \text{—} & 9 & 0 & \text{—} \\ 4 & 1 & 4 & \text{—} & 3 & \text{—} & 0 \end{bmatrix}$$

Solution. The SD matrix A_7 and the SP matrix P_7 are

$$A_7 = \begin{bmatrix} 0 & 1 & 3 & 5 & 5 & 1 & 2 \\ 1 & 0 & 2 & 4 & 4 & 2 & 1 \\ 3 & 2 & 0 & 2 & 5 & 4 & 3 \\ 5 & 4 & 2 & 0 & 3 & 6 & 5 \\ 5 & 4 & 5 & 3 & 0 & 6 & 3 \\ 1 & 2 & 4 & 6 & 6 & 0 & 3 \\ 2 & 1 & 3 & 5 & 3 & 3 & 0 \end{bmatrix} \quad \text{and} \quad P_7 = \begin{bmatrix} \text{—} & 2 & 2 & 2 & 2 & 6 & 2 \\ 1 & \text{—} & 3 & 3 & 7 & 1 & 7 \\ 2 & 2 & \text{—} & 4 & 4 & 2 & 2 \\ 3 & 3 & 3 & \text{—} & 5 & 3 & 3 \\ 7 & 7 & 4 & 4 & \text{—} & 7 & 7 \\ 1 & 1 & 1 & 1 & 1 & \text{—} & 1 \\ 2 & 2 & 2 & 2 & 5 & 2 & \text{—} \end{bmatrix}$$

5.6 Find the SD matrix and the SP matrix of the network for which the weight matrix is

$$A = \begin{bmatrix} 0 & — & 4 & 10 & 3 & — & — \\ — & 0 & -1 & -1 & 2 & 11 & 0 \\ — & 9 & 0 & 8 & 3 & 2 & 1 \\ — & 4 & 0 & 0 & 8 & 6 & 3 \\ — & 0 & 1 & 2 & 0 & 3 & -1 \\ — & -1 & -1 & 3 & 2 & 0 & 0 \\ — & 4 & 3 & — & — & 2 & 0 \end{bmatrix}$$

Solution. The SD matrix A_7 and the SP matrix P_7 are

$$A_7 = \begin{bmatrix} 0 & 3 & 2 & 2 & 3 & 4 & 2 \\ — & 0 & -1 & -1 & 2 & 1 & 0 \\ — & 1 & 0 & 0 & 3 & 2 & 1 \\ — & 1 & 0 & 0 & 3 & 2 & 1 \\ — & 0 & -1 & -1 & 0 & 1 & -1 \\ — & -1 & -2 & -2 & 1 & 0 & -1 \\ — & 1 & 0 & 0 & 3 & 2 & 0 \end{bmatrix} \quad \text{and} \quad P_7 = \begin{bmatrix} — & 5 & 5 & 5 & 5 & 5 & 5 \\ 1 & — & 3 & 4 & 5 & 3 & 7 \\ 1 & 6 & — & 6 & 5 & 6 & 7 \\ 1 & 3 & 3 & — & 3 & 3 & 3 \\ 1 & 2 & 2 & 2 & — & 2 & 7 \\ 1 & 2 & 2 & 2 & 2 & — & 2 \\ 1 & 6 & 6 & 6 & 6 & 6 & — \end{bmatrix}$$

5.7 In Problem 5.6, find an SP from vertex 4 to vertex 2.

Solution. The (4, 2)-entry in the path matrix is 3. So in an SP from 4 to 2, the first vertex after 4 is 3. The (3, 2) entry in the path matrix is 6, so the next vertex in the SP is 6. The (6, 2) entry in the path matrix is 2, so the last vertex in the SP is 2. Thus an SP from 4 to 2 is $4 \to 3 \to 6 \to 2$ with a weight of $0 + 2 + (-1) = 1$, which is the (4, 2) entry in the SD matrix.

5.8 Construct a shortest distance arborescence rooted at vertex 1 in the network of Problem 5.6, the path matrix.

Solution. One such arborescence is the following shortest path from the root to vertex 3: $1 \to 5 \to 7 \to 6 \to 2 \to 4 \to 3$.

5.9 Suppose the negative entry (-1) that appears in the fourth column of the weight matrix in problem 5.6 is replaced by a smaller negative entry (-3). Find a negative cycle in the modified network.

Solution. There are two negative cycles: $2 \to 4 \to 3 \to 6 \to 2$ and $2 \to 4 \to 3 \to 7 \to 6 \to 2$.

5.10 In Problem 5.6, obtain the distance matrix and the path matrix that will give the shortest distance between every pair of vertices using paths that will not use vertices 5, 6, and 7 as intermediate vertices. Use these two matrices to obtain a path of minimum length from vertex 7 to vertex 4 that does not use vertices 5 and 6 as intermediate vertices.

Solution. The matrices A_4 and P_4 will ignore the paths that contain 5, 6, and 7 as intermediate vertices. These matrices are

$$A_4 = \begin{bmatrix} 0 & 13 & 4 & 10 & 3 & 6 & 5 \\ — & 0 & -1 & -1 & 2 & 1 & 0 \\ — & 9 & 0 & 8 & 3 & 2 & 1 \\ — & 4 & 0 & 0 & 3 & 2 & 1 \\ — & 0 & -1 & -1 & 0 & 1 & -1 \\ — & -1 & -2 & -2 & 1 & 0 & -1 \\ — & 4 & 3 & 3 & 6 & 2 & 0 \end{bmatrix} \quad \text{and} \quad P_4 = \begin{bmatrix} — & 3 & 3 & 4 & 5 & 3 & 3 \\ 1 & — & 3 & 4 & 5 & 3 & 7 \\ 1 & 2 & — & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & — & 3 & 3 & 3 \\ 1 & 2 & 2 & 2 & — & 3 & 7 \\ 1 & 2 & 2 & 2 & 2 & — & 2 \\ 1 & 2 & 3 & 2 & 2 & 6 & — \end{bmatrix}$$

A path of minimum distance from 7 to 4 (which will not pass through 5 and 6) will have a weight of 3, which is the (7, 4) entry in the weight matrix. Using the path matrix, we see that this path is $7 \to 2 \to 1$ with weight $4 + (-1) = 3$. As we saw in Problem 5.6, the SP from 7 to 4 that allows any vertex to be an intermediate vertex has weight 0. Thus the path is $7 \to 6 \to 2 \to 4$ with total weight $2 + (-1) + (-1)$.

THE STEINER NETWORK PROBLEM

5.11 Prove Theorem 3: Let $\{1, 2, 3, \ldots, n\}$ be the vertex set of the complete graph G with a nonnegative weight function w satisfying the triangle property $w(\{i, j\}) \le w(\{i, k\}) + w(\{k, j\})$ for any three vertices i, j, and k. If W is a set of m vertices in network G, there exists a Steiner tree for W in the network that contains at most $(n - 2)$ Steiner points.

Solution. Let the number of Steiner points of W with respect to a tree T be p. Then T has $(m + p)$ vertices and $(m + p - 1)$ edges. If x is the average degree (in T) of a Steiner point and y is the average degree (in T) of a non-Steiner point, the degree sum $px + my$ is equal to $2(m + p - 1)$. At the same time, $x \ge 3$ (because of the triangle property), and y is at least 1. Hence, $p \le (m - 2)$.

5.12 Find a Steiner tree for the set $W = \{1, 2, 3, 4\}$ in the network shown in Fig. 5-4.

Solution. Here $m = 4$. The subsets of vertices to be considered are $\{1, 2, 3, 4\}$, $\{1, 2, 3, 5\}$, $\{1, 2, 3, 4, 6\}$, $\{1, 2, 3, 4, 7\}$, $\{1, 2, 3, 4, 5, 6\}$, $\{1, 2, 3, 4, 5, 7\}$, and $\{1, 2, 3, 4, 6, 7\}$. The weights of the minimum spanning trees induced by these sets on a complete network with the weight matrix as the SD matrix of the given network are 8, 8, 10, 10, 9, 9, and 11, respectively. So the weight of a Steiner tree is 8. Vertex 5 is the Steiner point, and the corresponding tree has $\{1, 5\}$, $\{2, 5\}$, $\{2, 3\}$, and $\{3, 4\}$ as edges with weights 1, 2, 2, and 3, giving a total of 8.

5.13 Find the Steiner points of the set $\{1, 3, 5\}$ in the network of Problem 5.5.

Solution. From the SD matrix, we see that the weights of the minimum spanning trees in G' induced by $\{1, 3, 5\}$, $\{1, 3, 5, 2\}$, $\{1, 3, 5, 4\}$, $\{1, 3, 5, 6\}$, and $\{1, 3, 5, 7\}$ are 8, 7, 8, and 9, respectively. So in G' the Steiner point is at vertex 2. The M.S.T. in G' consists of edges $\{1, 2\}$, $\{2, 3\}$, and $\{2, 5\}$. Edge $\{2, 5\}$ is the shortest path, $2 — 7 — 5$, in G. So the Steiner points are 2 and 7. The edges of the Steiner tree are $\{1, 2\}$, $\{2, 3\}$, $\{2, 7\}$, and $\{5, 7\}$.

FACILITY LOCATION PROBLEMS

5.14 Find the median of the network of Problem 5.5.

Solution. The row sums in the SD matrix are 17, 14, 19, 25, 26, 22, and 17. The minimum is for vertex 2. So the median is the set $\{2\}$.

5.15 Find the weighted median of the network of Problem 5.5 if the weights of the vertices are 1, 0, 3, 3, 3, 1, and 2, respectively.

Solution. The row sums in this case are 44, 34, 34, 36, 41, 55, and 38. The median is the set $\{2, 3\}$.

5.16 Find the center of the network of Problem 5.5.

Solution. The maximum entries in the rows of the SD matrix are 5, 4, 5, 6, 6, 6, and 5. The minimum is for row 2. So the center is the set $\{2\}$.

5.17 In a network, the minimum value of the eccentricity is the **radius $r(G)$,** and the maximum value is the **diameter $d(G)$.** Show that $r(G) \le d(G) \le 2r(G)$.

Solution. Of course, $r(G) \le d(G)$. Let the SD between two vertices u and v be denoted by $d(u, v)$. There is a vertex x such that $d(x, v) \le r(G)$ for every vertex v. There are also vertices p and q such that $d(p, q) = d(G)$. Now $d(p, q) \le d(p, w) + d(w, q) \le r(G) + r(G)$.

5.18 Show that if each edge of a tree has a nonnegative weight, the cardinality of its center (median) is at most 2.

Solution. If w is a vertex of degree 1 and if w is adjacent to v, the degree of w is at least 2. Moreover, $d(w, u) \ge d(v, u)$, where u is any other vertex. Thus $e(w) \ge e(v)$. Consequently, if T' is the tree obtained by deleting all vertices of degree 1 from T, both trees will have the same center. Then we delete all vertices of degree 1 from T'. Continue this process. Eventually, we get a tree with either one vertex or two vertices. The proof is analogous for median.

Supplementary Problems

5.19 List the arcs of the shortest distance arborescence rooted at vertex 1 of the network with $V = \{1, 2, 3, 4, 5, 6\}$ and $E = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 5), (3, 6), (4, 5), (5, 6)\}$ with weights 4, 7, 3, 3, 2, 2, 3, and 2, respectively.
Ans. (1, 2), (1, 3), (1, 4), (4, 5), and (5, 6)

5.20 List the arcs of the shortest distance arborescence rooted at vertex 1 of the network with $V = \{1, 2, 3, 4, 5, 6\}$ and $E = \{(1, 2), (1, 5), (2, 3), (2, 4), (2, 5), (3, 4), (4, 5), (4, 6), (5, 3)\}$ with weights 3, 7, 7, 2, 5, 3, 2, 5, and 1, respectively. *Ans.* (1, 2), (2, 4), (4, 5), (5, 3), and (4, 6)

5.21 Find the SD matrix A_5 and the SP matrix P_5 of the undirected network with $V = \{1, 2, 3, 4, 5\}$ and $E = \{\{1, 2\}, \{1, 5\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \{4, 5\}\}$ with weights 1, 4, 5, 1, 2, and 1, respectively.

Ans. $$A_5 = \begin{bmatrix} 0 & 1 & 4 & 2 & 3 \\ 1 & 0 & 3 & 1 & 2 \\ 4 & 3 & 0 & 2 & 3 \\ 2 & 1 & 2 & 0 & 1 \\ 3 & 2 & 3 & 1 & 0 \end{bmatrix}; \quad P_5 = \begin{bmatrix} 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 4 & 4 & 4 \\ 4 & 4 & 3 & 4 & 4 \\ 2 & 2 & 3 & 4 & 5 \\ 4 & 4 & 4 & 4 & 5 \end{bmatrix}$$

5.22 Find the SD matrix and the SP matrix of the network with the following weight matrix:

$$\begin{bmatrix} - & 15 & - & - & 9 & - \\ - & - & 35 & 3 & - & - \\ - & - & - & 6 & - & 21 \\ - & - & - & - & 2 & 7 \\ - & 4 & - & 2 & - & - \\ - & - & 5 & - & - & - \end{bmatrix}$$

Ans. $$A_6 = \begin{bmatrix} - & 13 & 23 & 11 & 9 & 18 \\ - & - & 15 & 3 & 5 & 10 \\ - & 12 & - & 6 & 8 & 13 \\ - & 6 & 12 & - & 2 & 7 \\ - & 4 & 14 & 2 & - & 9 \\ - & 17 & 5 & 11 & 13 & - \end{bmatrix}; \quad P_6 = \begin{bmatrix} - & 5 & 5 & 5 & 5 & 5 \\ - & - & 4 & 4 & 4 & 4 \\ - & 4 & - & 4 & 4 & 4 \\ - & 5 & 6 & - & 5 & 6 \\ - & 2 & 4 & 4 & - & 4 \\ - & 3 & 3 & 3 & 3 & - \end{bmatrix}$$

5.23 Find the weight of a Steiner tree of the set {1, 2, 3} in the network whose weight matrix is

$$\begin{bmatrix} - & 6 & 5 & 1 & 3 & - & - \\ 6 & - & - & - & 2 & 3 & - \\ 5 & - & - & 3 & - & 7 & 2 \\ 1 & - & 3 & - & 2 & - & 1 \\ 3 & 2 & - & 2 & - & - & 4 \\ - & 3 & 7 & - & - & - & 4 \\ - & - & 2 & 1 & 4 & 4 & - \end{bmatrix}$$

Ans. 8

Chapter 6

Flows, Connectivity, and Combinatorics

6.1 FLOWS IN NETWORKS AND MENGER'S THEOREM

The Maximum Flow Problem

A digraph with an integer-valued function c (known as the **capacity function**) defined on its set of arcs is called a **capacitated network.** Two vertices in the network are specially designated: the **source,** with indegree zero, and the **sink,** with outdegree zero. Every other vertex is an **intermediate vertex.** We assume that the set of vertices is $\{1, 2, \ldots, n\}$ in which vertex 1 is the source and vertex n is the sink. If $e = (i, j)$ is an arc in G, the integer $c(e) = c(i, j)$ is the **capacity** of the arc. It is assumed that the capacity of each arc is nonnegative. A **flow** f in the network is an integer-valued function defined on its set of arcs such that $0 \leq f(e) \leq c(e)$ for each arc e. The integer $f(e)$ is the **flow along arc e.** The sum of the flows along all the arcs directed to vertex i is the **inflow into i,** and the sum of the flows along all the arcs directed from vertex i is the **outflow from i.** A flow is a **feasible flow** if it satisfies the **conservation condition:** the inflow into i is equal to the outflow from i for every vertex i other than the source and the sink. If f is a feasible flow in a capacitated network G, the **value $f(G)$ of the flow** is the outflow from the source. A feasible flow in a capacitated network such that the value of the flow is as large as possible is called a **maximum flow** in the network. The problem of finding a feasible flow in a network such that its flow value is maximum is known as the maximum flow problem.

[If there are arcs in either direction between a pair of vertices in a network, a new vertex can be inserted on one of the two arcs, replacing that arc by two arcs in the same direction with the same capacity. So without loss of generality, it can be assumed that the digraph is asymmetric; that is, for distinct vertices i and j, not both (i, j) and (j, i) are arcs.]

Example 1. Figure 6-1 shows a digraph representing a capacitated network (with 1 as the source and 4 as the sink) and a feasible flow defined on its set of arcs. Along each arc are two integers separated by a comma. The first number is the flow along the arc, and the second is its capacity. The flow value is 3, and it is easy to see that it can be increased by four more units using arc (1, 4). So the feasible flow as depicted is not a maximum flow.

Cuts in a Capacitated Network

Consider any partition of the vertex set of a capacitated network $G = (V, E)$ into two sets S and T such that the source is in S and the sink is in T. The set $(S, T) = \{(i, j) : (i, j) \in E, i \in S, j \in T\}$ is called a **cut** (more appropriately, a **source-sink cut**) in the network since no flow can be sent from the source to the sink if all the arcs in the cut are deleted. The sum of the capacities of all the arcs in cut (S, T) is the **capacity $c(S, T)$** of the cut. A cut is called a **minimum cut** if its capacity does not exceed the capacity of any other cut. If f is a feasible flow in the network, the sum of the flows along all the arcs in cut (S, T) is the **flow $f(S, T)$** along the cut. Obviously, $0 \leq f(S, T) \leq c(S, T)$.

[Observe that every capacitated network (V, E) with source s and sink t always has a feasible flow and a source-sink cut: the cut $(\{s\}, V - \{s\})$ is a source-sink cut, and the trivial flow $f = 0$ is a feasible flow.]

Theorem 6.1. If f is any feasible flow in a capacitated network G and if (S, T) is any cut in the network, $f(G) = f(S, T) - f(T, S)$.

Corollary 1: The value $f(G)$ of any feasible flow f in a network is also equal to the inflow into the sink.

Corollary 2: If f is any feasible flow and if (S, T) is any cut, $f(G) \leq c(S, T)$. (See Solved Problems 6.2 and 6.3.)

Example 2. In the network shown in Fig. 6-2, both the inflow into the sink and the outflow from the source for the current flow f are 8, which is the current flow value. If $S = \{1, 2, 3\}$ and $T = \{4, 5, 6\}$, the flow value is also equal to $f(S, T) - f(T, S) = (3 + 4 + 7) - (0 + 6) = 8$. Furthermore, the flow value is less than or equal to the capacity

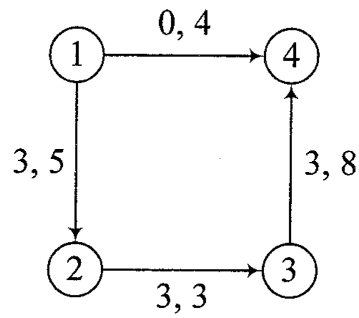


Fig. 6-1

$(7 + 4 + 8) = 19$ of this cut. Suppose $S = \{1, 4, 5\}$ and suppose T is its complement. The flow value again is $f(S, T) - f(T, S) = (5 + 0 + 1 + 6) - 4 = 8$, and the capacity of this cut is $(8 + 4 + 5 + 9) = 26$, which also exceeds the flow value.

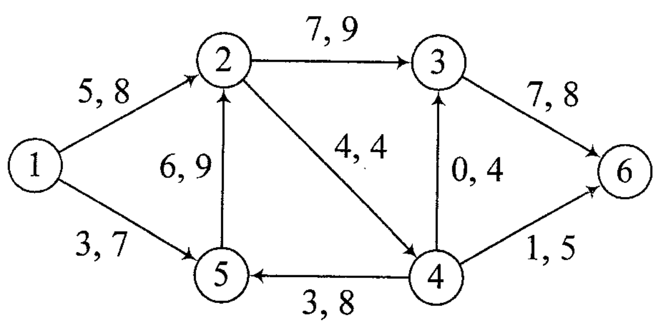


Fig. 6-2

If f is a feasible flow in a capacitated network, the arc (i, j) is **f-saturated** if $f(i, j) = c(i, j)$, is **f-zero** if $f(i, j) = 0$, and is **f-positive** if $f(i, j)$ is positive and less than $c(i, j)$.

Theorem 6.2. (*a*) If f is a feasible flow and if (S, T) is any cut, $f(G) = c(S, T)$ if and only if every arc in (S, T) is f-saturated and every arc in cut (T, S) is f-zero. (*b*) If f is a feasible flow and if (S, T) is any cut such that $f(G) = c(S, T)$, f is a maximum flow and (S, T) is a minimum cut. (See Solved Problem 6.4.)

Example 3. In the digraph shown in Fig. 6-3, the flow value is 12. The capacity of cut (S, T), where $S = \{1, 2, 3, 5\}$ and $T = \{4, 6\}$, is also 12. The arcs in (S, T) are (2, 4) and (3, 6). Both these arcs are f-saturated. The arcs in (T, S) are (4, 5) and (4, 3). Both these arcs are f-zero. The current flow is a maximum flow, and the cut is a minimum cut.

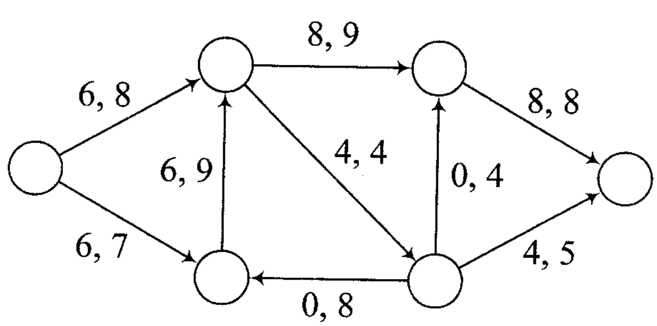


Fig. 6-3

Flow-Augmenting Paths

An alternating sequence P of vertices and arcs of the form $v_0, e_1, v_1, e_2, \ldots, e_k, v_k$ in which no vertex is repeated is called a **semipath** from v_0 to v_k. Arc e_i in this path is a **forward arc** in P if it is directed to v_i. Otherwise, it is a **backward arc** in P. A semipath is ***f*-unsaturated** if no forward arc is f-saturated and no backward arc is f-free. An f-unsaturated path from the source to the sink is called an ***f*-augmenting path.** For each arc e_i in an f-augmenting path P, define $\delta_i(P)$ to be $c(e_i) - f(e_i)$ if e_i is a forward arc and $f(e_i)$ if e_i is a backward arc. The **excess flow capacity of semipath P is $\delta(P)$** $= \min\{\delta_i(P) : e_i \in P\}$, which is a positive integer.

Example 4. In the semipath displayed in Fig. 6-4, the source is 1 and the sink is 9. The only backward arc is the arc from 3 to 4. The excess flow capacity of this path is the minimum in the set $\{8 - 3, 7 - 6, 2, 9 - 4\}$, which is 1.

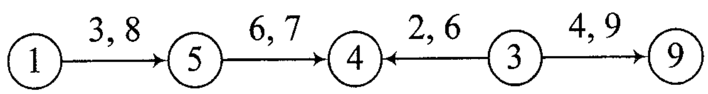


Fig. 6-4

If P is an f-augmenting path with excess flow capacity $\delta(P)$, a new feasible flow f' can be obtained by increasing the flow along each forward arc by $\delta(P)$ and at the same time decreasing the flow along each backward arc by $\delta(P)$ such that $f'(G) = f(G) + \delta(P)$. Once this is accomplished, the semipath is no longer an augmenting path. In the semipath of Example 4, we can increase the flow value by increasing the flow along each forward arc by one unit and at the same time decreasing the flow from the backward arc by one unit. In other words, if there is an f-augmenting path in a network, flow f is not a maximum flow since its flow value can be increased using this augmenting path. It turns out that the converse of this assertion is also true.

Theorem 6.3. A flow f in a capacitated network is a maximum flow if and only if there is no f-augmenting path in the network. (See Solved Problem 6.5.)

Theorem 6.4 **(Ford–Fulkerson Theorem).** In a capacitated network, the value of a maximum flow is equal to the capacity of a minimum cut. This theorem is also known as the **max-flow min-cut theorem.** (See Solved Problem 6.6.)

The Edmonds–Karp Algorithm to Solve the Maximum Flow Problem

Input: $G = (V, E)$ is a capacitated network with a capacity function c and an initial feasible flow f (which could be the trivial flow). The set V is $\{1, 2, \ldots, n\}$ in which the source is 1 and the sink is n. The capacity of each arc is assumed to be a positive integer.

Step 1. Construct a digraph $D(f) = (V, E')$ as follows: (a) If (i, j) is an f-saturated arc in E, (j, i) is an arc in E'. (b) If (i, j) is an f-zero arc in E, (i, j) is also an arc in E'. (c) If (i, j) is an f-positive arc in E, both (i, j) and (j, i) are arcs in E'.

Step 2. Starting from the source, apply a breadth first search (BFS) in $D(f)$ to obtain a directed path (with a minimum number of arcs) from the source to the sink. If there is no such path, go to step 4.

Step 3. The directed path in the digraph $D(f)$ from the source to the sink n obtained in step 2 is a semipath in G with positive excess flow capacity, so it is an f-augmenting path. Increase the flow value in the network by using this path. Go to step 1.

Step 4. The maximum flow value is the outflow from the source. Let S be the set of vertices that can be reached from vertex 1 in $D(f)$, and let T be its complement. (S, T) is a minimum cut. (See Solved Problem 6.7.)

Vertex-Capacitated Networks

In a capacitated network, each arc is associated with a positive integer known as its capacity. If, in addition, each vertex also is associated with a positive integer, the network is called a vertex-capacitated network. A feasible flow in the network is known as a **generalized feasible flow** if the outflow at each intermediate vertex does not exceed the capacity of that vertex. As before, there is a unique vertex known as the source and another unique vertex known as the sink. A set X of arcs and vertices is called a **generalized source-sink cut** if every directed path from the source to the sink will contain at least one element from X. In the network shown in Fig. 6-5, the set $\{(1, 2), (2, 3), (3, 6), 5\}$ is a generalized source-sink cut.

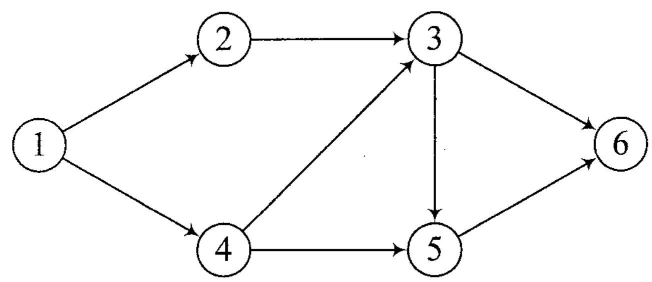


Fig. 6-5

The capacity of a generalized cut is the sum of the capacities of its elements. If the capacity of each vertex is infinite, no vertex can be an element of a minimum generalized cut. Suppose X is a minimum generalized cut and S is the set of all vertices that can be reached from the source without using an arc from X. Then $(S, V - S)$ is a cut such that every arc in it is an arc in X. Thus a minimum cut in the usual sense is a minimum cut in the generalized sense when the capacity of each vertex is infinite.

Theorem 6.5 **(Generalized Max-Flow Min-Cut Theorem).** The maximum value of a generalized source-sink flow in a vertex-capacitated network G is equal to the capacity of a minimum generalized source-sink cut. (See Solved Problem 6.11.)

Example 5. A generalized maximum flow and a generalized minimum cut in the network represented by Fig. 6-6 are as follows: the generalized cut is $\{(2, 4), 3\}$, and the flow is as indicated on the arcs.

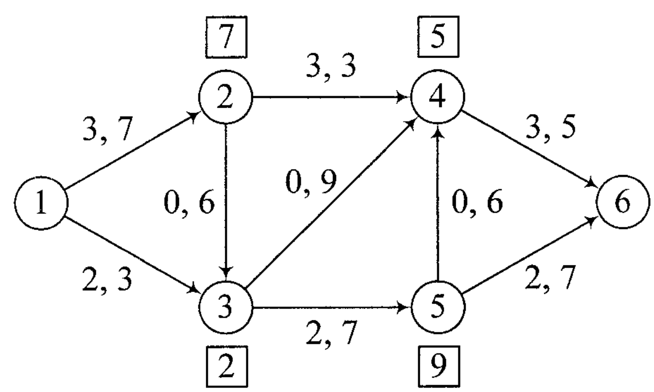


Fig. 6-6

Two paths from source s to sink t in a network are called **internally disjoint paths** from s to t if they have no vertices in common other than s and t. Any two paths are called **arc-disjoint paths (edge-disjoint paths** in the case of undirected graphs) if they have no arcs (edges) in common.

Theorem 6.6 **(Menger's Theorem).**

(*a*) *Vertex form for undirected graphs:* The maximum number of internally disjoint paths between any two nonadjacent vertices in a graph is equal to the minimum number of vertices whose deletion results in a graph in which there are no paths between those two vertices.

(*b*) *Edge form for undirected graphs:* The maximum number of edge-disjoint paths between two vertices s and t in a graph is equal to the minimum number of edges whose deletion results in a graph in which there are no paths between the two vertices.

(*c*) *Vertex form for directed graphs:* The maximum number of internally disjoint paths from vertex s to vertex t in a digraph in which there are no arcs from s to t is equal to the minimum number of vertices whose deletion results in a digraph in which there are no paths from s to t.

(*d*) *Arc form for directed graphs:* The maximum number of arc-disjoint paths from vertex s to vertex t in a digraph is equal to the minimum number of arcs whose deletion results in a digraph in which there are no paths from s to t.

(See Solved Problems 6.13 through 6.16.)

Theorem 6.7. The vertex form of Menger's theorem, the arc (edge) form of Menger's theorem, and the Ford–Fulkerson theorem are equivalent. (See Solved Problem 6.19.)

6.2 MORE ON CONNECTIVITY

Vertex Connectivity

Recall that a graph is connected if and only if there is a path between every pair of vertices in it. Since loops do not have any significance in flow problems as well as in problems related to connectivity, we assume that the graphs and digraphs we investigate in this chapter have no loops. A set W of vertices in a graph $G = (V, E)$ is a **separating set** (also known as **a vertex cut**) **of *G*** if $G - W$ has more than one component. If a separating set consists of a single vertex w, w is known as a **cut vertex** (or **articulation vertex**). The **connectivity number $\kappa(G)$** of a graph G is the minimum size of a separating set in it. Since a complete graph has no separating set, we adopt the convention that the connectivity number of a complete graph of order n is $(n - 1)$ for all n. A graph G is said to be ***k*-connected** if $\kappa(G) \geq k$. Thus K_n is $(n - 1)$-connected for all n, and a graph that is not complete is k-connected if and only if every separating set in it has at least k vertices. The connectivity number of a graph is zero if and only if it is either a disconnected graph or the trivial graph.

Theorem 6.8. A graph of order n is k-connected (where $1 \leq k \leq n - 1$) if the degree of each vertex is at least $(n + k - 2)/2$. (See Solved Problem 6.37.)

Example 6. In the graph of Fig. 6-7, there are six vertices. If we take $k = 2$, we see that the degree of each vertex is at least $(6 + 2 - 2)/2 = 3$. So the graph is 2-connected. Notice that the deletion of the vertices 2 and 5 results in a disconnected graph. Thus the connectivity number of the graph is 2. Since $(6 + 3 - 2)/2$ is more than 3, the graph is not 3-connected because there is a vertex of degree 3 in the graph.

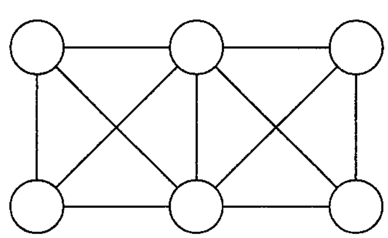


Fig. 6-7

A **nonseparable graph** is a connected nontrivial graph with no cut vertex. A **block** of a graph G is a maximal nonseparable subgraph H; that is, if H' is a nonseparable subgraph of G such that H is a subgraph of H', $H = H'$. Observe that a graph is 2-connected if and only if it is a block with at least two edges. K_2 is the only block that is not 2-connected. Any cycle in a graph is a block of that graph. If H is a subgraph of G, it is not necessary that $\kappa(H) \leq \kappa(G)$; consider the case that H is a block of a 1-connected graph G that is not 2-connected. Obviously, if H is a spanning subgraph of G, $\kappa(H) = \kappa(G)$. (See Problem 6.32 for a complete characterization of blocks due to Harary.)

As mentioned before, a graph is connected (that is, 1-connected) if and only if there is at least one path between every pair of vertices in it. More generally, a graph is k-connected if and only if there are at least k paths between every pair of vertices in it. The pleasant fact is that no two of these paths have any intermediate vertex in common. This remarkable characterization of k-connected graphs, which can be easily established using (the vertex form) of Menger's theorem, is due to Whitney.

Theorem 6.9 **(Whitney's Theorem).** A graph with at least $(k + 1)$ vertices is k-connected if and only if any two distinct vertices in the graph are connected by at least k internally disjoint paths. In particular, a graph with at least three vertices is a block if and only if every two vertices lie on a common cycle. (See Solved Problem 6.30.)

Example 7. In the graph of Fig. 6-7, the number of internally disjoint paths between every pair of vertices is 2. So, by Theorem 6.9, the graph is 2-connected but not 3-connected.

Edge Connectivity

A set F of edges in a graph G is a **disconnecting set** if $G - F$ has more than one component. If a disconnecting set consists of a single edge, that edge is called a **bridge** (also known as a **cut edge** or an **isthmus**). A graph is said to be **k edge connected** if every disconnecting set has at least k edges. The **edge-connectivity number $\lambda(G)$** of a graph G is the minimum size of a disconnecting set in it and, *by definition,* is zero when G is the trivial graph. Thus $\lambda(G)$ is zero if and only if G is disconnected or trivial, and it is k edge connected if and only if $\lambda(G) \geq k$.

The following result is known as **Whitney's inequality.** For any graph G, $\kappa(G) \leq \lambda(G) \leq \delta(G)$, where $\delta(G)$ is the minimum vertex degree of the graph. (See Solved Problems 2.11 and 2.12. So any k-connected graph is k edge connected.

Example 8. In the graph G of Fig. 6-7, the minimum degree $\delta(G)$ is 3, and $\kappa(G) = 2$. So $\lambda(G) = 2$ or 3. The graph does not become disconnected by removing any set of two edges, but it becomes disconnected if the three edges adjacent to a vertex of degree 3 are deleted. So $\lambda(G) = 3$.

The following characterization of k-edge-connected graphs is the "edge version" of Theorem 6.9.

Theorem 6.10. A graph is k edge connected if and only if any two distinct vertices in it are connected by at least k edge-disjoint paths. (See Solved Problem 6.39.)

Example 9. In the graph of Fig. 6-7, joining any two vertices are three edge-disjoint paths. This confirms that its edge-connectivity number is 3.

Cuts and Cut Sets

If the set V of vertices in a graph G is partitioned into two nonempty subsets, cut (S, T) of all edges of the graph joining vertices in S and vertices in T as defined in the previous section is indeed a disconnecting set of the graph. But an arbitrary disconnecting set need not be a cut. For example, the set of the three edges in the complete graph of order 3 is a disconnecting set but not a cut. A disconnecting set F of the graph G is called a **cut set** (also known as a **bond**) if no proper subset of F is a disconnecting set. Suppose F is such a minimal disconnecting set and H is a component obtained from G after deleting all the edges of F from the graph. If W

is the set of vertices of H, the minimality of F implies that F is cut $(W, V - W)$. Thus every cut set is a cut. But the converse is not true: the set of two edges in $K_{1,2}$ forms a cut, but it is not a cut set. Since every cut set is a cut as well as a minimal disconnecting set, the edge-connectivity number of a graph is the minimum size of a cut. So the following characterization of the class of k-edge-connected graphs is obvious.

Theorem 6.11. A graph is k edge connected if and only if the number of edges in any cut is at least k.

6.3 SOME APPLICATIONS TO COMBINATORICS

Matchings and Coverings

The number of vertices in a maximum independent set in graph G is denoted by $\alpha(G)$, and the number of vertices in a minimum vertex cover in G is denoted by $\beta(G)$. Likewise, the number of edges in a maximum matching in a graph G is denoted by $\alpha_1(G)$, and the number of edges in a minimum edge cover is denoted by $\beta_1(G)$. It has been proved that $\alpha(G) + \beta(G) = \alpha_1(G) + \beta_1(G) = n$, where n is the order of G. See Solved Problems 2.19, 2.20, 2.26, 2.27, and 2.28.

Theorem 6.12 **(Konig's Theorem).** In a bipartite graph G, $\alpha_1(G) = \beta(G)$. [Consequently, $\alpha(G) = \beta_1(G)$ if G has no vertex of degree 0. This equality is known as Konig's other theorem.] (See Solved Problems 6.44 and 6.46.)

Example 10. In the bipartite graph of Fig. 6-8, a maximum matching consists of three edges $\{1, 6\}$, $\{2, 8\}$, and $\{5, 9\}$. A minimum vertex cover consists of vertices 1, 5, and 8. The six vertices $\{2, 3, 4, 6, 7, 9\}$ form a maximum independent set, and the set of all edges in the graph except edge $\{5, 8\}$ forms a minimum edge cover.

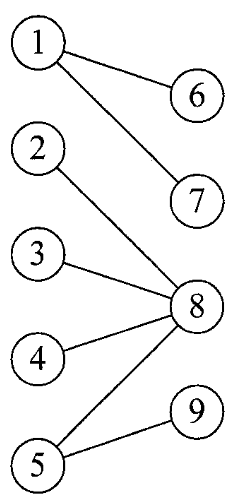


Fig. 6-8

In a bipartite graph (X, Y, E), a **complete matching from X to Y** is a matching M such that every vertex in X is incident to an edge in M, and a **perfect matching** is a matching that is complete from X to Y as well as from Y to X. If both X and Y have the same number of elements, a complete matching from one to the other is a perfect matching.

Systems of Distinct Representatives

Suppose we have a family of N sets. The possibility that two sets in the family are the same is not to be ruled out. In other words, the family could be just a collection, not a set. If it is possible to select one element from each set in the family such that the N selected elements constitute a set X, X is called a **system of distinct representatives (SDR)** of the family. For example, if the family consists of the sets $\{1, 2, 3\}$, $\{3, 4\}$, $\{1, 2, 3\}$, and $\{5, 6, 7\}$, one can select 1, 3, 2, and 5, respectively, from these four sets to form an SDR of the family.

Theorem 6.13 **(Hall's Marriage Theorem).** A family of N sets (not necessarily distinct) will have a system of distinct representatives if and only if the following condition, known as the **marriage condition (MC),** is satisfied: The union of any subfamily of k sets from the family should have at least k elements for every k from the set $\{1, 2, \ldots, N\}$. (Konig's theorem implies this theorem. See Solved Problem 6.47.)

This theorem, which was first proved by Philip Hall in 1935 settling the following issue known as the marriage problem, has several proofs. Suppose there are m women and n men in a party, where $m \leq n$. A necessary and sufficient requirement that we can form a set of m couples (each couple consisting of a woman and a man known to each other) is that every set of k women in the party collectively know at least k men in the party for every choice of k. The following theorem is a graph-theoretic version of Hall's marriage theorem.

Theorem 6.14. In the bipartite graph (X, Y, E), a complete matching from X to Y exists if and only if $|f(A)| \geq |A|$ for every subset A of X, where $f(A)$ is the set of those vertices in Y that are adjacent to at least one vertex in A.

The *P*-Rank of a Matrix

A **line** in a matrix A is either a row or a column. Suppose P is a property that an element in A may not have. A collection of elements in the matrix satisfying the property P is ***P*-independent** if no two elements in the collection lie on the same line. The ***P*-rank** of A is the number of elements in a largest P-independent collection in A.

Theorem 6.15 **(Konig–Egervary Theorem).** The P-rank of a matrix is equal to the minimum number of lines that contain all the elements of the matrix that possess the property P. [Notice that both this theorem and Konig's theorem (Theorem 6.12) make the same assertion: the former in the context of matrices, and the latter in the context of bipartite graphs. Hall's marriage theorem implies this theorem and therefore Konig's theorem. See Solved Problem 6.48.]

Example 11. In the matrix consisting of a few letters of the alphabet,

$$\begin{bmatrix} a & e & a & p \\ q & r & e & q \\ p & p & u & q \\ s & p & a & r \\ t & p & e & e \end{bmatrix}$$

an element is supposed to have the property P if it is a vowel. All the vowels in the matrix can be covered by three lines: row 1, row 5, and column 3. A largest P-independent set consists of the first vowel a in row 1, the vowel e from row 2, and the last vowel e from row 5. Thus the P-rank is 3, which is equal to the minimum number of lines needed to cover all the vowels in the matrix.

Theorem 6.16. Konig's theorem implies Menger's theorem. (See Solved Problem 6.49.)

Theorem 6.17 **(Konig's Marriage Theorem).** If a bipartite graph $G = (X, Y, E)$ is k-regular (where k is positive), there is a perfect matching in the graph. (Hall's marriage theorem implies this theorem. See Solved Problem 6.50.)

Partially Ordered Sets and Dilworth's Theorem

A **partially ordered set** (or **poset**) consists of a set X and an order relation $\leq$ among its elements satisfying the following three properties: (a) reflexivity: $x \leq x$ for every x in X, (b) antisymmetry: $x \leq y$ and $y \leq x$ imply that $x = y$, and (c) transitivity: $x \leq y$ and $y \leq z$ imply that $x \leq z$. Given a poset with a finite number of elements, one can construct a digraph G such that there is a one-to-one correspondence between the set of elements in the poset and the set of vertices of the digraph and such that an arc is drawn from the vertex that corresponds to element x to the vertex that corresponds to element y if and only if $x \leq y$. If all the arcs that are present in G due to transitivity are deleted, we get a (unique) subgraph H known as the **Hasse diagram** of the poset.

Example 12. The digraph of Fig. 6-9 represents a poset with 13 elements. There is an arrow from 1 to 6 and an arrow from 6 to 9. In addition to the relations $1 \leq 6$ and $6 \leq 9$, we also have, due to transitivity, the relation $1 \leq 9$, which is not explicitly shown by an arrow from 1 to 9.

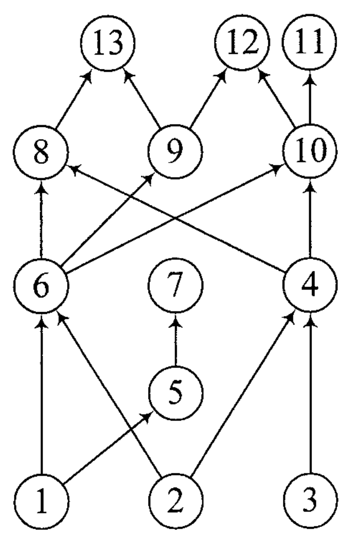


Fig. 6-9

Two elements x and y in a poset are **comparable** if either $x \leq y$ or $y \leq x$. Otherwise, they are **incomparable.** A set of elements belonging to a poset is called a **chain** (or **linear order**) in the poset if every pair in the set is a comparable pair. A set C of n elements in a poset is a chain if and only if the elements in C can be labeled as y_i $(i = 1, 2, \ldots, n)$ such that $y_1 \leq y_2 \leq \cdots \leq y_n$. An **antichain** in a poset, on the other hand, is a set in which no two elements are comparable.

Theorem 6.18 **(Dilworth's Theorem).** In a finite poset, the maximum size of an antichain is equal to the minimum number of chains into which the set of elements of the poset can be partitioned. (The Konig–Egervary theorem implies this theorem. See Solved Problem 6.51.)

Example 13. In the poset represented in Fig. 6-9, the set of the 13 elements can be partitioned into four chains $\{1, 5, 7\}$, $\{2, 6, 8, 13\}$, $\{3, 4, 10, 11\}$, and $\{9, 12\}$. The set $\{7, 8, 9, 10\}$ is an antichain with four elements in the poset.

Theorem 6.19. Dilworth's theorem implies Hall's marriage theorem. (See Solved Problem 6.52.)

Theorem 6.20 **(The Equivalence Theorem).** Menger's theorem, the Ford–Fulkerson theorem, Konig's theorem, the Konig–Egervary theorem, Dilworth's theorem, and Hall's marriage theorem are equivalent.

Some of the implications connecting these six famous theorems and resulting in their equivalence are indicated in Fig. 6-10.

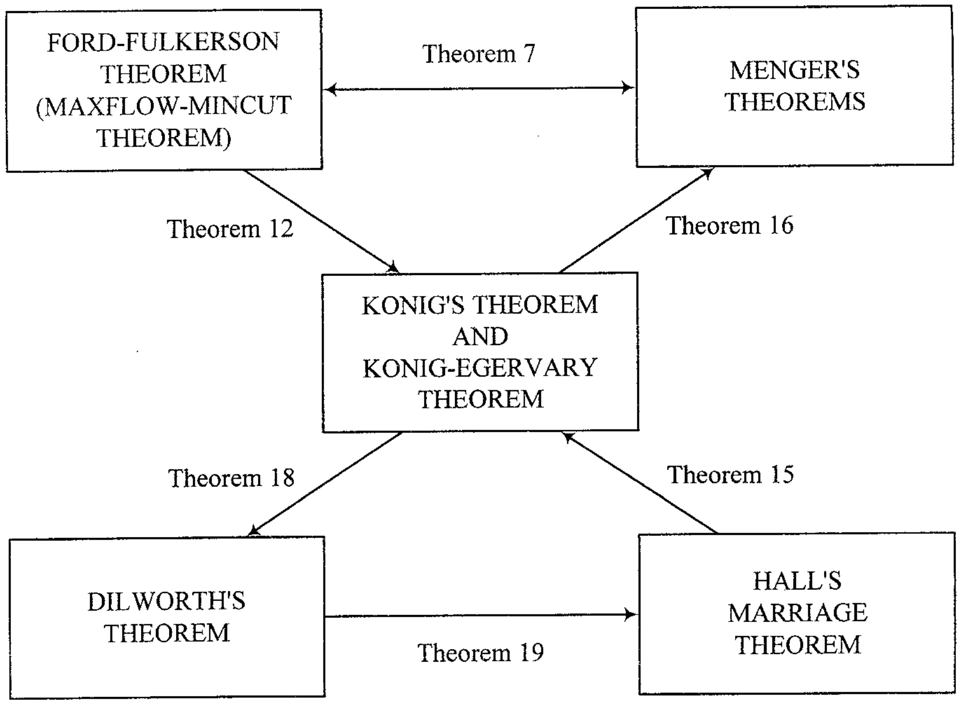


Fig. 6-10

Solved Problems

FLOWS IN NETWORKS AND MENGER'S THEOREM

6.1 In network G shown in Fig. 6-11, the flow and the capacity are as indicated on the arcs. Vertex 1 is the source, and vertex 6 is the sink. If $S = \{1, 4, 5\}$ and $T = \{2, 3, 6\}$, show that the flow value $f(G)$ is equal to $f(S, T) - f(T, S)$. Show that $f(G)$ does not exceed the capacity $c(S, T)$ of cut (S, T).

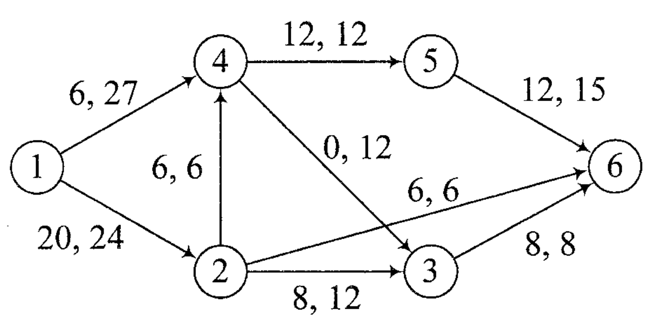


Fig. 6-11

Solution. Here the flow value $f(G)$ is $6 + 20 = 12 + 6 + 8 = 26$, $f(S, T) = 20 + 0 + 12 = 32$, and $f(T, S) = 6$. Thus $f(G) = f(S, T) - f(T, S)$. The capacity of cut (S, T) is $27 + 12 + 15 = 54$.

6.2 Prove Theorem 6.1: If f is any feasible flow in a capacitated network and if (S, T) is any cut in the network, $f(G) = f(S, T) - f(T, S)$.

Solution. The vertex set is $V = \{1, 2, \ldots, n\}$. S is any set of vertices that contains vertex 1 (the source), and T is its complement that contains vertex n (the sink). Notice that $\Sigma_j f(i, j) - \Sigma_j f(j, i)$ is $f(G)$ when $i = 1$. So $\Sigma_{i \in S} \Sigma_j f(i, j) - \Sigma_{i \in S} \Sigma_j f(j, i) = f(G)$. If i and j are both in S, the term $f(i, j)$ appears in the first summation

$\Sigma_{i \in S} \Sigma_j f(i, j)$ as well as in the second summation $\Sigma_{i \in S} \Sigma_j f(j, i)$. So it is enough if we let the subscript j vary for all j in T. Hence, $\Sigma_{i \in S} \Sigma_{j \in T} f(i, j) - \Sigma_{i \in S} \Sigma_{j \in T} f(j, i) = f(G)$.

6.3 Prove the two corollaries of Theorem 6.1. (*a*) Corollary 1: The value $f(G)$ of any feasible flow f in a network is also equal to the inflow into the sink. (*b*) Corollary 2: If f is any feasible flow and if (S, T) is any cut, $f(G) \leq c(S, T)$.

Solution. (*a*) In Theorem 6.1, let $S = V - \{n\}$. In that case $f(T, S)$ is zero, and $f(S, T)$ is the inflow into the sink. (*b*) $f(G) = f(S, T) - f(T, S) \leq f(S, T) \leq c(S, T)$.

6.4 Prove Theorem 6.2: (*a*) If f is a feasible flow and if (S, T) is any cut, $f(G) = c(S, T)$ if and only if every arc in (S, T) is f-saturated and every arc in cut (T, S) is f-zero. (*b*) If f is a feasible flow and if (S, T) is any cut such that $f(G) = c(S, T)$, f is a maximum flow and (S, T) is a minimum cut.

Solution.

(*a*) $f(G) = c(S, T)$ if and only if $f(S, T) - f(T, S) = c(S, T)$. This is possible if and only if every arc in cut (T, S) is f-zero and every arc in cut (S, T) is f-saturated.

(*b*) Let f' be a maximum flow and (S', T') be a minimum cut in the network. Then $f(G) \leq f'(G) \leq c(S', T') \leq c(S, T)$. If $f(G)$ and $c(S, T)$ are equal, we have the chain $f(G) = f'(G) = c(S', T') = c(S, T)$, which implies that f is indeed a maximum flow and (S, T) is a minimum cut.

6.5 Prove Theorem 6.3: A flow f in a capacitated network is a maximum flow if and only if there is no f-augmenting path in the network.

Solution. If there is an f-augmenting path in the network, the current flow value can be increased using this path; therefore, the current flow is not a maximum flow. So if the current flow f is a maximum flow, there are no f-augmenting paths. We now show that if there are no f-augmenting paths with respect to flow f, f is indeed a maximum flow. Let S be the set of all vertices i such that there is an f-unsaturated path from the source (vertex 1) to i. Obviously, 1 is a vertex in S and the sink (vertex n) is not a vertex in S. Thus there is a cut (S, T) in the network. Let (i, j) be any arc in this cut. Since there is no f-unsaturated path from the source to j, arc (i, j) is necessarily f-saturated. Likewise, any arc in cut (T, S) is f-zero. So the current flow value is equal to the capacity of cut (S, T). Hence, the current flow is a maximum flow.

6.6 Prove Theorem 6.4. (Ford–Fulkerson Theorem): In a capacitated network, the value of a maximum flow is equal to the capacity of a minimum cut.

Solution. Let f be a maximum flow, and let S be the set of vertices i such that there is an f-unsaturated path from the source to i. Then the complement T of S is nonempty; thus there is a cut (S, T). Each arc in this cut is f-saturated. Moreover, each arc in cut (T, S) is f-zero. So the flow value $f(G)$ is equal to $c(S, T)$, which is a minimum cut.

6.7 Let $G = (V, E)$ be a capacitated network with capacity function c and initial feasible flow f (which could be the trivial flow). The set V is $\{1, 2, \ldots, n\}$ in which the source is 1 and the sink is n. The capacity of each arc is assumed to be a positive integer. Construct a digraph $D(f) = (V, E')$ as follows. (1) If (i, j) is an f-saturated arc in E, (j, i) is an arc in E'. If (i, j) is an f-zero arc in E, (i, j) is also an arc in E'. (3) If (i, j) is an f-positive arc in E, both (i, j) and (j, i) are arcs in E'. Show that there is an f-augmenting path in the network if and only if there is a directed path from the source to the sink in $D(f)$, and show that a shortest source-sink path in $D(f)$ has the same length as a shortest f-augmenting path.

Solution. Let Q be a directed path in $D(f)$ of the form $v_0, e_1, v_1, e_2, \ldots, e_k, v_k$ in which no vertex is repeated. This alternating sequence of vertices and arcs forms a semipath Q' in the network. Arc e_i in Q from v_{i-1} to v_i corresponds to an arc in Q' that is either from v_{i-1} to v_i or in the opposite direction. In the former case, it is an f-unsaturated arc. In the latter case, it is an arc that is not f-zero. Thus no forward arc in Q' is f-saturated, and no backward arc is f-zero. So Q' is an f-unsaturated path. Similarly, it can be shown that any f-unsaturated path Q' in the network corresponds to a directed path Q in $D(f)$. Furthermore, both Q and Q' have the same number of arcs. So a shortest source-sink path in the digraph has the same number of arcs as a shortest f-augmenting path.

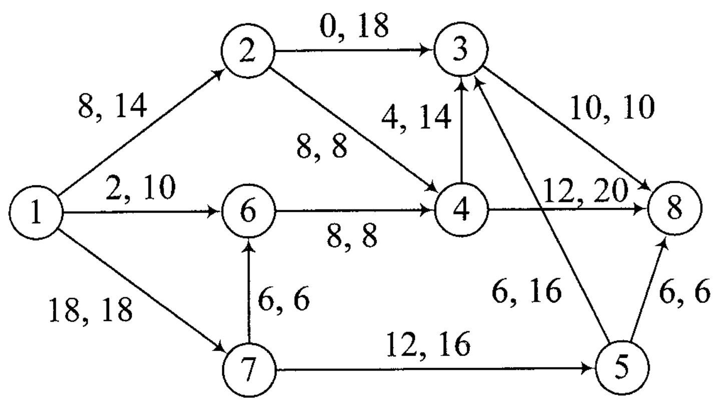


Fig. 6-12a

6.8 Obtain a maximum flow and a minimum cut in the network shown in Fig. 6.12(a).

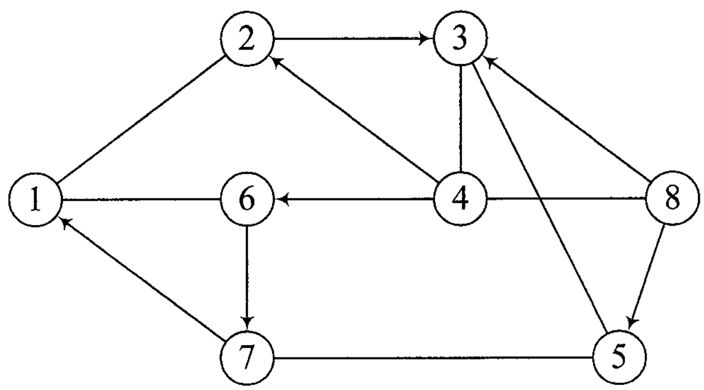


Fig. 6-12b

Solution. Here 1 is the source and 8 is the sink. The digraph $D(f)$ corresponding to the current flow in which the undirected edges are considered arcs in either direction is shown (for the sake of convenience) as a mixed graph in Fig. 6.12(b). In this digraph, $1 \to 2 \to 3 \to 4 \to 8$ is a directed path from the source to the sink, defining the flow augmenting (semi)path $1 \to 2 \to 3 \leftarrow 4 \to 8$ along which a flow of four units can be sent from 1 to 8. The updated network is shown in Fig. 6-12(c).

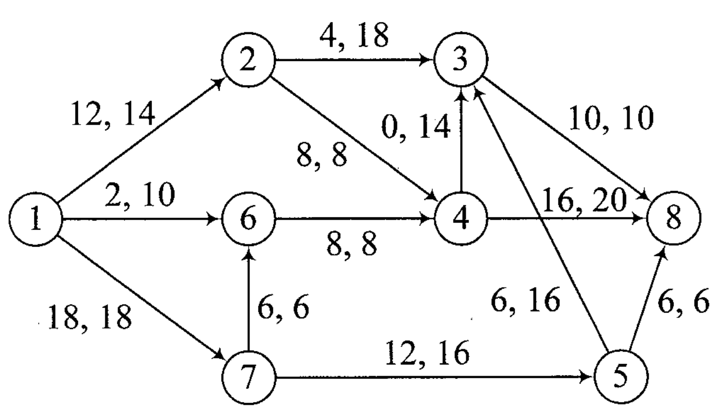


Fig. 6-12c

The digraph corresponding to this revised flow in which there is no directed path from 1 to 8 is shown in Fig. 6-12(d). So the updated flow is a maximum flow with flow value 32. In this digraph, the vertices not reachable from 1 are 4 and 8. Thus $T = \{4, 8\}$ and $S = V - T$. Cut $(S, T) = \{(2, 4), (6, 4), (3, 8), (5, 8)\}$ with a capacity of $8 + 8 + 10 + 6 = 32$ is a minimum cut.

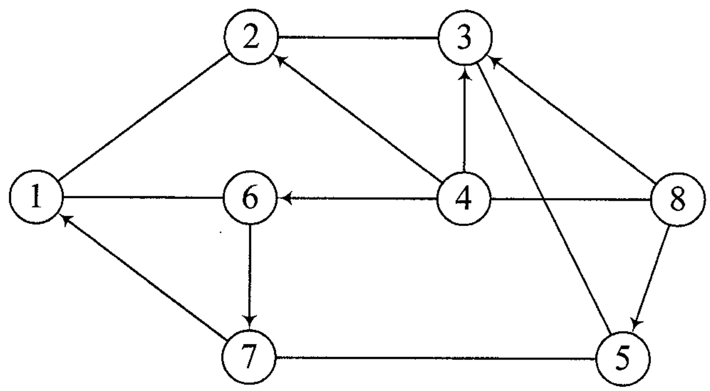


Fig. 6-12d

6.9 Obtain a maximum flow and a minimum cut in the network shown in Fig. 6-13(a).

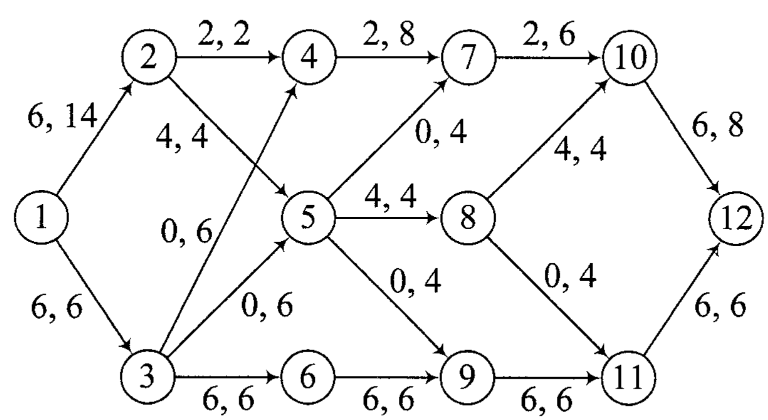


Fig. 6-13a

Solution. The current flow value from the source (vertex 1) to the sink (vertex 12) is 12. The mixed graph corresponding to this flow is shown in Fig. 6-13(b). In this mixed graph, there is no directed path from the source to the sink. So the current flow with flow value 12 is a maximum flow. The only vertex reachable from the source is vertex 2. Thus $S = \{1, 2\}$, and T is the remaining set of vertices. Cut (S, T), consisting of arcs (1, 3), (2, 4) and (2, 5) with capacity $6 + 2 + 4 = 12$, is a minimum cut.

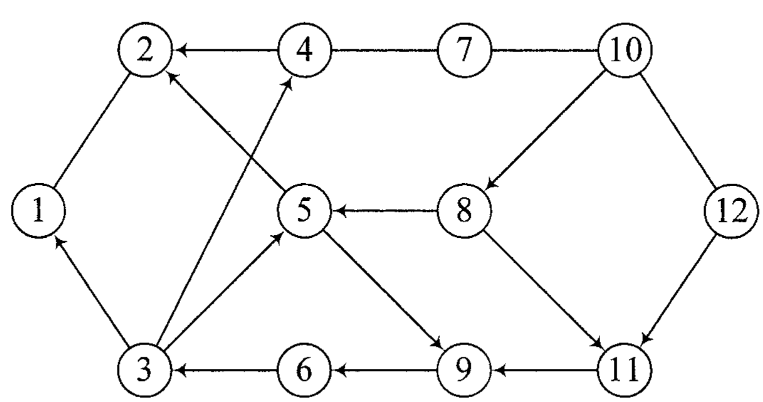


Fig. 6-13b

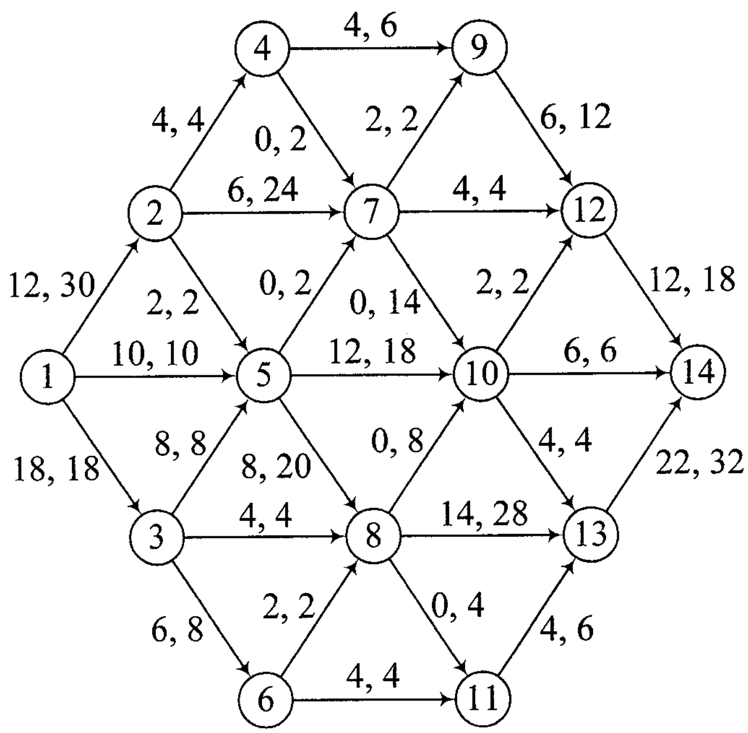


Fig. 6-14a

6.10 Obtain a maximum flow and a minimum cut in the network shown in Fig. 6-14(a).

Solution. The current flow value from the source (vertex 1) to the sink (vertex 14) is 40. A cursory look at the network (by someone who is not familiar with the theory) gives the (false) impression that it is not possible to

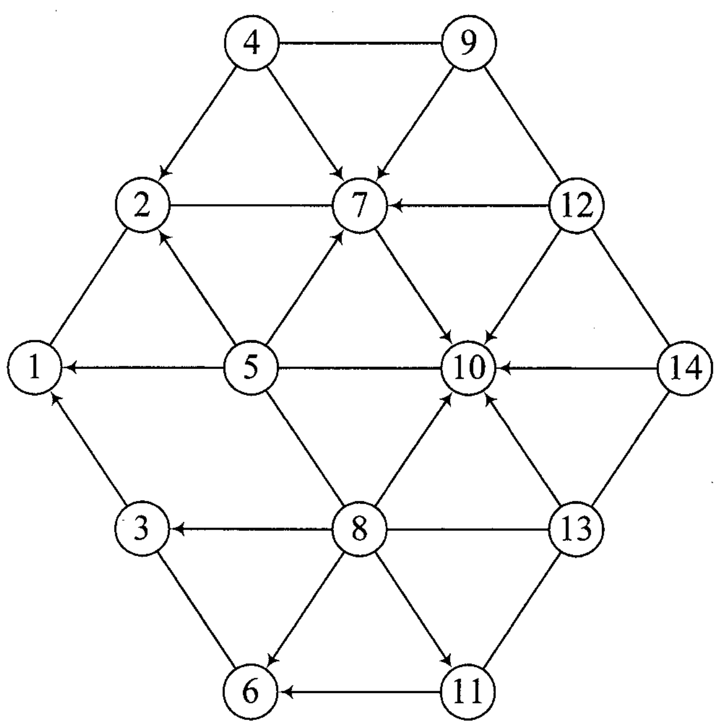


Fig. 6-14b

increase the flow value. That there is a directed path in the mixed graph shown in Fig. 6-14(*b*) from the source to the sink indicates that the current flow is not a maximum flow.

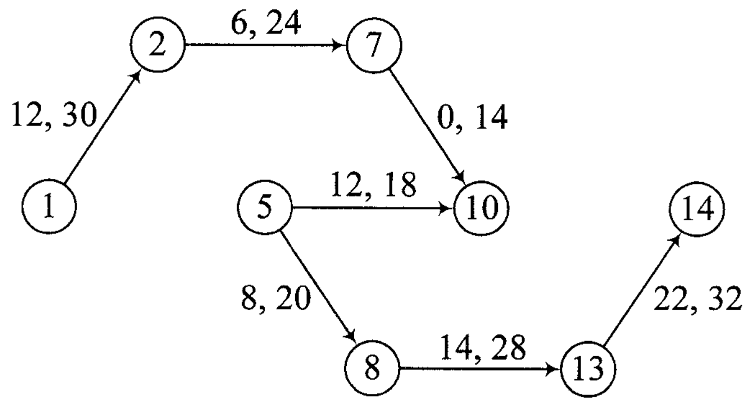


Fig. 6-14c

A directed path in the mixed graph from the source to the sink defines the semipath as shown in Fig. 6-14(*c*), which is a flow augmenting path. Using this semipath, the flow value can be increased by 10 more units. The updated network with a flow value of 50 units is shown in Fig. 6-14(*d*). The mixed graph corresponding to the updated flow in which there is no directed path from the source to the sink is shown in Fig. 6-14(*e*). So the updated flow is a maximum flow.

In the mixed graph, the set T contains vertices that are not reachable from the source are 4, 9, 12, and 14. The remaining vertices constitute set S. Cut (S, T), consisting of arcs (2, 4), (7, 9), (7, 12), (10, 12), (10, 14), and

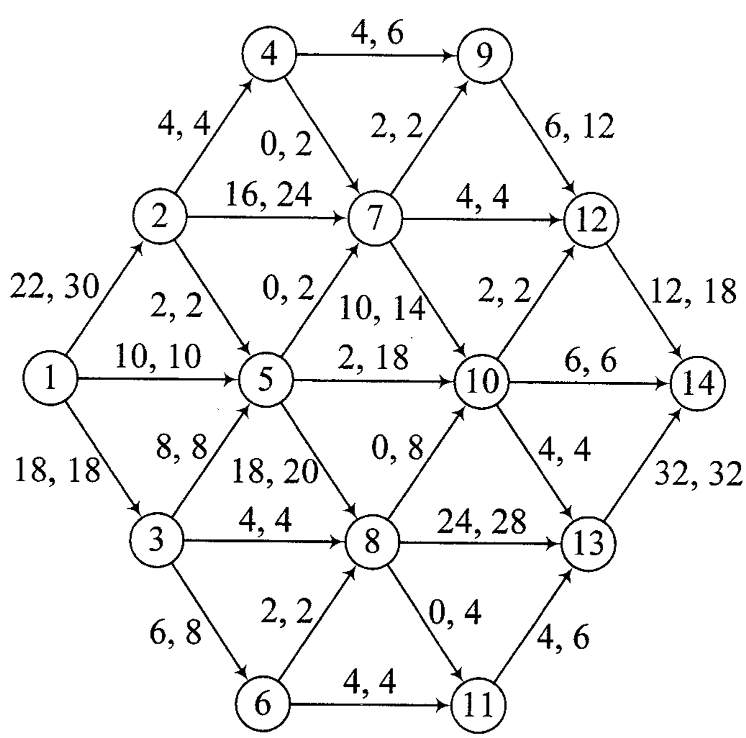


Fig. 6-14d

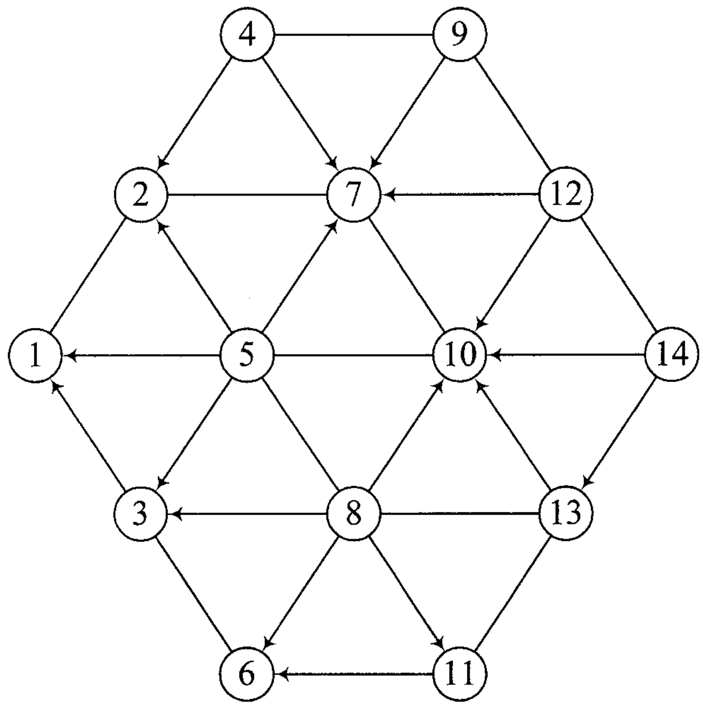


Fig. 6-14e

(13, 14), has a capacity of 4 + 2 + 4 + 2 + 6 + 32, which is equal to the current flow value. If these six arcs are deleted from the network, we get a digraph as shown in Fig. 6-14(f) in which the dotted lines are the deleted arcs. It is easy to see that there is no directed path from the source to the sink in the digraph without the dotted lines.

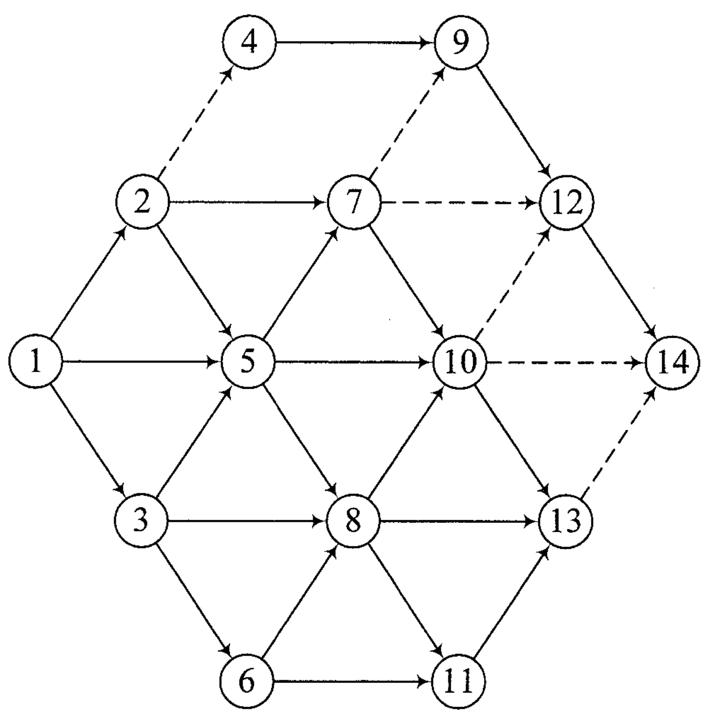


Fig. 6-14f

6.11 Prove Theorem 6.5: The maximum value of a generalized source-sink flow in a vertex-capacitated network G is equal to the capacity of a minimum generalized source-sink cut.

Solution. Let s be the source, and let t be the sink of G. Then we can construct an s-t network G' (which is not vertex-capacitated) as follows. Each intermediate vertex i of weight w_i is replaced by two intermediate vertices i' and i'' along with an arc (i', i'') of capacity w_i. At the same time, every arc (u, v) of the network is replaced by arc (u'', v'). (See Problem 6.12, which illustrates this construction.) Let $s = s' = s''$ and $t = t' = t''$. Any flow entering i' must pass through i'', and all flow leaving f'' must come from f. So there is a one-to-one correspondence between generalized flows in G and feasible flows in G'. Thus the maximum-flow minimum-cut theorem in G' implies the generalized theorem in the vertex-capacitated network G.

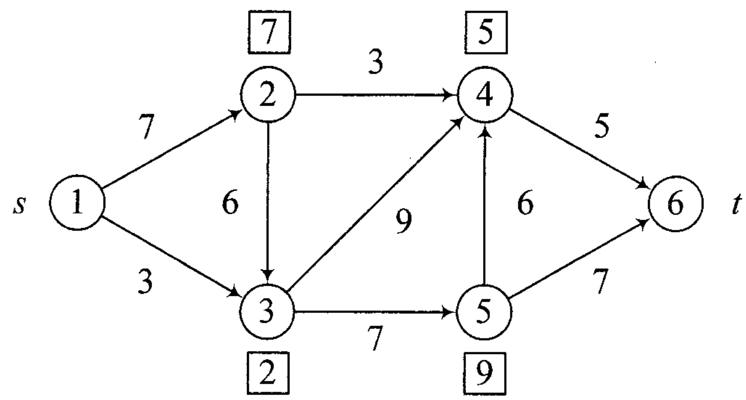


Fig. 6-15a

6.12 Obtain a generalized maximum flow and a generalized minimum cut in the vertex-capacitated network in which vertex 1 is the source and vertex 6 is the sink shown in Fig. 6-15(a).

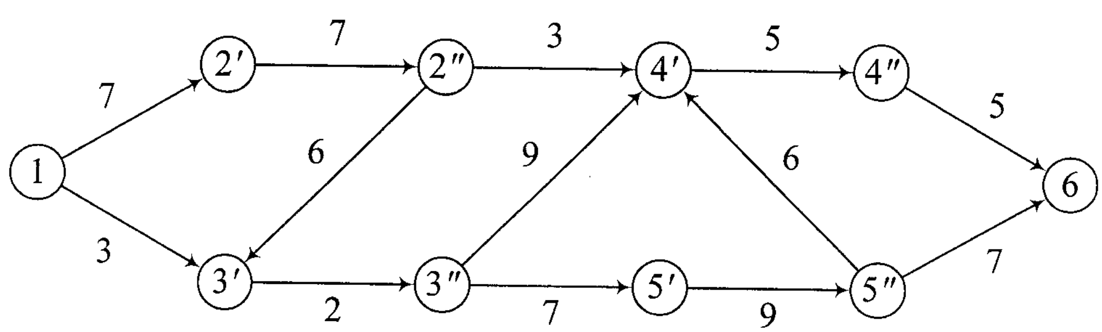


Fig. 6-15b

Solution. The expanded network G' is shown in Fig. 6-15(b). The maximum flow value in G' is 5. A minimum cut is $\{(2'', 4'), (3', 3'')\}$. Arc $(2', 4'')$ corresponds to arc (2, 4) in G. Arc $(3', 3'')$ corresponds to vertex 3 in G. Thus a generalized minimum cut consists of arc (2, 4) and vertex 3. The generalized maximum flow in the given network is shown in Fig. 6-15(c).

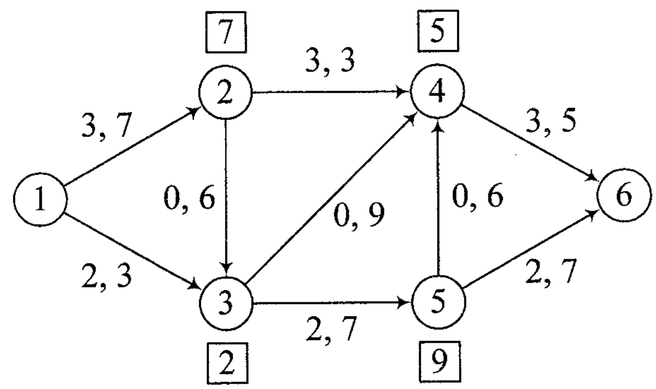


Fig. 6-15c

6.13 (*Menger's Theorem: Vertex Form for Directed Graphs*) Show that the maximum number of internally disjoint paths from vertex s to vertex t in a digraph in which there are no arcs from s to t is equal to the minimum number of vertices whose deletion results in a digraph in which there are no paths from s to t.

Solution. Without loss of generality, it can be assumed that the indegree of s and the outdegree of t are both zero. If there are k internally disjoint paths from s to t, the minimum number of vertices whose deletion will result in a digraph in which t is not reachable from s is obviously k. To prove the converse, let us assume that (1) there exists a set of k vertices whose deletion will result in a digraph in which sink t is not reachable from source s in the vertex-capacitated network, and (2) there will be at least one path from s to t in the network obtained after deleting any set of $(k - 1)$ vertices from the graph. The given network is considered a vertex-capacitated network in which the capacity of each vertex is 1 and the capacity of each arc is infinite. By our hypothesis, the (generalized) minimum cut value is k. So the (generalized) maximum flow is k. Since we can send only one unit of flow through a vertex from the source to the sink and since there is no arc from s to t, there should be exactly k internally disjoint paths from the source to the sink.

6.14 (*Menger's Theorem: Arc Form for Directed Graphs*) Show that the maximum number of arc-disjoint paths from vertex s to vertex t in a digraph is equal to the minimum number of arcs whose deletion results in a digraph in which there are no paths from s to t.

Solution. Without loss of generality, we can assume that the indegree of s as well as the outdegree of t are both zero. If there are k arc-disjoint paths from s to t, the minimum number of arcs whose deletion will result in a digraph in which t is not reachable from s is obviously k. To prove the converse, assume that (1) there exists a set of k arcs whose deletion results in a digraph in which there is no path from s to t and (2) there is a path from s to t in the digraph obtained by deleting any set of $(k - 1)$ arcs. Suppose the capacity of each arc is 1. So the minimum cut value is k, implying that the maximum flow value is also k. We now use induction on k to establish that there are k arc-disjoint paths from the source to the sink. Let I be the set of all positive integers n such that if there is a flow (with integer components) with flow value n, there are n arc-disjoint paths from s to t. Obviously, $1 \in I$. Suppose $(k - 1) \in I$, and suppose the flow value is k. Then there is a path P from s to t along which a flow of one unit can be sent. If we delete all the arcs belonging to P, the flow value in the resulting network is $(k - 1)$. By the induction hypothesis, there are $(k - 1)$ arc-disjoint paths from the source to the sink. These $(k - 1)$ paths, together with path P, constitute a set of k arc-disjoint paths. So $k \in I$. Thus there are k arc-disjoint paths from the source to the sink.

6.15 (*Menger's Theorem: Vertex Form for Undirected Graphs*) Show that the maximum number of internally disjoint paths between any two nonadjacent vertices s and t in a graph is equal to the minimum number of vertices whose deletion results in a graph in which there are no paths between those two vertices.

Solution. If G is the given graph, construct the associated digraph $D(G)$ by replacing each edge of the graph joining two vertices u and v by two arcs (u, v) and (v, u). Delete all arcs directed to s and all arcs directed from t.

Let the resulting digraph be G'. There is a one-to-one correspondence between the set of directed paths from s to t in G' and the set of paths between s and t in G. Hence, this theorem follows as a consequence of the result established in Problem 6.13.

6.16 (*Menger's Theorem: Edge Form for Undirected Graphs*) Show that the maximum number of edge-disjoint paths between two vertices s and t in a graph is equal to the minimum number of edges whose deletion results in a graph in which there are no paths between the two vertices.

Solution. If G is the given graph, construct the associated digraph $D(G)$ by replacing each edge of the graph joining two vertices u and v by two arcs (u, v) and (v, u). Delete all arcs directed to s and all arcs directed from t. Let the resulting digraph be G'. There is a one-to-one correspondence between the set of directed paths from s to t in G' and the set of paths between s and t in G. Hence, this theorem follows as a consequence of the result established in Problem 6.14.

6.17 Show that the vertex form of Menger's theorem implies the edge (arc) form.

Solution. Let G be any undirected graph in which s and t are two vertices. Introduce two new vertices x and y. Join x and s by constructing a new edge. Join t and y by constructing another new edge. The enlarged graph is G', and its line graph is $L(G')$. Let s' be the vertex in the line graph that corresponds to the new edge joining x and s in G'. Likewise, let t' be the vertex in the line graph that corresponds to the new edge joining t and y. Notice that s' and t' are not adjacent since s and t are distinct. (See Problem 6.18.) Any cut in G corresponds to a set of vertices in $L(G)$, the deletion of which results in a graph in which there is no path between s' and t'. Furthermore, any pair of edge-disjoint paths between s and t in G become two internally disjoint paths between s' and t' in $L(G)$. So the vertex form of Menger's theorem in the line graph implies its edge form in G. The proof for digraphs is the same.

6.18 Show that there are three edge-disjoint paths between s and t in the graph G shown in Fig. 6.16(a) by showing that there are three internally disjoint paths between s' and t' in the line graph $L(G')$ of G', where G' is obtained by enlarging G, as explained in Problem 6.17.

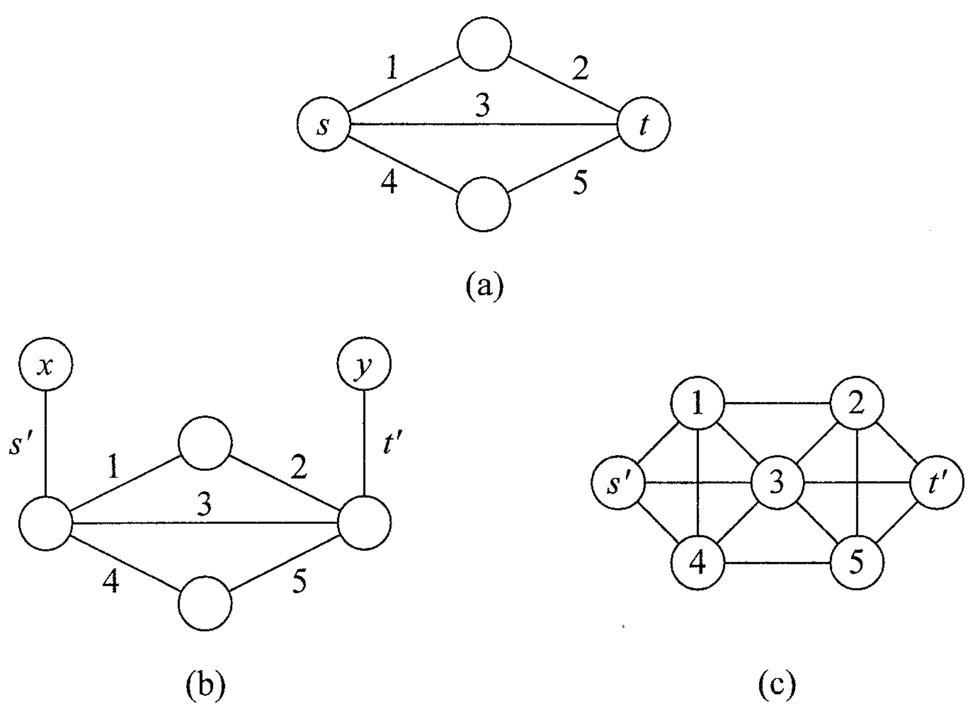


Fig. 6-16

Solution. The five edges of G are marked 1, 2, 3, 4, and 5. The three edge-disjoint paths between s and t are (*i*) the path consisting of the edges 1 and 2, (*ii*) the path consisting of the edge 3, and (*iii*) the path consisting of the edges 4 and 5. The enlarged graph G' is shown in Fig. 6-16(*b*). The edge joining the new vertex x and s becomes the source s' in $L(G')$, as shown in Fig. 6-16(*c*). Likewise, the edge joining the new vertex y and the sink t becomes the new sink t' in the line graph. There are three internally disjoint paths between s' and t' in the line graph: (*i*) s'—1—2—t' (*ii*) s'—3—t', and (*iii*) s'—4—5—t'.

6.19 Prove Theorem 6.7: The vertex form of Menger's theorem, the arc (edge) form of Menger's theorem, and the Ford–Fulkerson theorem are equivalent.

Solution. See Problem 6.13 for a proof that the Ford–Fulkerson theorem implies the vertex form of Menger's theorem. In Problem 6.17, it was shown that the vertex form implies the edge (arc) form. So it is enough if we prove that the edge (arc) form of Menger's theorem implies the Ford–Fulkerson theorem. Let G be a capacitated network with source s and sink t as specified in this chapter. The capacity of each arc joining vertex i and vertex j is a positive integer c_{ij}. Replace each such arc by c_{ij} arcs, resulting in a capacitated multigraph G' with unit capacity on each arc. The flow value p of a maximum flow in G is equal to the number of arc-disjoint paths from s to t in G'. The capacity q of a minimum cut in G is equal to the number of arcs in a minimum cut in G'. The arc form of Menger's theorem implies that $p = q$. In other words, the maximum flow value in G is equal to the capacity of a minimum cut in it. So the arc form of Menger's theorem implies the Ford–Fulkerson theorem.

MORE ON CONNECTIVITY

6.20 Show that if a simple graph of order n and size m has k components, $m \leq \frac{1}{2}(n - k)(n - k + 1)$.

Solution. The conclusion remains valid even if we assume that each component is a complete graph. Suppose H_i and H_j are two such components with n_i and n_j vertices, where $n_i \geq n_j \geq 1$. If we replace these two components by two complete graphs of order $(n_i + 1)$ and $(n_j - 1)$, respectively, the total number of vertices will remain unchanged but the number of edges will *increase* by $n_i - n_j + 1$. So the number of edges of a simple graph of order n with k components will be a *maximum* if there are $(k - 1)$ isolated vertices and one component that is a complete graph with $(n - k + 1)$ vertices with $\frac{1}{2}(n - k)(n - k + 1)$ edges.

6.21 Find the minimum number of edges needed to ensure that a simple graph is connected.

Solution. The graph consisting of two components K_{n-1} and K_1 is a disconnected graph of order n and size $\frac{1}{2}(n - 1)(n - 2)$. If m is the size of any simple graph of order n and if $m > \frac{1}{2}(n - k)(n - k + 1)$, the number of its components is $(k - 1)$ or less, as established in Problem 6.20. In particular, if $m > \frac{1}{2}(n - 2)(n - 2 + 1)$, the graph is connected. Thus a simple graph of order n and size m is connected if $m > \frac{1}{2}(n - 1)(n - 2)$.

6.22 Find the minimum number of edges in a k-connected graph.

Solution. If the graph G of order n and size m is k-connected, the degree of each vertex is at least k (see Theorem 6.9), and so $(2m)$ is at least (nk).

6.23 Exhibit a k-connected graph of order n and size m such that $(2m) = (nk)$ when (*a*) $k = 1$ and (*b*) $k = 2$.

Solution. (*a*) The simple graph of order 2 and size 1. (*b*) The graph $G = (V, E)$ with $V = \{1, 2, 3, 4\}$ and edges $\{1, 2\}$, $\{2, 3\}$, $\{3, 4\}$, and $\{4, 1\}$.

6.24 Exhibit a k-connected of order n and size m such that $(2m) = (nk) + 1$.

Solution. If G is a k-connected graph with n vertices and m edges and if (nk) is odd, $(2m)$ is at least $(nk) + 1$. It can be easily verified that the graph in Fig. 6-17 with five vertices and eight edges is a 3-connected graph.

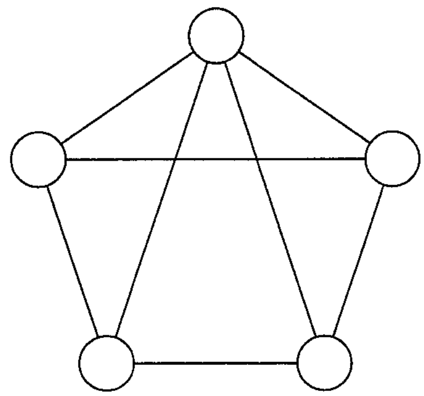


Fig. 6-17

6.25 If $1 \leq k < n$, the **Harary graph $H_{k,n}$** of order n is constructed as follows. The n vertices are placed on the circumference of a circle. (*a*) If $k = 2r$, join each vertex to the nearest r vertices in each direction around the circle. (*b*) If $k = 2r + 1$ and n is even, join each vertex to the nearest r vertices in each direction on the circle and also to the vertex exactly opposite to it. (*c*) Suppose $k = 2r + 1$ and n is odd. First the graph $H_{2r,n}$ is constructed as in part (*b*). Define $tn + i = i$ for any positive integer t, and using this (modulo) addition rule, construct additional edges by joining vertex i and vertex $\frac{1}{2}(n + 3)$ for $1 \leq i \leq (\frac{1}{2})(n + 1)$. Find the size of the Harary graph $H_{k,n}$.

Solution. In parts (*a*) and (*b*), the degree of each vertex is k, so the sum of the degrees of the n vertices is (nk). In part (*c*), there are $(n - 1)$ vertices of degree k and one vertex of degree $(k + 1)$. The sum of the degrees of the n vertices in this case is $(nk) + 1$. Thus $(2m) = (nk)$ or $(nk) + 1$, where m is the size of the graph.

6.26 Construct the Harary graphs $H_{k,n}$ for (*a*) $n = 6$, $k = 4$; (*b*) $n = 6$, $k = 5$; (*c*) $n = 7$, $k = 4$; and (*d*) $n = 7$, $k = 5$.

Solution. These graphs are shown in Figs. 6-18 through 6-21.

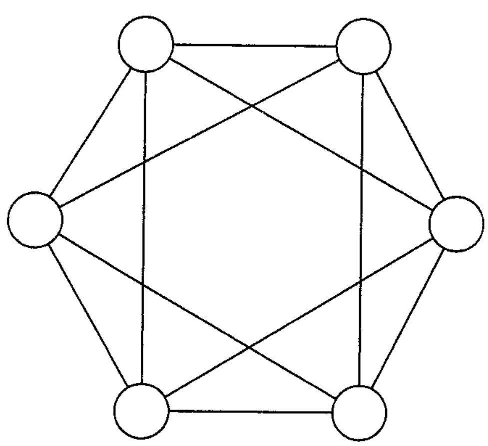


Fig. 6-18

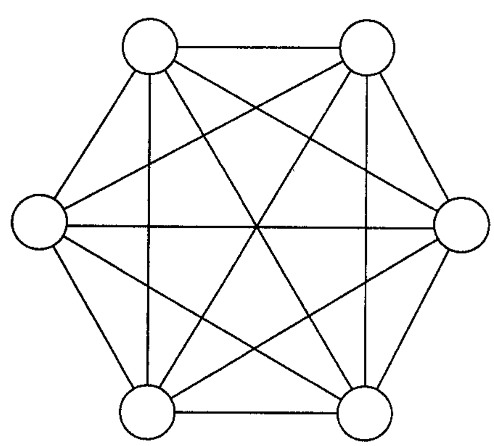


Fig. 6-19

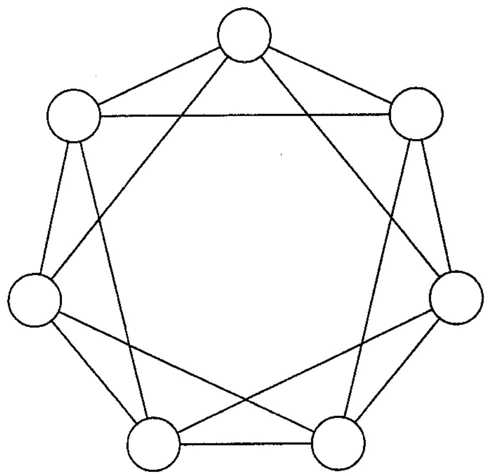


Fig. 6-20

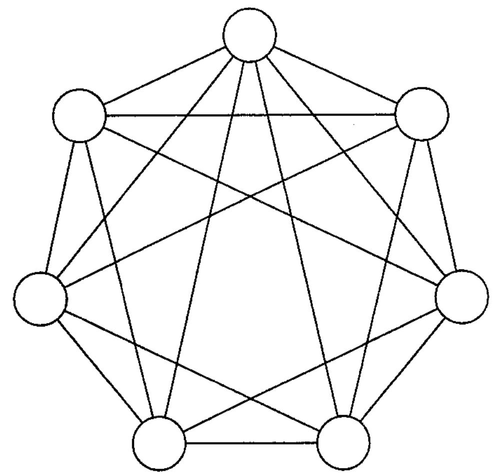


Fig. 6-21

6.27 Show that the Harary graph $K_{k,n}$ is k-connected.

Solution. Let $k = 2r$ and $H_{k,n}$ be denoted by G. Let V be the set of n vertices of G, and let W be any subset of V such that $|W| < 2r$. We shall establish that there is a path between every pair of vertices in the subgraph G' obtained after deleting W from G. Let p and q be any nonadjacent vertices in G'. Now the n vertices of G can be placed on the circumference of a circle. Let X be the set of vertices (of G) situated on the circle between p and q (excluding p and q) as we move clockwise from p to q. Likewise, let Y be vertices (of G) situated on the circle between p and q (excluding p and q) as we move counterclockwise from p to q. Since $|W \cap X| + |W \cap Y| < 2r$, at least one of the sets $W \cap X$ or $W \cap Y$ should have fewer than r elements. If $|W \cap X| < r$, there is a path between p and q in G' consisting of vertices exclusively from the set $(X - W)$ since every vertex v is adjacent (in G) to r vertices on either side of v. Thus G' is a connected graph. In other words, the graph G remains connected after deleting a set of $2r$ vertices from it. So $\kappa(G) \geq 2r$. But $\kappa(G) \leq 2r$ since the minimum degree of G is $2r$. Thus the Harary graph $H_{k,n}$ is k-connected when k is even. The proof is similar when k is odd.

6.28 Bondy's theorem shows that if k is a fixed positive integer less than n and if the degree vector $[d_1 \ \ d_2 \ \ \cdots \ \ d_n]$ (in nondecreasing order) of a simple graph G satisfies the inequality $d_j \geq (j + k - 1)$ whenever $1 \leq j \leq (n - 1 - d_{n-k+1})$, the graph G is k-connected.

Solution. Suppose G is not k-connected. Then $\kappa(G) < k$. So there exists a set S of s vertices of G (where $s < k$) such that the subgraph $G' = G - S$ has more than one component. Let H be a component of G' with minimum number of vertices, and let its order be j. If v is a vertex of H, its degree (in G) is at most $(j - 1 + s)$ since v is not adjacent to any vertex in any other component of G'. By our minimality assumption, $j \leq (n - s - j)$; consequently, the degree of v is at most $(n - j - 1)$. Next, suppose v is a vertex of G that is neither in S nor a vertex of H. In this case, the degree of v (in G) is also at most $(n - 1 - j)$. In other words, any vertex whose degree exceeds $(n - 1 - j)$ is necessarily an element of S. So there are at most s vertices whose degrees exceed $(n - 1 - j)$. Obviously, the vertex whose degree is d_{n-s} cannot be one such vertex. Hence, $d_{n-s} \leq (n - 1 - j)$.

$$\text{Now } s < k \Rightarrow s \leq (k - 1) \Rightarrow n - (k - 1) \leq n - s \Rightarrow d_{n-(k-1)} \leq d_{n-s}.$$

Hence, $d_{n-(k-1)} \leq (n - 1 - j)$; consequently, $j \leq n - 1 - d_{n-(k-1)}$. So by the hypothesis, we have the inequality $d_j \geq (j + k - 1)$. Since the degree (in G) of any vertex v in the component H is at most $(j - 1 + s)$, there are j vertices whose degrees cannot exceed this limit. In other words, $d_j \leq (j - 1 - s)$. Thus $(j + k - 1) \leq d_j \leq (j - 1 + s)$, which implies that $k \leq s$, contradicting the assumption that $s > k$.

6.29 Use Bondy's theorem to show that the simple graph with degree vector $[2 \ \ 3 \ \ 3 \ \ 3 \ \ 4 \ \ 4 \ \ 5]$ is a 2-connected graph.

Solution. (There is a simple graph G for which this vector is the degree vector. See Example 9 in Chapter 1.) Here $n = 7$, $k = 2$, and $n - 1 + d_{n-k+1} = 6 - d_6 = 2$. So the choices for j (to apply Bondy's theorem) are 1 and 2. For $j = 1$, $d_1 = 2$ and $1 + k - 1 = 2$. For $j = 2$, $d_2 = 3$ and $2 + k - 1 = 3$. Thus the required inequalities are satisfied. Hence, G is 2-connected.

6.30 Prove Theorem 6.9 (Whitney's theorem): A graph with at least $(k + 1)$ vertices is k-connected if and only if any two distinct vertices in the graph are connected by at least k internally disjoint paths. In particular, a graph with at least three vertices is a block if and only if every two vertices lie on a common cycle.

Solution. Let G be a k-connected graph. So $\kappa(G) \geq k$. Let p be the maximum number of internally disjoint paths between u and v. Suppose $p < k$. If u and v are not adjacent, $\kappa(G) = p < k$ by Menger's theorem (vertex form), which is a contradiction. Suppose u and v are adjacent. Let G' be the graph obtained after deleting the edge joining the vertices u and v. Then $\kappa(G') = p - 1 < k - 1$. So there exists a set S of s vertices ($s < k - 1$) such that $G' - S$ is a disconnected graph. So either $G - \{S \cup u\}$ or $G - \{S \cup v\}$ is disconnected, which implies that $\kappa(G) \leq s + 1 < (k - 1) + 1 = k$, arriving at the same contradiction as before. Conversely, assume that the number of internally disjoint paths between any pair of vertices is at least k. Suppose G is not k-connected. In that case, $\kappa(G) < k$. Obviously, the graph is not a complete graph. Let S be the set of vertices such that $G - S$ becomes disconnected. If u and v are vertices belonging to two different components, these two vertices cannot be adjacent vertices of the graph. So there are at least k internally disjoint paths between these two vertices. So by Menger's theorem, S should have at least k vertices. In other words, $\kappa(G) \geq k$.

6.31 Give an example of a k-connected graph such that the number of internally disjoint paths between any pair of vertices is equal to k.

Solution. In the complete bipartite graph $K_{n,n+1} = (X, Y, E)$ with n vertices in X, the maximum number of internally disjoint paths between a pair of two vertices in Y is n.

6.32 (*Harary's Characterization of Blocks of a Graph*) In a connected graph G with three or more vertices, the following statements are equivalent:
(1) G is a block. (2) If u and v are two distinct vertices of G, there is a cycle that passes contains those two vertices. (3) If u is a vertex and e is an edge, there is a cycle that contains u and e. (4) If e and f are two distinct edges, there is a cycle that contains those two edges. (5) If u and v are two vertices and e is an edge, there is a path between those two vertices that contains edge e. (6) For every three vertices, there is a path between two of them that contains the third. (7) For every three vertices, there is a path between two of them that does not contain the third.

Solution.

(1) implies (2): Suppose G is a block. Since G has more than two vertices, it is 2-connected. So by Whitney's theorem, there are at least internally disjoint paths between these two vertices, forming a cycle.

(2) implies (3): Let e be the edge joining v and w. By hypothesis, there is a cycle C that contains u and v. If C passes through w, we are done. Otherwise, let P be a path between w and u that does not pass through v. If P does not pass through any vertex of C, there are two cycles that contain v and edge e. Otherwise, let u' be the first vertex of P in cycle C. Then the path P, the path in C from u to v that does not contain u', and edge e together form the desired cycle.

(3) implies (4): Let e be the edge joining u and v. Then there is a cycle that passes through u and v that also contains edge f.

(4) implies (5): Let u and v be two vertices, and let e be an edge. Let e' be an edge adjacent to u, and let f' be an edge adjacent to v. Since, by assumption, there is a cycle that contains both these edges, there is a cycle that passes through u and v; therefore, the hypothesis of (2) is satisfied. So there exists a cycle C that contains u and edge e. Likewise, there is a cycle C' that contains v and x. We are done if u is in C' or when v is in C. Otherwise, we construct a path P starting from u using the vertices of C until we reach vertex w that belongs to C'. Then we continue the construction of the path from w to u using the vertices of C' and edge e.

(5) implies (6): Let u, v, and w be three distinct vertices. Suppose e is an edge adjacent to w. There is a path between u and v that contains edge e; therefore, the vertex w.

(6) implies (7): Let u, v, and w be three distinct vertices as before. There is a path P between u and v that passes through w by hypothesis. The subpath P' of P between u and w does not pass through v.

(7) implies (1): Let u and v be any two vertices. Suppose w is another vertex. Then there is a path P between u and v that does not pass through w. Consequently, there are at least two internally disjoint paths between u and v. So G is a block.

6.33 Let G be a k-connected graph and G' be the graph obtained from G by constructing a new vertex w and joining it to k or more vertices in G. Then G' is k-connected.

Solution. Let W be a separating set of the enlarged graph G'. If w is a vertex in W, $W' = W - v$ is a separating set of G, implying that W has at least $k + 1$ elements. If w is not a vertex in W, the set X of vertices adjacent to w is a subset of W. In this case, W has at least k vertices. Thus G' is k-connected.

6.34 *(Fan Lemma of Dirac)* A set of k paths from vertex v of a graph to each vertex in a set X of k vertices is called a **(v, X) fan of size k** if no two paths in the set have a vertex in common other than v. Show that a graph is k-connected if and only if it has at least $(k + 1)$ vertices and for any choice of a vertex v and any choice Y of vertices (where $v \notin Y$) with k or more vertices, there is a (v, X) fan of size k, where $X \subset Y$.

Solution. Let G be a k-connected graph. Enlarge the graph as in Problem 6.33 by constructing a new vertex w and joining w to each vertex in Y. The enlarged graph G' is also k-connected. So there are k internally disjoint paths between v and w. If we delete the new edges from these paths, we get a (v, X)-fan of size k. To prove the converse, assume that the graph G is not k-connected. So there is a separating set S (with less than k vertices) such that $G - S$ has more than one component. Let v and w be vertices belonging to different components of $G - S$. Let X be a set of k vertices that contain w and S but not v. In that case, every path between v and w will pass through a vertex belonging to S, showing that there is no (v, X) fan of size k, which is a contradiction.

6.35 *(Dirac's Theorem on* k*-Connectivity)* In a k-connected graph with three or more vertices, for every set of k vertices there is a cycle that will pass through these k vertices. (The converse is not true. A cycle with k vertices is not k-connected when $k > 3$.)

Solution. If $k = 2$, the result follows as an immediate consequence of Whitney's theorem. So let $k > 2$. Let W be any set of k vertices in the graph. Of all the cycles in the graph that have vertices in common with W, choose a cycle C that has the maximum number of vertices in common with W. Let m be the number of vertices common to C and W, and let $W' = \{w_1, w_2, \ldots, w_m\}$ be the set of common vertices. Suppose $m < k$. So there is a vertex w in $W - W'$. Since the graph is k-connected, there are k paths joining w to each vertex in W' such that no two paths have a vertex in common other than w, as established in Problem 6.34. Suppose w_i and w_j are adjacent vertices on C. Let Q_i be a path joining w and w_i, and let Q_j be a path joining w and w_j. If the only vertices in C are the vertices of W', it is possible to construct a cycle C' using these paths and the path between w_i and w_j obtained from C after deleting the edge joining these two vertices. Cycle C' has more vertices in common with W' than W, violating the maximality assumption. So there should be at least one vertex in C that should not belong to set W. Let w_{m+1} be a vertex in C that is not in W. The vertices w and w_{m+1} are distinct. The assumption that $m < k$ implies that $m + 1 \leq k$. Since G is k-connected, there exist paths P_i joining w and w_i $(i = 1, 2, \ldots, m + 1)$ such that no two of these $(m + 1)$ paths have a vertex in common other than vertex w. Let v_i be the first vertex in path P_i belonging to cycle C as we move from w to w_i.

Denote the subpath of P_i between w and v_i by P_i'. The $(m + 1)$ vertices v_i are distinct vertices on C, and the possibility that $v_i = w_i$ is not ruled out. Cycle C defines two paths between every pair of vertices v_j and v_k. Since C and W have exactly m vertices in common, there are integers j and k such that one of the paths defined by C between v_j and v_k has no vertex belonging to W. Let P be the other path between these two vertices. Then the three paths P, P_j', and P_k' together constitute a cycle that has more than m vertices belonging to the set W, which is a contradiction. So $m = k$, as we wished to prove.

6.36 Give an example of a k-connected graph such that an arbitrary set of $(k + 1)$ vertices need not lie on a cycle in the graph.

Solution. Consider the complete bipartite graph $K_{k,k+1} = (X, Y, E)$ with k vertices in X that is k-connected. There is no cycle that passes through all the vertices in the set Y.

6.37 Show that a graph of order n is k-connected (where $1 \leq k \leq n - 1$) if the degree of every vertex is at least $(n + k - 2)/2$.

Solution. If the graph is complete, it is k-connected for $k \leq (n - 1)$. Assume that the graph is not complete and not k-connected. So there is a disconnecting set S of s vertices such that $G - S$ is a disconnected graph. Let H be a component of $G - S$ with as few vertices as possible. If the order of H is r, $r \leq (n - s - r)$, which gives the upper bound $(n - s)/2$ for the order r. If v is any vertex of H, the degree of v in the graph G cannot exceed $(r - 1) - s$. Thus $\deg v \leq (n - s)/2 - 1 + s = (n + s - 2)/2 < (n + k - 2)/2$, violating the given inequality.

6.38 Show that the sufficient condition in Problem 6.37 for k-connectivity is not a necessary condition.

Solution. Here is a counterexample. For $n \geq 6$, let W_n be the graph obtained from the cyclic graph C_{n-1} with $(n - 1)$ vertices by introducing a new vertex and joining it to each vertex of the cycle. This graph is known as the **wheel** since the shape of the graph looks like a wheel if the vertices of the cycle are placed symmetrically on the circumference of a circle and the new vertex is placed at the center. The wheel thus defined is 3-connected; the minimum degree of a vertex is 3, which is less than $(n + 3 - 2)/2$ whenever $n > 5$.

6.39 Prove Theorem 6.10: A graph is k edge connected if and only if any two distinct vertices in it are connected by at least k edge-disjoint paths.

Solution. If there are k or more pairwise edge-disjoint paths between every pair of vertices, we need at least k edges to disconnect the graph, so the edge-connectivity number $\lambda(G)$ is at least k. Hence, the graph is k edge connected. Conversely, if the graph is k edge connected, it is not possible to disconnect it by deleting any set of $(k - 1)$ edges. Let p the maximum number of edge-disjoint paths between two vertices x and y in the graph. Now by Menger's theorem (edge form), there are p edges, the deletion of which will result in a disconnected graph. So it is possible to disconnect the graph by deleting p edges. Hence, $p > (k - 1)$. So the number of edge-disjoint paths between the two vertices is at least k.

6.40 (*Chvátal and Erdös Theorem*) Show that the graph G is Hamiltonian if $\kappa(G) \geq \alpha(G)$, where $\alpha(G)$ is its internal stability (independence) number.

Solution. The internal stability number $\alpha(G)$ is the maximum cardinality of a set of vertices in a graph such that no two vertices in it are adjacent. If $\alpha(G) = 1$, the graph is complete; therefore, it is Hamiltonian. Otherwise, $\kappa(G) = k \geq 2$, implying that G has at least one cycle. Let C be a cycle in G with the maximum number of vertices. Then C has at least k vertices since every set of k vertices belongs to some cycle, as proved in Problem 6.35. It can be shown that C is a Hamiltonian cycle. Suppose C is not a Hamiltonian cycle. If X is the set of all vertices of C, there exists a vertex $w \notin X$ and a (w, X) fan of size k as proved in Problem 6.34. If w is adjacent to any of the vertices in X, there exists a larger circle passing through all the vertices in X and vertex w, violating the maximality assumption. So w is not adjacent to any vertex in X. In that case, there is an independent set consisting of $(k + 1)$ vertices, contradicting the hypothesis.

6.41 Let $G = (V, E)$ be a connected graph, and let S be a proper subset of V. The subgraph induced by S is denoted by $G(S)$, and the subgraph induced by $T = V - S$ is denoted by $G(T)$. Show that the disconnecting set $D = (S, T)$ is a cut set if and only if both $G(S)$ and $G(T)$ are connected.

Solution. Suppose the two subgraphs are connected, and suppose D is not a cut set. Then there exists a proper subset D' of D that is a cut set. Let e be an edge in $D - D'$ joining vertex x in S and vertex y in T. If v is any vertex in $G(S)$, there is a path between v and x that passes through vertices from set S. Likewise, if w is any vertex in $G(T)$, there is a path between y and w consisting of vertices exclusively from T. In other words, when all the edges belonging to D' are deleted, the graph G still remains connected, implying that D' is not a disconnecting set. So D is a cut set. Conversely, suppose D is a cut set, and let e be an edge in D joining x in S and y in T. Then

$D' = D - e$ is not a cut set; therefore, $G - D'$ is a connected graph. Since e is the only edge between S and T in $G - D'$, there should be a path between every vertex in $G(S)$ and vertex x. Thus $G(S)$ and, similarly, $G(T)$ are connected graphs.

6.42 Let $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$, and for each i, let G_i be the subgraph obtained by deleting vertex v_i from G. Show that G is connected if and only if at least two of these subgraphs are connected.

Solution. Let G be a connected graph, and let T be any spanning tree in the graph. Any vertex v_i of degree 1 (in T) cannot be a cut vertex of the graph; therefore, G_i is a connected subgraph. But T has at least two vertices of degree 1. So two of these subgraphs are connected subgraphs. Conversely, suppose G_i and G_j are two connected subgraphs among these n subgraphs. Any vertex v_k other than these two is a vertex of both G_i and G_j. Hence, there is a path between v_j and v_k in the connected graph G_i, and there is a path between v_i and v_k in the connected graph G_j. So G is a connected graph.

6.43 Let $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$, and for each i, let G_i be the subgraph obtained by deleting vertex v_i from G. If each G_i is a connected graph with exactly one cycle, what can we say about G?

Solution. Notice that each G_i has $(n - 1)$ vertices and $(n - 1)$ edges. So the degree of each vertex in G is $m - (n - 1)$, where m is the size of G. Hence, $(2m) = (n)(m - n + 1)$. So $m(n - 2) = n(n - 1)$, implying that $n = 4$ and $m = 6$. Thus G is the complete graph with four vertices. So the only graph for which each subgraph G_i contains (is) a unique cycle is K_4.

SOME APPLICATIONS TO COMBINATORICS

6.44 Show that the max-flow min-cut theorem implies Konig's theorem.

Solution. Construct a digraph G' from the given bipartite graph $G = (X, Y, E)$ by converting each edge between vertex x in X and vertex y in Y to an arc (directed edge) from x to y and by introducing two vertices s and t such that there is an arc from s to every vertex in X and there is an arc from every vertex in Y to t. The capacity of any arc directed from s is 1. Likewise, the capacity of any arc directed to t is also 1. The capacity of every other arc is assumed to be infinite. Thus G' can be considered a capacitated network with s as the source and t as the sink. (See Problem 6.45.) Any feasible flow with flow value k obviously corresponds to a matching of cardinality k in the bipartite graph. So a maximum flow in G' defines a maximum cardinality matching in the bipartite graph G. Now if W is any covering in G, there is no arc in G' from a vertex in $(X - W)$ to a vertex in $(Y - W)$. So if $S = \{s\} \cup (X - W) \cup (Y \cap W)$ and $T = \{t\} \cup (Y - W) \cup (X \cap W)$, we have cut (S, T) in the network with capacity k. In other words, any covering of cardinality k defines a cut with cut value equal to k. On the other hand, any cut (S, T) with finite cut value k consists of k arcs, each of capacity 1. Let W_1 be the set of vertices in X that are adjacent to s, and let W_2 be the set of vertices in Y that are adjacent to t in this cut. Then the union of W_1 and W_2 is a covering. Thus any minimum cut in the digraph G' defines a minimum covering in the bipartite graph G. Thus the cardinality of a maximum matching in G is equal to the cardinality of a minimum covering in G since the former is equal to the maximum flow value in G' and the latter is equal to the minimum cut value in G'.

6.45 Illustrate Konig's theorem for the bipartite graph in Fig. 6-22(a) by converting it into a capacitated network as described in Problem 6.44.

Solution. It is easy to see that $M = \{\{x_1, y_2\}, \{x_3, y_3\}, \{x_4, y_4\}\}$ is a matching and $W = \{x_3, x_4, y_2\}$ is a covering. Both M and W have the same number of elements, so by Konig's theorem, the former is a maximum matching and the latter is a minimum covering. The network corresponding to the bipartite graph is shown in Fig. 6-22(b). Then $S = \{s, x_1, x_2, y_2\}$ and $T = \{t, x_3, x_4, y_1, y_3, y_4, y_5\}$. So cut (S, T) consists of arcs $\{s, x_3\}$, $\{s, x_4\}$ and $\{y_2, t\}$. The capacity of this cut is 3, which is also equal to the maximum flow value.

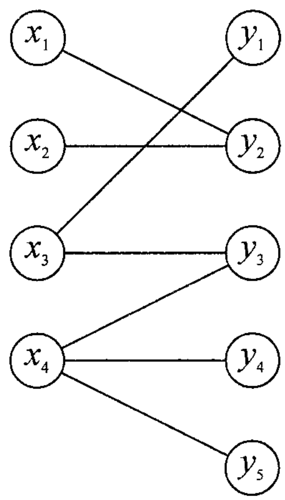


Fig. 6-22a

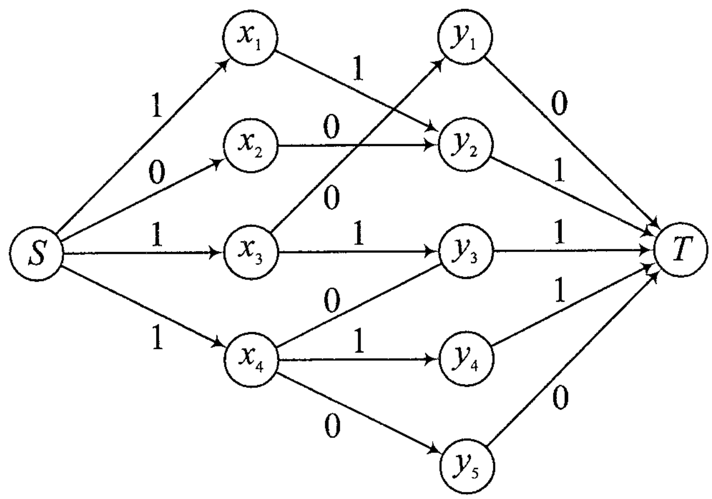


Fig. 6-22b

6.46 Show that Menger's theorem implies Konig's theorem.

Solution. For the given bipartite graph, we construct the directed graph G' as in Problem 6.44. By Menger's theorem, the maximum number of internally disjoint paths from s to t is equal to the minimum number of vertices whose deletion destroys all paths from s to t. But the former is the cardinality of a (maximum) matching, and the latter is the cardinality of a (minimum) covering.

6.47 Show that Konig's theorem implies Hall's marriage theorem.

Solution. Let $F = \{X_1, X_2, \ldots, X_m\}$ be a family of sets, and let $X = \{a_1, a_2, \ldots, a_n\}$ be the union of all the sets in the family. In the bipartite graph $G = (F, X, E)$, there is an edge between X_i and a_j if and only if a_j is an element of X_i. The marriage condition (M.C.) states that for any choice of a set F' of k vertices from F, the union of these k sets has at least k elements. Hall's theorem asserts that the M.C. is a necessary and sufficient condition for the existence of an SDR in the family. Obviously, the M.C. is a necessary condition. To prove the sufficiency part, let us assume that the M.C. holds but that the family has no SDR. In that case, let the largest subfamily that has an SDR consist of r vertices from F, where $r < m$. In other words, the size of a maximum matching is r; therefore, by Konig's theorem, there exists a covering W consisting of r vertices out of which at least one vertex is necessarily from X. Suppose $F - W = \{X_1, X_2, \ldots, X_k\}$. Then $|X \cap W|$ is $r - (m - \text{k})$. Now the M.C. implies that the cardinality of the union of the k sets in the subfamily $(F - W)$ is at least k. So $|X \cap W| \geq k$; hence, $r - (m - \text{k}) \geq k$, which contradicts the assumption that $r < m$.

6.48 Show that Hall's marriage theorem implies the Konig–Egervary theorem.

Solution. If the matrix is $m \times n$, construct a bipartite graph $G = (X, Y, E)$ with m vertices in $X = \{1, 2, \ldots, m\}$ corresponding to the rows and with n vertices in $Y = \{1, 2, \ldots, n\}$ corresponding to the n columns. Join vertex i in X to vertex j in Y if and only if the (i, j) element in the matrix has property P. Thus the Konig–Egervary theorem and Konig's theorem are equivalent. And Hall's marriage theorem implies the latter.

6.49 Show that Konig's theorem implies Menger's theorem.

Solution. Let x and y be two nonadjacent vertices in a graph $G = (V, E)$ with n vertices. The neighborhood $N(x)$ of x is the set of all vertices adjacent to x, and $N(y)$ is the set of all vertices adjacent to y. The intersection $N(x) \cap N(y)$ is denoted by T. Let $X = N(x) - T$ and $Y = N(y) - T$. We thus have a partition of V into four sets: T, X, Y, and U, where $U = V - T - X - Y$. Set S of vertices in the graph is called an x, y separating set if the deletion of S from the graph will result in a subgraph in which there is no path between x and y. Obviously, T is a subset of any such separating set. Suppose k is the size of a minimum x, y separating set. So there is a path between x and y when any set of $(k - 1)$ vertices is deleted, and there exists an x, y separating set of cardinality k.

Set S is the disjoint union of T and $(S - T)$. If $|T| = t$, there are t internally disjoint paths between x and y using the vertices in T. (This proof is due to Douglas West.) We distinguish between two cases:

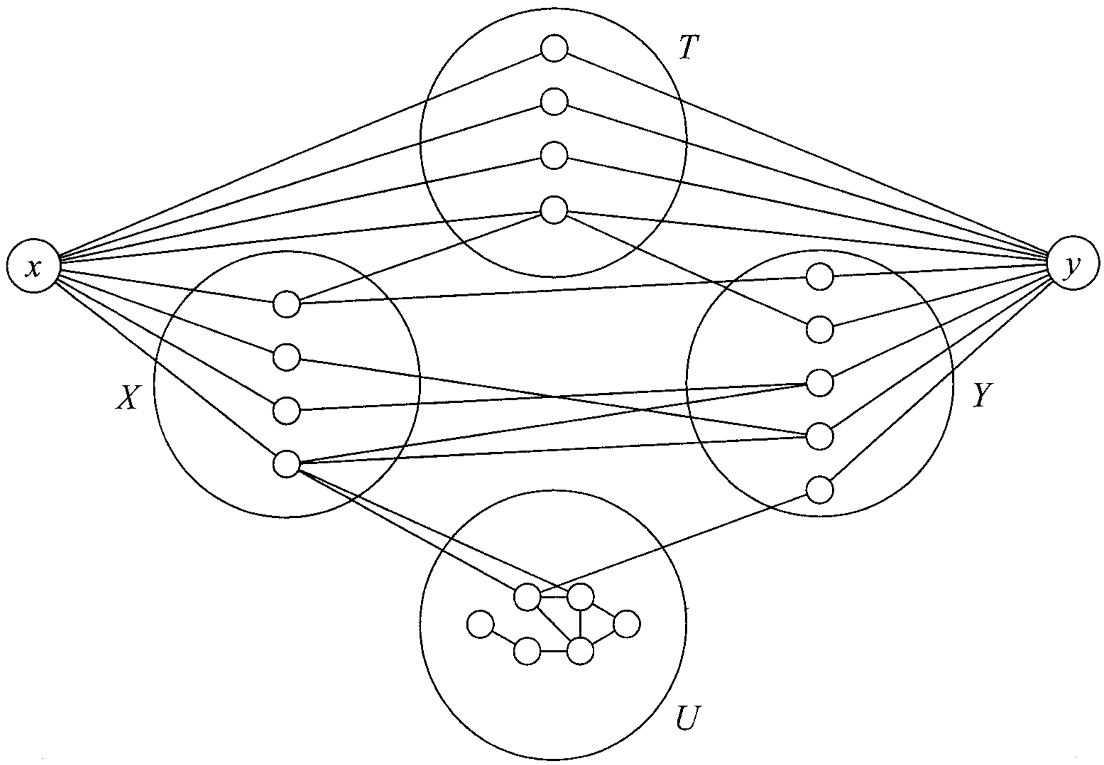


Fig. 6-23a

Case (i): If S is a minimum $x - y$ separating set, $S \cap U = \emptyset$. See Fig. 6-23(a). Let $G' = (X, Y, E')$ be the bipartite graph in which the edges are precisely those edges in G joining vertices in X and Y. See Fig. 6-23(b). Since $(S - T)$ is a set of vertices of minimum cardinality in the bipartite graph and since the exclusion of one of its vertices will no longer make S a separating set, $(S - T)$ is necessarily a minimum vertex cover in the bipartite graph. So, by Konig's theorem, there exists a (maximum) matching of size $(k - t)$ in the bipartite graph, giving $(k - t)$ internally disjoint paths between x and y using the vertices in $(S - T)$. Thus there are k internally disjoint paths between x and y, out of which t paths have two edges and the rest have three edges.

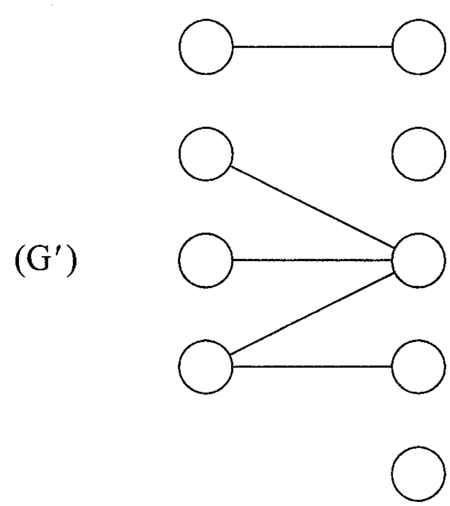


Fig. 6-23b

Case (ii): There is a minimum x, y separating set S of size k such that $S \cap U \neq \emptyset$. See Fig. 6-24. It can be proved by induction on n (the order of G) that there are k internally disjoint paths between x and y. Assume that Menger's theorem is true for all graphs of order less than n. Let $G(x)$ be the subgraph of G consisting of all paths between x and the vertices in S such that no two such paths have a vertex in common other than x. Construct the graph $G'(x)$ by introducing an artificial vertex x' and joining x' to each vertex in S. The size of a minimum x, x' separating set in $G'(x)$ cannot be more than k; if it is less than k, the minimality requirement regarding S will be violated. Since the paths chosen to construct $G(x)$ are pairwise disjoint (except at x), there is at least one vertex in $N(x)$ that is not a vertex of $G'(x)$. So its order is less than n. So by the induction hypothesis, there are k internally disjoint paths between x and x'. Hence, there are k pairwise disjoint paths between x and the vertices in S. Similarly, it is established that there are k such paths between y and the vertices in S. If we splice the k paths from x in $G(x)$ and the k paths from y in $G(y)$, we get k internally disjoint paths between x and y.

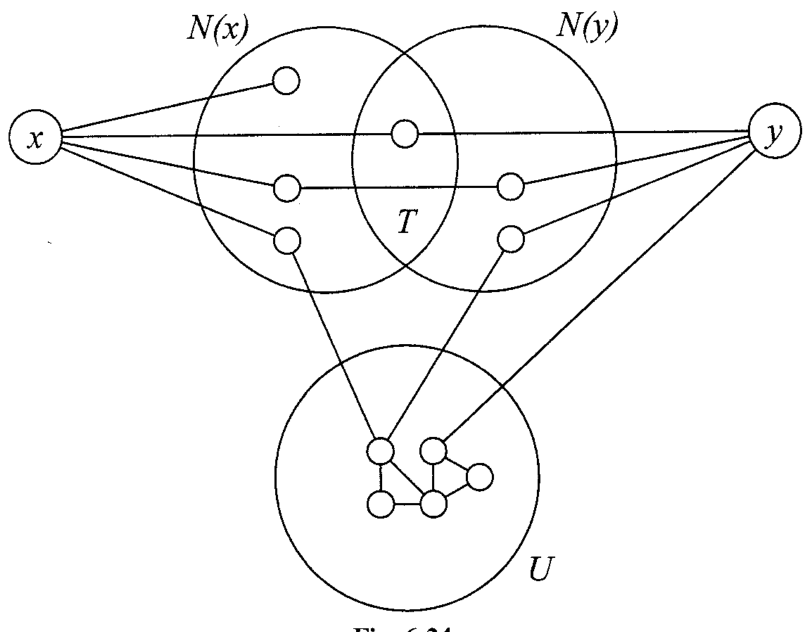


Fig. 6-24

6.50 Prove Theorem 6.17 (Konig's marriage theorem): If a bipartite graph $G = (X, Y, E)$ is k-regular (where k is positive), there is a perfect matching in that graph.

Solution. Let A be any subset of X, and let $f(A)$ be the set of vertices in Y that are adjacent to at least one vertex in A. Let E_1 be the set of edges adjacent to the vertices in A, and let E_2 be the set of edges adjacent to the vertices in $f(A)$.Then $|E_1| \leq |E_2|$. But $|E_1| = (k)|A|$ and $|E_2| = (k)|f(A)|$. Thus $|f(A)| \geq |A|$ for any set subset A of X, satisfying the hypothesis of Hall's theorem. So there is a complete matching from X and Y. But both X and Y have the same number of vertices since k is positive. So there is a perfect matching between X and Y.

6.51 Prove Theorem 6.18, (Dilworth's theorem): In a finite poset, the maximum size of an antichain is equal to the minimum number of chains into which the set of elements of the poset can be partitioned.

Solution. The finite poset $P = (\{x_1, x_2, \ldots, x_n\}, \leq)$ can be represented by an $n \times n$ matrix $\mathbf{A} = [a_{ij}]$, where $a_{ij} = 1$ if and only if $x_i < x_j$. So $a_{ij} = 1$ implies that $a_{ji} = 0$. Assume that a set of 1's in the matrix is an independent set if no two 1's belonging to the set are on the same row or column. Every chain in the poset consisting of two or more elements defines an independent set. Suppose a chain decomposition of P consists of p_i nonsingleton chains of length k_i $(i = 1, 2, \ldots, r)$ and q singleton chains. Then $n = \Sigma p_i q_i + q$. Now this decomposition will define an independent set in which the number of 1's will be m, where $m = \Sigma p_i(q_i - 1) = \Sigma p_i q_i - \Sigma p_i = n - q - p$, where $p = \Sigma p_i$. Since $p + q$ is the total number of chains, the cardinality of the poset is the sum of the number of chains in the decomposition and the number of 1's in the corresponding independent set. Therefore, if there exists a maximum independent set of size t (the term rank of the matrix), there is a minimum chain decomposition of q singleton chains and $n - t - q$ chains of length more than 1. Then by the Konig–Egervary theorem, the matrix can be covered by t lines and cannot be covered by less than t lines. This minimal cover corresponds to a set D of $n - t - q$ elements of P (one from each of the $n - t - q$ chains). The set D' of elements that constitute the singleton chains is of cardinality q. Then the union of D and D' is a set of incomparable elements of cardinality $n - t$. So there is a chain decomposition of the poset P consisting of $n - t$ chains and an antichain of cardinality $n - t$ in P.

6.52 Prove Theorem 6.19: Dilworth's theorem implies Hall's marriage theorem.

Solution. Suppose $\{A_i : i = 1, 2, \ldots, n\}$ is a family of subsets (not necessarily distinct) of a finite set $E = \{x_1, x_2, \ldots, x_m\}$ satisfying Hall's marriage condition. Consider the set $X = \{x_1, x_2, \ldots, x_m, A_1, A_2, \ldots, A_m\}$. Let $<$ be a strict partial order in X, where $x_i < A_j$ if and only if x_i is an element of A_j. In this poset, E is an antichain of cardinality m. Let D be an arbitrary antichain in the poset consisting of p elements from E and q elements from the family. Suppose $D = \{x_1, x_2, \ldots, x_p, A_1, A_2, \ldots, A_q\}$. Since none of the p elements can be an element of the union of the q sets in D, the union can have at most $m - p$ elements. But the marriage condition implies that this union of q sets has at least q elements. Thus $q \leq (m - p)$, implying that $p + q \leq m$. So E is a maximum antichain in the poset. By Dilworth's theorem, there is a partition of the poset consisting of m chains. Each chain consists of two elements: an element from the family and an element of E that belongs to it.

6.53 (*Mirsky's Theorem*) Show that in a finite poset, the maximum size of a chain is equal to the minimum number of disjoint antichains to which the poset can be partitioned.

Solution. Suppose the number of elements in a chain is p, and suppose the poset is partitioned into q antichains. Then $p \leq q$. So it is enough to show that if m is the number of elements in a largest chain, there exists a partition of the poset into m antichains. This can be done by induction on m. It is true when $m = 1$. Suppose it is true for $m - 1$. Let P be a poset that has a largest chain consisting of m elements. An element x is maximal if $x \leq y$ implies that $x = y$. The set X of all maximal elements is an antichain. In the subposet $P - X$, the number of elements in the largest chain is at most $n - 1$. The length of a largest chain in $P - X$ is $m - 1$. So by the induction hypothesis, $P - X$ can be partitioned into $m - 1$ pairwise disjoint antichains, which implies that P can be partitioned into m disjoint chains. So the theorem is true for m as well.

6.54 Verify Mirsky's theorem for the poset represented in Fig. 6-9.

Solution. The set $\{1, 6, 8, 13\}$ is a chain with maximum number of elements. The four antichains $\{1, 2, 3\}$, $\{4, 5, 6\}$, $\{11, 12, 13\}$, and $\{7, 8, 9, 10\}$ constitute a partition of the poset.

Supplementary Problems

6.55 Find the maximum flow value and a minimum cut in the network with six vertices with vertex 1 as the source and vertex 6 as the sink and with the following weight matrix:

$$\begin{bmatrix} - & 24 & - & 27 & - & - \\ - & - & 15 & 6 & - & 6 \\ - & - & - & - & - & 8 \\ - & - & 12 & - & 12 & - \\ - & - & - & - & - & 15 \\ - & - & - & - & - & - \end{bmatrix}$$

Ans. The maximum flow value is 26, and a minimum cut is $C(S, T)$, where S consists of vertices 1, 2, 3, and 4.

6.56 Find the maximum flow value and a minimum cut in the network with eight vertices with vertex 1 as the source and vertex 8 as the sink and with the following weight matrix:

$$\begin{bmatrix} - & 16 & 24 & 12 & - & - & - & - \\ - & - & - & - & 30 & - & - & - \\ - & - & - & - & 9 & 6 & 12 & - \\ - & - & - & - & - & - & 21 & - \\ - & - & - & - & - & 9 & - & 15 \\ - & - & - & - & - & - & - & 9 \\ - & - & - & - & - & - & - & 18 \\ - & - & - & - & - & - & - & - \end{bmatrix}$$

Ans. The maximum flow value is 42, and a minimum cut is $C(S, T)$, where T consists of only the sink.

6.57 Find the maximum flow value and a minimum cut in the network with 12 vertices with vertex 1 as the source and vertex 12 as the sink in which the arcs are (1, 2), (1, 3), (2, 4), (2, 5), (3, 4), (3, 5), (3, 6), (4, 7), (5, 7), (5, 8), (5, 9), (6, 9), (7, 10), (8, 10), (8, 11), (9, 11), (10, 12), and (11, 12) with weights 14, 6, 2, 4, 6, 6, 6, 8, 4, 8, 4, 6, 6, 4, 4, 6, 8, and 6, respectively.

Ans. The maximum flow value is 12, and a minimum cut is $C(S, T)$, where S consists of the source and vertex 2.

6.58 Find the maximum flow value and a minimum cut in the network with seven vertices with vertex 1 as the source and vertex 7 as the sink and with the following weight matrix:

$$\begin{bmatrix} - & 3 & 18 & - & 12 & - & - \\ - & - & - & 12 & 6 & - & - \\ - & - & - & - & - & 12 & - \\ - & - & - & - & - & - & 27 \\ - & 9 & - & 9 & - & 3 & - \\ - & - & - & - & 6 & - & 12 \\ - & - & - & - & - & - & - \end{bmatrix}$$

Ans. The maximum flow value is 27, and a minimum cut is $C(S, T)$, where S consists of the source and vertex 3.

6.59 Find the maximum flow value and a minimum cut in the network with eight vertices with vertex 1 as the source and vertex 8 as the sink and with the following weight matrix:

$$\begin{bmatrix} - & 14 & - & - & - & 10 & 18 & - \\ - & - & 18 & 8 & - & - & - & - \\ - & - & - & - & - & - & - & 10 \\ - & - & 14 & - & - & - & - & 20 \\ - & - & 16 & - & - & - & - & 6 \\ - & - & - & 8 & - & - & - & - \\ - & - & - & - & 16 & 6 & - & - \\ - & - & - & - & - & - & - & - \end{bmatrix}$$

Ans. The maximum flow value is 32, and a minimum cut is $C(S, T)$, where T consists of the sink and vertex 4.

6.60 Find the maximum flow value and a minimum cut in the network with 10 vertices with vertex 1 as the source and vertex 10 as the sink in which the arcs are (1, 2), (1, 3), (1, 4), (1, 5), (2, 6), (3, 2), (3, 7), (4, 7), (4, 8), (5, 4), (6, 7), (6, 10), (7, 10), (8, 9), (8, 10), (9, 5), and (9, 10) with weights 12, 12, 8, 8, 6, 10, 12, 10, 16, 10, 8, 8, 16, 10, 8, 8, and 6, respectively.

Ans. The maximum flow value is 38, and a minimum cut is $C(S, T)$, where S consists of the source and vertex 2.

6.61 Find a generalized minimum cut in the vertex capacitated network with vertices 1 (source), 2, 3, 4, 5, 6 (sink) with weights 0, 1, 1, 3, 2, 0 and with arcs (1, 2), (1, 3), (2, 4), (3, 2), (3, 5), (4, 3), (4, 6), (5, 4), and (5, 6) and with weights 5, 2, 3, 2, 5, 1, 7, 1, and 4, respectively.
Ans. The generalized minimum cut consists of vertices 2 and 3 and no arcs.

6.62 Verify Konig's theorem and Konig's "other" theorem in the case of the bipartite graph (V, W, E), where $V = \{1, 2, 3, 4, 5\}$, $W = \{6, 7, 8, 9\}$, and $E = \{1, 6\}, \{2, 6\}, \{3, 7\}, \{3, 9\}, \{4, 6\}, \{5, 7\}, \{5, 8\}$. [*Hint:* A maximum matching consists of three edges. A maximum independent set consists of six vertices.]

6.63 Construct the binary matrix corresponding to the bipartite graph in Problem 6.62 with five rows corresponding to the vertices in V and four columns corresponding to the vertices in W such that (i, j) entry in the matrix is positive if and only if there is an edge between i and j. Verify the Konig–Egervary theorem for this binary matrix. [*Hint:* The number of independent elements is 3.]

6.64 Show that the family $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ of sets, where $A_1 = \{a, b, c\}$, $A_2 = \{b, c\}$, $A_3 = \{c, e, f\}$, $A_4 = \{a, b\}$, $A_5 = \{a, c\}$, and $A_6 = \{d, e, f\}$, does not have an SDR. [*Hint:* Consider the union of sets A_1, A_2, A_4, and A_5.]

Chapter 7

Matchings and Factors

7.1 MORE ON MATCHINGS

A set M of edges in a graph G, as defined earlier, is a matching (or an independent edge set) in G if no two edges in M have a vertex in common. An edge in a matching in a graph is a **matched edge,** and an edge of the graph that is not in the matching is a **free edge.** A vertex that is incident to an edge of the matching M is called a **matched vertex** (with respect to M), and any other vertex is an **exposed vertex** with respect to the matching. A path between two vertices is an ***M*-alternating path** if its edges are alternately free and matched with respect to M. An ***M*-augmenting path** between two vertices u and v is an M-alternating path in which both u and v are exposed. A **maximum matching** (also known as a **maximum cardinality matching**) is a matching of maximum cardinality, whereas a **maximal matching** is one that is not a proper subset of another matching. Of course, a maximum matching is necessarily a maximal matching. The following characterization of a maximum matching in a graph is due to C. Berge.

Theorem 7.1 **(Berge's Theorem).** A matching M in a graph is a maximum matching if and only if there is no M-augmenting path in the graph. (see Solved Problem 7.3.)

In a bipartite graph $G = (X, Y, E)$, a matching is a complete matching from X to Y if every vertex in X is incident to an edge of the matching. It was shown in Chapter 6 that a necessary and sufficient condition for the existence of such a complete matching is that $|f(A)| \geq |A|$ for every subset of A of X, where $f(A)$ is the set of vertices that are adjacent to at least one vertex in A. Recall that this statement is the graph-theoretic formulation of Hall's marriage theorem; see Theorems 6.13 and 6.14. See also Solved Problem 7.4 for a proof that Berge's theorem implies Hall's marriage theorem.

A matching in a graph is a **perfect matching** if every vertex of the graph is incident to an edge in the matching. Given an arbitrary graph $G = (V, E)$, it is quite natural to ask whether it has a perfect matching so that the vertices in V can be grouped in pairs using this matching. Hall's marriage theorem implies that every k-regular bipartite graph has a perfect matching; see Theorem 6.17. A necessary and sufficient condition for an arbitrary graph to have a perfect matching was obtained by W. T. Tutte in 1947.

Theorem 7.2 **(Tutte's Theorem).** The graph $G = (V, E)$ has a perfect matching if and only if the number of odd components of $(G - S)$ does not exceed $|S|$ for every $S \subset V$. (See Solved Problem 7.8.)

7.2 THE OPTIMAL ASSIGNMENT PROBLEM

If each edge e of a bipartite graph is assigned a nonnegative weight $w(e)$, the problem of finding a matching M in G such that the sum $w(M)$ of the weights of the edges in M is as small as possible is known as the optimal assignment problem. It is assumed that $w(e)$ is a nonnegative integer. Without loss of generality, we may assume that the bipartite graph under consideration is $K_{n,n} = (X, Y, E)$ by introducing artificial vertices and artificial edges and by assigning the weight $+\infty$ to artificial edges. The **weight matrix** of the graph is $A = [a_{ij}]$, where a_{ij} is the weight of the edge joining vertex x_i in X to vertex y_j in Y. Thus a solution of the optimal assignment problem consists of a choice of n elements from the matrix such that (i) no two selected elements lie in the same row or same column and (ii) the sum of the n selected entries is as small as possible. A choice of n such elements then defines an optimum matching M, also known as an **optimal assignment.** If there exists a permutation P of the $n \times n$ identity matrix I such that the nonzero elements in P lie in the same position as n of the zeros in the weight matrix, the selection of these n elements from A will produce an optimal assignment M such that $w(M) = P.A$ (the "dot product" of the two matrices) is the sum of the n^2 pairwise products of the entries of the two matrices. In this case, we say A is matched with P.

Suppose we add an integer t to each entry in a row (or a column) of the weight matrix A, thereby modifying the weight matrix to a new matrix A'. Then $P.A' = P.A + t$. In other words, the choice of an optimal assignment is unaffected if we modify the weight matrix in this manner.

Thus given an arbitrary weight matrix A, we first try to find out whether we can obtain a modified matrix A' (with nonnegative integers) by systematically subtracting positive numbers from columns and rows such that A' can be matched with a permutation matrix. By subtracting the smallest number of a row from each entry in that row and by continuing this process for each row, we get a modified matrix in which each row has at least one zero. Then we can carry out the same procedure for each column. If we are able to obtain a modified matrix that can be matched with a permutation matrix, we are done.

Example 1. Matrix A given below is modified by subtracting 2 from row 1, 3 from row 2, 2 from row 3, and 2 from row 4. Then we subtract 3 from column 1. The modified matrix A' can be matched with the permutation matrix P. The three matrices are as follows:

$$A = \begin{bmatrix} 5 & 4 & 2 & 4 \\ 6 & 3 & 3 & 5 \\ 6 & 2 & 5 & 2 \\ 6 & 3 & 2 & 7 \end{bmatrix}, \quad A' = \begin{bmatrix} 0 & 2 & 0 & 2 \\ 0 & 0 & 0 & 2 \\ 1 & 0 & 3 & 0 \\ 1 & 1 & 0 & 5 \end{bmatrix}, \quad \text{and} \quad P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

If $X = \{x_1, x_2, x_3, x_4\}$ and $Y = \{y_1, y_2, y_3, y_4\}$ are the sets of vertices in the complete bipartite graph (X, Y, E) then a minimum weight matching M (corresponding to the nonzero entries in P) consisting of edges $\{x_1, y_1\}$, $\{x_2, y_2\}$, $\{x_3, y_4\}$, and $\{x_4, y_3\}$ with a total weight of $5 + 3 + 2 + 2 = A'.\mathrm{P} = 12$.

If an association like this exists between the modified matrix and a permutation matrix, an easy procedure to obtain the permutation matrix is as follows. Locate a row or a column with the smallest number of zeros. In a row, identify a zero entry in that row, draw a vertical line through that entry, and ignore that line in any future consideration. If a column is chosen, identify a zero entry in that column, draw a horizontal line through that entry, and ignore that line in any future consideration. Continue this process. If we are able to draw n lines like this, the n identified elements in the matrix will correspond to the nonzero entries of P.

Example 2. We can apply this procedure to obtain a minimum assignment for the weight matrix A and its modified matrix A':

$$A = \begin{bmatrix} 4 & 1 & 3 & 2 & 4 \\ 6 & 2 & 2 & 4 & 5 \\ 1 & 3 & 4 & 1 & 1 \\ 5 & 2 & 3 & 4 & 1 \\ 7 & 6 & 5 & 3 & 3 \end{bmatrix} \quad \text{and} \quad A' = \begin{bmatrix} 3 & 0 & 2 & 1 & 3 \\ 4 & 0 & 0 & 2 & 3 \\ 0 & 2 & 3 & 0 & 0 \\ 4 & 1 & 2 & 3 & 0 \\ 4 & 3 & 2 & 0 & 0 \end{bmatrix}$$

In row 1 of A', there is only one zero. The (1, 2) entry is identified, and a line is drawn along column 2 that is to be ignored in future considerations. At this stage, there is only one zero entry in column 3. So the (2, 3) entry is identified, and a line is drawn along row 2. Then we go to row 4, with a single zero. The (4, 5) entry is identified, and a line is drawn along column 5. Then the (5, 4) entry is identified in row 5, and a line is drawn along column 4. Then the (3, 1) entry is identified, and a line is drawn either along column 1 or along row 3. Thus we are able to draw five lines. The five identified entries of the matrix correspond to the nonzero entries of a permutation matrix. The weight of a minimum matching is therefore $1 + 2 + 1 + 3 + 1 = 8$.

Obviously, such an association between a matrix modified by this method and a permutation matrix need not always exist. For example, consider a modified matrix in which the only zero of row i and the only zero of row j both lie on column k. In cases like this, we have to redistribute the zeros of the modified matrix so that it can be associated with a permutation matrix. According to the Konig–Egervary theorem, the number of edges in maximum matching (using the zeros of the reduced matrix) is equal to the minimum number of lines that can be drawn to cover all the zeros of the matrix. Suppose the number of lines needed to cover the zeros is k, which is less than n. If t is the smallest uncovered entry, we subtract t from all the entries in each of the uncovered rows. This will convert the zero entries (in the covered columns) into negative entries. At this stage, we add t to all the entries in each covered column. Then we have an updated matrix with a redistribution of zeros. We continue this process until we get n lines and n entries.

Example 3. The matrix A and its modified matrix A' are

$$A = \begin{bmatrix} 4 & 9 & 3 & 11 & 4 \\ 9 & 8 & 3 & 10 & 8 \\ 7 & 5 & 3 & 8 & 6 \\ 9 & 5 & 3 & 4 & 6 \\ 10 & 11 & 7 & 10 & 11 \end{bmatrix}, \quad A' = \begin{bmatrix} 0 & 4 & 0 & 7 & 0 \\ 5 & 3 & 0 & 6 & 4 \\ 3 & 0 & 0 & 4 & 2 \\ 5 & 0 & 0 & 0 & 2 \\ 2 & 2 & 0 & 2 & 3 \end{bmatrix}, \quad \text{and} \quad P = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The zeros of A' can be covered by drawing four lines: row 1, row 4, column 2, and column 3. The smallest uncovered entry is 2. We subtract 2 from each entry in row 1 and from each entry in row 2. Then we add 2 to each entry in column 3 and to each entry in column 4. The corresponding permutation matrix P is shown above. The weight of an optimal assignment is $P.A' = 4 + 3 + 5 + 4 + 10 = 26$. (This method of solving the assignment problem is known as the **Hungarian method.**)

Example 4. We apply the Hungarian method to matrix A:

$$A = \begin{bmatrix} 2 & 2 & 2 & 5 & 2 \\ 1 & 3 & 3 & 1 & 4 \\ 6 & 4 & 7 & 4 & 2 \\ 5 & 6 & 5 & 5 & 2 \\ 9 & 8 & 5 & 5 & 2 \end{bmatrix}$$

Here $n = 5$. By subtracting the smallest element from each row we get modified matrix B, which has a zero in each line:

$$B = \begin{bmatrix} 0 & 0 & 0 & 3 & 0 \\ 0 & 2 & 2 & 0 & 3 \\ 4 & 2 & 5 & 2 & 0 \\ 3 & 4 & 3 & 3 & 0 \\ 7 & 6 & 3 & 3 & 0 \end{bmatrix}$$

Iteration 1: At least three lines are needed to cover the zeros of matrix B. Thus $k = 3$. The zeros of B are covered with three lines: row 1, row 2, and column 5. The smallest uncovered entry is $t = 2$. Subtract 2 from each entry of the uncovered rows and add 2 to each entry of the covered columns. We get matrix C:

$$C = \begin{bmatrix} 0 & 0 & 0 & 3 & 2 \\ 0 & 2 & 2 & 0 & 5 \\ 2 & 0 & 3 & 0 & 0 \\ 1 & 2 & 1 & 1 & 0 \\ 5 & 4 & 1 & 1 & 0 \end{bmatrix}$$

Iteration 2: At least four lines are needed to cover the zeros of the matrix C. Thus $k = 4$. The zeros of the matrix C can be covered with four lines: row 1, row 2, row 3, and column 5. The smallest uncovered entry is $t = 1$. Subtract 1 from each entry of the uncovered rows and add 1 to each entry of the covered columns. We get matrix D:

$$D = \begin{bmatrix} 0 & 0 & 0 & 3 & 3 \\ 0 & 2 & 2 & 0 & 6 \\ 2 & 0 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 4 & 3 & 0 & 0 & 0 \end{bmatrix}$$

Iteration 3: We need five lines to cover the zeros of D. Thus there is a permutation matrix P that can be matched with D. An optimal matching has weight $2 + 1 + 4 + 5 + 2 = 14$.

7.3 THE TRAVELING SALESPERSON PROBLEM (TSP)

A Hamiltonian cycle in a (directed) graph G is a (directed) cycle that passes through every vertex of G. The problem of finding a Hamiltonian cycle (if it exists) of minimum weight if each (arc) edge has a nonnegative weight is known as the optimal Hamiltonian problem (OHP). In other words, we are looking for a closed path of minimum weight that passes through each vertex *exactly* once. In many practical situations, a more meaningful question is related to the problem of finding a closed path of minimum weight that passes through each vertex *at least* once. This problem is known as the optimal salesperson problem (OSP). The OHP is usually known as the traveling salesperson problem (TSP).

In the weight matrix of a weighted graph (digraph) $G = (V, E)$ and in its shortest distance matrix, each entry is nonnegative and each diagonal entry is zero. Suppose $V = \{1, 2, \ldots, n\}$. Let $G' = (V, E')$ be the graph (digraph) in which there is an edge (arc) from i to j if and only if there is a path from i to j in G; in that case, the weight of the edge (arc) is the shortest distance from i to j. The (i, j) entry in the weight matrix will be undefined if there is no edge (arc) from i to j. Likewise, the (i, j) entry in the shortest distance matrix will be undefined if there is no path (directed path) from i to j. By replacing each diagonal entry and each undefined entry by ∞, let A be the $n \times n$ matrix obtained from the weight matrix, and by replacing each diagonal entry and each undefined entry by ∞, let D be the $n \times n$ matrix obtained from the shortest distance matrix. Matrix A will be used when we consider the OHP, whereas we use matrix D when the OSP is being considered.

Example 5. Suppose weight matrix A and shortest distance D (after replacing the diagonal entries by ∞) of the digraph $G = (V, E)$, where $V = \{1, 2, 3, 4\}$, are

$$A = \begin{bmatrix} \infty & 5 & 19 & 11 \\ \infty & \infty & 4 & 7 \\ \infty & 5 & \infty & 14 \\ 9 & \infty & 6 & \infty \end{bmatrix}, \qquad D = \begin{bmatrix} \infty & 5 & 9 & 11 \\ 16 & \infty & 4 & 7 \\ 21 & 5 & \infty & 12 \\ 9 & 11 & 6 & \infty \end{bmatrix}$$

A solution to the OHP (using matrix A) is $1 \to 2 \to 3 \to 4 \to 1$ with weight $5 + 4 + 14 + 9 = 32$. A solution to the OHP (using matrix D) is $1 \to 3 \to 2 \to 4 \to 1$ with weight $9 + 5 + 7 + 9 = 30$. A shortest path from 3 to 4 is $3 \to 2 \to 4$. Thus a solution to the OSP is $1 \to 2 \to 3 \to 2 \to 4 \to 1$, which is a closed path visiting vertex 2 twice.

A Branch and Bound Method to Solve TSP

Given a graph (or digraph) with $V = \{1, 2, \ldots, n\}$, we construct the complete bipartite graph (X, Y, E), where $X = \{x_i : i = 1, 2, \ldots, n\}$ and $Y = \{y_j : j = 1, 2, \ldots, n\}$ with the weight of the edge joining x_i and y_j the same as the (i, j) entry in A (or in D). Then we solve the optimal assignment problem for this matrix. If this optimal assignment (matching) in A defines a Hamiltonian cycle in G, the OHP is readily solved. Similarly, if the optimal assignment in D defines a Hamiltonian cycle in G', the OSP is readily solved. In many cases, however, this optimal assignment need not define a Hamiltonian cycle. Consider the simple problem where $n = 6$. Suppose an optimal matching in the bipartite graph is $\{(x_1, y_2), (x_2, y_3), (x_3, y_1), (x_4, y_5), (x_5, y_6), (x_6, y_4)\}$. Here we get two disjoint cycles (subtours) in the graph and not a Hamiltonian cycle. Such subtours have to be eliminated to get an optimal solution.

Suppose A is the $n \times n$ weight matrix of G with $w(A)$ as the weight of an optimal assignment in A. If this assignment gives a Hamiltonian cycle, we are done. Otherwise, $w(A)$ is a lower bound of weight Z of an optimal Hamiltonian cycle. If the graph is not Hamiltonian, Z is ∞. Suppose the optimal assignment for A gives a subtour passing through set $S = \{v_1, v_2, \ldots, v_k\}$ of k vertices. Any Hamiltonian cycle in G should contain at least one arc from a vertex in S to a vertex in $S' = V - S$. So we create k subproblems corresponding to the k vertices in S as follows. For the ith subproblem, replace the weight of each arc from v_i to every other vertex in S by ∞ so that these arcs will not be used, compelling us to use an arc (if it exists) from v_i to a vertex in $V - S$. The matrix thus obtained is the **branching matrix** for vertex v_i. We thus have k branching matrices and k optimal assignment problems. Each matrix will give an optimal assignment, the weight of which cannot be less than $w(A)$. Since any Hamiltonian cycle in G should contain an arc from a vertex in S to a vertex in S', it is enough if we examine only these k optimal solutions in our future computations. The subtour caused by set S is eliminated, and at the same time, no Hamiltonian cycles are lost in this branching process. If any of these subproblems gives a Hamiltonian cycle and if its weight does not exceed the minimum weight of the other subproblems, that cycle is optimal. Otherwise, we take the subproblem with the **smallest lower bound** (among the existing subproblems) and branch out again from its matrix using one of its subtours. This branch and bound method eventually leads us to an optimal solution if it exists or it leads us to the conclusion that there is no Hamiltonian cycle. See Example 6.

Example 6. Consider a problem in which the initial optimal assignment is non-Hamiltonian with weight 50. We write $w = 50\text{(NH)}$ as the lower bound for the first iteration in the algorithm. NH stands for non-Hamiltonian assignment, whereas H stands for Hamiltonian assignment. For this optimal assignment, suppose there is a subtour consisting of five vertices that gives the optimal assignments with weights 53(NH), 54(NH), 56(NH), 58(H), and ∞. We can ignore the Hamiltonian cycle with weight 58 as well as the assignment with ∞. We can also conclude that there is no Hamiltonian cycle with weight less than 53. It is not known whether there is a Hamiltonian cycle with weight less than 56 at this stage. So the current lower bound for the second iteration is 53. Suppose a subtour for the non-Hamiltonian subproblem (with weight 53) consists of three vertices, giving optimal assignments 55(NH), 57(H), and ∞. The assignments 57(H) and ∞ can be ignored. Then the current lower bound for the third iteration becomes 54. We now start with the matrix that gives the non-Hamiltonian assignment with weight 54, choose one of its subtours, and proceed as before. Suppose we have the optimal assignments 55(H), 55(NH), 59(H), and 61(NH) at this stage. The branch and bound procedure ends with the report that the optimal solution is the Hamiltonian cycle with weight 55 obtained in this branch. We thus have a **tree enumeration scheme** depicted by an arborescence with the root at the initial weight matrix and vertices at subsequent branching matrices. See Fig. 7-1.

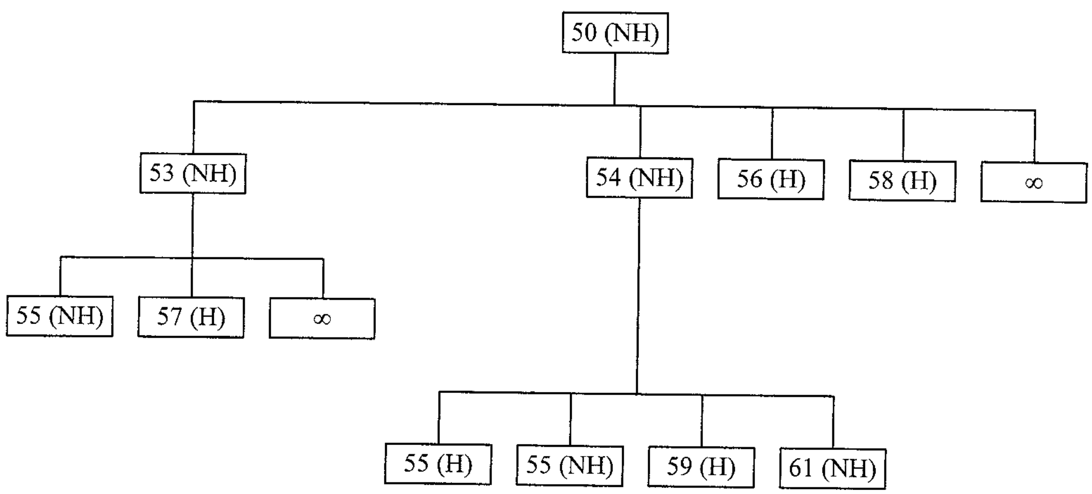


Fig. 7-1

Obtaining an Approximate Solution

Eliminating all the subtours is not an easy problem. In a typical problem, the number of such subtours could be very large, even when the number of vertices is not. So in many cases, it will be helpful to find a Hamiltonian cycle (assuming it exists) whose weight is as close as possible to the weight of an optimal Hamiltonian cycle. We can find an "approximate solution" of this kind in certain types of graphs. If we consider a complete undirected graph (with a nonnegative weight function defined on its set of edges) in which the edge joining every pair of vertices is a shortest path between them, it is possible to obtain a Hamiltonian cycle whose weight does not exceed twice the weight of an optimal Hamiltonian cycle. This quick method involves the following steps:

Step 1. Choose any vertex v_1 as the initial cycle C_1 with one vertex

Step 2. Let C_k be a cycle with k vertices. The vertices in this cycle are arranged on a line, with v_1 as the first vertex as well as the last vertex. From each vertex (except the first vertex) in this line, select a vertex not in C_k that is nearest to that vertex. Among those selected vertices, choose a vertex u that is closest to the vertices in C_k. Let v be the vertex that is adjacent to u.

Step 3. Let C_{k+1} be the cycle obtained by inserting u adjacent to v on its left on the line.

Step 4. Repeat steps 2 and 3 until all the vertices are included in the cycle.

(The weight of the Hamiltonian cycle thus obtained does not exceed twice the weight of an optimal Hamiltonian cycle. See Solved Problem 7.35 for a proof.)

Example 7. In the complete graph with five vertices whose weight matrix is

$$A = \begin{bmatrix} \infty & 3 & 3 & 2 & 7 \\ 3 & \infty & 3 & 4 & 5 \\ 3 & 3 & \infty & 1 & 4 \\ 2 & 4 & 1 & \infty & 5 \\ 7 & 5 & 4 & 5 & \infty \end{bmatrix}$$

the method described above can be used to obtain a Hamiltonian cycle as an approximate solution of the optimal Hamiltonian problem. Since the edge joining any two vertices is a shortest path between them, the shortest distance matrix is the same as the weight matrix.

We start from vertex 1. Vertex 4 is closest to it. So we have the cycle $1 \to 4 \to 1$. A vertex close to vertex 1 is vertex 2. The weight of the edge joining 1 and 2 is 3. The vertex closest to vertex 4 is vertex 3. The weight of the edge joining 3 and 4 is 1. So vertex 3 is selected and placed on the left of vertex 4. So the next cycle is $1 \to 3 \to 4 \to 1$.

The weights of the edges joining vertex 3, vertex 4, and vertex 1 to vertices not in the current cycle are 3, 4, and 3, respectively. An edge of minimum weight 3 could either be the one joining vertex 3 and vertex 2 or the one joining vertex 1 and vertex 2. We select vertex 2 and place it on the left of vertex 3. So the next cycle is $1 \to 2 \to 3 \to 4 \to 1$.

The remaining vertex is vertex 5, and it is closest to vertex 3. So vertex 5 is placed on the left of vertex 4. Thus $1 \to 2 \to 5 \to 3 \to 4 \to 1$ is an approximate solution with weight 15. So $\frac{15}{2}$ is a lower bound for the weight of an optimal Hamiltonian cycle.

7.4 FACTORS, FACTORIZATIONS, AND THE PETERSEN GRAPH

A nontrivial spanning subgraph of a graph G is called a **factor** of G. If H is a factor of G and if the degree (in H) of each vertex is a fixed positive integer k, H is called a ***k*-factor.** In other words, a k-factor of a graph is a k-regular spanning subgraph. A graph G has a 1-factor if and only if it has a perfect matching, and G has a connected 2-factor if and only if it is Hamiltonian. A necessary and sufficient condition (Tutte's theorem) for the existence of a 1-factor in G is that the number of odd components of $(G - S)$ does not exceed the cardinality of S, whereas a necessary condition (Theorem 3.4) for the existence of a connected 2-factor is that the number of components of $(G - S)$ does not exceed the cardinality of S, where S is any set of vertices of G. It is an immediate consequence of Hall's marriage theorem that any nontrivial regular bipartite graph has a 1-factor.

A set $\{H_i = (V, E_i) : i = 1, 2, \ldots, p\}$ of factors of a graph $G = (V, E)$ is a **factorization** of G if the set $\{E_i : i = 1, 2, \ldots, p\}$ forms a partition of E. If G has such a factorization, it can be **factored into** p factors. A factorization of G consisting of k-factors is called a ***k*-factorization.** If $\{H_1, H_2, \ldots, H_p\}$ is a k-factorization of G, we write $G = H_1 \oplus H_2 \oplus \cdots \oplus H_p$ and say that G is ***k*-factorable.** If a graph is k-factorable, it is a regular graph in which the degree of each vertex is a multiple of k. A subgraph H of a graph G is an **isofactor** of G if G has a factorization (consisting of at least two factors) such that each factor in the factorization is isomorphic to H. If G has an isofactor H, we say that G is ***H*-factorable** with an **isomorphic factorization** into the factor H.

The problem involving factors and factorizations of graphs that has received the most attention is the characterization of graphs that have a 1-factor and graphs that are 1-factorable. In the former case, even though we have Tutte's characterization of such graphs which besides being considered as one of the basic theorems in graph theory serves as an important tool in the investigation of factors and factorizations of graphs, no easily applicable criterion has been found to determine whether an arbitrary graph has a 1-factor. In the latter case, graphs that are 1-factorable have not yet been classified. Only certain classes of regular graphs are known to be 1-factorable.

Theorem 7.3. (a) A complete graph of even order is 1-factorable. (b) Any r-regular nontrivial bipartite graph is 1-factorable. (See Solved Problems 7.45 and 7.46.)

Theorem 7.4. A simple graph is 2-factorable if and only if it is r-regular, where r is even. (See Solved Problem 7.48.)

Theorem 7.5. (a) The complete graph of order $(2n + 1)$ can be factored into n Hamiltonian cycles. (b) The complete graph of order $2n$ can be factored into $(n - 1)$ Hamiltonian cycles and a 1-factor. (See Solved Problems 7.49 and 7.53.)

Trivially, every 1-regular graph has a 1-factor and is 1-factorable. On the other hand, a 2-regular graph G has a 1-factor if and only if every component of G is an even cycle; in that case, it is 1-factorable. Figure 7-2 shows that an arbitrary 3-regular (known as a **cubic graph**) graph need not have 1-factor. By deleting vertex v, we get three odd components, so by Tutte's theorem, this cubic graph has no perfect matching.

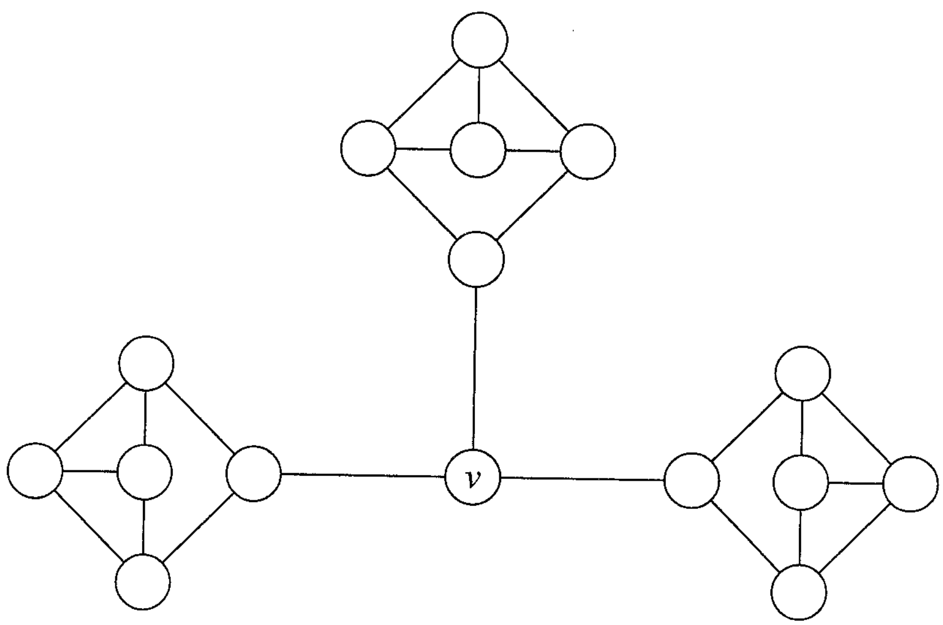


Fig. 7-2

When will a cubic graph have a 1-factor? The following theorem gives an easily verifiable sufficient condition.

Theorem 7.6 **(Petersen's Theorem).** A cubic graph in which no edge is a bridge can be factored into a 2-factor and a 1-factor. (See Solved Problem 7.59.)

Even though a bridgeless cubic graph has a 1-factor, it need not be a 1-factorable graph in general. The **Petersen graph** shown in Fig. 7-3 is a counterexample.

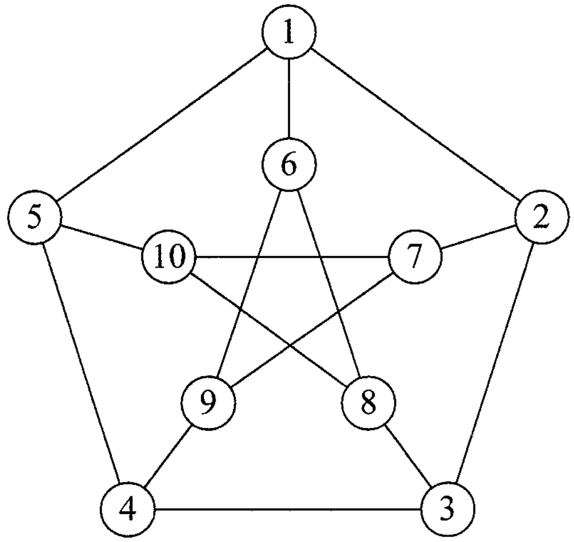


Fig. 7-3

Theorem 7.7. The Petersen graph is not 1-factorable. (See Solved Problem 7.60.)

Observe that the size of an isofactor of a graph G is a divisor of the size of G. So if the Petersen graph has an isofactor, it should have five or three edges. Now $5K_2$ is not an isofactor of the Petersen graph since it is not 1-factorable. It can be easily verified, however, that the graph shown in Fig. 7-4 with five edges is an isofactor of the Petersen graph.

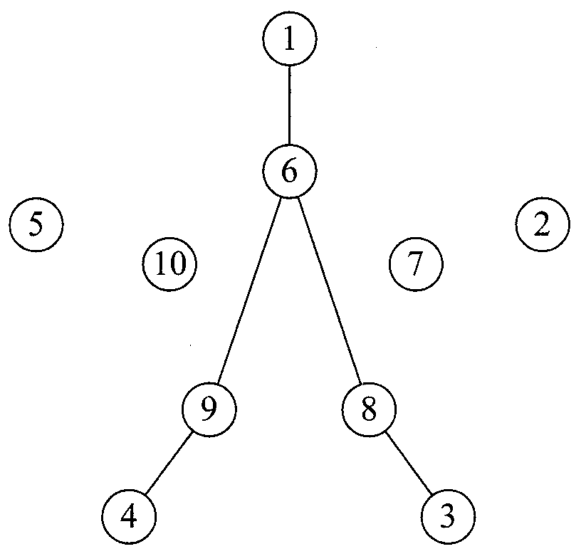


Fig. 7-4

Solved Problems

MORE ON MATCHINGS

7.1 Find the number of perfect matchings in $K_{n,n}$ and in K_{2n}.

Solution. In the case of bipartite graphs, a vertex can be arbitrarily chosen from one of its partite sets and can be matched with a vertex in the other set in n ways. Delete these two vertices, and then continue the process.

So in this case there are $n!$ perfect matchings. In the case of K_{2n}, suppose there are f_n perfect matchings. There are $(2n-1)$ ways of matching an arbitrary vertex with any one of the remaining vertices. So we have the recursion formula $f_n = (2n-1)f_{n-1}$ with $f_1 = 1$. Thus $f_n = 1 \cdot 3 \cdot 5 \cdot \ldots \cdot (2n-1)$.

7.2 If M is a matching in a graph G and if P is an M-augmenting path in G, show that the symmetric difference $(M \Delta P)$ is also a matching in G with one more edge than M.

Solution. Suppose e and f are any two edges in $(M \Delta P)$, the disjoint union of $(P - M)$ and $(M - P)$. If both e and f are in $P - M$ (or, for that matter, in $M - P$), they cannot have a vertex in common since P is an M-alternating path. Suppose e is in $(P - M)$ and f is in $(M - P)$. If e and f have vertex v in common, v is necessarily a terminal vertex of the M-augmenting path P, implying that v is not an exposed vertex, which is a contradiction. Thus no two edges in $(M \Delta P)$ have a vertex in common, so it is indeed a matching. Notice that the augmenting path P has k matched edges and $(k+1)$ free edges. If M has m edges, $(M - P)$ has $m - k$ edges and $(P - M)$ has $(2k+1) - k = (k+1)$ edges. Thus $(M \Delta P)$ has $m + 1$ edges.

7.3 Prove Theorem 7.2 (Berge's Theorem): A matching M in a graph $G = (V, E)$ is a maximum matching if and only if there is no M-augmenting path in G.

Solution. Suppose there exists an M-augmenting path P with respect to a maximum matching M. Then (as shown in Problem 7.2), the symmetric difference $(M \Delta P)$ is a matching that has more edges than the maximum matching, which is a contradiction. So if M is a maximum matching, there is no M-augmenting path. To prove the converse, it is enough to show that there is an M-augmenting path if the given matching M is not a maximum matching. Suppose M' is a maximum matching. Let $G' = (V, M \Delta M')$. In G', the degree of each vertex is either 1 or 2. So each component G is either an even cycle (with an equal number of edges belonging to both the matchings) or a path with edges alternately belonging to the two matchings. There is at least one component P (which is not a cycle) with an odd number of edges; otherwise, both M and M' will have the same cardinality. Since M' has more edges than M, the first edge and the last edge of P belong to M'. In that case, P is an M-augmenting path. In other words, if a matching M is not a maximum matching, there is always an M-augmenting path.

7.4 Show that Berge's theorem implies Hall's marriage theorem.

Solution. It is enough if we establish the following graph-theoretical analog (Theorem 6.14) of Hall's marriage theorem. In a bipartite graph $G = (X, Y, E)$, there is a complete matching from X to Y (matching each vertex in X to some vertex in Y) if and only if $f(A) \geq |A|$ for every $A \subset X$, where $f(A)$ is the set of vertices in Y that are adjacent to at least one vertex in A. Obviously, if there is a complete matching from X to Y, this inequality has to be satisfied. To prove the converse, assume that this inequality is satisfied for all A but that there is no complete matching from X to Y. Suppose M is a maximum matching from X to Y. Since there is no complete matching from X to Y, there is a vertex u in X that is exposed under M. Let S be the set of all vertices in X such that there is an M-alternating path from u, and let T be the set of all vertices in Y such that there is an M-alternating path from u. So every vertex in $S - u$ is matched with a vertex in T and vice versa; hence, $|T| = |S| - 1$, implying that $|T| < |S|$. If vertex v is in $f(S) - T$, there will be an M-augmenting path between u and v, violating the maximality of M. So $T = f(S)$, which implies that $|f(S)| = |T| < |S|$. But $|f(S)| \geq |S|$, according to the hypothesis.

7.5 If the order of a graph G is even and if S is any set of vertices of the graph, the number of odd components of the graph $(G - S)$ is odd if and only if $|S|$ is odd.

Solution. The total number of vertices in the even components is even. If the number of odd components is odd, the total number of vertices in the odd components is odd. Thus the total number of vertices in all the components together is odd. Hence, the number of vertices in S is odd since the order of the graph is even.

7.6 Suppose W is the set of all vertices of degree $(n-1)$ in a graph G of order n, where n is even. Show that G has a perfect matching if the number of odd components of $(G - W)$ does not exceed $|W|$ and if every component of $(G - W)$ is complete.

Solution. Each even component has a perfect matching. If one vertex is removed from each odd component, the remaining vertices in that component can be matched in pairs. Suppose the vertices removed from the set of odd components constitute a set of k vertices. By assumption, S has $k + q$ vertices, where $q \geq 0$. By hypothesis, each vertex in S is adjacent to every other vertex in the graph. Choose k vertices at random from S and match them arbitrarily in pairs with the k vertices chosen from the odd components. At this stage, there are q unmatched vertices forming set Q. Now q is always even since k is odd if and only if $k + q$ is odd. Moreover, every vertex in Q is adjacent to every other vertex in Q. Thus the set of vertices in Q can also be matched in pairs. So there is a perfect matching in G.

7.7 Show that if G' is the graph obtained from the graph G by joining two of its nonadjacent vertices so that G becomes a spanning subgraph of G', the number of odd components of $(G' - S)$ cannot exceed the number of odd components of $(G - S)$.

Solution. Let $o(G)$ denote the number of odd components of any graph. We have to prove that $o(G' - S) \leq o(G - S)$. Let u and v be two nonadjacent vertices in G, and let G' be the graph obtained from G by joining these two vertices. If either u or v belongs to S or if both u and v belong to the same component, both $(G - S)$ and $(G' - S)$ have the same number of odd components and the same number of even components. If these two vertices belong to two odd components in $(G - S)$, these two odd components join together to become an even component of $(G' - S)$; in this case, $o(G - S) = o(G' - S) + 2$. If u belongs to an odd component of $(G - S)$ and if v belongs to an even component of $(G - S)$, these two components join together as an odd component of $(G' - S)$; then $o(G - S) = o(G' - S)$. If both u and v belong to two different even components, $o(G - S) = o(G' - S)$ also. In other words, $o(G' - S) \leq o(G - S)$.

7.8 Prove Theorem 7.2 (Tutte's theorem): The graph $G = (V, E)$ has a perfect matching if and only if the number of odd components of $(G - S)$ does not exceed $|S|$ for every $S \subset V$.

Solution. Let $o(G)$ denote the number of odd components of G. Suppose M is a perfect matching in the graph. Then the order of G is even. Let S be any set of vertices of G. If S is the empty set, $o(G - S) = o(G) = 0 = |S|$. Suppose S is not empty. If $(G - S)$ has no odd components, $o(G - S) = 0 < |S|$. Consider any odd component H of $(G - S)$. Then there exists at least one vertex v in H that is not matched with any vertex in H. Since there is a perfect matching in G, there is a matched edge e joining v and vertex w that has to be in S. So for each odd component, there is a corresponding vertex in S. Thus if $o(G - S) = k$, $|S| \geq k$.

To prove the sufficiency part of the theorem, it has to be shown that there is a perfect matching in G if $o(G - S) \leq |S|$ for every S. This inequality implies that G has no odd component (by taking S as the empty set); hence, the order n of the graph is even. If the graph is complete, it has a perfect matching. So we assume that G is not complete. Suppose G' is the graph obtained from G by joining any two of its nonadjacent vertices so that G becomes a spanning subgraph of G'. Then, as established in Problem 7.7, $o(G' - S) \leq o(G - S)$. Hence, the inequality $o(G - S) \leq |S|$ implies that $o(G' - S) \leq o(G - S) \leq |S|$. Thus, without loss of generality, we assume that the graph G under consideration is maximal in the sense that there will be a perfect matching in the enlarged graph when any two nonadjacent vertices in G are joined by an edge. It is enough to show that there is a perfect matching in such as maximal graph G satisfying the inequality if $o(G - S) \leq |S|$ for every set S of vertices of G.

Let W be the set of vertices of degree $n - 1$ in G. If each component of $(G - W)$ is complete, there is a perfect matching in G, as proved in Problem 7.6. Suppose $(G - W)$ has a component H that is not complete. So there are three vertices x, y, and z in H such that x and z are not adjacent and y is adjacent to them both. Since y is not in W, its degree is less than $(n - 1)$, which implies that there is a vertex w that is not adjacent to y. This vertex w cannot be in W since the degree of each vertex in W is $(n - 1)$. So w is a vertex in one of the components of $(G - W)$. By the maximality assumption on G, we get a perfect matching M_1 in the enlarged graph G_1 obtained from G by joining x and z by edge e_1. Similarly, we get a perfect matching M_2 in the enlarged graph G_2 obtained from G by joining y and w by edge e_2. Obviously, e_1 is in $M_1 - M_2$ and e_2 is in $M_2 - M_1$. Let F be the spanning subgraph of G whose edges are either in M_1 or in M_2 but not in both. If the edge joining p and q is in both the matchings, the degree of p (and q) in F is 0. If the edge joining p and q is in M_1 but not in M_2, there exists a vertex r adjacent to p such that the edge joining p and r is in M_2. So the degree of p (and similarly of q) in F is 2. Thus the degree of each vertex in F is either 0 or 2. In other words, F is the disjoint union of cycles. Furthermore, in each cycle, the edges belonging to these two matchings occur alternately. Thus each cycle has an even number of edges. If edge e_1 belongs to cycle C_1 and edge e_2 belongs to another cycle C_2, a matching M in G, is readily obtained. M consists of all edges in C_1 that are matched under the M_2 matching and all other edges (from the other cycles) that are matched under the M_1 matching.

Finally, suppose both e_1 and e_2 belong to the same cycle C shown in Fig. 7-5. Every edge in C other than e_1 and e_2 is an edge of G. Edge a (in G) joining x and y cannot be a matched edge under either of these matchings. So a is not an edge of cycle C. Similarly, edge b joining y and z cannot be an edge of C. One of these two edges (say edge a) will divide C into two even cycles, with a as a common edge. Then we can get a matching of the vertices in C consisting of edge a, the edges under the matching M_1 on one of the cycles, and the edges under the matching M_2 in the other cycle. These matchings from C, together with the matchings under either M_1 or M_2, constitute a perfect matching in G. This completes the proof. [The proof of Tutte's theorem presented here is due to L. Lovasz (1973).]

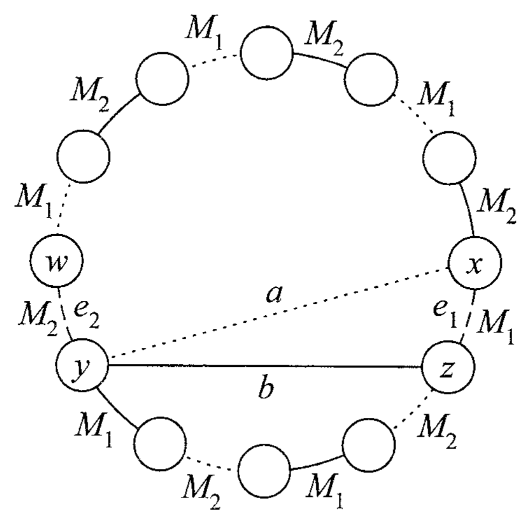


Fig. 7-5

7.9 Show that G has no perfect matching if and only if there exists a set S of vertices of the graph such that the number of odd components of $(G - S)$ is at least $|S| + 2$.

Solution. By Tutte's theorem, G has no perfect matching if and only if there exists a set S of vertices such that $o(G - S) > |S|$. But both $o(G - S)$ and $|S|$ have the same parity: either both are odd or both are even. Hence, $o(G - S) \geq |S| + 2$.

7.10 Show that the minimum number of vertices that cannot be matched in a graph of order n is t if and only if $o(G - S) \leq |S| + t$ for every set S of vertices of the graph.

Solution. Since $n - t$ is even, n + t is also even. Let $G' = (V', E')$ be the graph of order $n + t$ obtained by introducing a set W of t new vertices and by joining each new vertex to the other $n + t - 1$ vertices so that G is a subgraph of G'. Then there will be a matching M in G such that all but t vertices in it are matched if and only if G' has a perfect matching. This is equivalent to the assertion that $o(G' - S') \leq |S'|$, where S' is any subset of V' by Tutte's theorem. There are two cases to be considered: (1) V' is a subset of S'. Let $S = S' - V'$. Then $o(G' - S') = o(G - (S' - V')) = o(G - S) \leq |S| + t = |S'|$. (2) V' is not a subset of S'. In that case, $G' - S'$ is a connected graph. So $o(G' - S') \leq 1$, which implies that $o(G' - S') \leq |S'|$. Thus in either case G' has a perfect matching whenever $o(G - S) \leq |S| + t$ for every set S of vertices of the graph.

7.11 Show that a tree cannot have more than one perfect matching.

Solution. Suppose M and M' are two perfect matchings in a tree such that there is an edge e_1 joining vertices v_1 and v_2 in $M - M'$. Edge e_1 cannot be a terminal edge since a terminal edge of a tree has to be an edge in every perfect matching. So there is a nonterminal vertex v_3 and edge e_2 joining v_2 and v_3 such that e_2 is in $M' - M$. If this process continues, we get a cycle that will terminate at v_1. But the graph is acyclic.

7.12 A tree has a perfect matching if and only if it has exactly one odd component when an arbitrary vertex is deleted from it.

Solution. If v is a vertex in a tree T, each component of the disconnected graph is a subtree. If the tree has a perfect matching, the number of odd subtrees is at most 1 by Tutte's theorem. Suppose all subtrees are even. Now one edge incident to v in T is a matched edge. Suppose this matched edge is the edge joining v and w, where w is a vertex in one of the components. If we exclude these two matched vertices, the total number of remaining vertices in the tree is odd. This is a contradiction since T has a perfect matching. Conversely, suppose $o(T - v) = 1$ for every vertex v of tree T. Suppose there is no perfect matching. Let M be a maximum matching. So there is an unmatched vertex v such that no edge incident to v is a matched edge under M. By hypothesis, $T - v$ has exactly one odd component. In that odd component, the number of unmatched vertices is odd. In particular, there is an unmatched vertex w in that component such that there is an M-augmenting path from v to w, violating the maximality of M.

7.13 Show that Tutte's theorem implies Hall's marriage theorem.

Solution. We have to prove that there is a matching M in the bipartite graph $G = (X, Y, E)$ such that every vertex in X is incident to an edge from M if and only if $|f(A)| \geq |A|$ for every $A \subset X$, where $f(A)$ is the set of vertices adjacent to the set of vertices in A. If there is such a matching (known as a **complete matching from X to Y**), this equality obviously has to be satisfied for every choice of A. To prove the reverse implication, we construct the graph $G' = (V', E')$ as follows. Let $V' = X \cup Y$ if the order of G is even. Otherwise, construct a new vertex v; in that case, $V' = X \cup Y \cup \{v\}$. Construct an edge between every pair of vertices in Y. Also construct an edge between v and every vertex in Y. The set E' is the union of E and the set of new edges thus constructed. Obviously, G' has a perfect matching if and only if there is a complete matching from X to Y in the bipartite graph G. Suppose G' has no perfect matching. So by Tutte's theorem, there exists a set $S \subset V'$ such that $o(G' - S) \geq |S| + 2$.

Let $S_1 = S \cap X$ and $S_2 = S \cap Y$, and let A be the (possibly empty) subset of X such that $f(A) \subset S_2$. If we delete S from G', the vertices in $(A - S_1)$ become isolated vertices. Vertex v also becomes isolated. Because of the construction of new edges in Y, there is no other isolated vertex. Hence, the number of components of $G' - S$ is $|(A - S_1)| + 1$, which is the same as $|(A - S)| + 1$. Thus $|(A - S) + 1| \geq o(G' - S) \geq |S| + 2$. But by hypothesis, $|A| \leq |f(A)| \leq |S_2| \leq |S|$. The inequality $|A| \leq |S|$ violates the inequality $|(A - S)| + 1 \geq |S| + 2$. Hence, G' has a perfect matching.

7.14 The problem of finding a closed walk in an undirected connected weighted network that has at least two odd vertices such that the walk contains each edge at least once and such that its weight is a minimum is known as the undirected **Chinese postman problem (CPP).** Solve this problem if the number of odd vertices is exactly two.

Solution. If there are no odd vertices, the graph is Eulerian; therefore, any Eulerian circuit is a solution to the problem. If the graph is not Eulerian, it can be converted into an Eulerian graph by constructing additional edges so that the degree of each vertex is even. Construct an artificial edge between the two odd vertices so that the enlarged graph G' is Eulerian. Let the weight of this new edge be the shortest distance between these two vertices. Now locate an Eulerian circuit C in G'. Replace the new edge in C' by a shortest path between x and y. The closed trail thus constructed is a solution to the problem.

7.15 Find a solution to the CPP in the network shown in Fig. 7-6.

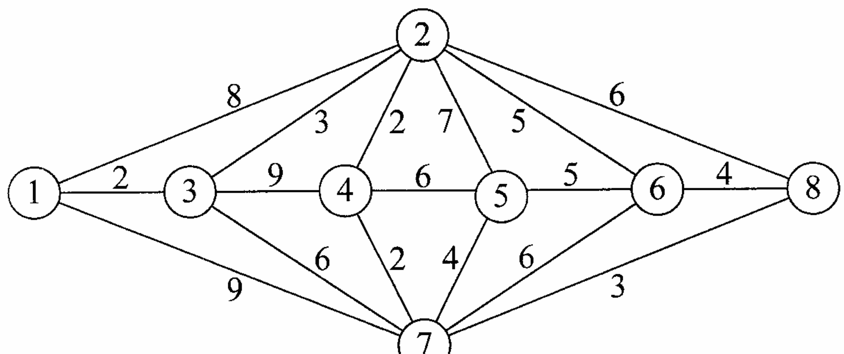


Fig. 7-6

Solution. Vertices 1 and 8 are odd. The shortest path connecting these two vertices is $1 \to 3 \to 2 \to 4 \to 7 \to 8$ with a total weight of 12. These five edges will have to be repeated. The total weight of all the edges in G is 87. Thus the weight of an optimal Chinese postman route is $87 + 12 = 99$. The enlarged Eulerian graph is shown in Fig. 7-7 in which the dotted lines indicate repeated edges. Starting from vertex 1, the postman goes to 3 and returns to 1. At this stage, both the edges joining 1 and 3 are deleted. After that, the vertices visited consecutively are 2, 3, 2, 4, 2, 5, 4, 3, 7, 4, 7, 5, 6, 7, 8, 2, 6, 8, 7, and finally 1. This closed walk contains 22 edges. The network has 17 edges, and five of them are being repeated.

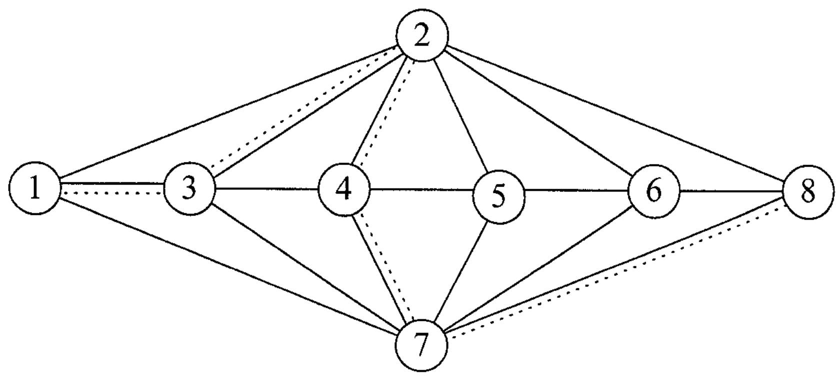


Fig. 7-7

7.16 Discuss the method for solving the CPP if the number of odd vertices is more than two.

Solution. Suppose the number of odd vertices is $2k$. Find the shortest distance between every pair of these odd vertices, and construct a complete graph with $2k$ vertices in which the weight of the edge between two vertices is the shortest distance between them. Suppose M is a minimum weight perfect matching in this complete graph. The edges belonging to M are edges in the shortest paths between pairs of odd vertices. All these edges become repeated edges in the given network. As a result, we have an enlarged network that is Eulerian. Any Eulerian circuit in the enlarged network defines an optimal route in the given network.

7.17 Obtain an optimal solution of the CPP in the network shown in Fig. 7-8.

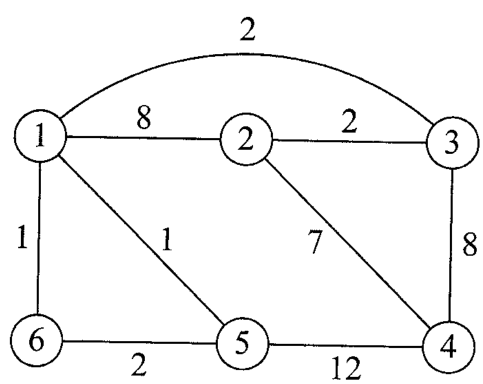


Fig. 7-8

Solution. The odd vertices form the set $\{2, 3, 4, 5\}$. The complete graph with these four vertices is displayed in Fig. 7-9, in which the weight of an edge between two vertices is the shortest distance between them in the given network. Of the three perfect matchings in this complete graph, the matching consisting of edges $\{2, 4\}$ and $\{3, 5\}$ with weight $7 + 3 = 10$ is the minimum weight perfect matching. Edge $\{2, 4\}$ in the complete graph represents the path $2 \to 4$, and this edge is to be repeated. Edge $\{3, 5\}$ in the complete graph represents the path $3 \to 1 \to 5$, and the two edges in this path are also repeated. The enlarged graph with repeated edges is shown in

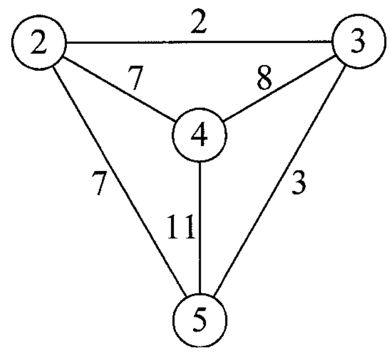


Fig. 7-9

Fig. 7-10. The Eulerian circuit $1 \to 2 \to 3 \to 1 \to 3 \to 4 \to 2 \to 4 \to 5 \to 1 \to 5 \to 6 \to 1$ in the enlarged graph is an optimal solution to the CPP in the given network with a total weight of 53.

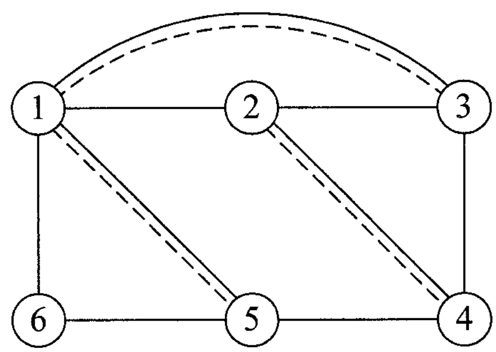


Fig. 7-10

7.18 A matrix $D = [d_{ij}]$ is **doubly stochastic** if each entry is nonnegative and if the sum of the entries in any row or column is equal to 1. Suppose $G = (X, Y, E)$ is a bipartite graph in which each vertex in X corresponds to a row of D and each vertex in Y corresponds to a column of D. Furthermore, there is an edge between vertex x_i in X and vertex y_j in Y if and only if the entry d_{ij} is positive. Show that there is a perfect matching in G.

Solution. If there is no perfect matching, there exists $A \subset X$ such that $|f(A)| < |A|$, where $f(A) = \{y \in Y : y$ is adjacent to at least one vertex in $A\}$, by Hall's marriage theorem. This inequality implies that some column sum is more than 1, violating the hypothesis. Thus the bipartite graph defined by a doubly stochastic matrix has a perfect matching.

7.19 A **permutation matrix** is a binary square matrix in which no two nonzero elements appear in the same row or same column. Prove the Birkhoff–von Neumann theorem: A square matrix D is doubly stochastic if and only if there exist nonnegative numbers r_i and permutation matrices P_i ($i = 1, 2, \ldots, k$) such that $D = r_1P_1 + r_2P_2 + \cdots + r_kP_k$ and the sum $\Sigma\, r_i$ is 1. (In other words, D is a **convex combination** of permutation matrices.)

Solution. Any convex combination of permutation matrices is a doubly stochastic matrix. To prove the converse, consider the bipartite graph G defined by the doubly stochastic matrix D. If D is a permutation matrix, we are done. Assume that this is not the case. Let M_1 be a perfect matching in G, and let P_1 be the permutation matrix that corresponds to this matching. Now each edge has a nonnegative weight d_{ij}, and since D is not a permutation matrix, there is a weight that is less than 1. Choose edge e_1 of minimum weight a_1 in the matching M. Then we can write $D = a_1P_1 + (1 - a_1)D_1$, where D_1 is doubly stochastic with more zero entries than D. At the next stage, the doubly stochastic matrix D_1 is expressed as $a_2P_2 + (1 - a_2)D_2$, where D_2 has more zeros than D_1. We continue this process until we reach a stage where the current matrix D_k is a permutation matrix, which is P_k. Thus we get a convex combination of permutation matrices.

7.20 Represent the following doubly stochastic matrix as a convex combination of permutation matrices:

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{4} & 0 & \frac{3}{4} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}$$

Solution. The bipartite graph is $G = (X, Y, E)$, where both X and Y have three vertices. There is no edge between x_1 and y_3. A perfect matching M_1 consists of edges $\{x_1, y_2\}$, $\{x_2, y_1\}$, and $\{x_3, y_3\}$. Edge $a_1 = \{x_2, y_1\}$ is an edge of minimum weight in M_1. The permutation matrix P_1 corresponding to M_1 has rows [0 1 0], [1 0 0], and [0 0 1]. The weight of a_1 is $\frac{1}{4}$. We write $D = (\frac{1}{4})P_1 + (\frac{3}{4})D_1$, where the rows of the matrix D_1 are $[\frac{2}{3}\ \frac{1}{3}\ 0]$, [0 0 1], and $[\frac{1}{3}\ \frac{2}{3}\ 0]$. Notice that D_1 has more zeros than D. A perfect matching M_2 in the bipartite graph defined by D_1 contains $\{x_1, y_2\}$, $\{x_2, y_3\}$, and $\{x_3, y_1]$ with weights $\frac{1}{3}$, 1, and $\frac{1}{3}$, respectively. Thus the permutation matrix P_2 has rows [0 1 0], [0 0 1], and [1 0 0], and $a_2 = \frac{1}{3}$. So we write $D_1 = (\frac{1}{3})P_2 + (\frac{2}{3})D_2$, where the rows of D_2 are [1 0 0], [0 0 1], and [0 1 0]. Since we have obtained a doubly stochastic matrix D_2 that is a permutation matrix, we write $D_2 = P_3$ and stop. Thus $(\frac{1}{4})P_1 + (\frac{3}{4})D_1 = (\frac{1}{4})P_1 + (\frac{3}{4})[(\frac{1}{3})P_2 + (\frac{2}{3})D_2] = (\frac{1}{4})P_1 + (\frac{1}{4})P_2 + (\frac{1}{2})P_3$ is a convex combination of the doubly stochastic matrix D.

7.21 Express the following double stochastic matrix D as a convex combination of permutation matrices:

$$D = \begin{bmatrix} 0.5 & 0 & 0 & 0.5 \\ 0 & 0.25 & 0.25 & 0.5 \\ 0.25 & 0.5 & 0.25 & 0 \\ 0.25 & 0.25 & 0.5 & 0 \end{bmatrix}$$

Solution. By using the procedure outlined in Problem 7.20, we have the representation $D = \frac{1}{2}P_1 + \frac{1}{4}P_2 + \frac{1}{4}P_3$, where the three permutation matrices are

$$P_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad \text{and} \quad P_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

THE OPTIMAL ASSIGNMENT PROBLEM

7.22 The weights of the edges of the complete bipartite graph $K_{5,5} = (X, Y, E)$ are

$$A = \begin{bmatrix} 8 & 3 & 2 & 10 & 5 \\ 10 & 7 & 10 & 6 & 6 \\ 4 & 9 & 4 & 2 & 9 \\ 8 & 10 & 5 & 3 & 3 \\ 9 & 5 & 8 & 5 & 9 \end{bmatrix}$$

Find a perfect matching of minimum weight in the bipartite graph. Here $X = \{x_i : i = 1, 2, 3, 4, 5\}$ *and* $Y = \{y_i : i = 1, 2, 3, 4, 5\}$.

Solution. By subtracting the smallest element in each row from all the elements of that row and doing likewise for all the columns as well, we get the following modified matrix B, which can be matched with the permutation matrix P (or P') such that the nonzero entries in P (or P') are matched with the zeros of B:

$$B = \begin{bmatrix} 4 & 1 & 0 & 8 & 8 \\ 2 & 1 & 4 & 0 & 0 \\ 0 & 7 & 2 & 0 & 7 \\ 3 & 7 & 2 & 0 & 0 \\ 2 & 0 & 3 & 0 & 4 \end{bmatrix}, \quad P = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad P' = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Notice that the dot products $A.P$ and $A.P'$ are both equal to 20, which is the weight of a minimum weight perfect matching. We may choose 4, 5, 2, 3, 6 (or 4, 5, 2, 6, 3) from columns 1, 2, 3, 4, and 5. An optimal matching consists of edges $\{x_1, y_3\}$, $\{x_2, y_5\}$, $\{x_3, y_1\}$, $\{x_4, y_4\}$, and $\{x_5, y_2\}$. Another optimal matching consists of edges $\{x_1, y_3\}$, $\{x_2, y_4\}$, $\{x_3, y_1\}$, $\{x_4, y_5\}$, and $\{x_5, y_2\}$.

7.23 The weights of the edges of the complete bipartite graph $K_{5,5} = (X, Y, E)$ are

$$A = \begin{bmatrix} 8 & 3 & 2 & 10 & 5 \\ 10 & 7 & 10 & 6 & 6 \\ 4 & 9 & 4 & 2 & 9 \\ 8 & 10 & 5 & 3 & 3 \\ 9 & 5 & 8 & 5 & 9 \end{bmatrix}$$

Find a perfect matching of maximum weight in the bipartite graph. Here $X = \{x_i : i = 1, 2, 3, 4, 5\}$ and $Y = \{y_i : i = 1, 2, 3, 4, 5\}$.

Solution. By subtracting each entry of the matrix from an integer that is greater than or equal to the largest entry in the matrix, this problem can be converted to a minimization problem. After subtracting each element from 10, we obtain the matrix C. Then C is modified by subtracting the smallest element from each row and each column. The modified matrix D and a matching permutation matrix P are

$$D = \begin{bmatrix} 2 & 7 & 8 & 0 & 5 \\ 0 & 3 & 0 & 4 & 4 \\ 5 & 0 & 5 & 7 & 0 \\ 2 & 0 & 5 & 7 & 7 \\ 0 & 4 & 1 & 4 & 0 \end{bmatrix}, \qquad P = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The weight of an optimal (maximum) weight matching is $A.P = 49$. The entries in D that correspond to the nonzero entries in P give the weights of the edges in a maximum weight matching. An optimal matching consists of edges $\{x_1, y_4\}$, $\{x_2, y_3\}$, $\{x_3, y_5\}$, $\{x_4, y_2\}$, and $\{x_5, y_1\}$.

7.24 The weights of the edges of the complete bipartite graph $K_{5,5} = (X, Y, E)$ are

$$A = \begin{bmatrix} 4 & 7 & 8 & 9 & 2 \\ 6 & 4 & 5 & 6 & 2 \\ 5 & 3 & 4 & 5 & 2 \\ 1 & 1 & 1 & 5 & 3 \\ 2 & 3 & 1 & 0 & 3 \end{bmatrix}$$

Find a perfect matching of minimum weight in the bipartite graph. Here $X = \{x_i : i = 1, 2, 3, 4, 5\}$ and $Y = \{y_i : i = 1, 2, 3, 4, 5\}$.

Solution. The zeros of the modified matrix B can be covered by drawing lines along the last two rows and a line along the last column. The smallest uncovered entry is 1, which is subtracted from the entries of each uncovered row. Then we add 1 to each element in the covered column. The zeros of the resulting matrix C can be covered by drawing lines along the last three rows and drawing a line along the last column. The smallest uncovered entry is 1, which is subtracted from the elements of each row. We add 1 to each element in the covered column. The resulting matrix is D.

$$B = \begin{bmatrix} 2 & 5 & 6 & 7 & 0 \\ 4 & 2 & 3 & 4 & 0 \\ 3 & 1 & 2 & 3 & 0 \\ 0 & 0 & 0 & 4 & 2 \\ 2 & 3 & 1 & 0 & 3 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 4 & 5 & 6 & 0 \\ 3 & 1 & 2 & 3 & 0 \\ 2 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 4 & 3 \\ 2 & 3 & 1 & 0 & 4 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 & 3 & 4 & 5 & 0 \\ 2 & 0 & 1 & 2 & 0 \\ 2 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 4 & 4 \\ 2 & 3 & 1 & 0 & 5 \end{bmatrix}$$

We need five lines to cover the zeros of this matrix. So at this stage, by identifying five of the zeros in D with the nonzero entries of a corresponding permutation matrix, we have an optimal solution that selects 4 from row 1, 2 from row 2, 3 from row 3, the third 1 from row 4, and the 0 from row 5. The weight of this assignment is $4 + 2 + 3 + 1 + 0 = 10$. An optimal matching consists of $\{x_1, y_1\}$, $\{x_2, y_5\}$, $\{x_3, y_2\}$, $\{x_4, y_3\}$, and $\{x_5, y_4\}$.

7.25 Four candidates A, B, C, and D were tested by a firm to fill three different types of jobs classified as P, Q, and R. A scored 10, 9, and 4 points, respectively, for these jobs. B scored 10, 6, and 8 points; C scored 9, 10, and 10 points; and D scored 8, 9, and 8 points. Based on these scores, the firm has to hire three of them such that the sum of the scores of those who were selected is a maximum. Show that the firm cannot come to an equitable decision about hiring these four candidates at this stage.

Solution. To obtain a square matrix, we introduce a nonexistent job S and assume that each candidate scored zero points when tested for this job. Then we have a candidate versus job matrix with four rows and four columns in which each entry in the last column is zero. Since we have a maximization problem, we subtract the largest entry from each entry and obtain matrix M, which can be modified into M', where M and M' are

$$M = \begin{bmatrix} 0 & 1 & 6 & 10 \\ 0 & 4 & 2 & 10 \\ 1 & 0 & 0 & 10 \\ 2 & 1 & 2 & 10 \end{bmatrix}, \qquad M' = \begin{bmatrix} 0 & 0 & 5 & 1 \\ 0 & 3 & 1 & 0 \\ 2 & 0 & 0 & 1 \\ 2 & 0 & 1 & 0 \end{bmatrix}$$

By identifying M' with an appropriate permutation matrix, we find two optimal solutions: (1) A gets P, B is not hired, C gets R, and D gets Q, with a total score of $10 + 0 + 10 + 9 = 29$. (2) A gets Q, B gets P, C gets R, and D is not hired, with a total score of $9 + 10 + 10 + 0 = 29$. Since there is no unique optimal solution, the firm cannot come to a definite and fair conclusion regarding the selection of candidates for these three jobs based on their test scores.

7.26 Show that the optimal assignment problem can be interpreted as a two-matroid intersection problem.

Solution. For every bipartite graph $G = (X, Y, E)$, we can define two partition matroids: one for X and the other for Y. These two are known as the left partition matroid and the right partition matroid. See Solved Problem 4.38. Any set I of edges of G is a matching if and only if I is an independent set in both these matroids. If both X and Y have n elements, the problem of finding a minimum weight matching in G is equivalent to the problem of finding a set I of cardinality n that is independent in both the matroids such that the weight of the sum of the edges in I is a minimum.

THE TRAVELING SALESPERSON PROBLEM (TSP)

7.27 Let C be a Hamiltonian cycle in an undirected network G. (*a*) If T is any minimum weight spanning tree in G, show that $w(T) \le w(C)$. (*b*) If in the set of edges of the graph incident at vertex v the edges p and q are of minimum weight, show that $w(T) + w(p) + w(q) \le w(C)$, where T is a minimum weight spanning tree in the $G - v$.

Solution.

(*a*) Suppose T' is the spanning tree obtained from C by deleting one of its edges. Then $w(T) \le w(T') \le w(C)$.

(*b*) Let e and f be the two edges of C adjacent to v in C. Then $C - \{e, f\}$ is a spanning tree in $G - v$. So $w(T) \le w(C) - w(e) - w(f)$. Hence, $w(T) + w(p) + w(q) \le w(T) + w(e) + w(f) \le w(C)$.

7.28 Obtain lower bounds for the weight of an optimal Hamiltonian cycle in the undirected network shown in Fig. 7-11.

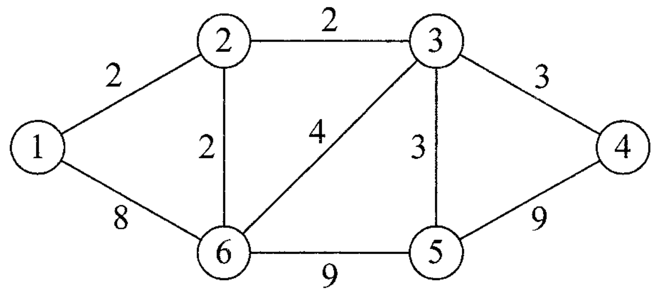


Fig. 7-11

Solution. The weight of a minimum weight spanning tree T in the network is 12, which is a lower bound for any Hamiltonian cycle in the network. If we delete vertex 1, the lower bound is $10 + 2 + 8 = 20$, as outlined in Problem 7.27. If we delete vertex 2, the lower bound is $18 + 2 + 2 = 22$. If we delete vertex 3, the lower bound is $22 + 2 + 3 = 27$. If we delete vertex 5, the lower bound is $9 + 3 + 9 = 21$. If we delete vertex 6, the lower bound is $9 + 3 + 9 = 21$. The best lower bound among these six is 22.

7.29 Using the optimal assignment method, obtain a lower bound for an optimal Hamiltonian cycle in the network shown in Fig. 7-11.

Solution. Since the undirected network has six vertices, we construct the symmetric complete bipartite graph $K_{6,6} = (X, Y, E)$ in which the weight of the edge between x_i and y_j is the weight of the edge in G between i and j. Then we have the symmetric 6×6 weight matrix A for this bipartite graph in which the diagonal entries as well as the entries corresponding to pairs of nonadjacent vertices are assigned the arbitrarily large (in this case) weight 10:

$$A = \begin{bmatrix} 10 & 2 & 10 & 10 & 10 & 8 \\ 2 & 10 & 2 & 10 & 10 & 2 \\ 10 & 2 & 10 & 3 & 3 & 4 \\ 10 & 10 & 3 & 10 & 9 & 10 \\ 10 & 10 & 3 & 9 & 10 & 9 \\ 8 & 2 & 4 & 10 & 9 & 10 \end{bmatrix}$$

The modified (after a few iterations) matrix B is

$$B = \begin{bmatrix} 2 & 0 & 7 & 1 & 1 & 0 \\ 0 & 14 & 5 & 7 & 7 & 0 \\ 8 & 6 & 13 & 0 & 0 & 2 \\ 2 & 8 & 0 & 1 & 0 & 2 \\ 2 & 8 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 2 \end{bmatrix}$$

An optimal assignment corresponds to the entries (1, 2), (2, 6), (6, 1), (3, 4), (4, 5), (5, 3) in these matrices with a total weight of 27. Thus 27 is a lower bound for any optimal Hamiltonian cycle in the network.

7.30 Obtain an optimal Hamiltonian cycle in the digraph whose weight matrix is

$$A = \begin{bmatrix} — & 5 & 19 & 11 \\ — & — & 4 & 7 \\ — & 5 & — & 14 \\ 9 & — & 6 & — \end{bmatrix}$$

Solution. The vertices are labeled 1, 2, 3, and 4. By inspection, we see that there are only two directed Hamiltonian cycles: $1 \to 2 \to 3 \to 4 \to 1$ with weight 32 and $1 \to 3 \to 2 \to 4 \to 1$ with weight 47. The former is the unique optimal Hamiltonian cycle.

7.31 Obtain a closed walk in the network of Problem 7.30 passing through each vertex as least once such that the sum of the weights of the edges in this walk is a minimum.

Solution. The shortest distance matrix D for this network is

$$D = \begin{bmatrix} - & 5 & 9 & 11 \\ 16 & - & 4 & 7 \\ 21 & 5 & - & 12 \\ 9 & 11 & 6 & - \end{bmatrix}$$

There are six Hamiltonian cycles (not listed here) with weights 30, 34, 38, 39, 47, and 48. The cycle $1 \to 2 \to 3 \to 4 \to 1$ with weight $5 + 4 + 12 + 9 = 30$ is the optimal cycle. The arc from 3 to 4 here is the shortest path $3 \to 2 \to 4$ in the original network. Thus an optimal closed walk is $1 \to 2 \to 3 \to 2 \to 4 \to 1$, which goes through vertex 2 twice and every other vertex once.

7.32 Find an optimal Hamiltonian cycle in a directed network whose weight matrix A is

$$A = \begin{bmatrix} - & 17 & 10 & 15 & 17 \\ 18 & - & 6 & 10 & 20 \\ 12 & 5 & - & 14 & 19 \\ 12 & 11 & 15 & - & 7 \\ 16 & 21 & 18 & 6 & - \end{bmatrix}$$

Solution. The vertices are labeled 1, 2, 3, 4, and 5. The optimal assignment gives the minimum as 46 with two subtours $1 \to 3 \to 2 \to 1$ and $4 \to 5 \to 4$. So we do not get a Hamiltonian cycle at this stage. If there is a Hamiltonian cycle in this graph, its weight should be at least 46. Thus we write $w(A) = 46(\text{NH})$ at the root of the tree of enumeration. Set S is the set of vertices of one of these subtours. We take S as the set consisting of vertices 4 and 5. Corresponding to vertex 4, we construct the matrix $A(4)$ by deleting the (4, 5) entry from A. Likewise, corresponding to vertex 5, we construct the matrix $A(5)$ by deleting the (5, 4) entry from A. The optimal assignment in $A(4)$ with weight 46 gives two subtours: $2 \to 3 \to 2$ and $1 \to 5 \to 4 \to 1$. So we write $w(A(4)) = 46(\text{NH})$. The optimal assignment in $A(5)$ with weight 48 gives a Hamiltonian cycle $1 \to 3 \to 2 \to 4 \to 5 \to 1$. So 48 is an upperbound for the weight of an optimal Hamiltonian cycle. At this stage, we branch from matrix $A(4)$ with $S = \{2, 3\}$, resulting in two matrices: $A(4{:}2)$ and $A(4{:}3)$. The optimal assignment in the former gives a Hamiltonian cycle $1 \to 3 \to 2 \to 5 \to 4 \to 1$ with weight 53, and the optimal assignment in the latter gives the Hamiltonian cycle $1 \to 5 \to 4 \to 2 \to 3 \to 1$ with weight 52. At this stage, the branch and bound search for an optimal Hamiltonian cycle ends with the conclusion that $1 \to 3 \to 2 \to 4 \to 5 \to 1$ is an optimal solution.

7.33 Obtain an optimal Hamiltonian cycle in the digraph whose weight matrix A is

$$A = \begin{bmatrix} - & 1 & - & - & 2 & - \\ 2 & - & 1 & - & 1 & 6 \\ - & 2 & - & 1 & - & - \\ - & - & 2 & - & - & 2 \\ 1 & - & - & 3 & - & 10 \\ 6 & - & 1 & 3 & - & - \end{bmatrix}$$

Solution.

Iteration 1: The optimal assignment is NH with weight 7 and with $1 \to 2 \to 5 \to 1$ as a subtour. We take $S = \{1, 2, 5\}$.

Iteration 2: We construct $A(1)$, $A(2)$, and $A(5)$. If we delete the appropriate entries from A to construct $A(1)$, there is no solution. So $w(A(1))$ is ∞. If we delete the appropriate entries from A to construct $A(2)$, the optimal assignment is NH with weight 11 and a subtour $1 \to 5 \to 1$. If we delete the appropriate entries from A to construct $A(2)$, the optimal assignment with weight 12 gives a Hamiltonian cycle $1 \to 5 \to 4 \to 6 \to 3 \to 2 \to 1$ with weight 12. So at the end of iteration, the conclusion is that 11 is a lower bound and 12 is an upper bound for the weight of an optimal solution. Now we branch out from matrix $A(2)$.

Iteration 3: We start with $A(2)$ and $S = \{1, 5\}$. When the appropriate deletions are made, there is no solution. The conclusion is that $1 \to 5 \to 4 \to 6 \to 3 \to 2 \to 1$ is an optimal Hamiltonian cycle with weight 12.

7.34 Obtain a closed walk of minimum weight in the digraph of Problem 7.33 that passes through each vertex at least once.

Solution. The shortest distance matrix D is

$$D = \begin{bmatrix} — & 1 & 2 & 3 & 2 & 5 \\ 2 & — & 1 & 2 & 1 & 4 \\ 4 & 2 & — & 1 & 3 & 3 \\ 6 & 4 & 2 & — & 5 & 2 \\ 1 & 2 & 3 & 3 & — & 5 \\ 5 & 3 & 1 & 2 & 4 & — \end{bmatrix}$$

The optimal assignment gives a Hamiltonian cycle $1 \to 2 \to 3 \to 4 \to 6 \to 5 \to 1$ with weight $1 + 1 + 1 + 2 + 4 + 1 = 10$. By replacing the arc from 6 to 5 in this cycle by the shortest path (in the network) $6 \to 3 \to 2 \to 5$, we get the optimal closed walk $1 \to 2 \to 3 \to 4 \to 6 \to 3 \to 2 \to 5 \to 1$ with weight 10.

7.35 Show that the weight of a Hamiltonian cycle obtained by the method of finding an approximate solution described in Section 7.3 does not exceed twice the weight of an optimal Hamiltonian cycle.

Solution. Let C: $1 \to 2 \to 3 \to \cdots (n - 1) \cdots 1$ be an optimal Hamiltonian cycle in the complete graph K_n in which the shortest distance between a pair of vertices i and j is the weight $w(i, j)$ of the edge between them, and let P be the optimal Hamiltonian path obtained from C after deleting its last edge. Suppose we start the approximation algorithm from vertex 1. If i is a vertex nearest to i, the algorithm will select the edge joining 1 and i, and we have cycle C_1: $1 \to i \to 1$ with weight $w(C_1) = w(1, i) + w(1, i)$. At the next stage, the algorithm will select a vertex that is nearest to either i or 1. Without loss of generality, assume that vertex j, which is nearest to vertex i, is chosen. At this stage, we have cycle C_2: $1 \to j \to i \to 1$ with weight $w(C_2)$. Then $w(C_2) - w(C_1) = w(1, j) + w(i, j) - w(1, i)$. But $w(1, j) - w(1, i) \le w(i, j)$, by our hypothesis. So $w(C_2) - w(C_1) \le 2w(i, j)$.

Now, while moving from C_1 to C_2, the approximation algorithm selected the edge joining i and j and ignored the edge joining i and $(i + 1)$ that appears in the optimal Hamiltonian path. [The possibility that j and $(i + 1)$ are the same is not ruled out.] So $w(i, j) \le w(i, i + 1)$. Hence, $w(C_2) - w(C_1) \le 2w(i, i + 1)$, where the edge joining i and $(i + 1)$ that appears in P is the "ignored" edge. More generally, $[w(C_{k+1}) - w(C_k)] \le$ (twice the weight of an ignored edge from P). By adding these $(n - 1)$ inequalities, we get the inequality $w(C_n) \le 2w(P) \le 2w(C)$, where C_n is a Hamiltonian cycle obtained by the approximation algorithm.

7.36 Use the approximation algorithm to obtain a Hamiltonian cycle in the complete graph whose weight matrix A is

$$A = \begin{bmatrix} — & 3 & 3 & 2 & 7 \\ 3 & — & 3 & 4 & 5 \\ 3 & 3 & — & 1 & 4 \\ 2 & 4 & 1 & — & 5 \\ 7 & 5 & 4 & 5 & — \end{bmatrix}$$

Solution. Let Z be the weight of an optimal Hamiltonian cycle in this network.

Starting from vertex 1, C_1: $1 \to 4 \to 1$, C_2: $1 \to 3 \to 4 \to 1$, C_3: $1 \to 2 \to 3 \to 4 \to 1$, and C_4: $1 \to 2 \to 5 \to 3 \to 4 \to 1$; $w(C_4) = 16 \le 2Z$.

Starting from vertex 2, C_1: $2 \to 1 \to 2$, C_2: $2 \to 4 \to 1 \to 2$, C_3: $2 \to 3 \to 4 \to 1 \to 2$, and C_4: $2 \to 5 \to 3 \to 4 \to 1 \to 2$; $w(C_4) = 15 \le 2Z$.

Starting from vertex 3, C_1: $3 \to 4 \to 3$, C_2: $3 \to 1 \to 4 \to 3$, C_3: $3 \to 2 \to 1 \to 4 \to 3$, and C_4: $3 \to 2 \to 1 \to 4 \to 5 \to 3$; $w(C_4) = 17 \le 2Z$.

Starting from vertex 4, C_1: $4 \to 3 \to 4$, C_2: $4 \to 3 \to 1 \to 4$, C_3: $4 \to 2 \to 3 \to 1 \to 4$, and C_4: $4 \to 2 \to 3 \to 1 \to 5 \to 4$; $w(C_4) = 22 \le 2Z$.

Starting from vertex 5, C_1: $5 \to 3 \to 5$, C_2: $5 \to 4 \to 3 \to 5$, C_3: $5 \to 1 \to 4 \to 3 \to 5$, and C_4: $5 \to 2 \to 1 \to 4 \to 3 \to 5$; $w(C_4) = 15 \leq 2Z$.

7.37 Using the branch and bound method, obtain an optimal Hamiltonian cycle in the network of Problem 7.36.

Solution. The optimal assignment (after redistributing the zeros of the weight matrix a couple of times in the spirit of the Konig–Egervary theorem) gives the Hamiltonian cycle $1 \to 2 \to 5 \to 4 \to 3 \to 1$ with weight 14. (Observe that twice the weight of an optimal Hamiltonian cycle is 30 and that the weight each Hamiltonian cycle obtained by the approximation method in Problem 7.36 does not exceed 30.)

7.38 Show that if T is a minimum weight spanning tree in the complete weighted graph G of order n in which the edge between any pair of vertices is a shortest path between them, it is possible to obtain a Hamiltonian cycle C such that $w(C) \leq 2w(T) \leq 2w(C')$, where C' is an optimal Hamiltonian cycle in G.

Solution. It has been already proved that $2w(T) \leq 2w(C')$. To establish the other inequality, we replace each edge of the tree T by two arcs directed in the opposite direction so that T becomes an Eulerian digraph with $2n$ arcs with a directed Eulerian circuit with weight $2w(T)$. The number of arcs is successively reduced by replacing by an arc (edge) that is not in the circuit two arcs of the circuit two as a time. This will not increase the weight of the path because of our hypothesis. Ultimately, we have a directed circuit with n arcs. See Problem 7.39.

7.39 Obtain a Hamiltonian cycle using the approximation method described in Problem 7.38 for the network in Problem 7.36.

Solution. The edges of a minimum spanning tree T are $\{1, 4\}$, $\{2, 3\}$, $\{3, 4\}$, and $\{3, 5\}$. Each edge is converted into two arcs (in opposite direction), as shown in Fig. 7-12. Arcs (1, 4) and (4, 3) are replaced by arc (1, 3). Arcs (5, 3) and (3, 2) are replaced by arc (5, 2). Arcs (2, 3) and (3, 4) are replaced by arc (2, 4). As a result, we have a directed Hamiltonian cycle C': $1 \to 3 \to 5 \to 2 \to 4 \to 1$ in which we use arcs (3, 5) and (4, 1) from the Eulerian circuit. In this process, we also dropped arcs (1, 4), (4, 3), (5, 6), (3, 2), (2, 3), and (3, 4) from the Eulerian circuit. Now $w(C') = w(1, 3) + w(3, 5) + w(5, 2) + w(2, 4) + w(4, 1) = 3 + 4 + 5 + 4 + 2 = 18 \leq [w(1, 4) + w(4, 3)] + w(3, 5) + [w(5, 3) + w(2, 3)] + [w(2, 3) + w(3, 4)] + w(4, 1) = 2w(T) = 20$.

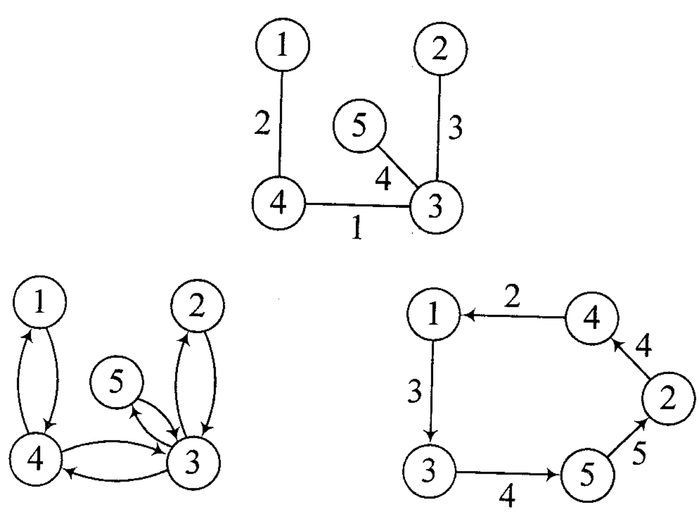


Fig. 7-12

7.40 Show that the problem of finding a directed Hamiltonian cycle in a digraph with n vertices is equivalent to finding a directed Hamiltonian path in a digraph with $(n + 1)$ vertices.

Solution. Suppose $G = (V, E)$ is a digraph with $V = \{1, 2, \ldots, n\}$. Construct a digraph $G' = (V, E')$ with $V = \{1, 2, \ldots, n, (n + 1)\}$ as follows. Both E and E' have the same number of arcs. For each arc in the digraph G directed to vertex 1 from vertex i, construct an arc in G' from i to the new vertex $(n + 1)$. Every other arc in E is an arc in E'. Then a Hamiltonian cycle starting from 1 and ending in 1 in G is equal to a Hamiltonian path from 1 to $(n + 1)$ in G', and vice versa.

7.41 Show that the TSP can be interpreted as a three-matroid intersection problem.

Solution. Given digraph G of n, construct digraph G' with one more vertex as described in Problem 7.40. Then define three matroids corresponding to G': the head-partition matroid, the tail-partition matroid, and the graphic matroid. An optimal Hamiltonian path in G is a set I of arcs of G' such that (1) the cardinality of I is n, (2) the weight of I is a minimum, and (3) I is an independent set in all the three matroids.

FACTORS, FACTORIZATIONS, AND THE PETERSEN GRAPH

7.42 Show that an Eulerian graph cannot have a bridge.

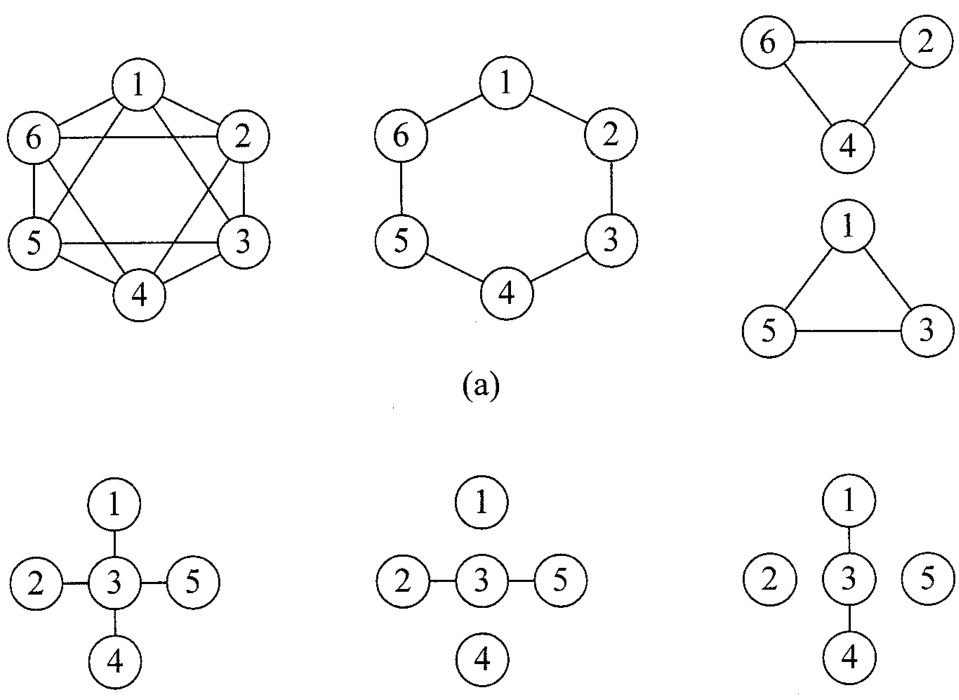
Solution. If a graph is Eulerian, it is a connected graph in which the degree of each vertex is even. Suppose it has a bridge. If this bridge is deleted, there will be two components, and in each component will be exactly one vertex of odd degree. But in any graph, the number of odd vertices is always even.

7.43 If a cubic graph has a bridge, it is not 1-factorable.

Solution. Suppose a cubic graph with a bridge is 1-factorable. The bridge will be in one of the three 3-factors. The removal of the bridge gives two components. Each component will have an even number of odd vertices and a vertex of degree 2. So there cannot be one more 1-factor since the number of vertices in each component is odd.

7.44 (*a*) Give an example of a factorization of a graph consisting of two 2-factors such that the two factors are not isomorphic. (*b*) Give an example of a factorization of a graph consisting of two isomorphic factors that are not regular.

Solution. (*a*) See Fig. 7-13(*a*). (*b*) See Fig. 7-13(*b*).



(a)

(b)

Fig. 7-13

7.45 Show that the complete graph of even order is 1-factorable.

Solution. Suppose the vertices are labeled 1, 2, 3, . . . , $2n$. Construct a regular polygon with $(2n - 1)$ vertices, and label these vertices 2, 3, . . . , $2n$ in the clockwise direction. Assume that the distance between two consecutive vertices in the cycle is one unit. The center of the polygon corresponds to vertex 1 of the complete graph. A 1-factor M_1 consisting of the edge joining 1 and 2 along with $(n - 1)$ more edges is constructed as follows. There are $n - 1$ vertices starting from 2 in the clockwise direction and $n - 1$ vertices in the counterclockwise direction, excluding vertex 2. Join the vertex located i units from vertex 2 in the clockwise direction and the vertex located i units from vertex 2 in the counterclockwise direction, where $i = 1, 2, \ldots, n$. By this method, $(n - 1)$ edges are obtained. The edge joining center vertex 1 and vertex 2 together with these $(n - 1)$ edges form a 1-factor. Then we locate vertices that are equidistant (in either direction) from vertex 3 on the cycle. This will give us a 1-factor consisting of the edge joining vertex 1 and vertex 3 along with $(n - 1)$ new edges. We continue this process. The last 1-factor consists of the edge joining vertex 1 and vertex $2n$ along with $(n - 1)$ new edges. Thus this construction defines $(2n - 1)$ perfect matchings. Since each 1-factor has n edges, the total number of edges in these $(2n - 1)$ matchings is $n(2n - 1)$, which is the total number of edges in the complete graph. Hence, the graph is 1-factorable.

7.46 A regular bipartite graph of degree r (where r is positive) is 1-factorable.

Solution. The proof is by induction on r. If $r = 1$, the result is true. Suppose it is true for $(r - 1)$, where $r \geq 2$. Consider any regular bipartite graph G of degree r. A consequence of Hall's marriage theorem is that there is a 1-factor F in G. Then the graph $G' = G - F$ is a regular bipartite graph of degree $(r - 1)$ that is 1-factorable by the induction hypothesis. Any 1-factorization of G' together with the 1-factor F gives a 1-factorization of G.

7.47 The k-cube Q_k is 1-factorable for every $k \geq 1$.

Solution. The k-cube Qk is a k-regular bipartite graph. (See Solved Problem 1.56.) So, by Problem 7.46, it is 1-factorable. A proof by induction is along the following lines. Q_k can be inductively defined as the Cartesian product $Q_{k-1} \times K_2$ that has 2^k vertices. The vertices of K_2 are labeled 1 and 2. For each vertex u in Q_{k-1}, the ordered pairs $(u, 1)$ and $(u, 2)$ define two adjacent vertices in the product since 1 and 2 are adjacent in K_2. By definition, the ordered pairs (u, i) and (v, i) are adjacent in the product if and only if u and v are adjacent in Q_{k-1}, where $i = 1$ or 2. Suppose the edge joining u and v is edge e in a perfect matching M (with 2^{k-1} edges) in Q_{k-1}. Then this matched edge defines two edges in Q_k: edge e_1 joining $(u, 1)$ and $(v, 1)$ and edge e_2 joining $(u, 2)$ and $(v, 2)$. Thus each matched edge in Q_{k-1} defines two edges in Q_k that have no vertex in common. So for every perfect matching in Q_{k-1}, we obtain a perfect matching in Q_k. If we consider the edge joining $(v, 1)$ and $(v, 2)$ as a matched edge for each vertex v of Q_{k-1}, there is one more perfect matching in Q_k.

7.48 Prove Theorem 7.4: A simple graph is 2-factorable if and only if it is r-regular, where r is even.

Solution. If a graph is 2-factorable, the degree of each vertex is obviously $2r$ for some positive integer r. Suppose the degree of each vertex of a simple graph $G = (V, E)$ of order n is $2r$, where r is positive. So there is an Eulerian circuit C in the graph. Consider this as a directed Eulerian circuit. If $V = \{v_1, v_2, \ldots, v_n\}$, construct a bipartite graph $G' = (X, Y, F)$, where $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$. The edges in F are defined as follows: join x_i and y_j by an edge if and only if there is an arc from v_i to v_j in the directed circuit C. The bipartite graph thus constructed is necessarily r-regular; therefore, it has a 1-factorization $F_1 \oplus F_2 \oplus \cdots \oplus F_r$, as proved in Theorem 7.3. For each factor F_k in this partition, define a bijection f_k on $\{1, 2, \ldots, n\}$ by $f_k(i) = j$ if and only if there is an edge in F_k between v_i and v_j. Since there is no loop in G, $f_k(i) \neq i$ for each i. Since G has no multiple edges, $f_k(i) = j$ implies that $f_k(j) = i$ for every i and j. In other words, each permutation under f_k is of length at least three, so it defines a cycle in the graph. Thus each 1-factor in the factorization of G' defines a collection of pairwise disjoint cycles in G; consequently, G is 2-factorable.

7.49 Show that a complete graph of order $(2n + 1)$ can be factored into n Hamiltonian cycles.

Solution. Let $V = \{v_i : i = 1, 2, \ldots, (2n + 1)\}$ be the set of vertices. Define path P_i between vertices v_1 and v_{i+n} passing through all vertices except v_{2n} as follows: $v_i \to v_{i-1} \to v_{i+1} \to v_{i-2} \to \cdots \to v_{i+n-1} \to v_{i+n}$, where all subscripts are taken as integers 1, 2, . . . , $2n$ (mod $2n$). There are n such edge disjoint paths. Construct the Hamiltonian cycle C_i from each path P_i by joining v_{2n} to the endpoints of each path P_i.

7.50 Obtain a 2-factorization of the complete graph with nine vertices.

Solution. The graph has 36 edges. Each Hamiltonian cycle in the graph will have nine edges. So the 2-factorization consists of four edge-disjoint Hamiltonian cycles in the graph. Here $2n + 1 = 9$, so $n = 4$. The subscripts of the vertices are the integers 1, 2, 3, . . . , 8 (mod 8). We construct four edge-disjoint paths:

P_1: $1 \to 8 \to 2 \to 7 \to 3 \to 6 \to 4 \to 5$

P_2: $2 \to 1 \to 3 \to 8 \to 4 \to 7 \to 5 \to 6$

P_3: $3 \to 2 \to 4 \to 1 \to 5 \to 8 \to 6 \to 7$

P_4: $4 \to 3 \to 5 \to 2 \to 6 \to 1 \to 7 \to 8$

Observe that each row has $2n$ entries. The first row starts from 1, and the entries in it alternately increase and decrease. The entries in each column increase (mod $2n$). Change each path into a Hamiltonian cycle by joining vertex 9 to the endpoints of each path.

7.51 Show that the complete graph of order $2n$ can be factored into n Hamiltonian paths; hence, prove it is 1-factorable.

Solution. This is an immediate consequence of the fact that the complete graph of order $(2n + 1)$ can be factored into n Hamiltonian cycles. Each path that leads to the construction of a Hamiltonian cycle in K_{2n+1} (see Problems 7.49 and 7.50) is, in fact, a Hamiltonian path in K_{2n}. Moreover, each such path defines a 1-factor in K_{2n}. Hence, it can be factored into n 1-factors.

7.52 Obtain a 1-factorization of the complete graph with eight vertices.

Solution. In the complete graph with vertices 1, 2, . . . , 9, if vertex 9 is deleted, we get four edge-disjoint Hamiltonian paths that constitute a factorization of K_8 into four 1-factors:

P_1: $1 \to 8 \to 2 \to 7 \to 3 \to 6 \to 4 \to 5$

P_2: $2 \to 1 \to 3 \to 8 \to 4 \to 7 \to 5 \to 6$

P_3: $3 \to 2 \to 4 \to 1 \to 5 \to 8 \to 6 \to 7$

P_4: $4 \to 3 \to 5 \to 2 \to 6 \to 1 \to 7 \to 8$

7.53 Show that a complete graph of order $2n$ can be factored into n Hamiltonian cycles and one 1-factor.

Solution. Suppose the vertices are labeled 1, 2, 3, . . . , $2n$. Construct a regular polygon with $(2n - 1)$ vertices, and label these vertices 2, 3, . . . , $2n$ in the clockwise direction. Assume that the distance between two consecutive vertices in the cycle is one unit. The center of the polygon corresponds to vertex 1 of the complete graph. A 1-factor M_1 consisting of the edge joining 1 and 2 along with $(n - 1)$ more edges is constructed as follows. There are $n - 1$ vertices starting from 2 in the clockwise direction and $n - 1$ vertices in the counterclockwise direction, excluding vertex 2. Join the vertex located i units from vertex 2 in the clockwise direction and the vertex located i units from vertex 2 in the counterclockwise direction, where $i = 1, 2, \ldots, n - 1$. By this method, $(n - 1)$ edges are obtained. The edge joining center vertex 1 and vertex 2 together with these $(n - 1)$ edges forms a 1-factor. Then we locate vertices that are equidistant (in either direction) from vertex 3 on the cycle. This will give us a 1-factor consisting of the edge joining vertex 1 and vertex 3 along with $(n - 1)$ new edges. We continue this process. If M_1 is the first matching, a Hamiltonian cycle C_1 starting from vertex $2n$ can be constructed by alternately using vertices on the clockwise direction and counterclockwise direction in this matching. Use the first $(n - 1)$ 1-factors to construct $(n - 1)$ Hamiltonian cycles. The next 1-factor and these $(n - 1)$ cycles give the desired factorization.

7.54 Obtain a factorization of the complete graph with eight vertices consisting of three Hamiltonian cycles and one 1-factor.

Solution. The vertices are 1, 2, . . . , 8. Keeping vertex 1 at the center of a polygon and the vertices in increasing order as corners in the clockwise direction, the following four 1-factors are obtained:

M_1: (1, 2), (3, 8), (4, 7), (5, 6)

M_2: (1, 3), (4, 2), (5, 8), (6, 7)

M_3: (1, 4), (5, 3), (6, 2), (7, 8)

M_4: (1, 5), (6, 4), (7, 3), (8, 2)

The first three matchings give the following Hamiltonian cycles:

C_1: $1 \to 2 \to 3 \to 8 \to 4 \to 7 \to 5 \to 6 \to 1$

C_2: $1 \to 3 \to 4 \to 2 \to 5 \to 8 \to 6 \to 7 \to 1$

C_3: $1 \to 5 \to 6 \to 4 \to 7 \to 3 \to 8 \to 2 \to 1$

These three cycles and the 1-factor M_4 constitute a partition consisting of three Hamiltonian cycles and one perfect matching.

7.55 If $0 \leq r < n$, rn is even if and only if there exists an r-regular graph G of order n.

Solution. If G is r-regular of order n, its size is $(rn)/2$, so rn has to be even. To prove the converse, we examine two cases.

Case (*i*): n is even. In this case, rn is even. Since n is even, $K_n = F_1 \oplus F_2 \oplus \cdots \oplus F_{n-1}$, where each F_i is a 1-factor. Then $G = F_1 \oplus F_2 \oplus \cdots \oplus F_r$ is an r-regular graph of order n.

Case (*ii*): n is odd. So r has to be even. In this case, K_n can be factored into $(n - 1)/2$ Hamiltonian cycles. So $K_n = F_1 \oplus F_2 \oplus \cdots \oplus F_{(n-1)/2}$, where each F_i is a 2-factor. Then $G = F_1 \oplus F_2 \oplus \cdots \oplus F_{r/2}$.

7.56 Show that a Hamiltonian cycle in a complete graph of odd order is an isofactor of the graph.

Solution. This is certainly true since any complete graph of order $(2n + 1)$ has a factorization consisting of n Hamiltonian cycles.

7.57 Show that a complete graph of odd order cannot be factored into Hamiltonian paths.

Solution. If the order of the graph is $(2n + 1)$, each Hamiltonian path will have $2n$ edges. The total number of edges in the graph is $n(2n + 1)$. Since $2n$ is not a divisor of $n(2n + 1)$, there is no factorization into Hamiltonian paths.

7.58 (*a*) Find two nonisomorphic-connected 1-factorable cubic graphs of the same order. (*b*) Find two nonisomorphic connected cubic graphs such that each can be factored into a 1-factor and a Hamiltonian cycle.

Solution. The two cubic graphs of order 6 shown in Fig. 7-14 are not isomorphic because one of them has a triangle as a subgraph while the other does not.

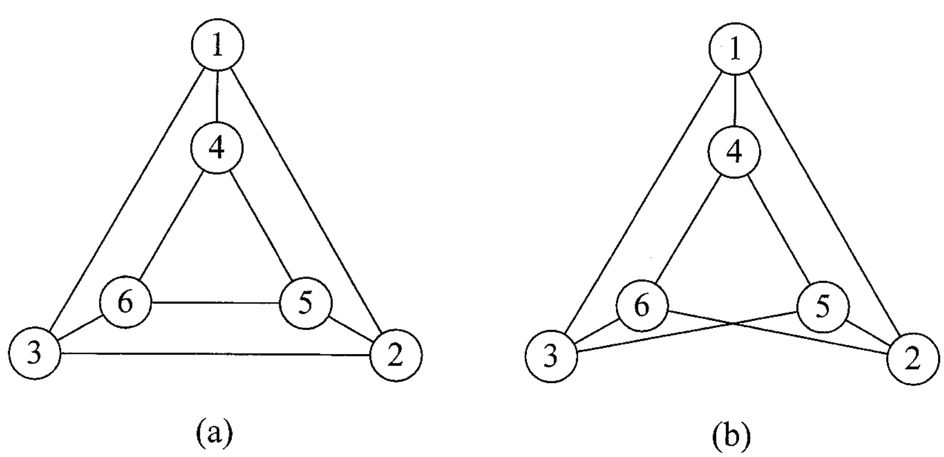


Fig. 7-14

(*a*) The graph in Fig. 7-14(*a*) can be factored into $M_1 = \{(1, 3), (2, 5), (4, 6)\}$, $M_2 = \{(3, 6), (4, 5), (1, 2)\}$, and $M_3 = \{(2, 3), (5, 6), (1, 4)\}$. The graph in Fig. 7-14(*b*) can be factored into $M_1 = \{(1, 3), (2, 5), (4, 6)\}$, $M_2 = \{(3, 6), (4, 5), (1, 2)\}$, and $M_3 = \{(3, 5), (2, 6), (1, 4)\}$.

(*b*) The graph in Fig. 7-14(*a*) can be factored into the Hamiltonian cycle $1 \to 2 \to 3 \to 6 \to 5 \to 4 \to 1$ and the 1-factor $M = \{(1, 3), (2, 5), (4, 6)\}$. The graph in Fig. 7-14(*b*) can be factored into the Hamiltonian cycle $1 \to 2 \to 5 \to 3 \to 6 \to 4 \to 1$ and $M = \{(1, 3), (2, 6), (4, 5)\}$.

7.59 Prove Theorem 7.6 (Petersen's Theorem): A cubic graph G in which no edge is a bridge can be factored into a 2-factor and a 1-factor.

Solution. If G has a 1-factor, the removal of the edges belonging to the 1-factor results in a 2-factor. So it is enough if we show that G has a 1-factor. Let $H_1, H_2, \ldots, H_k$ be the odd components of $G - S$, where S is any set of vertices. Since G has no bridges, the number of edges between S and any odd component H_i is more than 1. Suppose the number m_i of edges between S and H_i is an even number $2q$. If the number of vertices in H_i is $(2q + 1)$, the sum of the degrees of the vertices in H_i will be $3(2q + 1) - 2q$, which will be an odd number. Hence, m_i is at least 3 for each i. Thus $k \leq \frac{1}{3}(m)$, where $m = \{m_1 + m_2 + \cdots + m_k\}$. But $m \leq 3|S|$. So $k \leq |S|$, where k is the number of odd components of $G - S$. Hence, by Tutte's theorem, G has a 1-factor.

7.60 Prove Theorem 7.7: The Petersen graph is not 1-factorable.

Solution. The Petersen graph (see Fig. 7-3) has 15 edges. Vertices 1, 2, 3, 4, and 5 are the outer vertices; the others are the inner vertices. If it is 1-factorable, it will have three 1-factors each with five edges. So if any set F of five edges of the graph is chosen at random, there will be at least one 1-factor that contains at least two edges from F. Let F be the set of five edges joining an outer vertex and the unique inner vertex adjacent to it. Suppose H is the 1-factor that contains edge e that joins vertex 1 and vertex 6 and edge f that joins vertex 2 and vertex 7. But if these four vertices are deleted from the Petersen graph, the remaining six vertices cannot be matched in three pairs.

7.61 Give an example of a cubic bridgeless graph of order 10 that is 1-factorable.

Solution. The bridgeless cubic graph shown in Fig. 7-15 can obviously be factored into a 2-factor (consisting of two disjoint cycles) and a 1-factor. It also can be factored into three 1-factors as follows: $M_1 = \{(1, 2), (3, 4), (5, 6), (7, 8), (9, 10)\}$, $M_2 = \{(1, 3), (2, 4), (6, 7), (8, 9), (5, 10)\}$, and $M_3 = \{(1, 4), (2, 3), (6, 10), (7, 9), (5, 8)\}$.

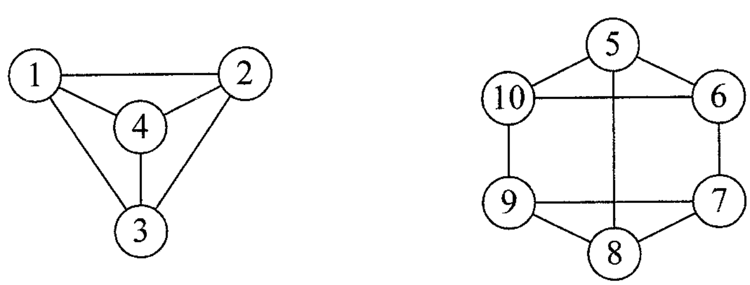


Fig. 7-15

7.62 Give an example of a 1-factorable Hamiltonian cubic graph of order 10.

Solution. The cycle $C: 1 \to 2 \to 3 \to 4 \to 5 \to 10 \to 9 \to 8 \to 7 \to 6 \to 1$ in Fig. 7-16 is a Hamiltonian cycle. The edges that are not in this cycle form a matching M, and thus the bridgeless cubic graph is factored into a 2-factor and a 1-factor. Furthermore, the graph can be factored into the three following 1-factors: arcs indicated by the labels A, B, and C.

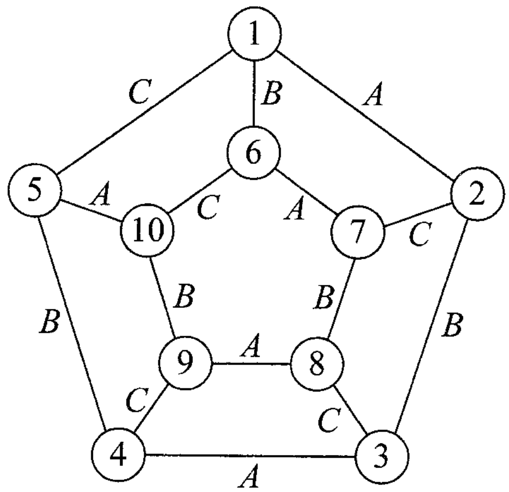


Fig. 7-16

7.63 Show that $K_{2n} = (2n - 1)H$ and $K_{n,n} = nH$, where $H = nK_2$.

Solution. The 1-factorable graph K_{2n} with $(n)(2n - 1)$ edges has an isomorphic factorization consisting of $(2n - 1)$ copies of its isofactor nK_2. Similarly, the 1-factorable bipartite graph $K_{n,n}$ with n^2 edges has an isomorphic factorization consisting of n copies of K_2.

7.64 Find an isofactor of the Petersen graph (with five sides) other than what is shown in Fig. 7-4.

Solution. The graph shown in Fig. 7-17 is another isofactor of the Petersen graph.

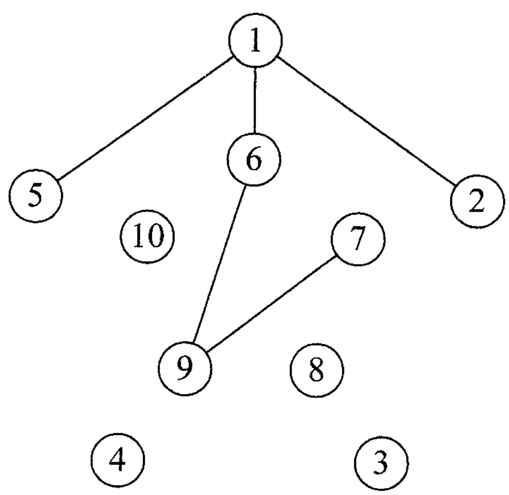


Fig. 7-17

7.65 Show that the Petersen graph is 3-connected.

Solution. Between any two vertices, there are three internally disjoint paths. So, by Whitney's theorem, the graph is 3-connected. Furthermore, the vertex connectivity number and the edge connectivity number are both 3 since the maximum degree is 3.

7.66 Find an isofactor of the Petersen graph with three edges.

Solution.

(*i*) A spanning H subgraph consisting of six vertices of degree 1 and four vertices of degree 0 is an isofactor with three sides. See Fig. 7-18. There are five copies of the isofactor, giving an isomorphic factorization of the Petersen graph. The edges of these five copies are marked A, B, C, D, and E.

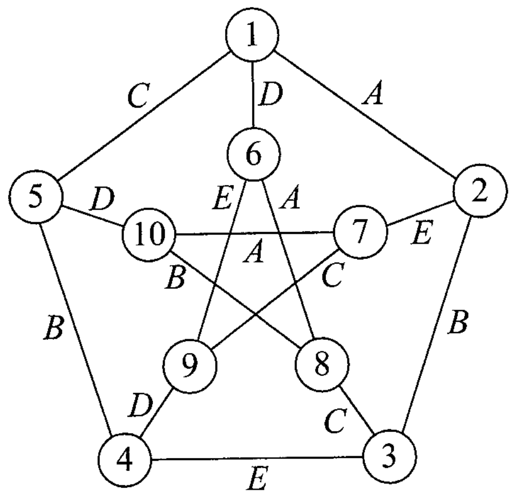


Fig. 7-18

(*ii*) A spanning subgraph with degree vector [2 1 1 1 1 0 0 0 0] is another isofactor with three sides, as shown in Fig. 7-19.

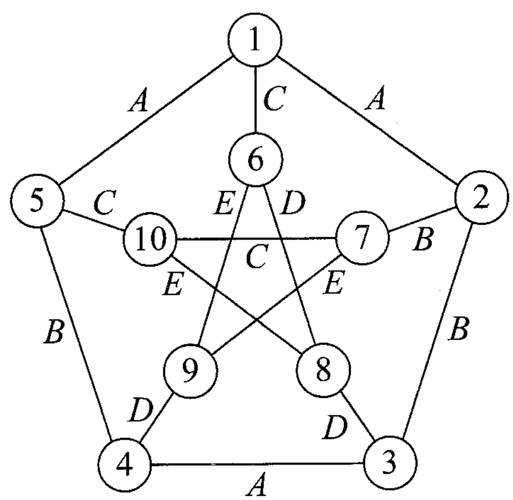


Fig. 7-19

7.67 (*Hartsfield-Ringel Theorem*) Show that the path with three edges is an isofactor of any cubic bridgeless graph.

Solution. Every bridgeless cubic graph can be factored into a 2-factor and a 1-factor. Suppose the edges belonging to a 1-factor are colored blue and the edges belonging to a 2-factor are colored red. So every vertex is incident to one blue edge and two red edges. The blue edges are labeled 1, 2, 3, . . . arbitrarily. Now we traverse the edges belonging to a cycle (a component of the 2-factor) in one direction. While doing so, each red edge from vertex u to vertex v in the cycle is assigned the same number as the label of the blue edge that is incident to vertex u. Then each number appears on three consecutive edges. Thus P_3 is an isofactor.

7.68 Obtain an isomorphic factorization of the Petersen graph such that each factor is a path consisting of three edges.

Solution. In Fig. 7-20, each edge joining an outer vertex to the corresponding inner vertex belongs to a 1-factor, and the five blue edges belonging to this 1-factor are labeled 1, 2, 3, 4, and 5. The 2-factor consists of two components: one is the cycle $A \to B \to C \to E \to A$ and the other is $F \to H \to J \to G \to I \to F$. As we move from vertex A in the counterclockwise direction, edge AB gets label 1. As we move from F in the counterclockwise direction, edge FH gets label 1. Thus the path $H \to F \to A \to B$ is one copy of the isofactor. The remaining four copies correspond to the four paths labeled 2, 3, 4, and 5 respectively.

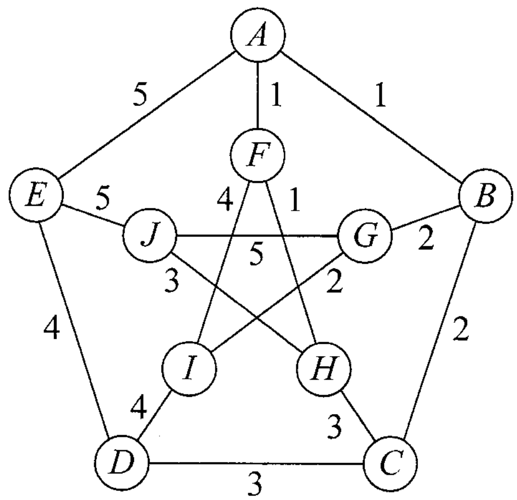


Fig. 7-20

7.69 Give examples of bipartite and nonbipartite cubic bridgeless graphs with a Hamiltonian cycle as a 2-factor.

Solution.

(*i*) The cube graph Q_3 with eight vertices [see Fig. 7-21(*a*)] is a bipartite bridgeless graph that can be factored into a Hamiltonian cycle and a 1-factor consisting of the pairs (1, 4), (2, 7), (3, 6), and (5, 8).

(*ii*) The cubic graph with eight vertices [see Fig. 7-21(*b*)] is a nonbipartite bridgeless graph that can be factored into a Hamiltonian cycle and a 1-factor consistng of the pairs (1, 3), (2, 4), (5, 7), and (6, 8).

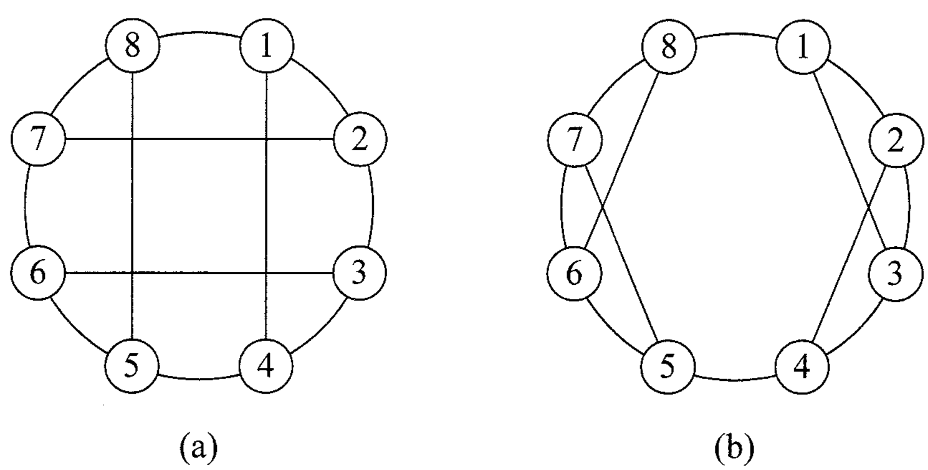


Fig. 7-21