11 Networks

11.1 FLOWS

Transportation networks, the means by which commodities are shipped from their production centres to their markets, can be most effectively analysed when they are viewed as digraphs that possess some additional structure. The resulting theory is the subject of this chapter. It has a wide range of important applications.

A *network* N is a digraph D (the *underlying digraph* of N) with two distinguished subsets of vertices, X and Y, and a non-negative integer-valued function c defined on its arc set A; the sets X and Y are assumed to be disjoint and nonempty. The vertices in X are the *sources* of N and those in Y are the *sinks* of N. They correspond to production centres and markets, respectively. Vertices which are neither sources nor sinks are called *intermediate vertices;* the set of such vertices will be denoted by I. The function c is the *capacity function* of N and its value on an arc a the *capacity* of a. The capacity of an arc can be thought of as representing the maximum rate at which a commodity can be transported along it.

We represent a network by drawing its underlying digraph and labelling each arc with its capacity. Figure 11.1 shows a network with two sources x_1 and x_2, three sinks y_1, y_2 and y_3, and four intermediate vertices v_1, v_2, v_3 and v_4.

If $S \subseteq V$, we denote $V \backslash S$ by $\bar{S}$. In addition, we shall find the following notation useful. If f is a real-valued function defined on the arc set A of N, and if $K \subseteq A$, we denote $\sum_{a \in K} f(a)$ by $f(K)$. Furthermore, if K is a set of arcs of the form $(S, \bar{S})$, we shall write $f^+(S)$ for $f(S, \bar{S})$ and $f^-(S)$ for $f(\bar{S}, S)$.

A *flow* in a network N is an integer-valued function f defined on A such that

$$0 \leq f(a) \leq c(a) \quad \text{for all} \quad a \in A \tag{11.1}$$

and

$$f^-(v) = f^+(v) \quad \text{for all} \quad v \in I \tag{11.2}$$

The value $f(a)$ of f on an arc a can be likened to the rate at which material is transported along a under the flow f. The upper bound in condition (11.1) is called the *capacity constraint;* it imposes the natural restriction that the rate of flow along an arc cannot exceed the capacity of the arc. Condition (11.2), called the *conservation condition,* requires that, for any intermediate vertex v, the rate at which material is transported into v is

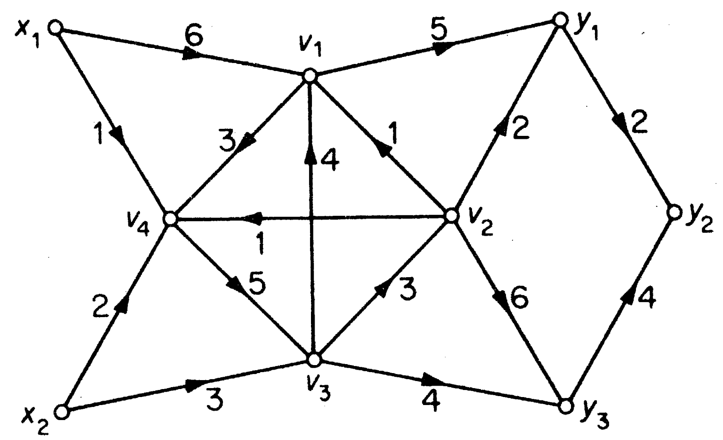


Figure 11.1. A network

equal to the rate at which it is transported out of v. Note that every network has at least one flow, since the function f defined by $f(a)=0$, for all $a \in A$, clearly satisfies both (11.1) and (11.2); it is called the *zero flow*. A less trivial example of a flow is given in figure 11.2. The flow along each arc is indicated in bold type.

If S is a subset of vertices in a network N and f is a flow in N, then $f^+(S)-f^-(S)$ is called the *resultant flow out of* S, and $f^-(S)-f^+(S)$ the *resultant flow into* S, relative to f. Since the conservation condition requires that the resultant flow out of any intermediate vertex is zero, it is intuitively clear and not difficult to show (exercise 11.1.3) that, relative to any flow f, the resultant flow out of X is equal to the resultant flow into Y. This common quantity is called the *value* of f, and is denoted by val f; thus

$$\text{val}\, f = f^+(X) - f^-(X)$$

The value of the flow indicated in figure 11.2 is 6.

A flow f in N is a *maximum flow* if there is no flow f' in N such that $\text{val}\, f' > \text{val}\, f$. Such flows are of obvious importance in the context of transportation networks. The problem of determining a maximum flow in an arbitrary network can be reduced to the case of networks that have just one

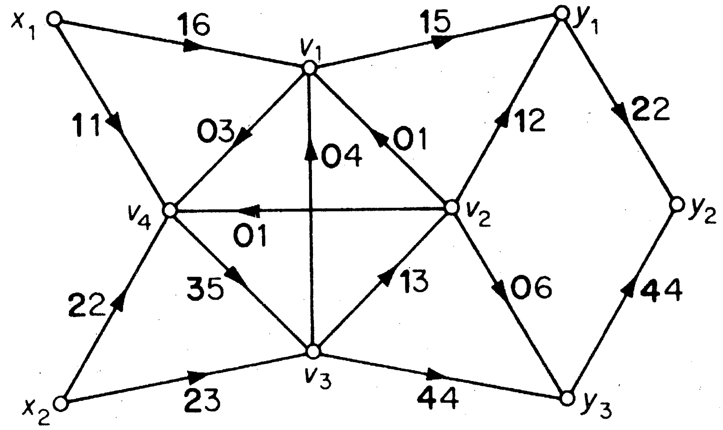


Figure 11.2. A flow in a network

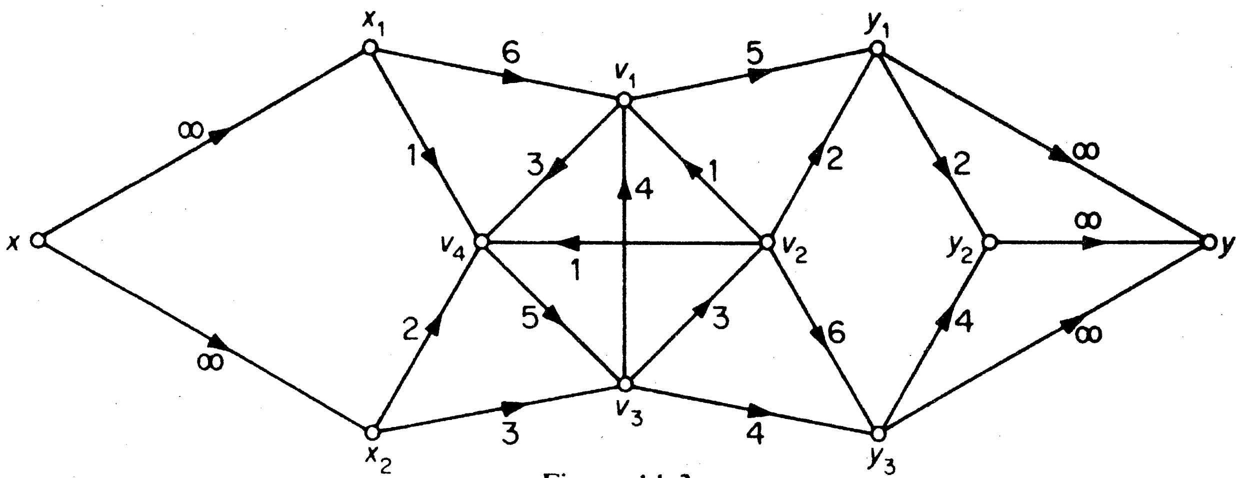


Figure 11.3

source and one sink by means of a simple device. Given a network N, construct a new network N' as follows:

(i) adjoin two new vertices x and y to N;
(ii) join x to each vertex in X by an arc of capacity ∞;
(iii) join each vertex in Y to y by an arc of capacity ∞;
(iv) designate x as the source and y as the sink of N'.

Figure 11.3 illustrates this procedure as applied to the network N of figure 11.1.

Flows in N and N' correspond to one another in a simple way. If f is a flow in N such that the resultant flow out of each source and into each sink is non-negative (it suffices to restrict our attention to such flows) then the function f' defined by

$$f'(a) = \begin{cases} f(a) & \text{if} \quad a \text{ is an arc of } N \\ f^+(v) - f^-(v) & \text{if} \quad a = (x, v) \\ f^-(v) - f^+(v) & \text{if} \quad a = (v, y) \end{cases} \tag{11.3}$$

is a flow in N' such that $\operatorname{val} f' = \operatorname{val} f$ (exercise 11.1.4*a*). Conversely, the restriction to the arc set of N of a flow in N' is a flow in N having the same value (exercise 11.1.4*b*). Therefore, throughout the next three sections, we shall confine our attention to networks that have a single source x and a single sink y.

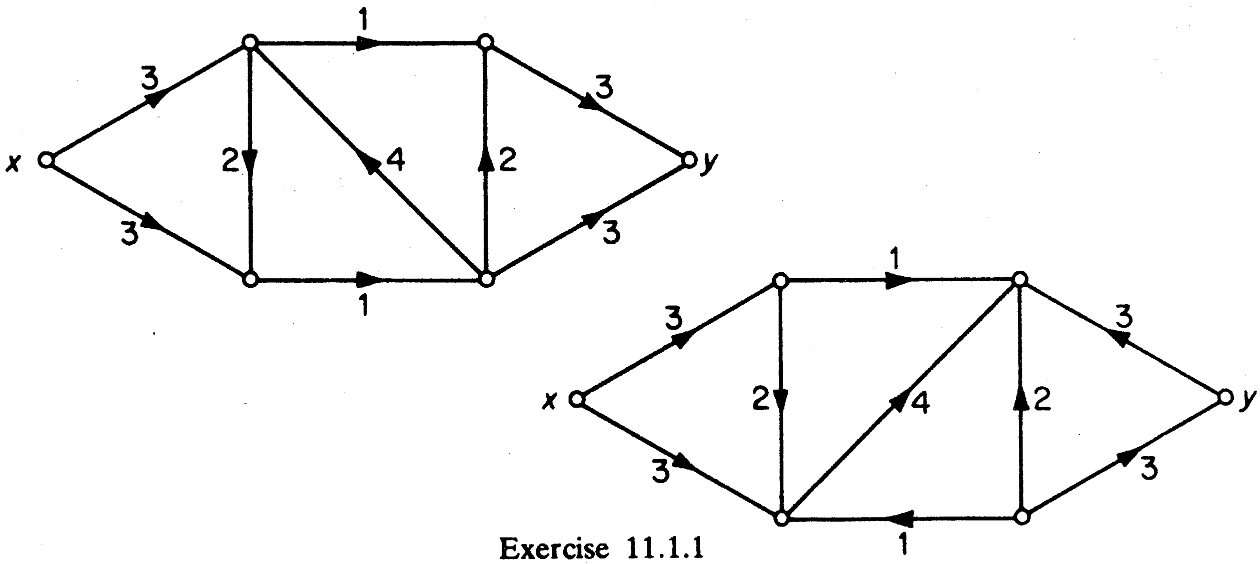
Exercises

11.1.1 For each of the following networks (see diagram, p. 194), determine all possible flows and the value of a maximum flow.

11.1.2 Show that, for any flow f in N and any $S \subseteq V$,

$$\sum_{v \in S} (f^+(v) - f^-(v)) = f^+(S) - f^-(S)$$

(Note that, in general, $\sum_{v \in S} f^+(v) \neq f^+(S)$ and $\sum_{v \in S} f^-(v) \neq f^-(S)$).



Exercise 11.1.1

11.1.3 Show that, relative to any flow f in N, the resultant flow out of X is equal to the resultant flow into Y.

11.1.4 Show that

(*a*) the function f' given by (11.3) is a flow in N' and that $\text{val}\, f' = \text{val}\, f$;

(*b*) the restriction to the arc set of N of a flow in N' is a flow in N having the same value.

11.2 CUTS

Let N be a network with a single source x and a single sink y. A *cut* in N is a set of arcs of the form $(S, \bar{S})$, where $x \in S$ and $y \in \bar{S}$. In the network of figure 11.4, a cut is indicated by heavy lines.

The *capacity* of a cut K is the sum of the capacities of its arcs. We denote the capacity of K by $\text{cap}\, K$; thus

$$\text{cap}\, K = \sum_{a \in K} c(a)$$

The cut indicated in figure 11.4 has capacity 16.

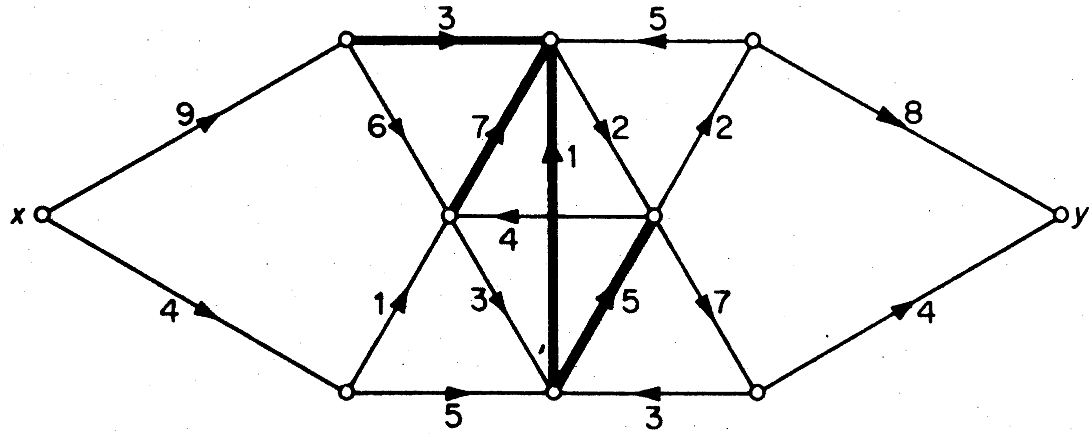


Figure 11.4. A cut in a network

Lemma 11.1 For any flow f and any cut $(S, \bar{S})$ in N

$$\text{val}\, f = f^+(S) - f^-(S) \tag{11.4}$$

Proof Let f be a flow and $(S, \bar{S})$ a cut in N. From the definitions of flow and value of a flow, we have

$$f^+(v) - f^-(v) = \begin{cases} \text{val}\, f & \text{if} \quad v = x \\ 0 & \text{if} \quad v \in S\backslash\{x\} \end{cases}$$

Summing these equations over S and simplifying (exercise 11.1.2), we obtain

$$\text{val}\, f = \sum_{v \in S} (f^+(v) - f^-(v)) = f^+(S) - f^-(S) \quad \square$$

It is convenient to call an arc a *f-zero* if $f(a) = 0$, *f-positive* if $f(a) > 0$, *f-unsaturated* if $f(a) < c(a)$ and *f-saturated* if $f(a) = c(a)$.

Theorem 11.1 For any flow f and any cut $K = (S, \bar{S})$ in N

$$\text{val}\, f \leq \text{cap}\, K \tag{11.5}$$

Furthermore, equality holds in (11.5) if and only if each arc in $(S, \bar{S})$ is f-saturated and each arc in $(\bar{S}, S)$ is f-zero.

Proof By (11.1)

$$f^+(S) \leq \text{cap}\, K \tag{11.6}$$

and

$$f^-(S) \geq 0 \tag{11.7}$$

We obtain (11.5) by substituting inequalities (11.6) and (11.7) in (11.4). The second statement follows, on noting that equality holds in (11.6) if and only if each arc in $(S, \bar{S})$ is f-saturated, and equality holds in (11.7) if and only if each arc in $(\bar{S}, S)$ is f-zero $\square$

A cut K in N is a *minimum cut* if there is no cut K' in N such that $\text{cap}\, K' < \text{cap}\, K$. If f^* is a maximum flow and $\tilde{K}$ is a minimum cut, we have, as a special case of theorem 11.1, that

$$\text{val}\, f^* \leq \text{cap}\, \tilde{K} \tag{11.8}$$

Corollary 11.1 Let f be a flow and K be a cut such that $\text{val}\, f = \text{cap}\, K$. Then f is a maximum flow and K is a minimum cut.

Proof Let f^* be a maximum flow and $\tilde{K}$ a minimum cut. Then, by (11.8),

$$\text{val}\, f \leq \text{val}\, f^* \leq \text{cap}\, \tilde{K} \leq \text{cap}\, K$$

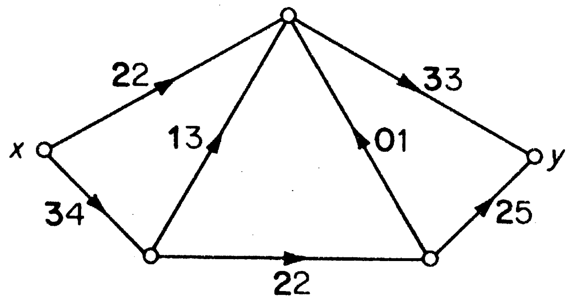
Since, by hypothesis, $\text{val}\, f = \text{cap}\, K$, it follows that $\text{val}\, f = \text{val}\, f^*$ and $\text{cap}\, K = \text{cap}\, \tilde{K}$. Thus f is a maximum flow and K is a minimum cut $\square$

In the next section, we shall prove the converse of corollary 11.1, namely that equality always holds in (11.8).

Exercises

11.2.1 In the following network:

(*a*) determine all cuts;
(*b*) find the capacity of a minimum cut;
(*c*) show that the flow indicated is a maximum flow.



11.2.2 Show that, if there exists no directed (x, y)-path in N, then the value of a maximum flow and the capacity of a minimum cut are both zero.

11.2.3 If $(S, \bar{S})$ and $(T, \bar{T})$ are minimum cuts in N, show that $(S \cup T, \overline{S \cup T})$ and $(S \cap T, \overline{S \cap T})$ are also minimum cuts in N.

11.3 THE MAX-FLOW MIN-CUT THEOREM

In this section we shall present an algorithm for determining a maximum flow in a network. Since a basic requirement of any such algorithm is that it be able to decide when a given flow is, in fact, a maximum flow, we first look at this question.

Let f be a flow in a network N. With each path P in N we associate a non-negative integer $\iota(P)$ defined by

$$\iota(P) = \min_{a \in A(P)} \iota(a)$$

where

$$\iota(a) = \begin{cases} c(a) - f(a) & \text{if } a \text{ is a forward arc of } P \\ f(a) & \text{if } a \text{ is a reverse arc of } P \end{cases}$$

As may easily be seen, $\iota(P)$ is the largest amount by which the flow along P can be increased (relative to f) without violating condition (11.1). The path P is said to be *f-saturated* if $\iota(P) = 0$ and *f-unsaturated* if $\iota(P) > 0$ (or, equivalently, if each forward arc of P is f-unsaturated and each reverse arc of P is f-positive). Put simply, an f-unsaturated path is one that is not being used to its full capacity. An *f-incrementing* path is an f-unsaturated path from the source x to the sink y. For example, if f is the flow indicated in the network of figure 11.5*a*, then one f-incrementing path is the path $P = xv_1v_2v_3y$. The forward arcs of P are (x, v_1) and (v_3, y) and $\iota(P) = 2$.

The existence of an f-incrementing path P in a network is significant since it implies that f is not a maximum flow; in fact, by sending an additional flow of $\iota(P)$ along P, one obtains a new flow $\hat{f}$ defined by

$$\hat{f}(a) = \begin{cases} f(a) + \iota(P) & \text{if } a \text{ is a forward arc of } P \\ f(a) - \iota(P) & \text{if } a \text{ is a reverse arc of } P \\ f(a) & \text{otherwise} \end{cases} \tag{11.9}$$

for which $\operatorname{val}\hat{f} = \operatorname{val} f + \iota(P)$ (exercise 11.3.1). We shall refer to $\hat{f}$ as the *revised flow based on P*. Figure 11.5*b* shows the revised flow in the network of figure 11.5*a*, based on the f-incrementing path $xv_1v_2v_3y$.

The rôle played by incrementing paths in flow theory is analogous to that of augmenting paths in matching theory, as the following theorem shows (compare theorem 5.1).

Theorem 11.2 A flow f in N is a maximum flow if and only if N contains no f-incrementing path.

Proof If N contains an f-incrementing path P, then f cannot be a maximum flow since $\hat{f}$, the revised flow based on P, has a larger value.

Conversely, suppose that N contains no f-incrementing path. Our aim is to show that f is a maximum flow. Let S denote the set of all vertices to which x is connected by f-unsaturated paths in N. Clearly $x \in S$. Also, since N has no f-incrementing path, $y \in \bar{S}$. Thus $K = (S, \bar{S})$ is a cut in N. We shall show that each arc in $(S, \bar{S})$ is f-saturated and each arc in $(\bar{S}, S)$ is f-zero.

Consider an arc a with tail $u \in S$ and head $v \in \bar{S}$. Since $u \in S$, there exists an f-unsaturated (x, u)-path Q. If a were f-unsaturated, then Q could be extended by the arc a to yield an f-unsaturated (x, v)-path. But $v \in \bar{S}$, and so there is no such path. Therefore a must be f-saturated. Similar reasoning shows that if $a \in (\bar{S}, S)$, then a must be f-zero.

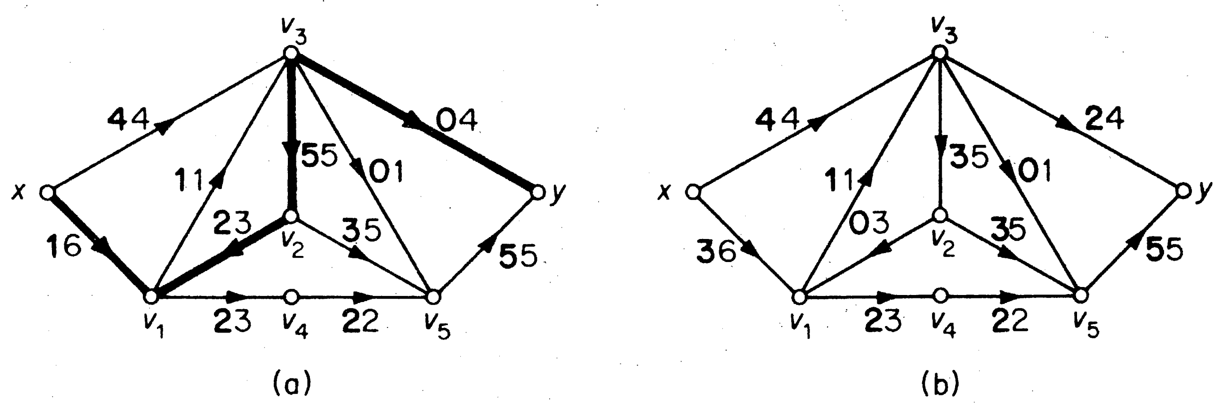


Figure 11.5. (*a*) An f-incrementing path P; (*b*) revised flow based on P

On applying theorem 11.1, we obtain

$$\text{val } f = \text{cap } K$$

It now follows from corollary 11.1 that f is a maximum flow (and that K is a minimum cut) □

In the course of the above proof, we established the existence of a maximum flow f and a minimum cut K such that val f = cap K. We thus have the following theorem, due to Ford and Fulkerson (1956).

Theorem 11.3 In any network, the value of a maximum flow is equal to the capacity of a minimum cut.

Theorem 11.3 is known as the *max-flow min-cut theorem*. It is of central importance in graph theory. Many results on graphs turn out to be easy consequences of this theorem as applied to suitably chosen networks. In sections 11.4 and 11.5 we shall demonstrate two such applications.

The proof of theorem 11.2 is constructive in nature. We extract from it an algorithm for finding a maximum flow in a network. This algorithm, also due to Ford and Fulkerson (1957), is known as the *labelling method*. Starting with a known flow, for instance the zero flow, it recursively constructs a sequence of flows of increasing value, and terminates with a maximum flow. After the construction of each new flow f, a subroutine called the *labelling procedure* is used to find an f-incrementing path, if one exists. If such a path P is found, then $\hat{f}$, the revised flow based on P, is constructed and taken as the next flow in the sequence. If there is no such path, the algorithm terminates; by theorem 11.2, f is a maximum flow.

To describe the labelling procedure we need the following definition. A tree T in N is an *f-unsaturated tree* if (i) $x \in V(T)$, and (ii) for every vertex v of T, the unique (x, v)-path in T is an f-unsaturated path. Such a tree is shown in the network of figure 11.6.

The search for an f-incrementing path involves growing an f-unsaturated tree T in N. Initially, T consists of just the source x. At any stage, there are two ways in which the tree may grow:

1. If there exists an f-unsaturated arc a in $(S, \bar{S})$, where $S = V(T)$, then both a and its head are adjoined to T.

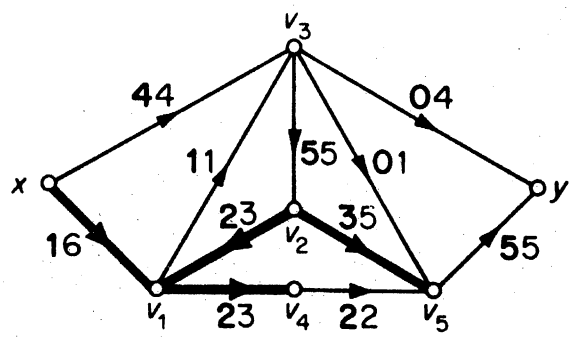


Figure 11.6. An f-unsaturated tree

2. If there exists an f-positive arc a in $(\bar{S}, S)$, then both a and its tail are adjoined to T.

Clearly, each of the above procedures results in an enlarged f-unsaturated tree.

Now either T eventually reaches the sink y or it stops growing before reaching y. The former case is referred to as *breakthrough;* in the event of breakthrough, the (x, y)-path in T is our desired f-incrementing path. If, however, T stops growing before reaching y, we deduce from theorem 11.1 and corollary 11.1 that f is a maximum flow. In figure 11.7, two iterations of this tree-growing procedure are illustrated. The first leads to breakthrough; the second shows that the resulting revised flow is a maximum flow.

The labelling procedure is a systematic way of growing an f-unsaturated tree T. In the process of growing T, it assigns to each vertex v of T the label $l(v) = \iota(P_v)$, where P_v is the unique (x, v)-path in T. The advantage of this labelling is that, in the event of breakthrough, we not only have the f-incrementing path P_y, but also the quantity $\iota(P_y)$ with which to calculate the revised flow based on P_y. The labelling procedure begins by assigning to the source x the label $l(x) = \infty$. It continues according to the following rules:

1. If a is an f-unsaturated arc whose tail u is already labelled but whose head v is not, then v is labelled $l(v) = \min\{l(u), c(a) - f(a)\}$.
2. If a is an f-positive arc whose head u is already labelled but whose tail v is not, then v is labelled $l(v) = \min\{l(u), f(a)\}$.

In each of the above cases, v is said to be labelled *based on u*. To *scan* a labelled vertex u is to label all unlabelled vertices that can be labelled based on u. The labelling procedure is continued until either the sink y is labelled (breakthrough) or all labelled vertices have been scanned and no more vertices can be labelled (implying that f is a maximum flow).

A flow diagram summarising the labelling method is given in figure 11.8.

It is worth pointing out that the labelling method, as described above, is *not* a good algorithm. Consider, for example, the network N in figure 11.9. Clearly, the value of a maximum flow in N is $2m$. The labelling method will use the labelling procedure $2m + 1$ times if it starts with the zero flow and alternates between selecting *xpuvsy* and *xrvuqy* as an incrementing path; for, in each case, the flow value increases by exactly one. Since m is arbitrary, the number of computational steps required to implement the labelling method in this instance can be bounded by no function of ν and ε. In other words, it is not a good algorithm.

However, Edmonds and Karp (1970) have shown that a slight refinement of the labelling procedure turns it into a good algorithm. The refinement suggested by them is the following: in the labelling procedure, scan on a 'first-labelled first-scanned' basis; that is, before scanning a labelled vertex

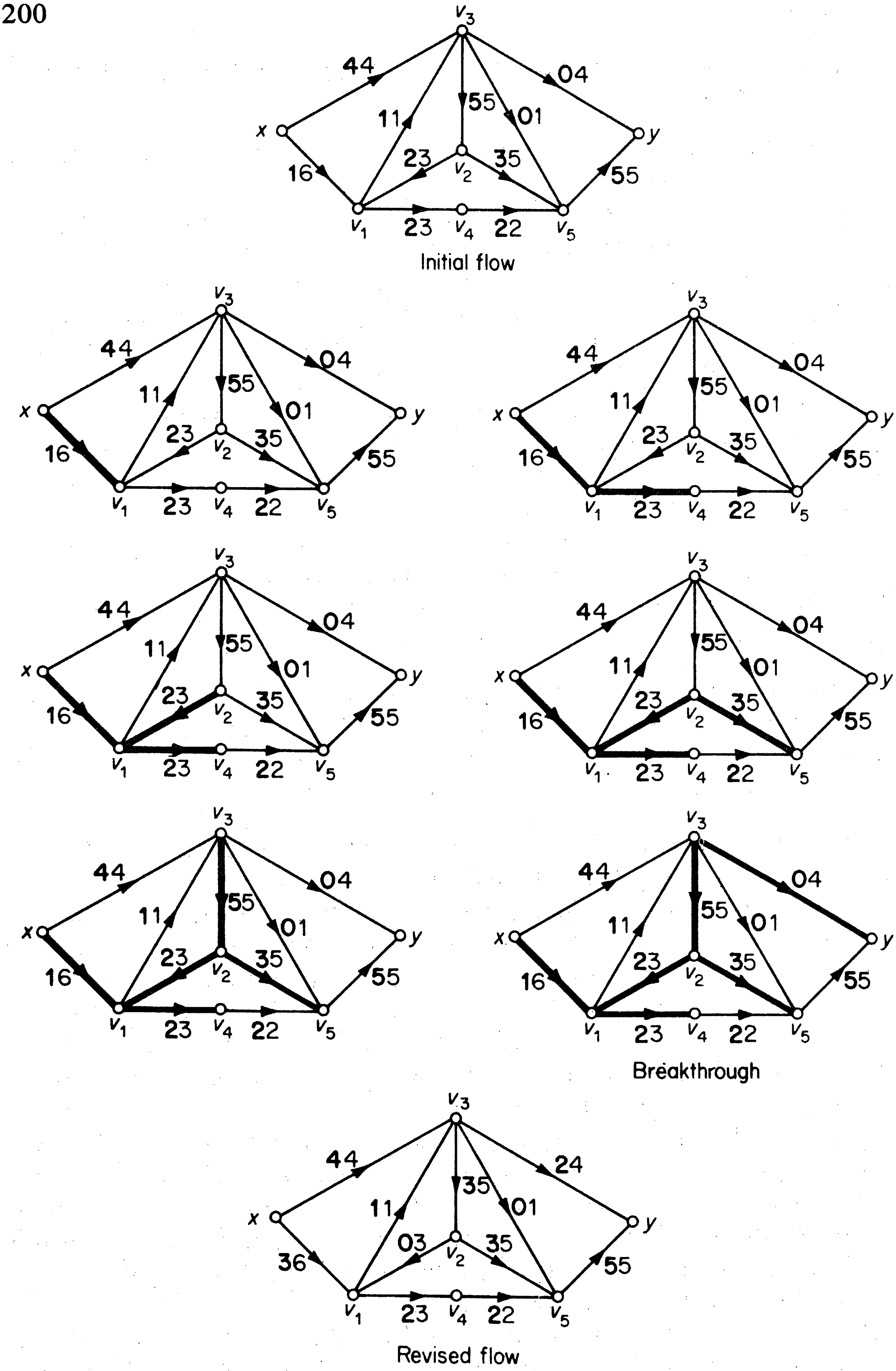


Figure 11.7

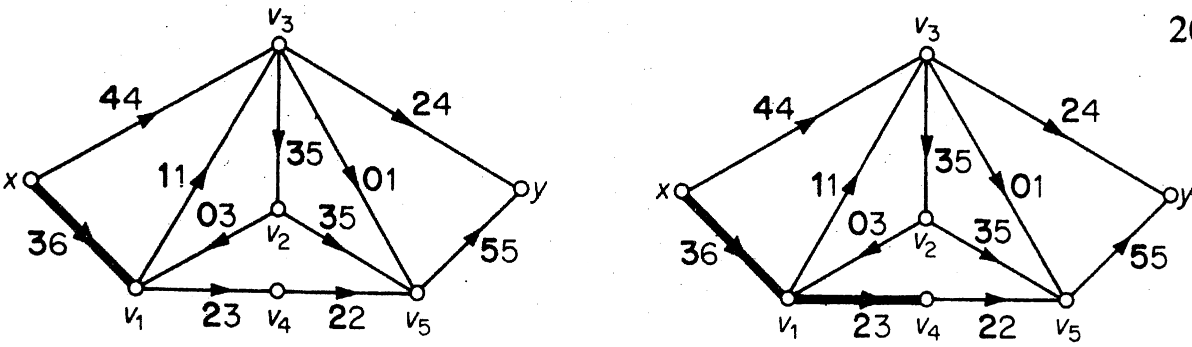


Figure 11.7. (Cont'd)

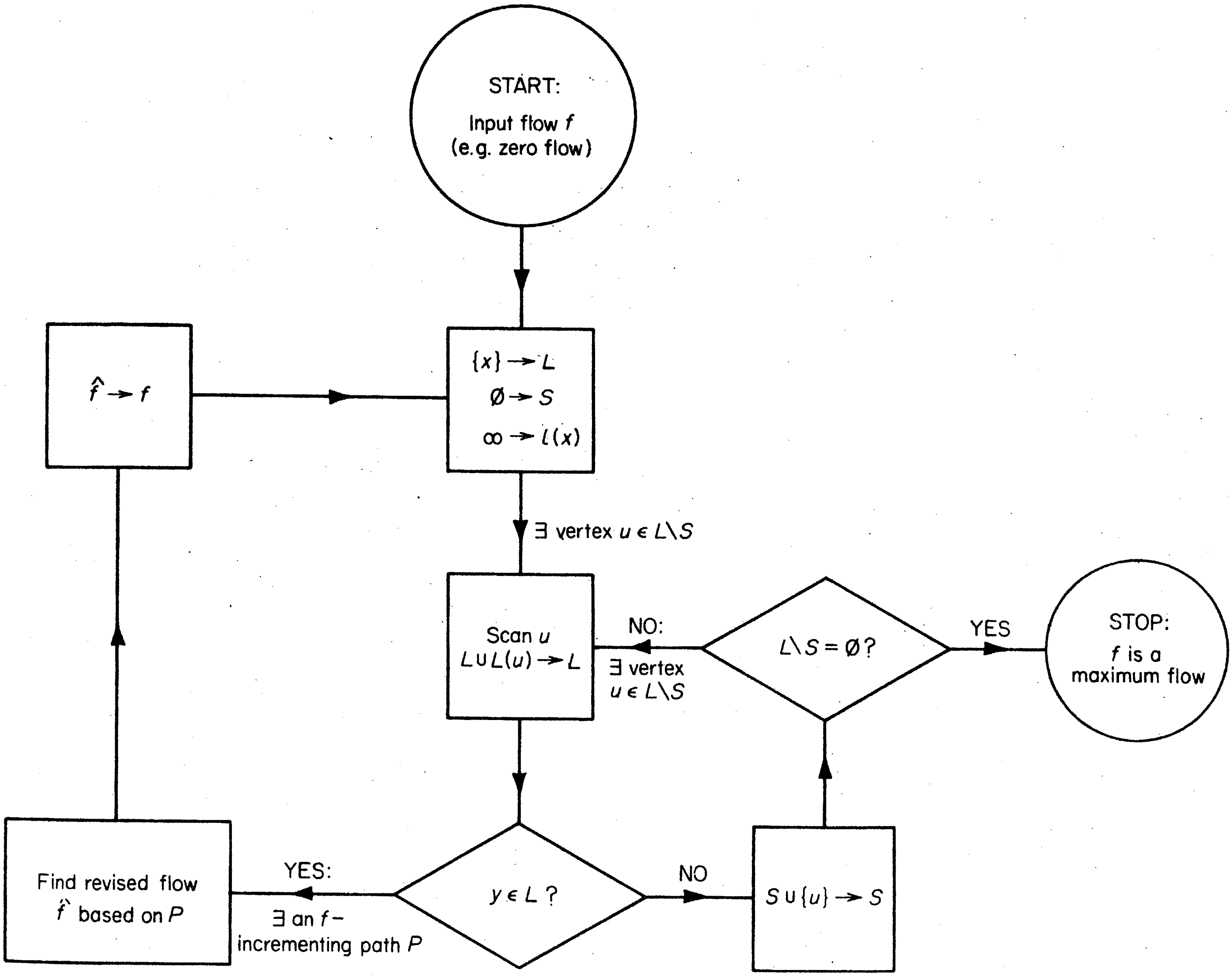


Figure 11.8. The labelling method (L, set of labelled vertices; S, set of scanned vertices; $L(u)$, set of vertices labelled during scanning of u)

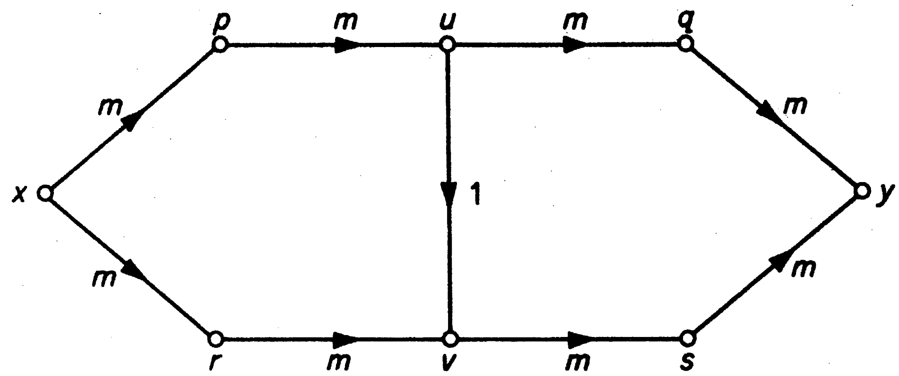


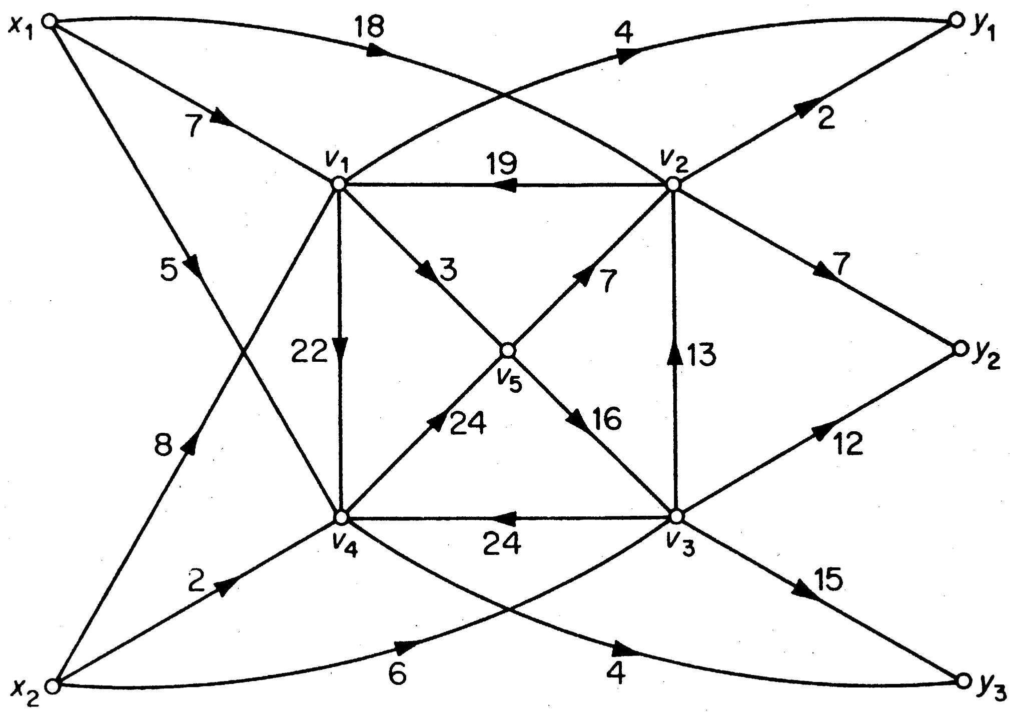
Figure 11.9

u, scan the vertices that were labelled before u. It can be seen that this amounts to selecting a shortest incrementing path. With this refinement, clearly, the maximum flow in the network of figure 11.9 would be found in just two iterations of the labelling procedure.

Exercises

11.3.1 Show that the function $\hat{f}$ given by (11.9) is a flow with $\text{val}\,\hat{f} = \text{val}\,f + \iota(P)$.

11.3.2 A certain commodity is produced at two factories x_1 and x_2. The commodity is to be shipped to markets y_1, y_2 and y_3 through the network shown below. Use the labelling method to determine the maximum amount that can be shipped from the factories to the markets.



11.3.3 Show that, in any network N (with integer capacities), there is a maximum flow f such that $f(a)$ is an integer for all $a \in A$.

11.3.4 Consider a network N such that with each arc a is associated an integer $b(a) \le c(a)$. Modify the labelling method to find a maximum flow f in N subject to the constraint $f(a) \ge b(a)$ for all $a \in A$ (assuming that there is an initial flow satisfying this condition).

11.3.5* Consider a network N such that with each intermediate vertex v is associated a non-negative integer $m(v)$. Show how a maximum flow f satisfying the constraint $f^-(v) \le m(v)$ for all $v \in V \backslash \{x, y\}$ can be found by applying the labelling method to a modified network.

APPLICATIONS

11.4 MENGER'S THEOREMS

In this section, we shall use the max-flow min-cut theorem to obtain a number of theorems due to Menger (1927); two of these have already been mentioned in section 3.2. The following lemma provides a basic link.

Lemma 11.4 Let N be a network with source x and sink y in which each arc has unit capacity. Then

(a) the value of a maximum flow in N is equal to the maximum number m of arc-disjoint directed (x, y)-paths in N; and
(b) the capacity of a minimum cut in N is equal to the minimum number n of arcs whose deletion destroys all directed (x, y)-paths in N.

Proof Let f^* be a maximum flow in N and let D^* denote the digraph obtained from D by deleting all f^*-zero arcs. Since each arc of N has unit capacity, $f^*(a) = 1$ for all $a \in A(D^*)$. It follows that

(i) $d^+_{D^*}(x) - d^-_{D^*}(x) = \text{val} f^* = d^-_{D^*}(y) - d^+_{D^*}(y)$;
(ii) $d^+_{D^*}(v) = d^-_{D^*}(v)$ for all $v \in V \backslash \{x, y\}$.

Therefore (exercise 10.3.3) there exist $\text{val} f^*$ arc-disjoint directed (x, y)-paths in D^*, and hence also in D. Thus

$$\text{val} f^* \le m \tag{11.10}$$

Now let $P_1, P_2, \ldots, P_m$ be any system of m arc-disjoint directed (x, y)-paths in N, and define a function f on A by

$$f(a) = \begin{cases} 1 & \text{if } a \text{ is an arc of } \bigcup_{i=1}^{m} P_i \\ 0 & \text{otherwise} \end{cases}$$

Clearly f is a flow in N with value m. Since f^* is a maximum flow, we have

$$\text{val} f^* \ge m \tag{11.11}$$

It now follows from (11.10) and (11.11) that

$$\text{val}\, f^* = m$$

Let $\tilde{K} = (S, \bar{S})$ be a minimum cut in N. Then, in $N - \tilde{K}$, no vertex of $\bar{S}$ is reachable from any vertex in S; in particular, y is not reachable from x. Thus $\tilde{K}$ is a set of arcs whose deletion destroys all directed (x, y)-paths, and we have

$$\text{cap}\, \tilde{K} = |\tilde{K}| \geq n \tag{11.12}$$

Now let Z be a set of n arcs whose deletion destroys all directed (x, y)-paths, and denote by S the set of all vertices reachable from x in $N - Z$. Since $x \in S$ and $y \in \bar{S}$, $K = (S, \bar{S})$ is a cut in N. Moreover, by the definition of S, $N - Z$ can contain no arc of $(S, \bar{S})$, and so $K \subseteq Z$. Since $\tilde{K}$ is a minimum cut, we conclude that

$$\text{cap}\, \tilde{K} \leq \text{cap}\, K = |K| \leq |Z| = n \tag{11.13}$$

Together, (11.12) and (11.13) now yield

$$\text{cap}\, \tilde{K} = n \quad \square$$

Theorem 11.4 Let x and y be two vertices of a digraph D. Then the maximum number of arc-disjoint directed (x, y)-paths in D is equal to the minimum number of arcs whose deletion destroys all directed (x, y)-paths in D.

Proof We obtain a network N with source x and sink y by assigning unit capacity to each arc of D. The theorem now follows from lemma 11.4 and the max-flow min-cut theorem (11.3) $\square$

A simple trick immediately yields the undirected version of theorem 11.4.

Theorem 11.5 Let x and y be two vertices of a graph G. Then the maximum number of edge-disjoint (x, y)-paths in G is equal to the minimum number of edges whose deletion destroys all (x, y)-paths in G.

Proof Apply theorem 11.4 to $D(G)$, the associated digraph of G (exercise 10.3.6) $\square$

Corollary 11.5 A graph G is k-edge-connected if and only if any two distinct vertices of G are connected by at least k edge-disjoint paths.

Proof This follows directly from theorem 11.5 and the definition of k-edge-connectedness $\square$

We now turn to the vertex versions of the above theorems.

Theorem 11.6 Let x and y be two vertices of a digraph D, such that x is not joined to y. Then the maximum number of internally-disjoint directed (x, y)-paths in D is equal to the minimum number of vertices whose deletion destroys all directed (x, y)-paths in D.

Proof Construct a new digraph D' from D as follows:

(i) split each vertex $v \in V\backslash\{x, y\}$ into two new vertices v' and v'', and join them by an arc (v', v'');
(ii) replace each arc of D with head $v \in V\backslash\{x, y\}$ by a new arc with head v', and each arc of D with tail $v \in V\backslash\{x, y\}$ by a new arc with tail v''. This construction is illustrated in figure 11.10.

Now to each directed (x, y)-path in D' there corresponds a directed (x, y)-path in D obtained by contracting all arcs of type (v', v''); and, conversely, to each directed (x, y)-path in D, there corresponds a directed (x, y)-path in D' obtained by splitting each internal vertex of the path. Furthermore, two directed (x, y)-paths in D' are arc-disjoint if and only if the corresponding paths in D are internally-disjoint. It follows that the maximum number of arc-disjoint directed (x, y)-paths in D' is equal to the maximum number of internally-disjoint directed (x, y)-paths in D. Similarly, the minimum number of arcs in D' whose deletion destroys all directed (x, y)-paths is equal to the minimum number of vertices in D whose deletion destroys all directed (x, y)-paths (exercise 11.4.1). The theorem now follows from theorem 11.4 □

Theorem 11.7 Let x and y be two nonadjacent vertices of a graph G. Then the maximum number of internally-disjoint (x, y)-paths in G is equal to the minimum number of vertices whose deletion destroys all (x, y)-paths.

Proof Apply theorem 11.6 to $D(G)$, the associated digraph of G □

The following corollary is immediate.

Corollary 11.7 A graph G with $\nu \geq k+1$ is k-connected if and only if any two distinct vertices of G are connected by at least k internally-disjoint paths.

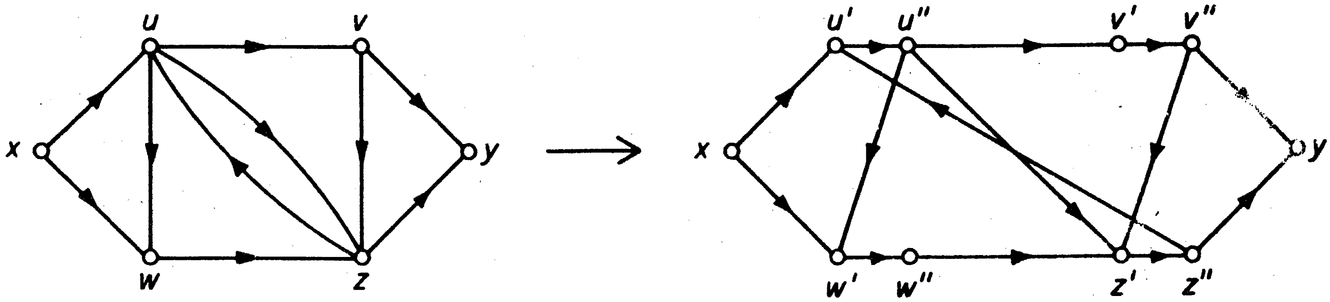


Figure 11.10

Exercises

11.4.1 Show that, in the proof of theorem 11.6, the minimum number of arcs in D' whose deletion destroys all directed (x, y)-paths is equal to the minimum number of vertices in D whose deletion destroys all directed (x, y)-paths.

11.4.2 Derive König's theorem (5.3) from theorem 11.7.

11.4.3 Let G be a graph and let S and T be two disjoint subsets of V. Show that the maximum number of vertex-disjoint paths with one end in S and one end in T is equal to the minimum number of vertices whose deletion separates S from T (that is, after deletion no component contains a vertex of S and a vertex of T).

11.4.4* Show that if G is k-connected with $k \geq 2$, then any k vertices of G are contained together in some cycle. (G. A. Dirac)

11.5 FEASIBLE FLOWS

Let N be a network. Suppose that to each source x_i of N is assigned a non-negative integer $\sigma(x_i)$, called the *supply* at x_i, and to each sink y_j of N is assigned a non-negative integer $\partial(y_j)$, called the *demand* at y_j. A flow f in N is said to be *feasible* if

$$f^+(x_i) - f^-(x_i) \leq \sigma(x_i) \quad \text{for all} \quad x_i \in X$$

and

$$f^-(y_j) - f^+(y_j) \geq \partial(y_j) \quad \text{for all} \quad y_j \in Y$$

In other words, a flow f is feasible if the resultant flow out of each source x_i relative to f does not exceed the supply at x_i, and the resultant flow into each sink y_j relative to f is at least as large as the demand at y_j. A natural question, then, is to ask for necessary and sufficient conditions for the existence of a feasible flow in N. Theorem 11.8, due to Gale (1957), provides an answer to this question. It says that a feasible flow exists if and only if, for every subset S of V, the total capacity of arcs from S to $\bar{S}$ is at least as large as the net demand of $\bar{S}$.

For any subset S of V, we shall denote $\sum_{v \in S} \sigma(v)$ by $\sigma(S)$ and $\sum_{v \in S} \partial(v)$ by $\partial(S)$.

Theorem 11.8 There exists a feasible flow in N if and only if, for all $S \subseteq V$

$$c(S, \bar{S}) \geq \partial(Y \cap \bar{S}) - \sigma(X \cap \bar{S}) \tag{11.14}$$

Proof Construct a new network N' from N as follows:

(i) adjoin two new vertices x and y to N;
(ii) join x to each $x_i \in X$ by an arc of capacity $\sigma(x_i)$;

(iii) join each $y_j \in Y$ to y by an arc of capacity $\partial(y_j)$;
(iv) designate x as the source and y as the sink of N'.

This construction is illustrated in figure 11.11.

It is not difficult to see that N has a feasible flow if and only if N' has a flow that saturates each arc of the cut $(Y, \{y\})$ (exercise 11.5.1). Now a flow in N' that saturates each arc of $(Y, \{y\})$ clearly has value $\partial(Y) = \text{cap}\,(Y, \{y\})$, and is therefore, by corollary 11.1, a maximum flow. It follows that N has a feasible flow if and only if, for each cut $(S \cup \{x\}, \bar{S} \cup \{y\})$ of N'

$$\text{cap}\,(S \cup \{x\}, \bar{S} \cup \{y\}) \geq \partial(Y) \tag{11.15}$$

But conditions (11.14) and (11.15) are precisely the same; for, denoting the capacity function in N' by c', we have

$$\begin{aligned}\text{cap}\,(S \cup \{x\}, \bar{S} \cup \{y\}) &= c'(S, \bar{S}) + c'(S, \{y\}) + c'(\{x\}, \bar{S}) \\ &= c(S, \bar{S}) + \partial(Y \cap S) + \sigma(X \cap \bar{S}) \quad \square\end{aligned}$$

There are many applications of theorem 11.8 to problems in graph theory. We shall discuss one such application.

Let $\mathbf{p} = (p_1, p_2, \ldots, p_m)$ and $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ be two sequences of non-negative integers. We say that the pair $(\mathbf{p}, \mathbf{q})$ is *realisable by a simple bipartite graph* if there exists a simple bipartite graph G with bipartition $(\{x_1, x_2, \ldots, x_m\}, \{y_1, y_2, \ldots, y_n\})$, such that

$$d(x_i) = p_i \quad \text{for} \quad 1 \leq i \leq m$$

and

$$d(y_j) = q_j \quad \text{for} \quad 1 \leq j \leq n$$

For example, the pair $(\mathbf{p}, \mathbf{q})$, where

$$\mathbf{p} = (3, 2, 2, 2, 1) \quad \text{and} \quad \mathbf{q} = (3, 3, 2, 1, 1)$$

is realisable by the bipartite graph of figure 11.12.

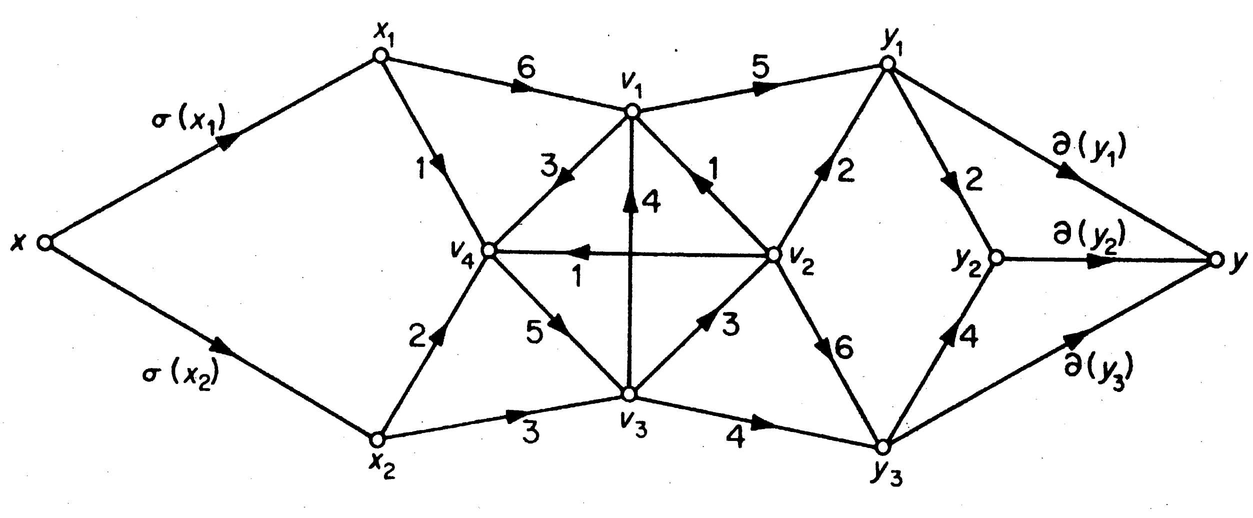


Figure 11.11

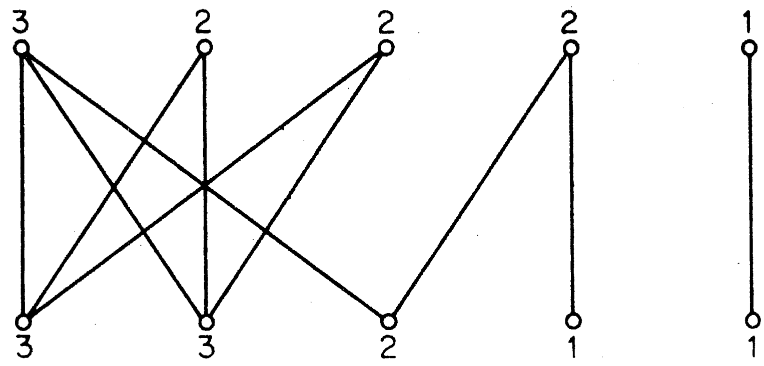


Figure 11.12

An obvious necessary condition for realisability is that

$$\sum_{i=1}^{m} p_i = \sum_{j=1}^{n} q_j \tag{11.16}$$

However, (11.16) is not in itself sufficient. For instance, the pair $(\mathbf{p}, \mathbf{q})$, where

$$\mathbf{p} = (5, 4, 4, 2, 1) \quad \text{and} \quad \mathbf{q} = (5, 4, 4, 2, 1)$$

is not realisable by any simple bipartite graph (exercise 11.5.2). In the following theorem we present necessary and sufficient conditions for the realisability of a pair of sequences by a simple bipartite graph. The order of the terms in the sequences clearly has no bearing on the question of realisability, and we shall find it convenient to assume that the terms of $\mathbf{q}$ are arranged in nonincreasing order

$$q_1 \geq q_2 \geq \ldots \geq q_n \tag{11.17}$$

Theorem 11.9 Let $\mathbf{p} = (p_1, p_2, \ldots, p_m)$ and $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ be two sequences of non-negative integers that satisfy (11.16) and (11.17). Then $(\mathbf{p}, \mathbf{q})$ is realisable by a simple bipartite graph if and only if

$$\sum_{i=1}^{m} \min\{p_i, k\} \geq \sum_{j=1}^{k} q_j \quad \text{for} \quad 1 \leq k \leq n \tag{11.18}$$

Proof Let $X = \{x_1, x_2, \ldots, x_m\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$ be two disjoint sets, and let D be the digraph obtained from the complete bipartite graph with bipartition (X, Y) by orienting each edge from X to Y. We obtain a network N by assigning unit capacity to each arc of D and designating the vertices in X and Y as its sources and sinks, respectively. We shall assume, further, that the supply at source x_i is p_i, $1 \leq i \leq m$, and that the demand at sink y_j is q_j, $1 \leq j \leq n$.

Now, to each spanning subgraph of D, there corresponds a flow in N which saturates precisely the arcs of the subgraph, and this correspondence is clearly one-one. In view of (11.16), it follows that $(\mathbf{p}, \mathbf{q})$ is realisable by a simple bipartite graph if and only if the network N has a feasible flow. We now use theorem 11.8.

For any set S of vertices in N, write

$$I(S)=\{i \mid x_i \in S\} \quad \text{and} \quad J(S)=\{j \mid y_j \in S\}$$

Then, by definition,

$$\left.\begin{array}{c} c(S, \bar{S})=|I(S)|\,|J(\bar{S})| \\ \sigma(X \cap \bar{S})=\sum_{i \in I(\bar{S})} p_i \quad \text{and} \quad \partial(Y \cap \bar{S})=\sum_{j \in J(\bar{S})} q_j \end{array}\right\} \tag{11.19}$$

Suppose that N has a feasible flow. By theorem 11.8 and (11.19)

$$|I(S)|\,|J(\bar{S})| \geq \sum_{j \in J(\bar{S})} q_j - \sum_{i \in I(\bar{S})} p_i$$

for any $S \subseteq X \cup Y$. Setting $S=\{x_i \mid p_i > k\} \cup \{y_j \mid j > k\}$, we have

$$\sum_{i \in I(S)} \min\{p_i, k\} \geq \sum_{j=1}^{k} q_j - \sum_{i \in I(\bar{S})} \min\{p_i, k\}$$

Since this holds for all values of k, (11.18) follows.

Conversely, suppose that (11.18) is satisfied. Let S be any set of vertices in N. By (11.18) and (11.19)

$$c(S, \bar{S}) \geq \sum_{i \in I(S)} \min\{p_i, k\} \geq \sum_{j=1}^{k} q_j - \sum_{i \in I(\bar{S})} \min\{p_i, k\} \geq \partial(Y \cap \bar{S}) - \sigma(X \cap \bar{S})$$

where $k=|J(\bar{S})|$. It follows from theorem 11.8 that N has a feasible flow □

We conclude by looking at theorem 11.9 from the viewpoint of matrices. With each simple bipartite graph G having bipartition $(\{x_1, x_2, \ldots, x_m\}, \{y_1, y_2, \ldots, y_n\})$, we can associate an $m \times n$ matrix $\mathbf{B}$ in which $b_{ij}=1$ or 0, depending on whether $x_i y_j$ is an edge of G or not. Conversely, every $m \times n$ (0, 1)-matrix corresponds in this way to a simple bipartite graph. Thus theorem 11.9 provides necessary and sufficient conditions for the existence of an $m \times n$ (0, 1)-matrix $\mathbf{B}$ with row sums $p_1, p_2, \ldots, p_m$ and column sums $q_1, q_2, \ldots, q_n$.

There is a simple way of visualising condition (11.18) in terms of matrices. Let $\mathbf{B}^*$ denote the (0, 1)-matrix in which the p_i leading terms in each row i are ones, and the remaining entries are zeros, and let $p_1^*, p_2^*, \ldots, p_n^*$ be the column sums of $\mathbf{B}^*$. The sequence $\mathbf{p}^*=(p_1^*, p_2^*, \ldots, p_n^*)$ is called the *conjugate* of $\mathbf{p}$. The conjugate of (5, 4, 4, 2, 1) is (5, 4, 3, 3, 1), for example (see figure 11.13).

Now consider the sum $\sum_{j=1}^{k} p_j^*$. Row i of $\mathbf{B}^*$ contributes $\min\{p_i, k\}$ to this sum. Therefore the left-hand side of (11.18) is equal to $\sum_{j=1}^{k} p_j^*$, and (11.18) is

$$\begin{array}{cc|ccccc|} & & \multicolumn{5}{c}{\mathbf{p}^*} \\ & & 5 & 4 & 3 & 3 & 1 \\ & 5 & 1 & 1 & 1 & 1 & 1 \\ & 4 & 1 & 1 & 1 & 1 & 0 \\ \mathbf{p} & 4 & 1 & 1 & 1 & 1 & 0 \\ & 2 & 1 & 1 & 0 & 0 & 0 \\ & 1 & 1 & 0 & 0 & 0 & 0 \end{array}$$

Figure 11.13

equivalent to the condition

$$\sum_{j=1}^{k} p_j^* \geq \sum_{j=1}^{k} q_j \quad \text{for} \quad 1 \leq k \leq n$$

This formulation of theorem 11.9 in terms of (0, 1)-matrices is due to Ryser (1957). For other applications of the theory of flows in networks, we refer the reader to Ford and Fulkerson (1962).

Exercises

11.5.1 Show that the network N in the proof of theorem 11.8 has a feasible flow if and only if N' has a flow that saturates each arc of the cut $(Y, \{y\})$.

11.5.2 Show that the pair $(\mathbf{p}, \mathbf{q})$, where

$$\mathbf{p} = (5, 4, 4, 2, 1) \quad \text{and} \quad \mathbf{q} = (5, 4, 4, 2, 1)$$

is not realisable by any simple bipartite graph.

11.5.3 Given two sequences, $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ and $\mathbf{q} = (q_1, q_2, \ldots, q_n)$, find necessary and sufficient conditions for the existence of a digraph D on the vertex set $\{v_1, v_2, \ldots, v_n\}$, such that (i) $d^-(v_i) = p_i$ and $d^+(v_i) = q_i$, $1 \leq i \leq n$, and (ii) D has a (0, 1) adjacency matrix.

11.5.4* Let $\mathbf{p} = (p_1, p_2, \ldots, p_m)$ and $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ be two nonincreasing sequences of non-negative integers, and denote the sequences $(p_2, p_3, \ldots, p_m)$ and $(q_1 - 1, q_2 - 1, \ldots, q_{p_1} - 1, q_{p_1+1}, \ldots, q_n)$ by $\mathbf{p}'$ and $\mathbf{q}'$, respectively.

(*a*) Show that $(\mathbf{p}, \mathbf{q})$ is realisable by a simple bipartite graph if and only if the same is true of $(\mathbf{p}', \mathbf{q}')$.

(*b*) Using (*a*), describe an algorithm for constructing a simple bipartite graph which realises $(\mathbf{p}, \mathbf{q})$, if such a realisation exists.

11.5.5 An $(m+n)$-regular graph G is *(m,n)-orientable* if it can be oriented so that each indegree is either m or n.

(a)* Show that G is (m,n)-orientable if and only if there is a partition (V_1, V_2) of V such that, for every $S \subseteq V$,

$$|(m-n)(|V_1 \cap S| - |V_2 \cap S|)| \leq |[S, \bar{S}]|$$

(b) Deduce that if G is (m,n)-orientable and $m > n$, then G is also $(m-1, n+1)$-orientable.

REFERENCES

Edmonds, J. and Karp, R. M. (1972). Theoretical improvements in algorithmic efficiency for network flow problems. *J. Assoc. Comput. Mach.*, **19,** 248–64

Ford, L. R. Jr. and Fulkerson, D. R. (1956). Maximal flow through a network. *Canad. J. Math.*, **8,** 399–404

Ford, L. R. Jr. and Fulkerson, D. R. (1957). A simple algorithm for finding maximal network flows and an application to the Hitchcock problem. *Canad. J. Math.*, **9,** 210–18

Ford, L. R. Jr. and Fulkerson, D. R. (1962). *Flows in Networks*, Princeton University Press, Princeton

Gale, D. (1957). A theorem on flows in networks. *Pacific J. Math.*, **7,** 1073–82

Menger, K. (1927). Zur allgemeinen Kurventheorie. *Fund. Math.*, **10,** 96–115

Ryser, H. J. (1957). Combinatorial properties of matrices of zeros and ones. *Canad. J. Math.*, **9,** 371–77