13

Flows in Networks

13.1 Transportation Networks and Flows

Graphs are used to model situations in which a commodity is transported from one location to another. A common example is the water supply, where the pipelines are edges, vertices represent water users, pipe joins, and so on. In the example of an airline, given in Section 2.3, we can interpret freight or passengers as commodities to be transported. Highway systems can be thought of as transporting cars. In many examples it is natural to interpret some or all edges as directed (some roads are one-way, water can flow only in one direction at a time in a given pipe, and so on). A common feature of transportation systems is the existence of a *capacity* associated with each edge — the maximum number of cars that can use a road in an hour, the maximum amount of water that can pass through a pipe, and so on.

Example. A communications network connects the n centers $x_1, x_2, \ldots, x_n$. The maximum number of messages, c_{ij}, that can be sent from x_i to x_j per minute depends on the number of lines between x_i and x_j. Given the $n \times n$ matrix $C = (c_{ij})$, it is important to to know the maximum amount of information that can be transmitted between a given pair of centers per minute. The appropriate model is a graph with vertices representing centers, edges representing direct communication lines and capacities representing the maximum rates of information transfer. This is an example where directions are not attached to the edges.

Normally there will exist several places where new material can enter the system. These will be modeled by vertices called *sources*. Material leaves at vertices called *sinks*. The usual model has one source and one sink; it will be shown that this involves no loss of generality. We are primarily concerned with the amount of material that flows through the system; the classical problem is to maximize this quantity.

We define a *transportation network* to be a digraph with two distinguished vertices called the *source* and the *sink*, and with a weight c defined on its arcs. $c(x, y)$ is called the *capacity* of the arc xy.

Example. An oil company pumps crude oil from three wells w_1, w_2 and w_3 to a central terminal t. The oil passes through a network of pipelines that has pumping stations at all three wells and also four intermediate stations p_1, p_2, p_3 and p_4. The digraph of this network, along with the capacities of the pipelines (in units of ten thousand barrels) is shown in Figure 13.1. There are three sources, w_1, w_2, w_3, and one sink, t.

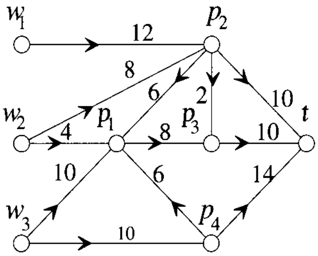


Fig. 13.1. Network for the oil pipeline problem

This example is not a transportation network, because of the multiple sources. The usual way to deal with this case is to add a new (dummy) vertex, say s, together with arcs of infinite capacity from s to every source. The new network has one source, s. Multiple sinks are handled similarly. The resulting single-source, single-sink network is often called the *augmented* or *completed* network.

Sometimes there is a capacity constraint on the vertices of a system. For example, say vertex x can process at most c units of a commodity per day. To embody this constraint, replace x by two vertices x_1 and x_2. All arcs that previously led into x now lead into x_1, all those that previously were directed out of x now leave x_2, and there is an arc from x_1 to x_2 with capacity $c(x_1, x_2) = c$. The most common type of vertex capacity is when the vertex x is a source, and in that case one can simply set $c(s, x) = c$ when defining the new source s.

Example (continued). Suppose the maximum possible amounts that oil wells w_1, w_2 and w_3 can produce are 120,000, 100,000, and 105,000 barrels per day respectively, and p_2 can process at most 150,000 barrels each day. Then an appropriate network is shown in Figure 13.2.

A *flow* f of *value* $v = v(f)$ on a transportation network with source s and sink t is a weight f satisfying

$$0 \leq f(x, y) \leq c(x, y) \text{ for every arc } xy, \tag{13.1}$$

and

$$\sum_{y \in A(s)} f(s, y) - \sum_{z \in B(s)} f(z, s) = v, \tag{13.2}$$

$$\sum_{y \in A(t)} f(t, y) - \sum_{z \in B(t)} f(z, t) = -v, \tag{13.3}$$

$$\sum_{y \in A(x)} f(x, y) - \sum_{z \in B(x)} f(z, x) = 0 \text{ for other vertices } x. \tag{13.4}$$

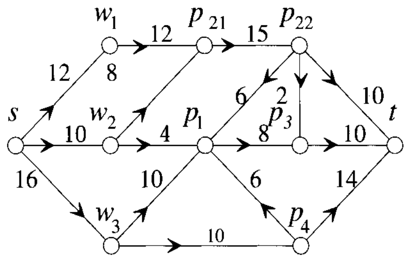


Fig. 13.2. Augmented network for the oil pipeline problem

The flow f is called a *flow from s to t*.

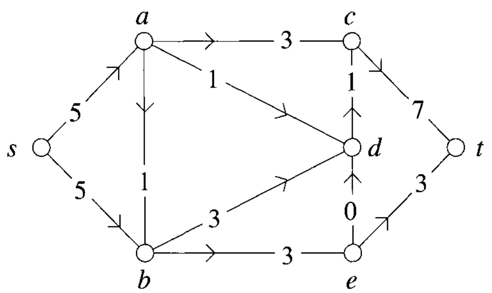
The quantity $\sum_{y \in A(x)} f(x, y) - \sum_{z \in B(x)} f(z, x)$ is called the *net flow at x*. The source and sink of a transportation network are often called *terminal* vertices, and the other vertices are *interior*, so (13.4) says that the net flow at any interior vertex is zero.

Recall that $[A, B]$ denotes the set of all edges of a graph with one endpoint in A and the other in B. The same notation is used in digraphs. If f is a function defined on edges, it is common to write $f[A, B]$ for $\sum_{x \in A} \sum_{y \in B} f(x, y)$. In that notation, the net flow at x is

$$F(x) = f[x, V] - f[V, x],$$

where V is the vertex-set of the network. (One could replace V by $V - x$, but as there is no flow from x to itself, this makes no difference.)

Example. The following diagram shows a real-valued function f on a network with source s and sink t. For convenience we assume that the capacity of each arc is sufficiently large that (13.1) is satisfied.



To verify that f is a flow, it suffices to check the net flow at each vertex:

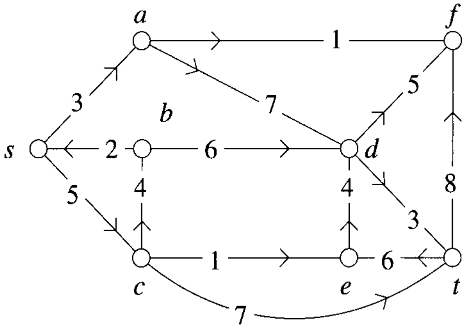
$$\begin{array}{lllllll}
F(s) & = & f(s,a) + f(s,b) & = & 5+5 & = & 10 \\
F(a) & = & f(a,b) + f(a,c) + f(a,d) - f(s,a) & = & 1+3+1-5 & = & 0 \\
F(b) & = & f(b,d) + f(b,e) - f(s,b) - f(a,b) & = & 3 = 3-5-1 & = & 0 \\
F(c) & = & f(c,t) - f(a,c) - f(d,c) & = & 7-3-4 & = & 0 \\
F(d) & = & f(d,c) - f(a,d) - f(b,d) - f(e,d) & = & 4-1-3-0 & = & 0 \\
F(e) & = & f(e,d) - f(e,t) - f(b,e) & = & 0+3-3 & = & 0 \\
F(t) & = & -f(c,t) - f(e,t) & = & -7-3 & = & -10
\end{array}$$

so f is a flow from s to t of value $v(f) = 10$.

Generalization to flows in networks with multiple sources and sinks will be considered in Exercise 13.2.3.

Exercises 13.1

A13.1.1 The network N is shown below. The numbers on each arc give the value of the function g.



(i) Give an example of a directed (s, t)-path.
(ii) Give an example of an (s, t)-path that is not a directed (s, t)-path.
(iii) Consider the subsets $X = \{s, a, d\}$, $Y = \{c, e, f\}$ and $Z = \{c, t\}$. Find:

(a) $[X, f]$; (b) $[X, e]$; (c) $g[X, Y]$;
(d) $g[X, Z]$; (e) $g[X, Y \cap Z]$; (f) $g[X, Y \cup Z]$.

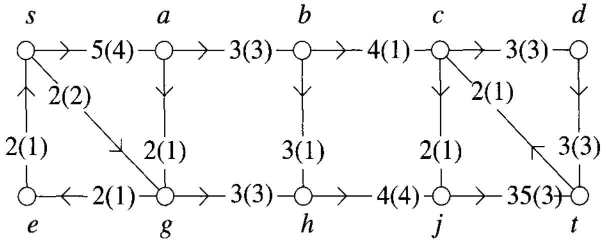
13.1.2 Water is sent from the main dam D to a metropolitan reservoir R through a pipeline network containing five pumping stations P_1, P_2, P_3, P_4 and P_5. The maximum number of gallons that can flow between the various stations per day is given by the table

	D	P_1	P_2	P_3	P_4	P_5	R
D	0	50	40	35	0	0	0
P_1	0	0	20	0	40	0	0
P_2	0	0	0	0	0	33	39
P_3	0	0	0	0	0	30	0
P_4	0	0	0	0	0	0	40
P_5	0	0	0	0	0	0	50
R	0	0	0	0	0	0	0.

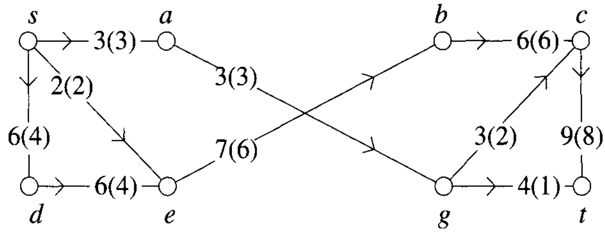
The problem is to maximize the amount of water flowing from the dam to the reservoir per day.
(i) Construct the network.
(ii) Suppose a new reservoir Q is added to the network, and pipelines are constructed joining Q to P_3, P_4 and P_5 with maximum daily flow capacities of 10, 15 and 20 million gallons respectively. How does the network in Part (i) change? What optimization problems arise?
(iii) Suppose P_4 has a maximum capacity of 55 million gallons per day. How can the network in (i) be changed to reflect this?

A13.1.3 In the following networks two numbers are assigned to each arc: the capacity is followed by the value, in brackets, of a function f. In each case, is f a flow from s to t? If f is a flow, what is its value?

(i)



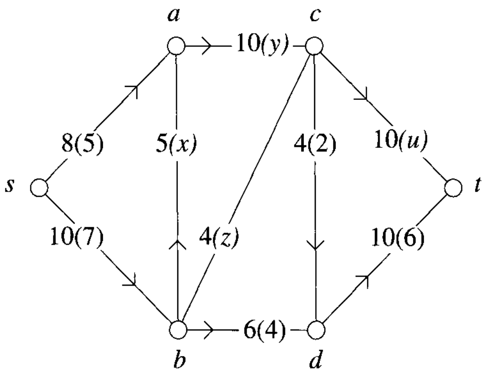
(ii)



13.1.4 Suppose g is a real-valued function defined on the arcs of a network N, and X, Y and Z are subsets of the vertex-set of N. Show that

$$g[X, Y \cup Z] = g[X, Y] + g[X, Z] - g[X, Y \cap Z].$$

13.1.5 In the following network, the two numbers on an arc again represent the capacity and the value, in brackets, of a nonnegative function f.

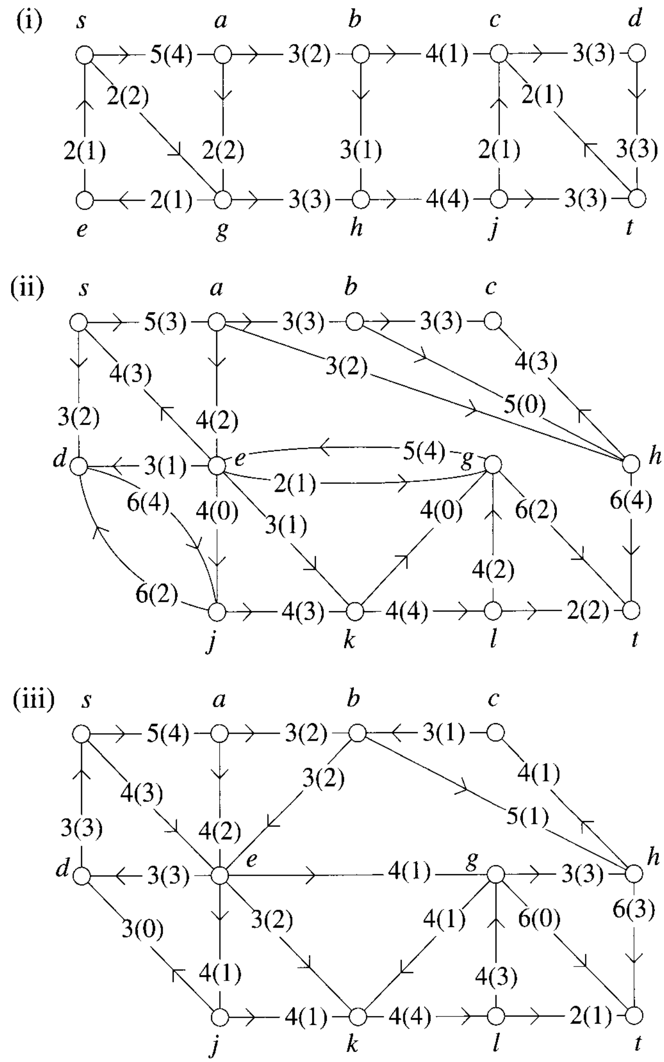


(i) List all directed paths from s to t.

(ii) Suppose $u = 5$. Can f define a flow from s to t?

(iii) Describe the possible values of x, y, z and u such that f is a flow from s to t.

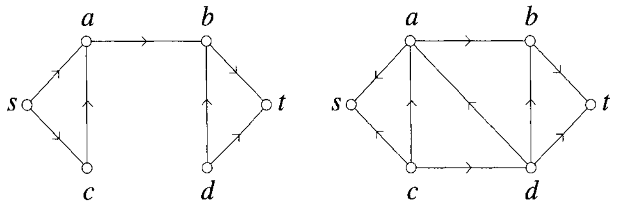
13.1.6 Repeat Exercise 13.1.3 for these networks:



13.2 Maximal Flows

In most flow problems, the main object is to find the maximum value of a flow in a given network, and to find a flow that attains the maximum value. It is moreover desirable to find an efficient algorithm for constructing such a flow.

A restriction on the maximum flow value is illustrated in Figure 13.3. In Figure 13.3(a), if an amount v is input at s and output at t, then the flow in the arc ab must be v also. A similar situation arises in Figure 13.3(b); the total flowing through the network must pass through ab or cd, and the flow through those arcs must also counterbalance any flow in da. In other words,



(a) (b)

Fig. 13.3. Restrictions on the maximum flow.

$$v = f(a, b) + f(c, d) - f(d, a).$$

To generalize these examples, observe that in each case the arcs discussed were those in the set $[S, T]$, where $S = \{s, a, c\}$ and $T = \{b, d, t\}$. The observation was that the value of the flow equaled the net flow from S to T. This applies in general.

If A and B are disjoint sets containing all vertices of a graph or digraph G, the set $[A, B]$ is often called a cut. We define a *separating cut* in a transportation network to be one in which the source and sink are in different parts; conventionally, if $[S, T]$ is a separating cut, then $s \in S$ and $t \in T$ (that is, the two parts of the cut are written in order, with the part containing the source being written first).

Lemma 13.1. *If $[S, T]$ is a separating cut in a transportation network, and f is a flow of value v from s to f, then*

$$v = f[S, T] - f[T, S], \tag{13.5}$$

and

$$v \leq c[S, T]. \tag{13.6}$$

Proof. Denote the set of vertices of the network by V. Since t is not in S, then $f[x, V] - f[V, x] = 0$ for every $x \in S$, except for $f[s, V] - f[V, s] = v$. So

$$\sum_{x \in S} f[x, V] - f[V, x] = v,$$

i.e., $f[S, V] - f[V, S] = v$, and since $V = S \cup T$,

$$f[S, S \cup T] - f[S \cup T, S] = v. \tag{13.7}$$

Since S and T are disjoint, $f[S, S \cup T] = f[S, S] + f[S, T]$, and similarly $f[S \cup T, S] = f[S, S] + f[T, S]$. So (13.7) becomes

$$f[S, T] - f[T, S] = v. \tag{13.8}$$

Now f is nonnegative, so $f[T, S] >\geq 0$. Also $f[S, T] \geq c[S, T]$. So

$$f[S, T] - f[T, S] \geq c[S, T],$$

establishing the lemma. □

It is clear that a finite network contains only a finite number of separating cuts, so there will be a well-defined minimum among the capacities of separating cuts. Any

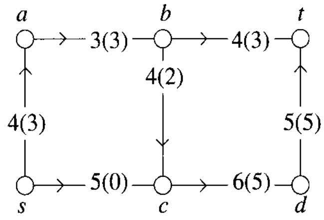


Fig. 13.4. A network with a flow

separating cut realizing this capacity will be called *minimal*. Similarly, if there is a maximum flow value, any flow attaining that value will be called *maximal*.

Example. Figure 13.4 shows a transportation network with a flow f on it. On each arc is shown the capacity, followed by the flow in parentheses. The flow f has value 6. This is not the maximum possible, because the flow g with

$$g(s, c) = 4, g(b, c) = 1, g(b, t) = 2,$$

and $g = f$ elsewhere, has value 7.

The network contains a cut of value 7, namely $[\{s, a\}, \{b, c, d, t\}]$. By Lemma 13.1, no flow can have value greater than 7. So the maximum has been attained, and g is a maximal flow.

It should be observed that g is not unique. Figure 13.5 shows a set of values on the arcs of the network that form a flow of value 7 for any real x, $0 \leq x \leq 1$. So the network has infinitely many maximal flows.

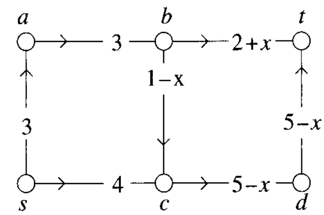


Fig. 13.5. Maximal flows in the network of Figure 13.4

In the preceding example, the flow g was obtained from the flow f by changing the flow in the three arcs sc, cb and bt, and those arcs form a path from s to t. We shall now generalize this example.

Suppose f is a flow from s to t in a transportation network. Consider a path P,

$$P = (x_0, x_1, \dots, x_n),$$

where $s = x_0$ and $t = x_n$. This path is *not* directed, so the edge joining x_i to x_{i+1} might be the arc $x_i x_{i+1}$ or the arc $x_{i+1} x_i$; say the arc is *forward* if its direction is from

x_i to x_{i+1}, and *backward* otherwise. Loosely speaking, forward arcs are those in the direction from s to t along the path.

The path P is called an *augmenting path* for P if it has the following properties:

1. If $x_i x_{i+1}$ is a forward arc, then $f(x_i, x_{i+1}) < c(x_i, x_{i+1})$;
2. If $x_i x_{i+1}$ is a backward arc, then $f(x_i, x_{i+1}) > 0$.

One commonly calls an arc "full" if the flow in it equals its capacity, and "empty" if its flow is zero. So an augmenting path is one that contains no full arcs in the forward direction (the direction from s to t), and no empty backward arcs.

Example (continued). The flow shown in Figure 13.4 has value 6. Since arc ab is operating at capacity, it cannot be in any augmenting path, and neither can dt, but there is one augmenting path, namely $scbt$.

Lemma 13.2. *If a transportation network with a flow f of value v has an augmenting path, then it has a flow whose value is greater than v.*

Proof. Suppose $(x_0, x_2, \dots, x_n)$ is an augmenting path, with $x_0 = s$ and $x_n = t$. If the arc joining x_i to x_{i+1} is $x_i x_{i+1}$, then define

$$\delta_i = c(x_i, x_{i+1}) - f(x_i, x_{i+1});$$

if it is $x_{i+1}x_i$, then

$$\delta_i = c(x_{i+1}, x_i).$$

Finally define $\delta = \min_{1 \le i < n} \delta_i$.

A new flow g is now constructed: if $x_i x_{i+1}$ is a forward arc of the augmenting path, then

$$g(x_i, x_{i+1}) = f(x_, x_{i+1}) + \delta;$$

if $x_i x_{i+1}$ is a backward arc of the augmenting path, then

$$g(x_{i+1}, x_i) = f(x_{i+1}, x_i) - \delta;$$

and in all other cases, when xy does not lie on the augmenting path,

$$g(x, y) = f(x, y).$$

It is clear that

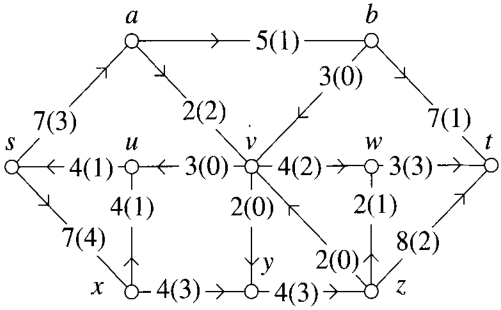
$$0 \le g(x, y) \le c(x, y)$$

for every arc xy. If x is not a vertex on the augmenting path, then the net flow at x is unchanged, so it is still zero. Now consider the effect of the change in flow in the arc $x_i x_{i+1}$. If the arc is directed *out of* x_i, then the flow in the arc is *increased* by δ, and if the direction is *into* x_i, the flow is *decreased* by δ. In either case, the net flow out of x_i is increased by δ, and the net flow out of x_{i+1} is decreased by the same amount. If $i \ne 0$ or n, the net effect on x_i of the flow changes in $x_{i-1}x_i$ and $x_i x_{i+1}$ is zero. The net flow out of the source $x_0 (= s)$ is increased by δ, and the net flow into the sink $x_n (= t)$ is decreased by δ. So g is a flow, of value $v + \delta$. □

Exercises 13.2

13.2.1 Suppose a network has v vertices, a single source and a single sink. How many separating cuts does it have?

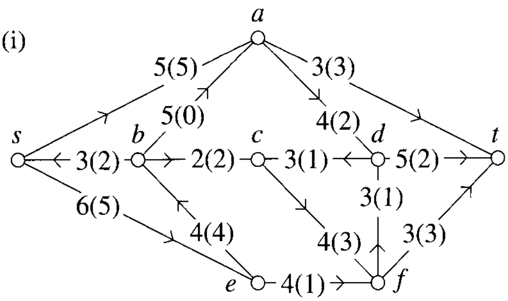
A13.2.2 On the following network, arc capacities shown and values of an arc weight f in parentheses.



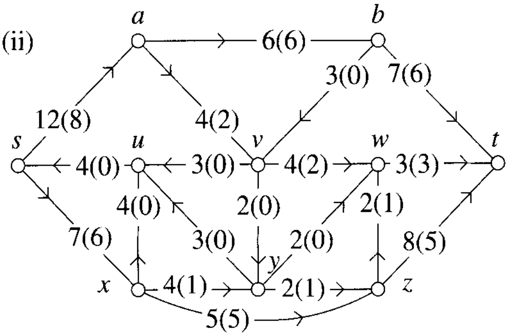
(i) Verify that f is a flow.
(ii) Find the value of the flow.
(iii) Find an augmenting path in the network.
(iv) Find a new flow of greater value than the original.
(v) If the flow is not maximal, find a better flow.
(vi) Find a maximal flow in the network.

13.2.3 Repeat the preceding exercise for the networks shown below.

(i)



(ii)



13.2.4 Consider a network N with a capacity function c whose vertex-set V contains a set S of p sources and a set T of q sinks, where S and T are disjoint; say

$$\begin{aligned} S &= \{s_1, s-2, \ldots, s_p\}, \\ T &= \{t_1, t_{,2}, \ldots, t_q\}. \end{aligned}$$

Define a *flow* g on N to be a function that satisfies (13.1), satisfies (13.4) for every vertex x not in $S \cup T$, and for which

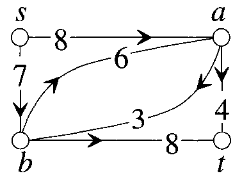
$$\begin{aligned} g[S, V] - g[V, S] &= v, \\ g[T, V] - g[V, T] &= -v \end{aligned}$$

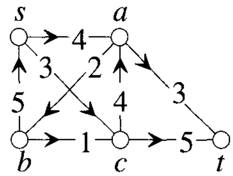
for some nonnegative real number v. In the completed network (with source s and sink t), a flow f is defined from g as follows. If x and y are vertices of the original network, then $f(x, y) = g(x, y)$.

$$\begin{aligned} g(s, s_i) &= f[s_i, V] - f[V, s_i] \text{ for every } i, \\ g(t_j, t) &= f[V, t_j] - f[t_j, V] \text{ for every } j. \end{aligned}$$

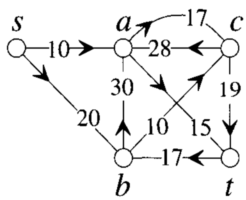
Assuming that $c(s, s_i) \geq c[s_i, V]$ and $c(t_j, t) \geq c[V, t_j]$ for every i and j, prove that f is a flow in the completed network, and that f is maximal if and only if g is maximal in N.

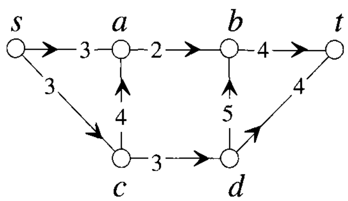
13.2.5 In the following networks, the number on each arc represents the capacity of that arc. In each case find all cuts separating s and t and their capacities. What is the minimum cut capacity of the network? Determine the maximum flow in the network.

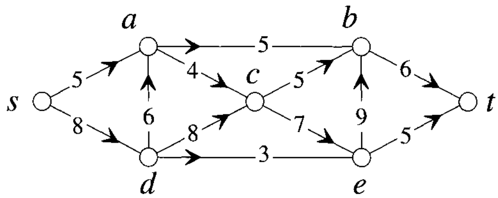
(i) 

(ii) 

13.2.6 Repeat the preceding exercise for these networks:

(i) 

(ii) 

(iii) 

A13.2.7 Suppose vertex x in a transportation network N has the property that no more than d units of material can flow through x per unit time. (The vertex might represent a pump in a sewage system, for example.) How could you model this feature in the network?

13.3 The Max Flow Min Cut Theorem

Lemma 13.1 tells us that the value of any flow in a network must be equal to or less than the capacity of any separating cut. So the maximum flow value is no greater than the minimum capacity of a separating cut. Ford and Fulkerson [45] showed that equality can be attained. We prove this result, using Lemma 13.2 and the following, which is essentially its converse.

Lemma 13.3. *If a transportation network has a flow f of value v from source s to sink t, then either the network contains an augmenting path for f or else it contains a separating cut whose capacity is v.*

Proof. We construct a series $S_0, S_1 \ldots$. of sets of vertices of the network

$$S_0 = \{s\}.$$

When $k > 0$, S_k consists of all vertices y that do not belong to $S_0, S_1, \ldots$ or S_{k-1} such that either there is an arc xy such that $f(x, y) < c(x, y)$, or else there is an arc yx such that $f(y, x) > 0$, for some vertex $x \in S_{k-1}$. In words, y must be a new vertex (not already in any of the S_i) such that either there is an edge into y from S_{k-1} that is not full, or there is an edge from y to S_{k-1} that is not empty.

The number of vertices is finite, so the number of (nonempty) sets constructed is finite. Either $t \in S_n$ for some n or not.

If $t \in S_n$, there must be a vertex $x_{n-1} \in S_{n-1}$ such that either $f(x_{n-1}, t) < c(x_{n-1}, t)$, or $f(t, x_{n-1}) > 0$. Similarly there must be a vertex $x_{n-2} \in S_{n-2}$ such that either $f(x_{n-2}, x_{n-1}) < c(x_{n-2}, x_{n-1})$, or $f(x_{n-1}, x_{n-2}) > 0$. Continuing in this way , s is eventually reached, and the sequence

$$s, x_1, x_2, \ldots, x_{n-1}, t$$

forms an augmenting path for f.

If $t \notin S_n$ for any n, write S for the union of the nonempty S_i, and t for its complement. Then $[S, T]$ is a separating cut. If $x \in S$ and $y \in T$, then any arc xy satisfies $f(x, y) < c(x, y)$, and any arc yx satisfies $f(y, x) > 0$, or else y would be in S. So

$$f[S, T] - f[T, S] = c[S, T] - 0 = c[S, T],$$

and thus

$$c[S, T] = v$$

from (13.5). □

Theorem 13.4. [45] *In any transportation network, the maximum flow value equals the minimum capacity of any separating cut.*

Proof. Suppose v is the maximum flow value, and suppose f is a flow of value v. By Lemma 13.2, f has no augmenting path. So Lemma 13.3 shows that the network contains a separating cut of capacity v. Lemma 13.1 says that no separating cut can have a capacity less than v. So v equals the minimum capacity. □

Since a flow that attains the theoretical maximum value is called a *maximal flow*, and a separating cut that has the minimum capacity is called a *minimal cut*, Theorem 13.4 is usually called the *Max Flow Min Cut Theorem.*

In proving this result, we have essentially used the following characterization of maximal flows:

Theorem 13.5. *A flow is maximal if and only if it has no augmenting path.*

The following characterization of minimal cuts is left as an exercise:

Theorem 13.6. *A separating cut* $[S, T]$ *is minimal if and only if every maximal flow makes every edge of* $[S, T]$ *full and every edge of* $[T, S]$ *empty.*

Exercises 13.3

13.3.1 Prove Theorem 13.6.

13.3.2 Suppose $[S, T]$ and $[X, Y]$ are two minimal cuts in a network N. Prove that both $[S \cup X, T \cap Y]$ and $[S \cap X, T \cup Y]$ are also minimal cuts of N.

13.3.3 Refer to Exercise 13.2.4. A *separating cut* in a network with a set S of sources and a set T of sinks is defined by a partition $\{X, Y\}$ where $S \subseteq X$ and $T \subseteq Y$. State and prove the appropriate version of the Max Flow Min Cut Theorem for such networks.

13.3.4 Consider a transportation network N with vertex-set and arc-set $V(N)$ and $A(N)$. A set D of arcs is called a *blocking set* if every directed path from s to t must contain an arc of D.

(i) Prove that every separating cut in N is a blocking set.

(ii) Given a blocking set D in N, a set S is constructed as follows:

1. $s \in S$;
2. if $x \in S$ and $xy \in A$ but $xy \notin D$, then $y \in S$;
3. every member of S can be found using rules 1 and 2.

Prove that $[S, \overline{S}]$ is a subset of D, and is a cut in N.

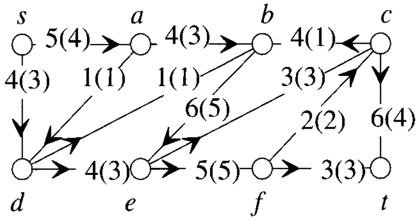
(iii) A blocking set D is called *minimal* if no proper subset D' of D is a blocking set. Prove that every minimal blocking set is a separating cut.

13.4 The Max Flow Min Cut Algorithm

Suppose one wishes to find a maximal flow in a network. One technique is to start with any flow (if necessary, use the trivial case of zero flow in each arc). If the given flow admits of an augmenting path, then by Lemma 13.2 it can be improved. Find such a path and find the improved flow. Then repeat the procedure for the new flow. Continue until no augmenting path exists. Theorem 13.5 says that the resulting flow is maximal.

Example. Consider the network shown in Figure 13.6(a). As usual, the capacity is shown on each arc, followed by a flow f, of value 7, in parentheses. It will be observed that this flow has an augmenting path, s, a, b, c, t, with $\delta = 1$. If the flow is augmented accordingly, adding 1 to $f(s, a)$, $f(a, b)$ and $f(c, t)$ and subtracting 1 from $f(c, b)$, the new flow (shown in Figure 13.6(b)) has value 8. It can be shown that this flow is maximal.

(a)



(b)

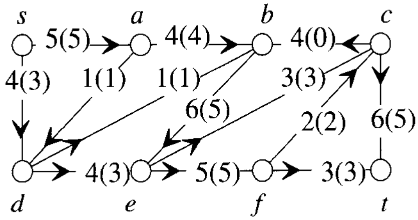


Fig. 13.6. Augmenting a flow

Clearly it is desirable to have an algorithm that will either find an augmenting path or prove that no such path exists. Such an algorithm was constructed by Ford and Fulkerson [46]. Given a transportation network N and a flow f on N, it either produces a flow g on N with value greater than that of f, or proves that f is maximal.

To each vertex u, the algorithm assigns a label of the form (z^+, δ) or (z^-, δ), where δ is a positive real number or ∞. If one of these labels is assigned to a vertex u, this means that we can construct an (undirected) (s, u)-path P in which:

(i) $c(x, y) - f(x, y) \geq \delta$ for every forward arc of P;
(ii) $f(y, x) \geq \delta$ for every backward arc of P.

The z in the label is a vertex adjacent to u; $+$ means that the forward arc zu is the last edge in the path; $-$ means it is the backward arc uz.

The labeling process ends when either the sink t is labeled or no further labels can be assigned. (Labeling of the sink is called *breakthrough*.) If termination occurs because t has been labeled, then f has an augmenting path, as described in (i) and

(ii) above. If t does not receive a label, there is no possible augmenting path and f is maximal.

To simplify the description of labeling, we use the following definition: to *scan* a labeled vertex z means to label every unlabeled vertex y that is adjacent to x and satisfies either $f(x, y) < c(x, y)$ or $f(y, x) > 0$.

Labeling Algorithm:

1. Label the source vertex s with $(-, \infty)$.
2. Select any labeled, unscanned vertex x. Suppose it is labeled $(z^+, \varepsilon(x))$ or $(z^-, \varepsilon(s))$. (In this notation, we could say $\varepsilon(s) = \infty$.) Scan x and assign labels according to the rules:
   - if xy is an arc in which $f(x, y) < c(x, y)$ and y is unlabeled, assign y the label $(x^+, \varepsilon(y))$, where $\varepsilon(y) = \min\{\varepsilon(x), c(x, y) - f(x, y)\}$;
   - if yx is an arc with $f(y, x) > 0$ and y is unlabeled, assign y the label $(x^-, \varepsilon(y))$, where $\varepsilon(y) = \min\{\varepsilon(x), f(y, x)\}$.
3. Repeat Step 2 until either t is labeled (breakthrough), or until no more labels can be assigned and t is unlabeled. In the latter case there is no augmenting path. If breakthrough occurs, then f admits a flow augmenting path, which can be constructed by backtracking from t.

Example. The first diagram in Figure 13.4 shows a network with a flow f of value 8 from s to t. As usual, on each edge the number in parentheses indicates the edge flow and the other number indicates the edge capacity. The labeling algorithm will be used to find an augmenting path.

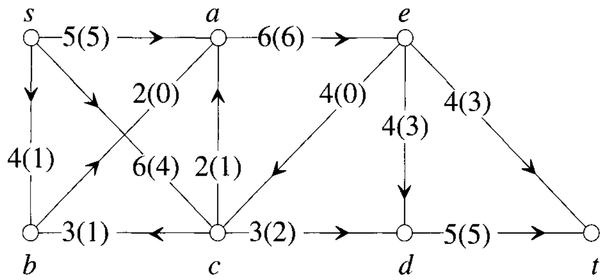
The construction of an augmenting path is illustrated in the remainder of Figure 13.4. The labeling algorithm terminates in breakthrough. Backtracking from t obtains the augmenting path; the arcs of this path are indicated in the final diagram by heavy lines.

When the labeling routine ends in breakthrough, an improved flow is constructed. This is done using another algorithm, the Flow Augmentation Algorithm. This algorithm takes a flow augmenting path P and processes the vertices along the path sequentially. It increases the flow along each forward edge of P by $\varepsilon(t)$ and decreases the flow along each reverse edge of P by $\varepsilon(t)$.

Flow Augmentation Algorithm:

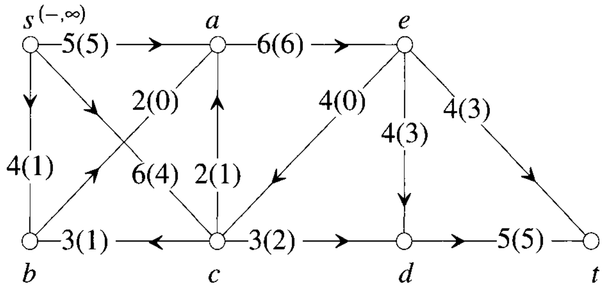
1. First process t:
   - if t is labeled $(y^+, \varepsilon(t))$, define $g(y, t) = f(y, t) + \varepsilon(t)$;
   - if t is labeled $(y^-, \varepsilon(t))$, define $g(y, t) = f(t, y) - \varepsilon(t)$;
   - next, process vertex y.

2. To process vertex u, where $u \neq t$:
   - if u is labeled $(x^+, \varepsilon(u))$, define $g(x, u) = f(x, u) + \varepsilon(t)$;
   - if u is labeled $(x^-, \varepsilon(u))$, define $g(u, x) = f(u, x) - \varepsilon(t)$;
   - next, process vertex x.

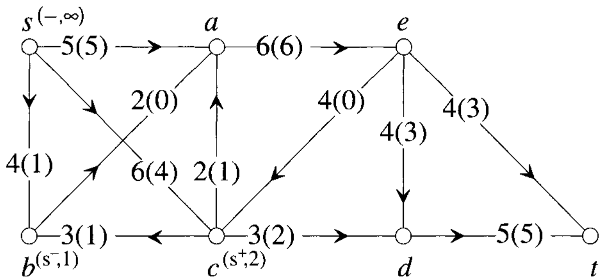


Network for which the maximal flow is to be found

Initial configuration:



After scanning s:



After scanning b:

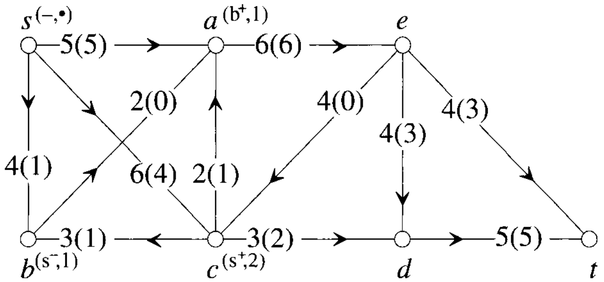
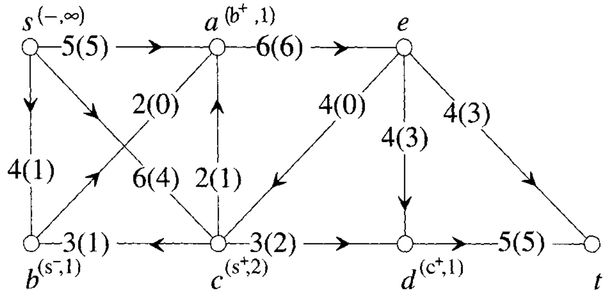
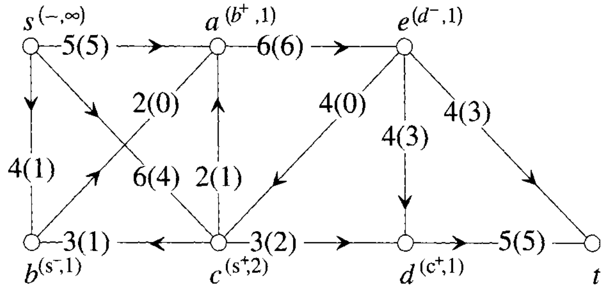


Fig. 13.7. The Labeling Algorithm

After scanning a (no result) and c:



After scanning d:



After scanning e; breakthrough occurs:

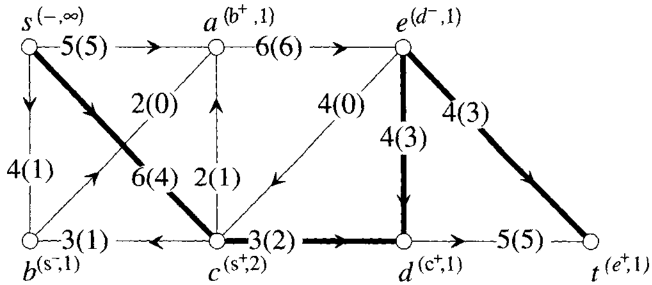


Figure 13.7, continued

3. Repeat Step 2 until the source vertex s is reached. If we take $g(x, y)$ to be equal to $f(x, y)$ for all edges not on the augmenting path, then g defines a flow from s to t of value $v(f) + \varepsilon(t)$.

Example (continued). Applying the flow augmentation algorithm to the flow augmenting path found in our earlier example, we get the revised flow indicated in Figure 13.8.

If the labeling algorithm does not reach breakthrough, then the current flow f is maximal. According to Theorem 13.4 the value v of f is equal to the capacity

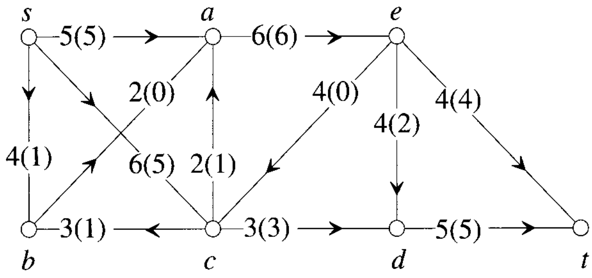


Fig. 13.8. Augmented flow for the example

of the minimal cut separating s and t. A minimal cut (S, T) can be identified from the labeling process: take S as the set of all vertices that receive labels and T as the set of unlabeled vertices. As an example, when the labeling routine is applied to the network of Figure 13.8, it terminates with the labeling shown in Figure 13.9. Define $S = \{s, a, b, c\}$ (the set of labeled vertices) and $T = \{d, e, t\}$. Then

$$[S, T] = \{ae, cd\}$$

is a cut with capacity 9, the value of the current flow. So $[S, T]$ is a minimal cut.

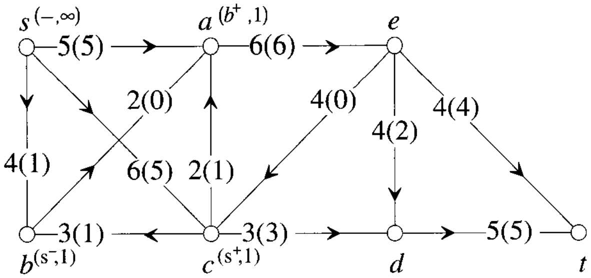


Fig. 13.9. Finding the minimal cut

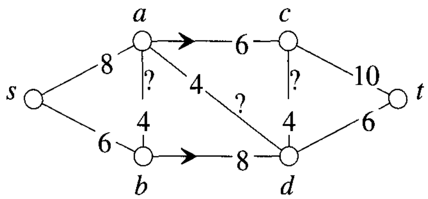
Exercises 13.4

A13.4.1 Prove that the flow in Figure 13.6(b) is maximal.

A13.4.2 In Exercise 13.1.1, if g is interpreted as a capacity function, find a maximal flow in the network.

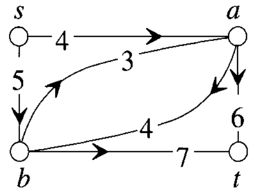
13.4.3 Apply the labeling algorithm to find maximal flows in the networks in Exercises 13.2.2, 13.2.3, 13.2.5 and 13.2.6. (Compare your results with the results for those exercises.)

H13.4.4 In the following street network, the numbers are the traffic flow capacities. The problem is to place one-way signs on the streets not already oriented (the streets marked ?) so as to maximize the traffic flow from s to t. Solve this problem using the labeling algorithm.

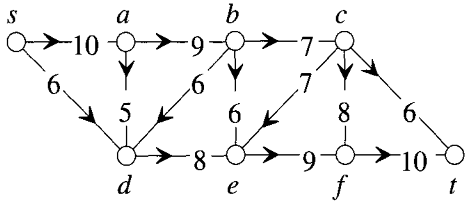


13.4.5 For each network, find a maximal flow from s to t, starting from a zero flow.

(i)

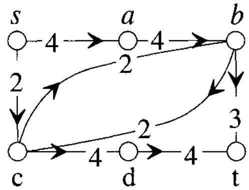


(ii)

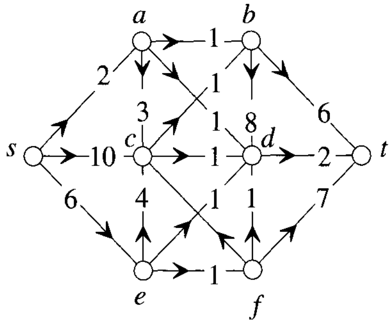


13.4.6 Repeat the preceding exercise for these networks.

(i)



(ii)



13.4.7 If the maximum flow algorithm discovers a flow augmenting path that contains an edge directed back along the path, then some flow will be removed from this backward edge and rerouted.

(i) Is is possible for the maximum flow algorithm never to reroute any flow? If so, what conditions generate such a situation?

(ii) Under what conditions can you be certain that the flow assigned to a specific edge will not be rerouted during a subsequent iteration of the maximum flow algorithm?

13.5 Supply and Demand Problems

Consider the situation where goods are made in several factories and shipped to retailers. Often they will be sent to intermediate depots, such as distributors and warehouses. The number of items that can be shipped per day over part of the route will be restricted and some places cannot be reached directly from others.

Every factory will have a certain maximum amount that it can produce, and every retailer will have a certain minimum amount that it needs. The total of these needs is called the *demand* in the system. The basic problem here is whether or not the factories can produce at such a level as will cause supply to meet demand.

To model this situation, define a *supply-demand network* to be a transportation network with a set X of sources (suppliers) $x_1, x_2, \ldots, x_\alpha$ and a set Y of sinks (retailers) $y_1, y_2, \ldots, y_\beta$; $X \cap Y = \phi$. With every source x_i is associated a positive real number $a(x_i)$, the maximum input at x_i, and with every sink y_j is associated a positive real number $b(y_j)$, the demand at y_j. If $x \notin X$, then $a(x) = 0$ and if $y \notin Y$, then $b(y) = 0$. (For convenience it has been assumed that no factory also acts as a retailer. In practice this is not an important restriction.)

It is said that *supply can meet demand* if the network has a flow whose output is at least equal to the demand at each sink. Clearly, such a flow must meet the following constraints:

$$f[(x, V] - f[V, x] \le a(x) \;\text{ if }\; x \in X; \tag{13.9}$$

$$f[V, x] - f[x, V] \ge b(x) \;\text{ if }\; x \in Y; \tag{13.10}$$

$$f[x, V] - f[V, x] = 0 \;\text{ for other }\; x; \tag{13.11}$$

$$0 \le f(x, y) \le c(x, y). \tag{13.12}$$

Suppose S and T are sets of vertices that partition the vertex-set V. The total demand from the retailers that are members of T is $b(T)$, and the maximum amount that can be produced by the suppliers in T is $a(T)$. The rest of the demand must be supplied from vertices in S. Given a flow f, the net amount that flows from S to T is $f[S, T] - f[T, S]$, and this cannot exceed $c[S, T]$. So, if supply is to meet demand,

$$b(T) - a(T) \le c[S, T]$$

for every such partition S, T. We now prove that this necessary condition is also sufficient.

Theorem 13.7. *If N is a supply-demand network with vertex-set V, then supply can meet demand if and only if*

$$b(T) - a(T) \leq c[S, T] \tag{13.13}$$

for every partition $\{S, T\}$ of V.

Proof. We know the condition is necessary. To prove sufficiency, suppose N satisfies (13.13) for every subset S. We construct a new network N', whose vertices and arcs are the vertices and arcs of N together with two new vertices s and t and arcs sx for all x in S and yt for all y in T. The capacity function c' is defined by

$$\begin{array}{lcll} c'(s, x) & = & a(x) & \text{if } x \in S, \\ c'(y, t) & = & b(y) & \text{if } x \in T, \\ c'(x, y) & = & c(x, y) & \text{if } x, y \in V. \end{array}$$

N' is treated as a network with one source s and one sink t.

Define $Q = \{t\}$, and $P = \{s\} \cup V$. Then $[P, Q]$ is a separating cut in N'. Select any separating cut $[G, H]$ in N' and write S for $G\backslash\{s\}$ and T for $H\backslash\{t\}$ (so that T is the complement of S *as a set of vertices of the original network* N). We evaluate $c'[G, H] - c'[P, Q]$. We use equations (13.1) to (13.4) and various other facts: $c'[s, Q]$ is zero because there is no edge st; $c'[S, Q] = b(S)$, and $c'[P, Q] = b(V)$; $c'[s, T] = a(T)$.

$$\begin{aligned} c'[G, H] - c'[P, Q] &= c'[S \cup \{s\}, T \cup Q] - b(V) \\ &= c'[S, T] + c'[s, T] + c'[S, Q] - b(V) \\ &= c[S, T] + a(T) + b(S) - b(V). \end{aligned} \tag{13.14}$$

Now S and T are disjoint, so $b(S) + b(T) = b(S \cup T) = b(V)$, since all members of T are in $S \cup T$. Therefore $b(S) - b(V)$ equals $-b(T)$, and (13.14) is

$$c[S, T] + a(T) - b(T).$$

This expression is nonnegative by (13.13), so

$$c'[G, H] \leq c'[P, Q].$$

Thus the separating cut $[P, Q]$ is minimal.

Suppose f' is a maximal flow in N'. Then f' must make every edge in $[P, Q]$ full and $f'(x, t) = c'(x, t) = b(x)$. The function f, defined on N by putting $f(x, y) = f'(x, y)$, is a flow that satisfies (13.11) and (13.12). If x is in T then (13.13) yields

$$f'[x, V] = f'(x, t) - f'[V, x] = 0,$$

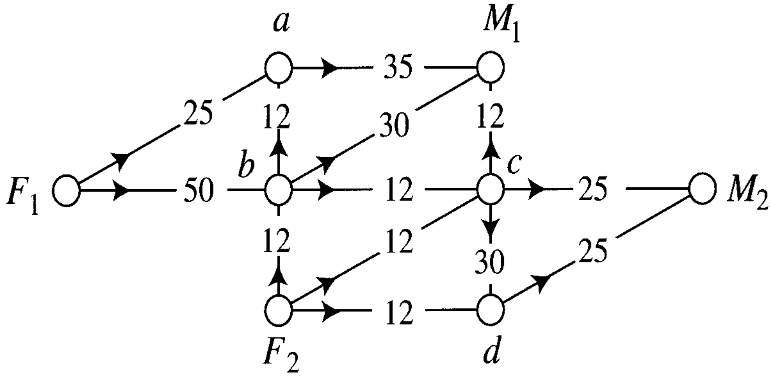
since there is no edge xs, sx or tx; so

$$f[V, x] - f(x, V)] = f'(x, t) = b(x).$$

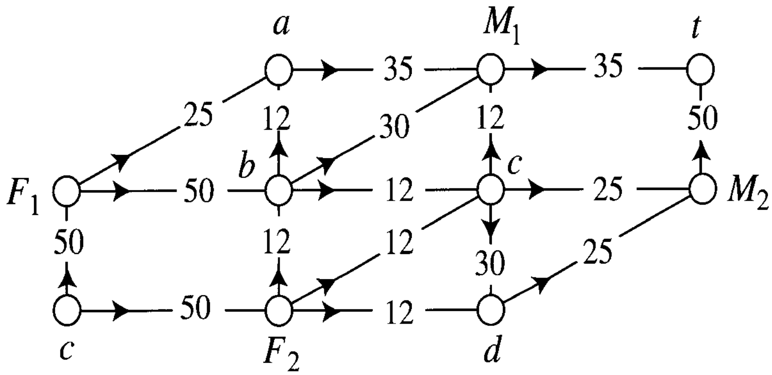
Similarly, one can prove that

$$f[x, V] - f[V, x] = f'(s, x) \leq a(x)$$

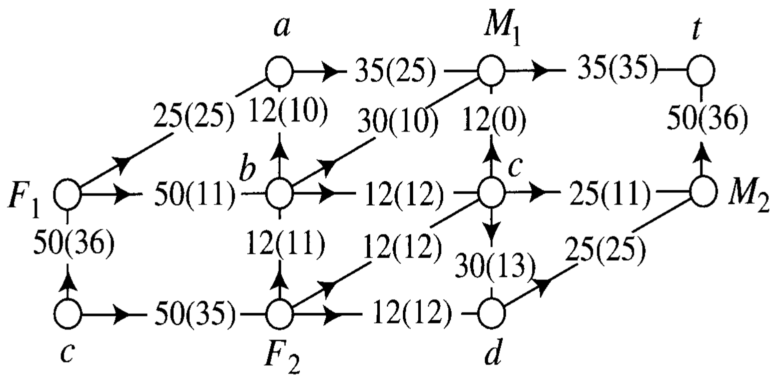
when x lies in S. So supply meets demand under the flow f. □



(a) The distribution network



(b) The transportation network



(c) A maximal flow

Fig. 13.10. Figures for the example

This theorem provides a method to find out whether supply can meet demand in a network. First set up the network N', and then find a maximal flow in it. If the flow fills all the edges into t, then it induces a solution to the supply-demand problem; if not, then the problem is insoluble.

Example. A company has two factories F_1 and F_2 producing a commodity sold at two retail outlets M_1 and M_2. The product is marketed by four distributors a, b, c and d. Each factory can produce 50 items per week. The weekly demand at M_1 and M_2 are 35 and 50 units respectively. The distribution network is given in Figure 13.10(a); the number on an arc indicates its weekly capacity. Can the weekly demands at the retail outlets be met?

The problem amounts to maximizing the flow from s to t in the network of Figure 13.10(b). Applying the algorithm yields the maximum flow f given in Figure 13.10(c). (Verifying this is left as an exercise.)

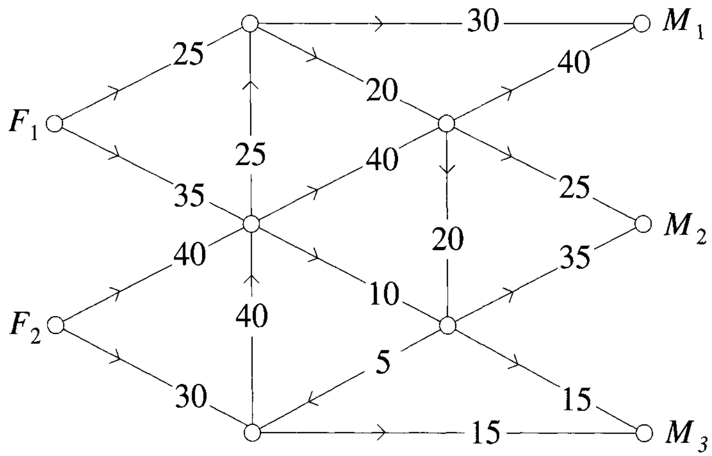
Since $v(f) = 71$ is less than the requirement, the demand cannot be met. The labeled vertices at the conclusion of the algorithm are s, F_1, F_2, a, b and M_1 If $S = \{F_1, F_2, a, b, M_1\}$ and $T = \{c, c, M_2\}$, then $a(T) = 0$, $b(T) = 85$ and $c(S, T) = 36$. Thus (13.13) is violated.

Note that when the problem is not feasible a violation of (13.13) can always be found, as occurred in this example.

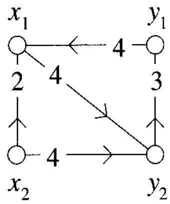
Exercises 13.5

13.5.1 Verify that the flow f given in Figure 13.10(c) is a maximal flow for the network.

13.5.2 Two factories F_1 and F_2 produce a commodity that is required at three markets M_1, M_2 and M_3. The commodity is transported from the factories to the markets through the network shown below. (Capacities are indicated.) Use the maximal flow algorithm to determine the maximum amount of the commodity that can be supplied to the markets from the factories.



13.5.3 A supply-demand network is shown below. It has two factories x_1 and x_2 and two retailers y_1 and y_2. Capacities are shown in thousands of units per week. x_1 can output 6000 units per week, x_2 can output 8000 units per week, y_1 needs 7000 units per week and y_2 needs 4000 units per week. Can supply meet demand?



13.5.4 A manufacturing company has two factories F_1 and F_2 producing a certain commodity that is required at three retail outlets or markets M_1, M_2 and M_3. Once produced, the commodity is stored at one of the five company warehouses W_1, W_2, ..., W_5 from where it is distributed to the various retail outlets. Because of location, it is not feasible to move the commodity from any factory to any warehouse, and from any warehouse to any outlet. Information is given in Figure 13.11; the maximum weekly amount of the commodity that can be moved from F_i to W_j, and from W_j to M_k, are given by the appropriate entries in the matrices, and movement of the commodity is possible through the network shown.
Factory 1 has a weekly production capacity of 60 units and Factory 2 has a weekly production rate of 40 units. Using an appropriate flow algorithm, determine the maximum amount of the commodity that can be supplied to the markets. (Assume that the demand is unlimited.)

	W_1	W_2	W_3	W_4	W_5
F_1	25	35	0	0	0
F_2	0	40	30	0	0

	M_1	M_2	M_3
W_1	30	0	0
W_2	0	0	0
W_3	0	0	15
W_4	40	25	0
W_5	0	35	15

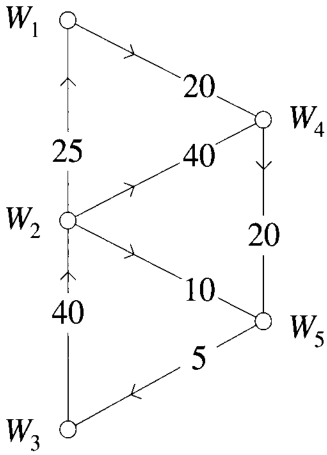


Fig. 13.11. Matrices and network for Exercise 13.5.4