called maximum-flow problems. Several specialized algorithms exist to solve maximum-flow problems. In this section, we begin by showing how linear programming can be used to solve a maximum-flow problem. Then we discuss the Ford-Fulkerson (1962) method for solving maximum-flow problems.

**LP Solution of Maximum-Flow Problems**

**Example 3**

**Maximum Flow**

Sunco Oil wants to ship the maximum possible amount of oil (per hour) via pipeline from node so to node si in Figure 6. On its way from node so to node si, oil must pass through some or all of stations 1, 2, and 3. The various arcs represent pipelines of different diameters. The maximum number of barrels of oil (millions of barrels per hour) that can be pumped through each arc is shown in Table 8. Each number is called an arc capacity. Formulate an LP that can be used to determine the maximum number of barrels of oil per hour that can be sent from so to si.

**Solution**

Node so is called the source node because oil flows out of it but no oil flows into it. Analogously, node si is called the sink node because oil flows into it and no oil flows out of it. For reasons that will soon become clear, we have added an artificial arc a0 from the sink to the source. The flow through a0 is not actually oil, hence the term artificial arc.

To formulate an LP that will yield the maximum flow from node so to si, we observe that Sunco must determine how much oil (per hour) should be sent through arc (i, j). Thus, we define

\[ x_{ij} = \text{millions of barrels of oil per hour that will pass through arc (i, j)} \]

As an example of a possible flow (termed a feasible flow), consider the flow identified by the numbers in parentheses in Figure 6.

\[ x_{so,1} = 2, \quad x_{13} = 0, \quad x_{12} = 2, \quad x_{3,si} = 0, \quad x_{2,si} = 2, \quad x_{si,so} = 2, \quad x_{so,2} = 0 \]

**Table 8**

Arc Capacities for Sunco Oil

<table>
<thead>
<tr>
<th>Arc</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>(so, 1)</td>
<td>2</td>
</tr>
<tr>
<td>(so, 2)</td>
<td>3</td>
</tr>
<tr>
<td>(1, 2)</td>
<td>3</td>
</tr>
<tr>
<td>(1, 3)</td>
<td>4</td>
</tr>
<tr>
<td>(3, si)</td>
<td>1</td>
</tr>
<tr>
<td>(2, si)</td>
<td>2</td>
</tr>
</tbody>
</table>
For a flow to be feasible, it must have two characteristics:

\[ 0 \leq \text{flow through each arc} \leq \text{arc capacity} \quad (1) \]

and

\[ \text{Flow into node } i = \text{flow out of node } i \quad (2) \]

We assume that no oil gets lost while being pumped through the network, so at each node, a feasible flow must satisfy (2), the conservation-of-flow constraint. The introduction of the artificial arc \( a_0 \) allows us to write the conservation-of-flow constraint for the source and sink.

If we let \( x_0 \) be the flow through the artificial arc, then conservation of flow implies that \( x_0 = \text{total amount of oil entering the sink} \). Thus, Sunco's goal is to maximize \( x_0 \) subject to (1) and (2):

\[
\begin{align*}
\text{max } z &= x_0 \\
\text{s.t. } & x_{so,1} \leq 2 \quad \text{(Arc capacity constraints)} \\
& x_{so,2} \leq 3 \\
& x_{12} \leq 3 \\
& x_{2,i} \leq 2 \\
& x_{13} \leq 4 \\
& x_{3,si} \leq 1 \\
& x_0 = x_{so,1} + x_{so,2} \quad \text{(Node so flow constraint)} \\
& x_{so,1} = x_{12} + x_{13} \quad \text{(Node 1 flow constraint)} \\
& x_{so,2} + x_{12} = x_{2,si} \quad \text{(Node 2 flow constraint)} \\
& x_{13} = x_{3,si} \quad \text{(Node 3 flow constraint)} \\
& x_{3,si} + x_{2,si} = x_0 \quad \text{(Node si flow constraint)} \\
& x_f \geq 0
\end{align*}
\]

One optimal solution to this LP is \( z = 3, x_{so,1} = 2, x_{13} = 1, x_{12} = 1, x_{so,2} = 1, x_{3,si} = 1, x_{2,si} = 2, x_0 = 3 \). Thus, the maximum possible flow of oil from node so to si is 3 million barrels per hour, with 1 million barrels each sent via the following paths: so–1–2–si, so–1–3–si, and so–2–si.

The linear programming formulation of maximum flow problems is a special case of the minimum-cost network flow problem (MCNFP) discussed in Section 8.5. A generalization of the transportation simplex (known as the network simplex) can be used to solve MCNFPs.

Before discussing the Ford–Fulkerson method for solving maximum-flow problems, we give two examples for situations in which a maximum-flow problem might arise.

**Example 4: Airline Maximum-Flow**

Fly-by-Night Airlines must determine how many connecting flights daily can be arranged between Juneau, Alaska, and Dallas, Texas. Connecting flights must stop in Seattle and then stop in Los Angeles or Denver. Because of limited landing space, Fly-by-Night is limited to making the number of daily flights between pairs of cities shown in Table 9.

Set up a maximum-flow problem whose solution will tell the airline how to maximize the number of connecting flights daily from Juneau to Dallas.
Table 9

<table>
<thead>
<tr>
<th>Cities</th>
<th>Maximum Number of Daily Flights</th>
</tr>
</thead>
<tbody>
<tr>
<td>Juneau–Seattle (J, S)</td>
<td>3</td>
</tr>
<tr>
<td>Seattle–L.A. (S, L)</td>
<td>2</td>
</tr>
<tr>
<td>Seattle–Denver (S, De)</td>
<td>3</td>
</tr>
<tr>
<td>L.A.–Dallas (L, D)</td>
<td>1</td>
</tr>
<tr>
<td>Denver–Dallas (De, D)</td>
<td>2</td>
</tr>
</tbody>
</table>

Figure 7
Network for Fly-by-Night Airlines

Solution
The appropriate network is given in Figure 7. Here the capacity of arc (i, j) is the maximum number of daily flights between city i and city j. The optimal solution to this maximum flow problem is z = x_j = 3, x_{J,S} = 3, x_{S,L} = 1, x_{S,De} = 2, x_{L,D} = 1, x_{De,D} = 2. Thus, Fly-by-Night can send three flights daily connecting Juneau and Dallas. One flight connects via Juneau–Seattle–L.A.–Dallas, and two flights connect via Juneau–Seattle–Denver–Dallas.

Example 5
Matchmaking

Five male and five female entertainers are at a dance. The goal of the matchmaker is to match each woman with a man in a way that maximizes the number of people who are matched with compatible mates. Table 10 describes the compatibility of the entertainers. Draw a network that makes it possible to represent the problem of maximizing the number of compatible pairings as a maximum-flow problem.

Solution
Figure 8 is the appropriate network. In Figure 8, there is an arc with capacity 1 joining the source to each man, an arc with capacity 1 joining each pair of compatible mates, and an arc with capacity 1 joining each woman to the sink. The maximum flow in this network is the number of compatible couples that can be created by the matchmaker. For ex-

Table 10
Compatibilities for Matching

<table>
<thead>
<tr>
<th></th>
<th>Leni Anderson</th>
<th>Meryl Streep</th>
<th>Katharine Hepburn</th>
<th>Linda Evans</th>
<th>Victoria Principal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kevin Costner</td>
<td>-</td>
<td>C</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Burt Reynolds</td>
<td>C</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Tom Selleck</td>
<td>C</td>
<td>C</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Michael Jackson</td>
<td>C</td>
<td>C</td>
<td>-</td>
<td>-</td>
<td>C</td>
</tr>
<tr>
<td>Tom Cruise</td>
<td>-</td>
<td>-</td>
<td>C</td>
<td>C</td>
<td>C</td>
</tr>
</tbody>
</table>

Note: C indicates compatibility.
ample, if the matchmaker pairs KC and MS, BR and LA, MJ and VP, and TC and KH, a flow of 4 from source to sink would be obtained. (This turns out to be a maximum flow for the network.)

To see why our network representation correctly models the matchmaker's problem, note that because the arc joining each woman to the sink has a capacity of 1, conservation of flow ensures that each woman will be matched with at most one man. Similarly, because each arc from the source to a man has a capacity of 1, each man can be paired with at most one woman. Because arcs do not exist between noncompatible mates, we can be sure that a flow of \( k \) units from source to sink represents an assignment of men to women in which \( k \) compatible couples are created.

Solving Maximum-Flow Problems with LINGO

The maximum flow in a network can be found using LINDO, but LINGO greatly lessens the effort needed to communicate the necessary information to the computer. The following LINGO program (in the file Maxflow.ling) can be used to find the maximum flow from source to sink in Figure 6.

```lingo
MODEL:
1)SETS:
  2)NODES/1...5/;
3)ARCS(NODES,NODES)/1,2 1,3 2,3 2,4 3,5 4,5 5,1/;
4)CAP,FLOW;
5)ENDSETS
6)MAX=FLOW (5,1);   
7)@FOR (ARCS(I,J): FLOW(I,J)<CAP(I,J));
8)@FOR (NODES(I):@SUM (ARCS(I,J): FLOW(I,J)));
9)@SUM (ARCS(I,J): FLOW(I,J));
10)DATA:
11)CAP=2,3,3,4,2,1,2000;
12)ENDDATA
END
```

If some nodes are identified by numbers, then LINGO will not allow you to identify other nodes with names involving letters. Thus, we have identified node 1 in line 2 with node so in Figure 6 and node 5 in line 2 with node st. Also nodes 1, 2, and 3 in Figure 6 correspond to nodes 2, 3, and 4, respectively, in line 2 of our LINGO program. Thus, line 2 defines the nodes of the flow network. In line 3, we define the arcs of the network by listing them (separated by spaces). For example, 1, 2 represents the arc from the source to node 1 in Figure 6 and 5,1 is the artificial arc. In line 4, we indicate that an arc capacity and a flow are associated with each arc. Line 5 ends the definition of the relevant sets.

In line 6, we indicate that our objective is to maximize the flow through the artificial arc (this equals the flow into the sink). Line 7 specifies the arc capacity constraints; for
each arc, the flow through the arc cannot exceed the arc’s capacity. Lines 8 and 9 create the conservation of flow constraints. For each node I, they ensure that the flow into node I equals the flow out of node I.

Line 10 begins the DATA section. In line 11, we input the arc capacities. Note that we have given the artificial arc a large capacity of 1,000. Line 12 ends the DATA section and the END statement ends the program. Typing GO yields the solution, a maximum flow of 3 previously described. The values of the variable FLOW(I,J) give the flow through each arc.

Note that this program can be used to find the maximum flow in any network. Begin by listing the network’s nodes in line 2. Then list the network’s arcs in line 3. Finally, list the capacity of each arc in the network in line 11, and you are ready to find the maximum flow in the network! In Line 6 we must list the arc that defines the maximum flow in the network.

**The Ford–Fulkerson Method for Solving Maximum-Flow Problems**

We assume that a feasible flow has been found (letting the flow in each arc equal zero gives a feasible flow), and we turn our attention to the following important questions:

**Question 1** Given a feasible flow, how can we tell if it is an optimal flow (that is, maximizes \( x_0 \))?  

**Question 2** If a feasible flow is nonoptimal, how can we modify the flow to obtain a new feasible flow that has a larger flow from the source to the sink?

First, we answer question 2. We determine which of the following properties is possessed by each arc in the network:

**Property 1** The flow through arc \((i, j)\) is below the capacity of arc \((i, j)\). In this case, the flow through arc \((i, j)\) can be increased. For this reason, we let \( I \) represent the set of arcs with this property.

**Property 2** The flow in arc \((i, j)\) is positive. In this case, the flow through arc \((i, j)\) can be reduced. For this reason, we let \( R \) be the set of arcs with this property.

As an illustration of the definitions of \( I \) and \( R \), consider the network in Figure 9. The arcs in this figure may be classified as follows: \((s_0, 1)\) is in \( I \) and \( R \); \((s_0, 2)\) is in \( I \); \((1, s_i)\) is in \( R \); \((2, s_i)\) is in \( R \); and \((2, 1)\) is in \( R \).

We can now describe the Ford–Fulkerson labeling procedure used to modify a feasible flow in an effort to increase the flow from the source to the sink.

**Step 1** Label the source.

**Step 2** Label nodes and arcs (except for arc \( a_0 \)) according to the following rules: (1) If node \( x \) is labeled, then node \( y \) is unlabeled and arc \((x, y)\) is a member of \( I \); then label node \( y \) and arc \((x, y)\). In this case, arc \((x, y)\) is called a forward arc. (2) If node \( y \) is unlabeled, node \( x \) is labeled and arc \((y, x)\) is a member of \( R \); label node \( y \) and arc \((y, x)\). In this case, \((y, x)\) is called a backward arc.

**Figure 9** Illustration of \( I \) and \( R \) arcs
Step 3 Continue this labeling process until the sink has been labeled or until no more vertices can be labeled.

If the labeling process results in the sink being labeled, then there will be a chain of labeled arcs (call it $C$) leading from the source to the sink. By adjusting the flow of the arcs in $C$, we can maintain a feasible flow and increase the total flow from source to sink. To see this, observe that $C$ must consist of one of the following:

Case 1 $C$ consists entirely of forward arcs.

Case 2 $C$ contains both forward and backward arcs.\(^\dagger\)

In each case, we can obtain a new feasible flow that has a larger flow from source to sink than the current feasible flow. In Case 1, the chain $C$ consists entirely of forward arcs. For each forward arc in $C$, let $f(x, y)$ be the amount by which the flow in arc $(x, y)$ can be increased without violating the capacity constraint for arc $(x, y)$. Let

$$k = \min_{(x, y) \in C} f(x, y)$$

Then $k > 0$. To create a new flow, increase the flow through each arc in $C$ by $k$ units. No capacity constraints are violated, and conservation of flow is still maintained. Thus, the new flow is feasible, and the new feasible flow will transport $k$ more units from source to sink than does the current feasible flow.

We use Figure 10 to illustrate Case 1. Currently, 2 units are being transported from source to sink. The labeling procedure results in the sink being labeled by the chain $C = (s_0, 1) - (1, 2) - (2, s_4)$. Each arc is in $I$, and $f(s_0, 1) = 5 - 2 = 3, f(1, 2) = 3 - 2 = 1$, and $f(2, s_4) = 4 - 2 = 2$. Hence, $k = \min(3, 1, 2) = 1$. Thus, an improved feasible flow can be obtained by increasing the flow on each arc in $C$ by 1 unit. The resulting flow transports 3 units from source to sink (see Figure 11).

In Case 2, the chain $C$ leading from the source to the sink contains both backward and forward arcs. For each backward arc in $C$, let $r(x, y)$ be the amount by which the flow through arc $(x, y)$ can be reduced. Also define

$$k_1 = \min_{x, y \in C \cap R} r(x, y) \quad \text{and} \quad k_2 = \min_{x, y \in C \cap U} f(x, y)$$

\(^\dagger\)Because we exclude arcs $a_0$ from the labeling procedure, no chain made entirely of backward arcs can lead from source to sink.

8.3 Maximum-Flow Problems 425
Of course, both $k_1$ and $k_2$ and $\min(k_1, k_2)$ are $> 0$. To increase the flow from source to sink (while maintaining a feasible flow), decrease the flow in all of $C$'s backward arcs by $\min(k_1, k_2)$ and increase the flow in all of $C$'s forward arcs by $\min(k_1, k_2)$. This will maintain conservation of flow and ensure that no arc capacity constraints are violated. Because the last arc in $C$ is a forward arc leading into the sink, we have found a new feasible flow and have increased the total flow into the sink by $\min(k_1, k_2)$. We now adjust the flow in the arc $a_0$ to maintain conservation of flow. To illustrate Case 2, suppose we have found the feasible flow in Figure 12. For this flow, $(s_0, 1) \in R$; $(s_0, 2) \in \mathcal{I}; (1, 3) \in \mathcal{I}; (1, 2) \in \mathcal{I}$ and $R; (2, s_1) \in R$; and $(3, s_1) \in \mathcal{I}$.

We begin by labeling arc $(s_0, 2)$ and node 2 (thus $(s_0, 2)$ is a forward arc). Then we label arc $(1, 2)$ and node 1. Arc $(1, 2)$ is a backward arc, because node 1 was unlabeled before we labeled arc $(1, 2)$, and arc $(1, 2)$ is in $R$. Nodes $s_0$, 1, and 2 are labeled, so we can label arc $(1, 3)$ and node 3. [Arc $(1, 3)$ is a forward arc, because node 3 has not yet been labeled.] Finally we label arc $(3, s_1)$ and node $s_1$. Arc $(3, s_1)$ is a forward arc, because node $s_1$ has not yet been labeled. We have now labeled the sink via the chain $C = (s_0, 2) - (1, 2) - (1, 3) - (3, s_1)$. With the exception of arc $(1, 2)$, all arcs in the chain are forward arcs. Because $h(s_0, 2) = 3; h(1, 3) = 4; h(3, s_1) = 1$; and $r(1, 2) = 2$, we have

$$\min_{(x, y) \in C \cap R} r(x, y) = 2 \quad \text{and} \quad \min_{(x, y) \in C \cap \mathcal{I}} h(x, y) = 1$$

Thus, we can increase the flow on all forward arcs in $C$ by 1 and decrease the flow in all backward arcs by 1. The new result, pictured in Figure 13, has increased the flow from source to sink by 1 unit (from 2 to 3). We accomplish this by diverting 1 unit that was transported through the arc $(1, 2)$ to the path $1 – 3 – s_1$. This enabled us to transport an extra unit from source to sink via the path $s_0 - 2 - s_1$. Observe that the concept of a backward arc was needed to find this improved flow.

If the sink cannot be labeled, then the current flow is optimal. The proof of this fact relies on the concept of a cut for a network.

**Definition**: Choose any set of nodes $V'$ that contains the sink but does not contain the source. Then the set of arcs $(i, j)$ with $i$ not in $V'$ and $j$ a member of $V'$ is a cut for the network.

**Figure 12**: Illustration of Case 2 of Labeling Method

Flow from source to sink = 2
Chain is $(s_0, 2) - (1, 2) - (1, 3) - (3, s_1)$

**Figure 13**: Improved Flow from Source to Sink: Case 2

Flow from source to sink = 3

426 Chapter 8 Network Models
The capacity of a cut is the sum of the capacities of the arcs in the cut.

In short, a cut is a set of arcs whose removal from the network makes it impossible to travel from the source to the sink. A network may have many cuts. For example, in the network in Figure 14, \( V' = \{1, st\} \) yields the cut containing the arcs \((so, 1)\) and \((2, st)\), which has capacity \(2 + 1 = 3\). The set \( V'' = \{1, 2, st\} \) yields the cut containing the arcs \((so, 1)\) and \((so, 2)\), which has capacity \(2 + 8 = 10\).

Lemma 1 and Lemma 2 indicate the connection between cuts and maximum flows.

**Lemma 1.**

The flow from source to sink for any feasible flow is less than or equal to the capacity of any cut.

**Proof.** Consider an arbitrary cut specified by a set of nodes \( V' \) that contains the sink but does not contain the source. Let \( V \) be all other nodes in the network. Also let \( x_{ij} \) be the flow in arc \((i, j)\) for any feasible flow and \( f \) be the flow from source to sink for this feasible flow. Summing the flow balance equations (flow out of node \( i \) = flow into node \( i = 0 \)) over all nodes \( i \) in \( V \), we find that the terms involving arcs \((i, j)\) having \( i \) and \( j \) both members of \( V' \) will cancel, and we obtain

\[
\sum_{i \in V'} \sum_{j \in V'} x_{ij} = f
\]

Now the first sum in (3) equals the capacity of the cut. Each \( x_{ij} \) is non-negative, so we see that \( f \leq \text{capacity of the cut} \), which is the desired result.

Lemma 1 is analogous to the weak duality result discussed in Chapter 6. From Lemma 1, we see that the capacity of any cut is an upper bound for the maximum flow from source to sink. Thus, if we can find a feasible flow and a cut for which the flow from source to sink equals the capacity of the cut, then we have found the maximum flow from source to sink.

Suppose that we find a feasible flow and cannot label the sink. Let \( \text{CUT} \) be the cut corresponding to the set of unlabeled nodes.

**Lemma 2.**

If the sink cannot be labeled, then

\[
\text{Capacity of CUT} = \text{current flow from source to sink}
\]

**Proof.** Let \( V' \) be the set of unlabeled nodes and \( \bar{V} \) be the set of labeled nodes. Consider an arc \((i, j)\) such that \( i \) is in \( \bar{V} \) and \( j \) is in \( V' \). Then we know that \( x_{ij} = \text{capac-} \)

8.3 Maximum-Flow Problems
Summary and Illustration of the Ford–Fulkerson Method

**Step 1** Find a feasible flow (setting each arc’s flow to zero will do).

**Step 2** Using the labeling procedure, try to label the sink. If the sink cannot be labeled, then the current feasible flow is a maximum flow; if the sink is labeled, then go on to step 3.

**Step 3** Using the method previously described, adjust the feasible flow and increase the flow from the source to the sink. Return to step 2.

To illustrate the Ford–Fulkerson method, we find the maximum flow from source to sink for Sunco Oil, Example 3 (see Figure 6). We begin by letting the flow in each arc equal zero. We then try to label the sink—label the source, and then arc (so, 1) and node 1; then label arc (1, 2) and node 2; finally, label arc (2, st) and node st. Thus, C = (so, 1) → (1, 2) → (2, st). Each arc in C is a forward arc, so we can increase the flow through each arc in C by min (2, 3, 2) = 2 units. The resulting flow is pictured in Figure 15.

As we saw previously (Figure 12), we can label the sink by using the chain C = (so, 2) → (1, 2) → (1, 3) → (3, st). We can increase the flow through the forward arcs (so, 2), (1, 3), and (3, st) by 1 unit and decrease the flow through the backward arc (1, 2) by 1 unit. The resulting flow is pictured in Figure 16. It is now impossible to label the sink. Any attempt to label the sink must begin by labeling arc (so, 2) and node 2; then we could label arc (1, 2) and arc (1, 3). But there is no way to label the sink.

We can verify that the current flow is maximal by finding the capacity of the cut corresponding to the set of unlabeled vertices (in this case, si). The cut corresponding to si is the set of arcs (2, st) and (3, st), with capacity 2 + 1 = 3. Thus, Lemma 1 implies that any feasible flow can transport at most 3 units from source to sink. Our current flow transports 3 units from source to sink, so it must be an optimal flow.

Another example of the Ford–Fulkerson method is given in Figure 17. Note that without the concept of a backward arc, we could not have obtained the maximum flow of 7.
**FIGURE 16**
Network for Sunco Oil (Optimal Flow)

Flow from source to sink = 3
Since sink cannot be labeled, this is an optimal flow.

**FIGURE 17**
Example of Ford-Fulkerson Method

a. Original network

- Label sink by so = 3 - si (adds 3 units of flow using only forward arcs)

b. Label sink by so = 3 - si (adds 3 units of flow using only forward arcs)

c. Label sink by so = 1 - i - 2 - 3 - si (adds 2 units of flow using only forward arcs)

d. Label sink by so = 2 - 1 - si (adds 2 units of flow using backward arc (1, 2)); maximum flow of ? has been obtained.

units from source to sink. The minimum cut (with capacity 7, of course) corresponds to nodes 1, 3, and si and consists of arcs (so, 1), (so, 3) and (2, 3).

**PROBLEMS**

**Group A**

1-3 Figures 18–20 show the networks for Problems 1–3. Find the maximum flow from source to sink in each network. Find a cut in the network whose capacity equals the maximum flow in the network. Also, set up an LP that could be used to determine the maximum flow in the network.

**FIGURE 18**
Network for Problem 1

- Maximum-Flow Problems 429
used to determine whether the packages can be loaded so that no truck carries two packages of the same type.

7 Four workers are available to perform jobs 1–4. Unfortunately, three workers can do only certain jobs: worker 1, only job 1; worker 2, only jobs 1 and 2; worker 3, only job 2; worker 4, any job. Draw the network for the maximum-flow problem that can be used to determine whether all jobs can be assigned to a suitable worker.

8 The Hatfields, Montagues, McCoys, and Capulets are going on their annual family picnic. Four cars are available to transport the families to the picnic. The cars can carry the following number of people: car 1, four; car 2, three; car 3, three; and car 4, four. There are four people in each family, and no car can carry more than two people from any one family. Formulate the problem of transporting the maximum possible number of people to the picnic as a maximum-flow problem.

9–10 For the networks in Figures 23 and 24, find the maximum flow from source to sink. Also find a cut whose capacity equals the maximum flow in the network.

Group B

11 Suppose a network contains a finite number of arcs and the capacity of each arc is an integer. Explain why the Ford–Fulkerson method will find the maximum flow in the finite number of steps. Also show that the maximum flow from source to sink will be an integer.

12 Consider a network flow problem with several sources and several sinks in which the goal is to maximize the total flow into the sinks. Show how such a problem can be converted into a maximum flow problem having only a single source and a single sink.

4–5 For the networks in Figures 21 and 22, find the maximum flow from source to sink. Also find a cut whose capacity equals the maximum flow in the network.

6 Seven types of packages are to be delivered by five trucks. There are three packages of each type, and the capacities of the five trucks are 6, 4, 5, 4, and 3 packages, respectively. Set up a maximum-flow problem that can be
13 Suppose the total flow into a node of a network is restricted to 10 units or less. How can we represent this restriction via an arc capacity constraint? (This still allows us to use the Ford–Fulkerson method to find the maximum flow.)

14 Suppose as many as 300 cars per hour can travel between any two of the cities 1, 2, 3, and 4. Set up a maximum-flow problem that can be used to determine how many cars can be sent in the next two hours from city 1 to city 4. (Hint: Have portions of the network represent \( i = 0, \)
\( j = 1, \) and \( t = 2 \).)

15 Fly-by-Night Airlines is considering flying three flights. The revenue from each flight and the airports used by each flight are shown in Table 11. When Fly-by-Night uses an airport, the company must pay the following landing fees (independent of the number of flights using the airport): airport 1, $300; airport 2, $700; airport 3, $500. Thus, if flights 1 and 3 are flown, a profit of $900 + $800 - $300 - $700 - $500 = $200 will be earned. Show that for the network in Figure 25 (maximum profit) = (total revenue from all flights) - (capacity of minimal cut). Explain how this result can be used to help Fly-by-Night maximize profit (even if it has hundreds of possible flights). (Hint: Consider any set of flights \( F \) (say flights 1 and 3). Consider the cut corresponding to the sink, the nodes associated with the flights not in \( F \), and the nodes associated with the airports not used by \( F \). Show that (capacity of this cut) = (revenue from flights not in \( F \)) + (costs associated with airports used by \( F \)).)

16 During the next four months, a construction firm must complete three projects. Project 1 must be completed within three months and requires 8 months of labor. Project 2 must be completed within four months and requires 10 months of labor. Project 3 must be completed at the end of two months and requires 12 months of labor. Each month, 8 workers are available. During a given month, no more than 6 workers can work on a single job. Formulate a maximum-flow problem that could be used to determine whether all three projects can be completed on time. (Hint: If the maximum flow in the network is 30, then all projects can be completed on time.)

**Figure 25**
Network for Problem 15

---

### 8.4 CPM and PERT

Network models can be used as an aid in scheduling large complex projects that consist of many activities. If the duration of each activity is known with certainty, then the critical path method (CPM) can be used to determine the length of time required to complete a project. CPM can also be used to determine how long each activity in the project can be delayed without delaying the completion of the project. CPM was developed in the late 1950s by researchers at DuPont and Sperry Rand.

If the duration of the activities is not known with certainty, the Program Evaluation and Review Technique (PERT) can be used to estimate the probability that the project will be completed by a given deadline. PERT was developed in the late 1950s by consultants working on the development of the Polaris missile. CPM and PERT were given a major share of the credit for the fact that the Polaris missile was operational two years ahead of schedule.

CPM and PERT have been successfully used in many applications, including:

1. Scheduling construction projects such as office buildings, highways, and swimming pools

---

**Table 11**

<table>
<thead>
<tr>
<th>Flight</th>
<th>Revenue ($)</th>
<th>Airport Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>900</td>
<td>1 and 2</td>
</tr>
<tr>
<td>2</td>
<td>600</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>800</td>
<td>2 and 3</td>
</tr>
</tbody>
</table>