(d) Let u be a vertex in D different from x and let v be the unique vertex in D such that (v, u) is an arc in A. Let T' be the subtree of T induced by all directed paths in T having u as their initial vertex, i.e., induced by u and all those vertices having u as an ancestor. Using the fact that the edge vu is not a bridge in G, show that there must be an edge in G joining some vertex w in T' to a vertex y not in T'. Prove also that this edge wy is oriented from w to y in D and that y is an ancestor of u.

(e) Prove that if u is a vertex in D different from x then it has a reachable ancestor.

(f) Prove that x is reachable from every other vertex of D and hence that D is strongly connected.

Chapter 8

Networks

8.1 Flows and Cuts

A manufacturer in New Zealand wants to export several boxes of one of his products, clockwork kiwifruit, to a department store in Taiwan. There are various channels through which the boxes can be sent and the digraph of Figure 8.1 represents these, with vertex s as the manufacturer and t the department store. The numbers assigned to each arc represent the maximum loads which each of the corresponding channels can handle.

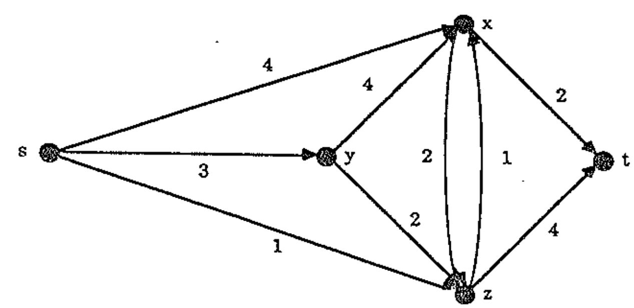


Figure 8.1: A clockwork kiwifruit export network.

The manufacturer wishes to find the maximum number of boxes he can send through the network of channels and "middle men" x, y and z to t so that he never exceeds the permitted capacity of any channel.

Using this example as motivation we now define the concept of a network.

A **network** N is a weakly connected simple digraph in which every arc a of N has been assigned a non-negative integer $c(a)$, called the **capacity** of a.

A vertex s of a network N is called a **source** if it has indegree 0 while a vertex t of N is called a **sink** if it has outdegree 0. Any other vertex of N is called an **intermediate** vertex.

It follows that any arc incident with a source s goes *from* s *to* another vertex and any arc incident with a sink t goes *from* some vertex *to* t.

We will assume from now on that any network N we consider *has exactly one source and exactly one sink*.

Roughly speaking, the source of any network can be thought of as the manufacturer and the sink as the market. The intermediate vertices represent middle men, while the capacity on each arc is the maximum amount of goods that can be sent from the tail of the arc to its head. In Figure 8.1, s, being the only vertex with zero indegree, is the source and similarly t is the sink.

Given any vertex u of the network N we let denote the set of arcs going *into* u and going *out of* u by $I(u)$ and $O(u)$ respectively.

> A **flow** in a network N from the source s to the sink t is a function f which assigns a non-negative integer to each of the arcs a in N such that
> (i) (**capacity constraint**) $f(a) \le c(a)$ for each arc a,
> (ii) the total flow into the sink t equals the total flow out of the source s and
> (iii) (**flow conservation**) for any intermediate vertex x, the total flow into x equals the total flow out of x.

To be more specific, (ii) means that for the source s and the sink t,

$$\sum_{a \in O(s)} f(a) = \sum_{a \in I(t)} f(a),$$

while (iii) means that if x is a vertex different from s and t then

$$\sum_{a \in O(x)} f(a) = \sum_{a \in I(x)} f(a).$$

For example, Figure 8.2 shows a flow for the network of Figure 8.1.

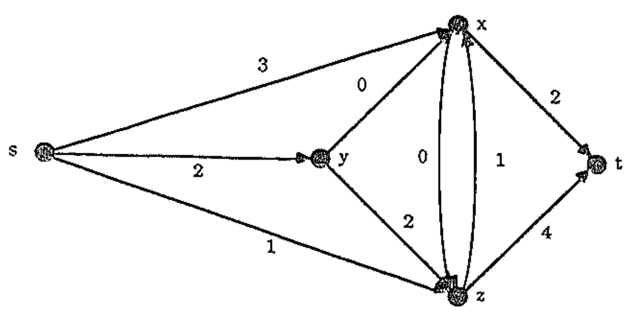


Figure 8.2: A flow for the network of Figure 8.1.

In this example we have

$$\begin{array}{lclclclcl}
\sum_{a \in O(s)} f(a) & = & 3+2+1 & = & 6 & = & 2+4 & = & \sum_{a \in I(t)} f(a), \\
\sum_{a \in O(x)} f(a) & = & 2+1 & = & 3 & = & 3+0 & = & \sum_{a \in I(x)} f(a), \\
\sum_{a \in O(y)} f(a) & = & 0+2 & = & 2 & = & & & \sum_{a \in I(y)} f(a), \\
\sum_{a \in O(z)} f(a) & = & 0+4 & = & 4 & = & 1+2+1 & = & \sum_{a \in I(z)} f(a),
\end{array}$$

and for each arc a of the network, $0 \le f(a) \le c(a)$.

> The number
> $$d = \sum_{a \in O(s)} f(a) = \sum_{a \in I(t)} f(a),$$
> where s and t are the source and sink of the network N, is called the **value** of the flow f.

Thus the flow in our example above has value 6.

Let f be a flow on the network $N = (V, A)$ and for any proper subset X of the vertex set V of N let $\overline{X}$ denote the complement of X in V, i.e., $\overline{X} = V - X$.

If X contains the source s but not the sink t, then intuitively we would expect the *net* flow from the vertices in X to those not in X, i.e., to $\overline{X}$, to equal d, the value of the flow. We illustrate this with the network in Figure 8.3 where the first number assigned to each arc a is its capacity $c(a)$, while the second is $f(a)$, the flow across a.

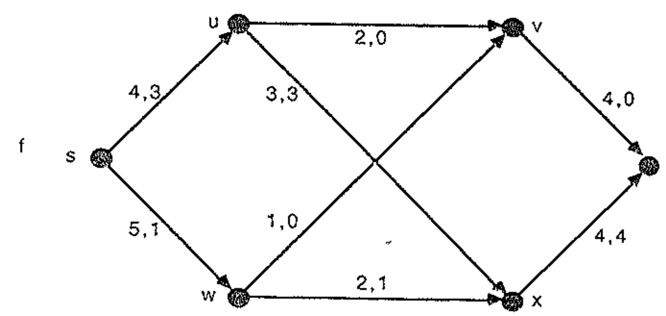


Figure 8.3: A network with a flow.

Here if we take $X = \{s, u, x\}$, so that $\overline{X} = \{v, w, t\}$, then the arcs from vertices in X to vertices in $\overline{X}$ are uv, sw and xt while there is only one arc from vertices in $\overline{X}$ to vertices in X, namely wx. Thus the *net* flow from vertices in X to vertices in $\overline{X}$ is

$$f(uv) + f(sw) + f(xt) - f(wx) = 0 + 1 + 4 - 1 = 4,$$

which is the value of the flow f, as expected. (The value of the flow is given by $f(su) + f(sw) = 3 + 1$, or, alternatively, by $f(vt) + f(xt) = 0 + 4$.)

Also, intuitively we would expect the net flow from any such subset X to $\overline{X}$ not to exceed the capacity of the arcs from X to $\overline{X}$, where by this we mean the sum of the capacities of each arc going from a vertex in X to a vertex in $\overline{X}$. For example for our subset X in the network of Figure 8.3, the capacity of the arcs from X to $\overline{X}$ is

$$c(uv) + c(sw) + c(xt) = 2 + 5 + 4 = 11,$$

which is certainly larger than the value of the flow, namely 4.

The preceding ideas are generalized in our first result below. We first fix some notation.

> If X and Y are any two subsets of vertices of the network N we let $A(X,Y)$ denote the set of arcs from vertices in X to vertices in Y.
>
> If g is any function which assigns non-negative integers to the arcs of the network N (for example, g could be the capacity function c or a flow f), then for any two subsets of vertices X, Y of N we define
>
> $$g(X,Y) = \sum_{a \in A(X,Y)} g(a).$$
>
> In other words, $g(X,Y)$ is the sum of the values of the function g on each arc from a vertex in X to a vertex in Y.

> A **cut** is a set of arcs $A(X,\overline{X})$ where the source s is in X and the sink t is in $\overline{X}$.

Thus, for the subset $X = \{s,u,x\}$ of Figure 8.3, $A(X,\overline{X})$ is a cut and for the capacity function c,

$$c(X,\overline{X}) = \sum_{a \in A(X,\overline{X})} c(a) = c(uv) + c(sw) + c(xt) = 2 + 5 + 4 = 11.$$

Theorem 8.1 *Let f be a flow on a network $N = (V,A)$ and let f have value d. If $A(X,\overline{X})$ is a cut in N then*

$$d = f(X,\overline{X}) - f(\overline{X},X)$$

and

$$d \leq c(X,\overline{X}).$$

In other words, the total flow **out** *of X minus the total flow* **into** *X, i.e., the* **net** *flow out of X, equals d, the value of the flow, and this never exceeds the total capacity of the arcs from X to $\overline{X}$.*

Proof From the definition of flow, for the source s we have

$$f(\{s\},V) = d \text{ and } f(V,\{s\}) = 0,$$

while, for any vertex u different from both s and the sink t,

$$f(\{u\},V) = \sum_{a \in O(u)} f(a) = \sum_{a \in I(u)} f(a) = f(V,\{u\}),$$

i.e., $f(\{u\},V) - f(V,\{u\}) = 0$ for $u \neq s,t$.

Thus, for our cut $A(X,\overline{X})$, we have

$$\sum_{x \in X} \{f(\{x\},V) - f(V,\{x\})\} = f(\{s\},V) - f(V,\{s\}) + 0 = d - 0 + 0 = d,$$

i.e., $f(X,V) - f(V,X) = d$. However

$$f(X,V) = f(X, X \cup (\overline{X})) = f(X,X) + f(X,\overline{X})$$

and similarly

$$f(V,X) = f(X,X) + f(\overline{X},X).$$

Thus

$$d = f(X,V) - f(V,X) = f(X,X) + f(X,\overline{X}) - f(X,X) - f(\overline{X},X) = f(X,\overline{X}) - f(\overline{X},X),$$

i.e., $d = f(X,\overline{X}) - f(\overline{X},X)$, thus establishing the first part of the Theorem.

Moreover, since for each arc a of N we have $f(a) \leq c(a)$ (from the definition of a flow), we get $f(X,\overline{X}) \leq c(X,\overline{X})$ and so

$$d = f(X,\overline{X}) - f(\overline{X},X) \leq f(X,\overline{X}) \leq c(X,\overline{X})$$

giving $d \leq c(X,\overline{X})$, as required. □

The second part of the theorem tells us that the value of *any* flow is less than or equal to the capacity of the arcs from X to $\overline{X}$ for *any* cut $A(X,\overline{X})$. Thus, if f is a flow with value d, we have

$$d \leq \min\{c(X,\overline{X}) : A(X,\overline{X}) \text{ is any cut}\}.$$

We are particularly interested in flows having values equal to the upper bound imposed by the last inequality.

> A flow with value equal to
>
> $$\min\{c(X,\overline{X}) : A(X,\overline{X}) \text{ is any cut}\}$$
>
> is called a **maximal** (or **maximum**) **flow**.

In the network of Figure 8.3, if $A(X,\overline{X})$ is a cut then s must be in X, t in $\overline{X}$ and each of the four intermediate vertices u, v, w and x can be either in X or $\overline{X}$. It follows that there are $2^4 = 16$ possible cuts in this network. (More generally, if there are n intermediate vertices in the network N then N will have 2^n cuts.) If we examine the capacity of each of these cuts in turn it turns out that the cut $A(X,\overline{X})$ where

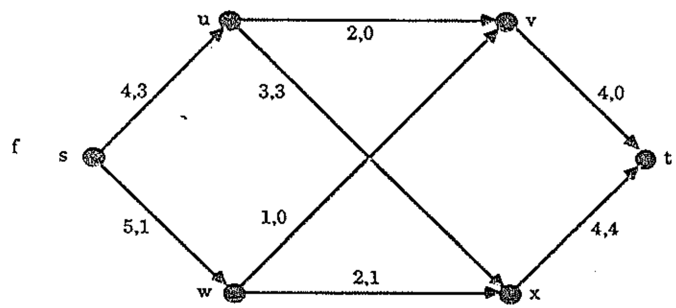


Figure 8.4: This is just Figure 8.3 in a more convenient place.

$X = \{s, u, w, x\}$ (and so $\overline{X} = \{v, t\}$) has capacity $c(uv)+c(wv)+c(xt) = 2+1+4 = 7$ and that this is the smallest capacity of any of the possible cuts. Hence, by the above inequality, any flow on the network must have value at most 7.

We now try to construct a maximal flow on our network. We increase the given f by steps. Figure 8.4 shows f.

For example, the flow on the directed path $s\ u\ v\ t$ is not at a maximum since the flow's value on each of su, uv, vt can be increased by 1 and still remain within the capacity of each arc. This gives a new flow f_1, shown in Figure 8.5.

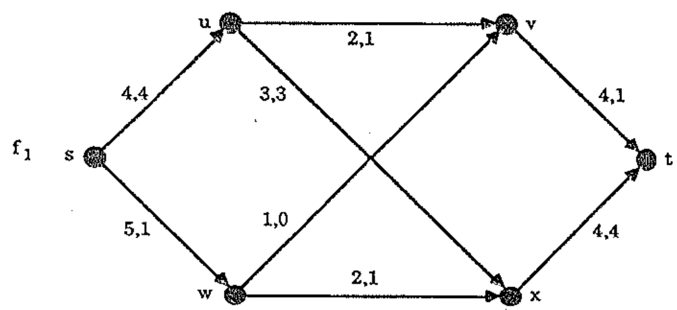


Figure 8.5: Flow f_1 has value 5, an improvement on f's value.

The value of this new flow is $4 + 1 = 5$, one more than the value of f. Likewise f_1 can be increased since the flow on the directed path $s\ w\ v\ t$ is not at a maximum but can be increased by 1 by increasing the flow on each of its arcs sw, wv and vt by 1. This gives a new flow f_2, shown in Figure 8.6, with value $4+2 = 6$.

Now each of the four possible directed paths from s to t, namely $s\ u\ v\ t$, $s\ u\ x\ t$, $s\ w\ x\ t$ and $s\ w\ v\ t$, have an arc whose flow equals the capacity of the arc: $s\ u\ v\ t$ has su, as does $s\ u\ x\ t$, while $s\ w\ x\ t$ has xt and $s\ w\ v\ t$ has wv. Thus we may suspect that f_2 is a flow of maximal value, i.e., no flow can have value more than 6. However we must still investigate the possibilty that the flow can be increased by some other kind of adjustment. In fact, if we increase the flow on sw, wx, uv and vt each by 1 and

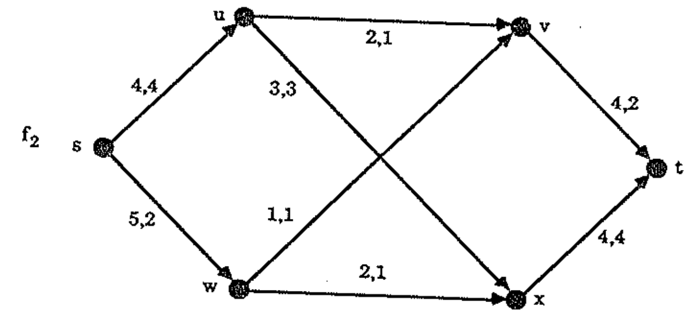


Figure 8.6: Flow f_2 has value 6, an improvement on f_1's value.

decrease the flow on ux by 1 we get a new flow f_3, shown in Figure 8.7, with value $3+4 = 7$. Moreover, since there is a cut with capacity 7, f_3 is a maximal flow.

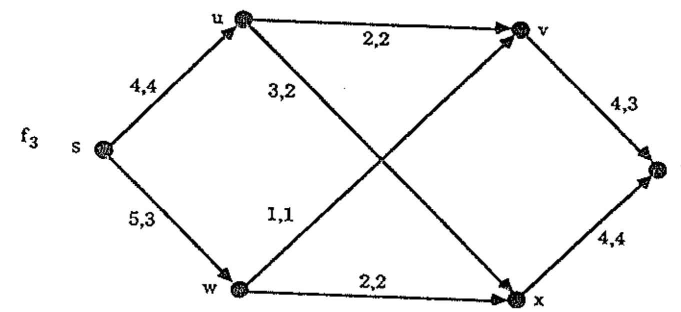


Figure 8.7: Flow f_3 has value 7, an improvement on f_2's value, and is a maximal flow.

We note that each of the adjustments made in the process of obtaining f_3 did in fact give a flow according to our definition: if we increased the flow of an arc *into* an intermediate vertex z then we made a similar increase on an arc *out* of z or a corresponding *decrease* on another arc *into* z, so that flow conservation was maintained, while flow increase out of source s was matched by a flow increase into sink t, so that condition (ii) of the definition was also respected.

The main result of this chapter assures us that for any network there is *always* a maximal flow. Furthermore the proof of the result yields an algorithm which constructs such a maximal flow. Before we state and prove it we introduce some more terminology.

Given any trail $W = v_0v_1 \dots v_k$ in the underlying graph G of a network N, then the associated arcs in N are either of the form $v_{i-1}v_i$ or of the form v_iv_{i-1}. An arc of the first form is called a **forward arc** of W while one of the second form is called a **reverse arc** of W.

If f is a flow in N we associate to the trail W, in the underlying graph G, a non-negative integer $i(W)$, called the **increment** of W, defined by

$$i(W) = \min\{i(a) : a \text{ is an arc associated with } W\},$$

where

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

For example, for the flow f on the network N of Figure 8.4, shown again in Figure 8.8, the walk $W = s\ w\ x\ u\ v$ has forward arcs sw, wx and uv while the remaining associated arc xu is a reverse arc of W. Thus

$$\begin{array}{lllll} i(sw) = c(sw) - f(sw) & = 5 - 1 = 4, & i(wx) = c(wx) - f(wx) & = 2 - 1 = 1, \\ i(ux) = f(ux) & = 3, & i(uv) = c(uv) - f(uv) & = 2 - 0 = 2, \end{array}$$

and so $i(W) = \min\{4, 1, 3, 2\} = 1$.

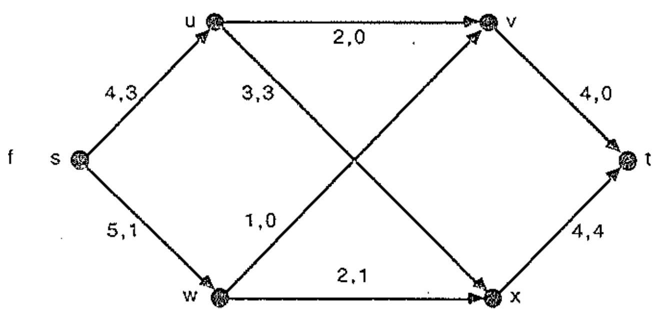


Figure 8.8: This is just Figure 8.4 in a more convenient place.

The amount $i(W)$ is the largest by which the flow f along W can be increased subject to the capacity constraint. For example, for $W = s\ w\ x\ u\ v$ above, we can increase the flow on sw, wx and uv by 1 and make the corresponding decrease by 1 on the arc xu (and this gives the largest increase possible on W). (See Figure 8.9.)

The walk W is said to be ***f*-saturated** if $i(W) = 0$ and ***f*-unsaturated** if $i(W) > 0$.

This simply means that f-unsaturated walks are precisely those that are not being used to their full capacity. Thus, for example, our walk $W = s\ w\ x\ u\ v$ above is f-unsaturated.

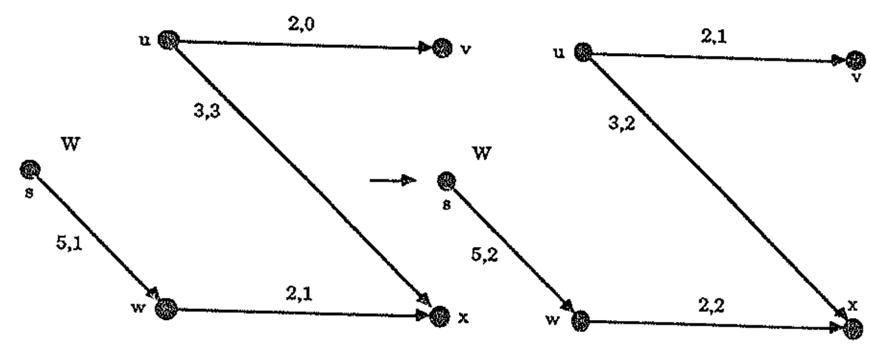


Figure 8.9: The flow is increased along W by $i(W) = 1$.

An ***f*-incrementing** walk is an f-unsaturated walk from the source s to the sink t.

We are now ready for our main result, due to Ford and Fulkerson [23]. We shall interrupt its proof at several stages to illustrate it using our favourite network, that of Figure 8.8 (or is it Figure 8.4?). Each of these interruptions begins and ends with the symbol ■.

Theorem 8.2 (The Max-Flow, Min-Cut Theorem) *Let N be a network with capacity function c. Then there exists a maximum flow in N, i.e., there exists a flow f in N with value*

$$\min\{c(X, \overline{X}) : A(X, \overline{X}) \text{ is a cut}\}.$$

Proof We already know from Theorem 8.1 that for any flow f in N with value d we have

$$d \le \min\{c(X, \overline{X}) : A(X, \overline{X}) \text{ is a cut}\}.$$

Given an arbitrary flow f, we let X be the set of vertices z in N such that either $z = s$ or in the underlying graph G of N there is an f-unsaturated walk $W = v_0 \dots v_k$ from s to z (so that $s = v_0$, $z = v_k$).

■ Thus, for our example N of Figure 8.8, the set X consists of:
s, the source,
u, since, for the walk $W = su$, $i(W) = i(su) = c(su) - f(su) = 4 - 3 > 0$,
w, since, for the walk $W = sw$, $i(W) = i(sw) = c(sw) - f(sw) = 5 - 1 > 0$,
v, since, for the walk $W = suv$,

$$\begin{aligned} i(W) &= \min\{i(su), i(uv)\} = \min\{c(su) - f(su), c(uv) - f(uv)\} \\ &= \min\{4 - 3, 2 - 0\} > 0, \end{aligned}$$

x, since, for the walk $W = swx$,

$$\begin{aligned} i(W) &= \min\{i(sw), i(wx)\} \\ &= \min\{c(sw) - f(sw), c(wx) - f(wx)\} \\ &= \min\{5 - 1, 2 - 1\} > 0, \end{aligned}$$

t, the sink, since for the walk $W = suvt$,

$$\begin{aligned} i(W) &= \min\{i(su), i(uv), i(vt)\} \\ &= \min\{c(su) - f(su), c(uv) - f(uv), c(vt) - f(vt)\} \\ &= \min\{4-3, 2-0, 4-0\} > 0, \end{aligned}$$

i.e., $X = \{s, u, w, v, x, t\}$. ■

Now, returning to the proof, either the sink t is in X or it is in $\overline{X}$. Let us suppose first that t is in X. Then there must be an f-unsaturated walk W from s to t, i.e., an f-incrementing walk. Choose such a walk W and let

$$i(W) = \epsilon,$$

so that $\epsilon > 0$.

■ Thus, in our example, we take W as $suvt$, so that $i(W) = \epsilon = 1$. ■

We now define a new function f_1 on the arcs a of N by

$$f_1(a) = \begin{cases} f(a) + \epsilon & \text{if } a \text{ is a forward arc in } W \\ f(a) - \epsilon & \text{if } a \text{ is a reverse arc in } W \\ f(a) & \text{if } a \text{ is any other arc in } N. \end{cases}$$

Then f_1 is a flow with value $d+\epsilon$. (We leave the proof of this as Exercise 8.1.2.) Since ϵ is, by its definition, a positive integer, we have increased the flow f, with value d, to a new flow f_1 with value $d+\epsilon$.

This new flow f_1 is called the **revised flow based on (the f-incrementing walk)** W.

■ In our example, f_1 is actually the flow f_1 given in Figure 8.5 since

$$\begin{aligned} f_1(su) &= f(su) + \epsilon = 3+1 = 4, & f_1(sw) &= f(sw) = 1, \\ f_1(uv) &= f(uv) + \epsilon = 0+1 = 1, & f_1(ux) &= f(ux) = 3, \\ f_1(vt) &= f(vt) + \epsilon = 0+1 = 1, & f_1(wv) &= f(wv) = 0, \\ f_1(wx) &= f(wx) = 1, & f_1(xt) &= f(xt) = 4, \end{aligned}$$

and the value of this new flow f_1 is $d + \epsilon = 4 + 1 = 5$. ■

This procedure of increasing the flow is always possible provided the sink t is in the set X. Thus we may repeat the process, progressively revising the flow based on incrementing walks until we reach a stage when t is not in the associated set X. At this stage, there is no longer an incrementing walk available to us. Also, since $t \notin X$, $A(X, \overline{X})$ is a cut.

■ In our example, the set X associated with f_1 (see Figure 8.5) consists of:
s, the source,
w, since, for the walk $W = sw$, $i(W) = i(sw) = c(sw) - f_1(sw) = 5 - 1 > 0$,
x, since, for the walk $W = swx$,

$$\begin{aligned} i(W) &= \min\{i(sw), i(wx)\} = \min\{c(sw) - f_1(sw), c(wx) - f_1(wx)\} \\ &= \min\{5-1, 2-1\} > 0, \end{aligned}$$

v, since, for the walk $W = swv$,

$$\begin{aligned} i(W) &= \min\{i(sw), i(wv)\} = \min\{c(sw) - f_1(sw), c(wv) - f_1(wv)\} \\ &= \min\{5-1, 1-0\} > 0, \end{aligned}$$

u, since, for the walk $W = swvu$,

$$\begin{aligned} i(W) &= \min\{i(sw), i(wv), i(vu)\} \\ &= \min\{c(sw) - f_1(sw), c(wv) - f_1(wv), f_1(uv)\} \\ &= \min\{5-1, 1-0, 1\} > 0, \end{aligned}$$

t, since, for the walk $W = swvt$,

$$\begin{aligned} i(W) &= \min\{i(sw), i(wv), i(vt)\} \\ &= \min\{c(sw) - f_1(sw), c(wv) - f_1(wv), c(vt) - f_1(vt)\} \\ &= \min\{5-1, 1-0, 4-1\} > 0, \end{aligned}$$

i.e., $X = V$, as for f.

Taking the walk $W = swvt$ we get $\epsilon = 1$. This is used to define a new function f_2 on the arcs of N by

$$\begin{aligned} f_2(sw) &= f_1(sw) + \epsilon = 1+1 = 2, & f_2(wv) &= f_1(wv) + \epsilon = 0+1 = 1, \\ f_2(vt) &= f_1(vt) + \epsilon = 0+1 = 1, \end{aligned}$$

and, for every other arc a, $f_2(a) = f_1(a)$. In fact, f_2 is just the flow f_2 given in Figure 8.6.

We now find the set X associated with f_2. It consists of (see Figure 8.6) the following vertices:
s, the source,
w, since, for the walk $W = sw$, $i(W) = i(sw) = c(sw) - f_2(sw) = 5 - 2 > 0$,
x, since, for the walk $W = swx$,

$$\begin{aligned} i(W) &= \min\{i(sw), i(wx)\} = \min\{c(sw) - f_2(sw), c(wx) - f_2(wx)\} \\ &= \min\{5-2, 2-1\} > 0, \end{aligned}$$

u, since, for the walk $W = swxu$,

$$\begin{aligned} i(W) &= \min\{i(sw), i(wx), i(xu)\} \\ &= \min\{c(sw) - f_2(sw), c(wx) - f_2(wx), f_2(ux)\} \\ &= \min\{5-2, 2-1, 3\} > 0, \end{aligned}$$

v, since, for the walk $W = swxuv$,

$$\begin{aligned} i(W) &= \min\{i(sw), i(wx), i(xu), i(uv)\} \\ &= \min\{c(sw) - f_2(sw), c(wx) - f_2(wx), f_2(ux), c(uv) - f_2(uv)\} \\ &= \min\{5-2, 2-1, 3, 2-1\} > 0, \end{aligned}$$

t, since for the walk $W = swxuvt$,

$$\begin{aligned} i(W) &= \min\{i(sw), i(wx), i(xu), i(uv), i(vt)\} \\ &= \min\{c(sw) - f_1(sw), c(wx) - f_2(wx), f_2(ux), c(uv) - f_2(uv), c(vt) - f_2(vt)\} \\ &= \min\{5-2, 2-1, 3, 2-1, 4-2\} > 0, \end{aligned}$$

i.e., $X = V$ again.

Then taking the walk $W = swxuvt$ from s to t we get $\epsilon = 1$. This is used to define a new function f_3 on the arcs of N by

$$\begin{aligned} f_3(sw) &= f_2(sw)+\epsilon = 2+1=3, & f_3(wx) &= f_2(wx)+\epsilon = 1+1=2,\\ f_3(ux) &= f_2(ux)-\epsilon = 3-1=2, & f_3(uv) &= f_2(uv)+\epsilon = 1+1=2,\\ f_3(vt) &= f_2(vt)+\epsilon = 2+1=3, \end{aligned}$$

and $f_3(a) = f_2(a)$ for all other arcs a. In fact, f_3 is just the flow f_3 given in Figure 8.7.

We now find the associated set X. It consists of (see Figure 8.7) the vertices:
s, the source,
w, since, for the walk $W = sw$, $i(W) = i(sw) = c(sw) - f_3(sw) = 5 - 3 > 0$,
and no other vertices since all the other arcs incident with s or w go from s or w to another vertex and their current flow value equals their capacity value, i.e.,

$$X = \{s, w\}.$$

Since $t \notin X$, $A(X,\overline{X})$ is a cut. ■

Meanwhile, back at the proof of the Theorem, we have also reached a flow, call it f', which has associated set X with $t \notin X$.

Then $A(X,\overline{X})$ is a cut. Now if the vertex x is in X then, by the definition of X, there is an f'-unsaturated walk $W = v_0 \ldots v_k$ from the source s to x, (so that $s = v_0$ and $v_k = x$). Suppose that y is a vertex not in X, i.e., $y \in \overline{X}$. Then, if there is an arc xy from x to y satisfying $f'(xy) < c(xy)$, the walk $W_1 = v_0 \ldots v_k y$ from s to y would also be f-unsaturated, implying that y is in X, not $\overline{X}$, a contradiction. Similarly, if there is an arc yx from y to x satisfying $f'(yx) > 0$ then the walk $W_2 = v_0 \ldots v_k y$ from s to y would be f-unsaturated, again giving a contradiction. Thus any arc of the form xy where $x \in X$ and $y \in \overline{X}$ must have $f'(xy) = c(xy)$, while any arc of the form yx where $x \in X$ and $y \in \overline{X}$ must have $f'(yx) = 0$. This shows that

$$f'(X,\overline{X}) = c(X,\overline{X}) \text{ while } f(\overline{X},X) = 0.$$

Now, if f' has value d then, since $A(X,\overline{X})$ is a cut, we have, by Theorem 8.1,

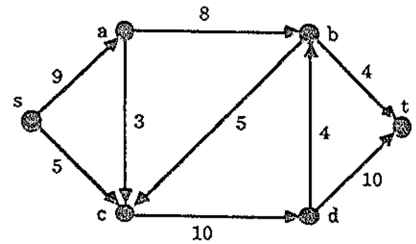
$$d = f'(X,\overline{X}) - f'(\overline{X},X),$$

Thus $d = c(X,\overline{X}) - 0 = c(X,\overline{X})$. In other words, the value of our flow f' equals the capacity of cut $A(X,\overline{X})$. It follows that f' is a maximal flow, completing the proof.

■ In our example, $f' = f_3$ and the set X was found to be $\{s, w\}$. The arcs from X to $\overline{X}$, i.e., $A(X,\overline{X})$, are su, wv, wx and so $c(X,\overline{X}) = c(su)+c(wv)+(wx) = 4+1+2 = 7$, which equals the value of the flow f_3, as expected. ■ □

Exercises for Section 8.1

8.1.1 For the two networks of Figure 8.10, list all the cuts and find a minimum cut.



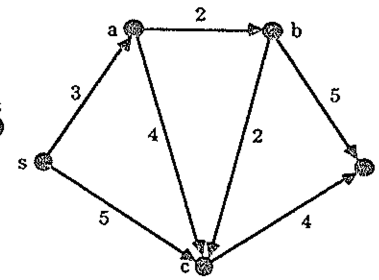


Figure 8.10: Two networks.

8.1.2 Let f be a flow on a network N and let W be an f-incrementing walk. Let f_1 be the revised flow based on W as defined in the proof of Theorem 8.2. Prove that f_1 is indeed a flow on N. (This fills a gap left in the proof of the Theorem.)

8.1.3 In this exercise we consider networks with several sources and several sinks and indicate how to modify these so the the theory of the single source, single sink situation can be applied.

Let N be a network with sources $s_1, \ldots, s_k$ and sinks $t_1, \ldots, t_l$. Then a flow in N from these sources to these sinks is a function f which assigns non-negative real numbers to each of the arcs a in N such that

(a) $f(a) \le c(a)$ for each arc a,

(b) the total flow out of the sources $s_1, \ldots, s_k$ equals the total flow into the sinks $t_1, \ldots, t_l$, and

(c) for any vertex x which is neither a source nor a sink the total flow into x equals the total flow out of x.

The value d of f is defined to be the total flow out of the sources and so by (b),

$$d = \sum_{i=1}^{k} \sum_{a \in O(s_i)} f(a) = \sum_{j=1}^{l} \sum_{a \in I(t_j)} f(a).$$

From N we now create a network N^* with a single source s and a single sink t. We do this by simply introducing s and t as two new vertices, joining an arc a_i from s to each s_i and joining an arc b_j from each t_j to t, these new arcs having infinite capacity (or, in practice, a capacity which is the sum of all the capacities of the arcs already in N).

Now let f^* be a flow in our new network N^*, and let f be the function obtained by restricting f^* to the arcs of N.

(a) Show that f is a flow in N with the same value as f^*.

(b) Show that if f^* is a maximal flow in N^* then f is a maximal flow in N.

8.1.4 Let f be a flow in a network N and let $A(X,\overline{X})$ be a cut in N.

(a) Prove that, if

$$\begin{aligned} f(a) &= c(a) \quad \text{for every } a \in A(X,\overline{X}), \text{ and} \\ f(a) &= 0 \quad \text{for every } a \in A(\overline{X},X), \end{aligned}$$

then f is a maximum flow and $A(X,\overline{X})$ is a minimum cut.

(b) Conversely, prove that if f is a maximum flow in N and $A(X,\overline{X})$ is a minimum cut then

$$\begin{aligned} f(a) &= c(a) \quad \text{for every } a \in A(X,\overline{X}), \text{ and} \\ f(a) &= 0 \quad \text{for every } a \in A(\overline{X},X). \end{aligned}$$

8.1.5 Let f_1 and f_2 be flows in a network N and let $A(X,\overline{X})$ be a cut in N.

(a) Show that if f_1 and f_2 are both maximum flows then we need not have $f_1(a) = f_2(a)$ for every $a \in A(X,\overline{X})$ and every $a \in A(\overline{X},X)$.

(b) If f_1 and f_2 agree on both $A(X,\overline{X})$ and $A(\overline{X},X)$, are both f_1 and f_2 maximum flows?

8.2 The Ford and Fulkerson Algorithm

We now present an algorithm, due to Ford and Fulkerson [24], and based on the proof of the Max-Flow Min-Cut Theorem, which constructs a maximal flow. It uses a labelling technique to produce a maximal flow. Starting with a known flow, for example the flow which has 0 assigned to each arc (i.e., the **zero flow**), it recursively constructs a sequence of flows of increasing value, terminating with a maximal flow.

To describe the labelling technique we need the following definition.

> An **f-unsaturated tree** of the network N (with respect to the flow f) is a subtree T of the underlying graph G of N such that
> (i) the source s is a vertex of T,
> (ii) for every vertex v of T the unique $s-v$ path in T is an f-unsaturated path.

In Figure 8.11 an example of an f-unsaturated tree is given by the shaded edges.

The sequence of flows of increasing value is constructed using f-incrementing walks, which are found by "growing" f-unsaturated trees. The growing procedure is as follows.

(i) Initially the f-unsaturated tree T consists of just the source s,

(ii) If X is the set of vertices of T at any given stage, an arc a is adjoined to T in either of the two following ways, *provided this process does not create cycles, (i.e., provided the end result is still a tree)*:

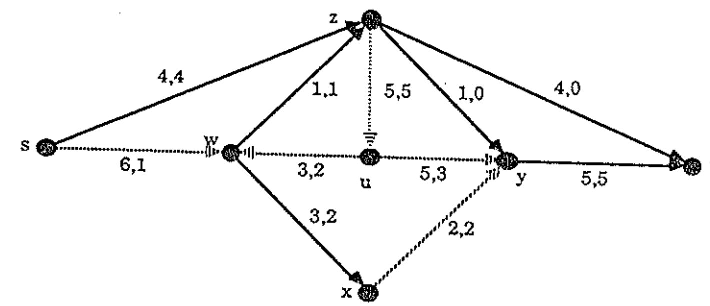


Figure 8.11: An f-unsaturated tree.

(a) if there is an arc a in $A(X,\overline{X})$ such that $f(a) < c(a)$ then adjoin both a and its head to T,
(b) if there is an arc a in $A(\overline{X},X)$ such that $f(a) > 0$ then adjoin both a and its tail to T.

Under this construction our tree T either eventually grows out as far as the sink t or it stops short of t. If it does grow out as far as t we say we have **breakthrough** and in this case the $s-t$ path in T is an f-incrementing path. On the other hand, if T stops growing before it reaches t then, as the last part of the proof of the theorem shows, our flow f must be a maximal flow.

The actual labelling technique assigns labels to the vertices of an f-unsaturated tree T as it grows. For each vertex v of T the label $\lambda(v) = i(P_v)$ where P_v is the unique path in T from the source s to v. The advantage of this labelling is that if breakthrough does occur then not only do we have an f-incrementing path P_t from the source s to the sink t but also we have calculated $\lambda(t) = i(P_t)$, the increment used to obtain our revised flow based on P_t. The labelling procedure begins by assigning to the source s the label $\lambda(s) = \infty$. 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

$$\lambda(v) = \min\{\lambda(u), c(a) - f(a)\}.$$

(2) If a is an arc with $f(a) > 0$ whose head u is already labelled but whose tail v is not, then v is labelled

$$\lambda(v) = \min\{\lambda(u), f(a)\}.$$

In both cases we say that v is 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 t is labelled (i.e., we have breakthrough) or all labelled vertices have been scanned and no more vertices can be labelled (i.e., the flow f is a maximal flow).

We now give a formal statement of the algorithm. It uses the zero flow as the initial flow on the network N.

The Ford and Fulkerson algorithm

Step 1. For each arc xy in N set $f(xy) = 0$.

Step 2. Set $A' = \emptyset$. (A' is the set of arcs in the unsaturated tree.)

For the source s of N, set $\lambda(s) = \infty$.

Set $L = \{s\}$. (L is the set of labelled vertices.)

Set $S = \emptyset$. (S is the set of vertices in L which have been scanned.)

Step 3. Let F be the set of arcs xy in N such that $c(xy) > f(xy)$. (F consists of forward arcs.)

Let R be the set of arcs xy in N such that $f(xy) > 0$. (R consists of reverse arcs.)

Step 4. If $L - S = \emptyset$ go to Step 10.

Step 5. Choose $x \in L - S$.

Step 6. If there is *no* $y \in \overline{L}$ such that $xy \in F$ and the set of arcs $A' \cup \{xy\}$ induces an underlying tree, then go to Step 7.

If there *is* a $y \in \overline{L}$ such that $xy \in F$ and the set of arcs $A' \cup \{xy\}$ induces an underlying tree then label y by

$$\lambda(y) = \min\{\lambda(x), c(xy) - f(xy)\}.$$

Change A' to $A' \cup \{xy\}$ and L to $L \cup \{y\}$.

Now repeat this step.

Step 7. If there is *no* $y \in \overline{L}$ such that $yx \in R$ and the set of arcs $A' \cup \{yx\}$ induces an underlying tree, then go to Step 8.

If there *is* a $y \in \overline{L}$ such that $yx \in R$ and the set of arcs $A' \cup \{yx\}$ induces an underlying tree then label y by

$$\lambda(y) = \min\{\lambda(x), f(yx)\}.$$

Change A' to $A' \cup \{yx\}$ and L to $L \cup \{y\}$.

Now repeat this step.

Step 8. Change S to $S \cup \{x\}$. If $t \notin L$ return to Step 4.

Step 9. (We reach this step when the sink t has been labelled, i.e., when we have breakthrough.)

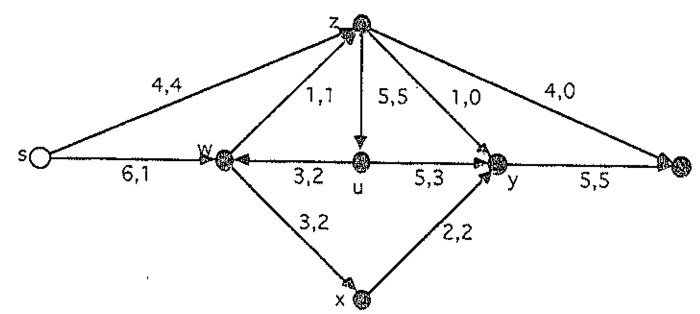
Using a backtracking procedure, identify the incrementing walk W from s to t in N using the arcs from A' and for each a in W change $f(a)$ to $f(a) + \lambda(t)$ if $a \in F$ and to $f(a) - \lambda(t)$ if $a \in R$.

Return to Step 2.

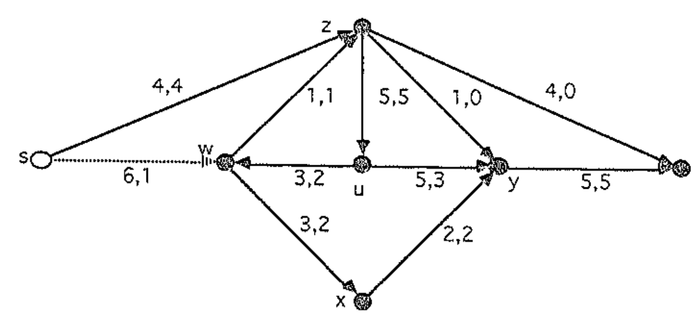
Step 10. (We reach this step when all labelled vertices have been scanned and there is no breakthrough.)

The values $f(xy)$ for each arc xy in N give a maximum flow in N and the set of vertices L gives a minimum cut.

We now illustrate the algorithm in Figures 8.12 – 8.16 using the network N of Figure 8.11, starting with the flow f given there (instead of the zero flow as used in the algorithm). At each stage the unsaturated tree is shown having shaded edges, the vertex being scanned is in white and the label assignment is shown below the diagram. To save space, we omit listing the specific steps of the algorithm used for the example and also recording the changes to the sets A', L, and S, but these should be clear to the reader from the diagrams.

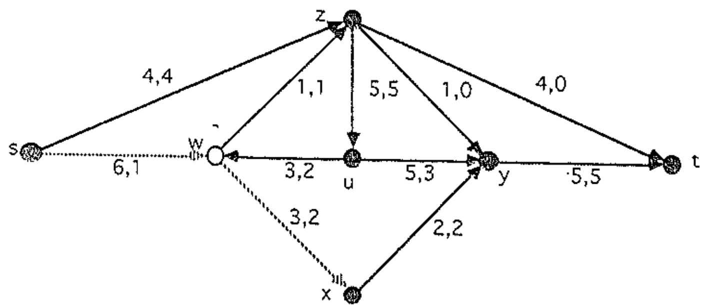


labelled vertices	s
label	∞

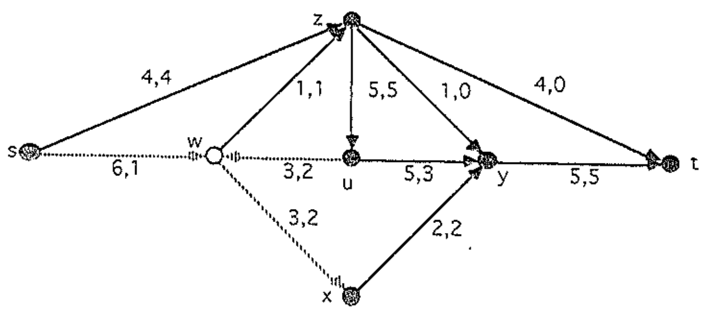


labelled vertices	s	w
label	∞	$\min\{\infty, 6-1\} = 5$

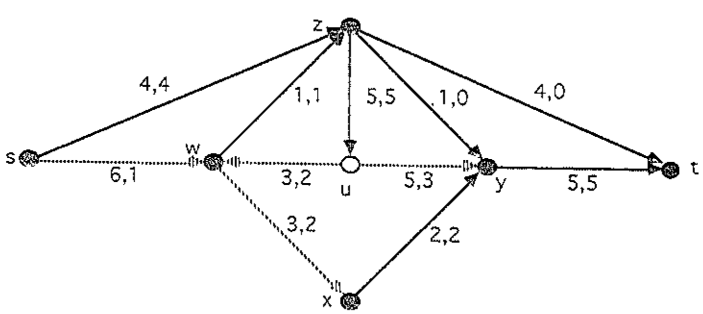
Figure 8.12



labelled vertices	s	w	x
label	∞	5	min{5, 3 − 2} = 1

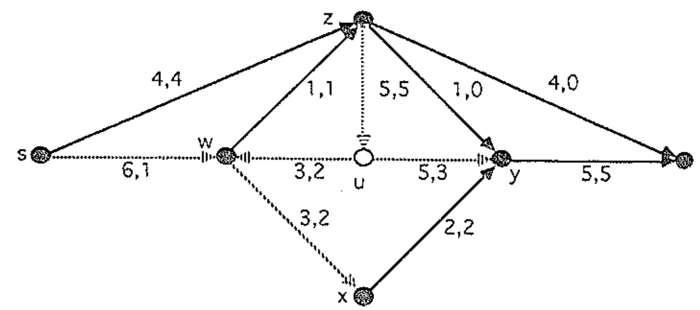


labelled vertices	s	w	x	u
label	∞	5	1	min{5, 2} = 2

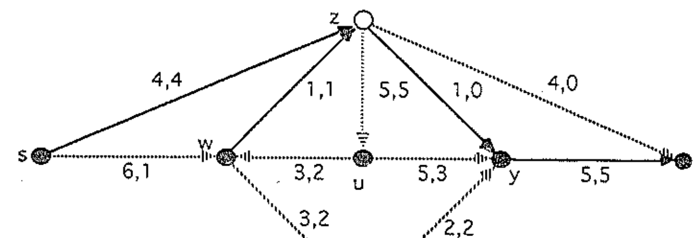


labelled vertices	s	w	x	u	y
label	∞	5	1	2	min{2, 5 − 3} = 2

Figure 8.13



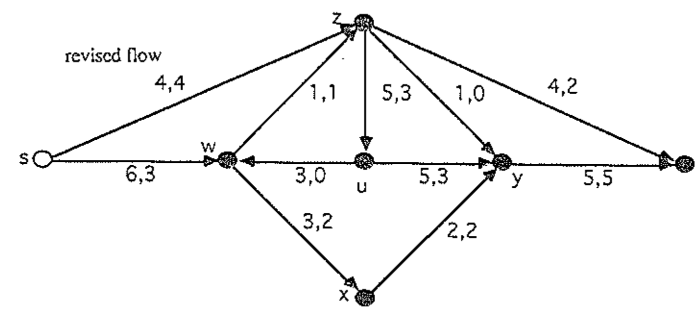
labelled vertices	s	w	x	u	y	z
label	∞	5	1	2	2	min{2, 5} = 2



Aha! - BREAKTHROUGH!
(with increment 2).

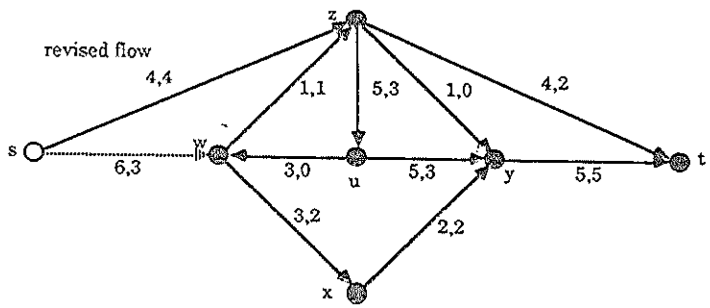
labelled vertices	s	w	x	u	y	z	t
label	∞	5	1	2	2	2	min{2, 4 − 0} = 2

revised flow



labelled vertices	s
label	∞

Figure 8.14



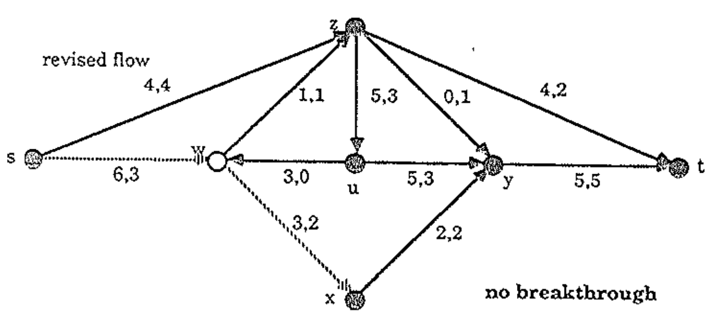
labelled vertices	s	w
label	∞	$\min\{\infty, 6-3\} = 3$

Figure 8.15

The following diagram is our last since, in the next stage, scanning all the labelled vertices, i.e., s, w and x, gives no new labelled vertices. Since all labelled vertices have been scanned, i.e., $L = S$ in the terminology of the algorithm, and we have no breakthrough, the flow shown in this last diagram is a maximal flow. It has value $f(sz) + f(sw) = 4 + 3 = 7$. Moreover the set $X = \{s, w, x\}$ of labelled vertices gives a minimal cut $A(X, \overline{X})$, with capacity given by

$$c(sz) + c(wz) + c(xy) = 4 + 1 + 2 = 7,$$

which is the flow's value (as expected).



labelled vertices	s	w	x
label	∞	3	$\min\{3, 3-2\} = 1$

Figure 8.16

Exercises for Section 8.2

8.2.1 Use the Ford and Fulkerson algorithm to find a maximal flow and a cut with capacity equal to this flow for both of the networks in Figure 8.10. (Draw the unsaturated trees at each stage of their growth and identify the sets A', L, S, F and R as defined in the step-wise presentation.)

8.2.2 Use the Ford and Fulkerson algorithm to find a maximal flow and a cut with capacity equal to this flow for each of the networks in Figure 8.17.

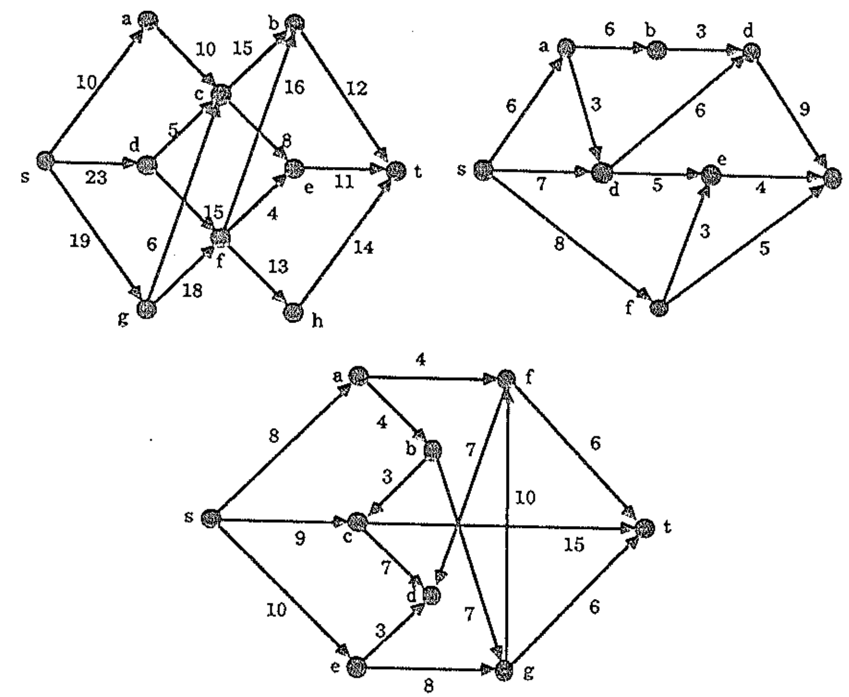


Figure 8.17: Two networks.

8.2.3 The two networks of Figure 8.18 have several sources and sinks. Use the technique described in Exercise 8.1.2 together with the Ford and Fulkerson algorithm to find maximal flows for both networks. What is the value of each of your flows? What is the total value of each of your flows out of each individual source and into each individual sink?

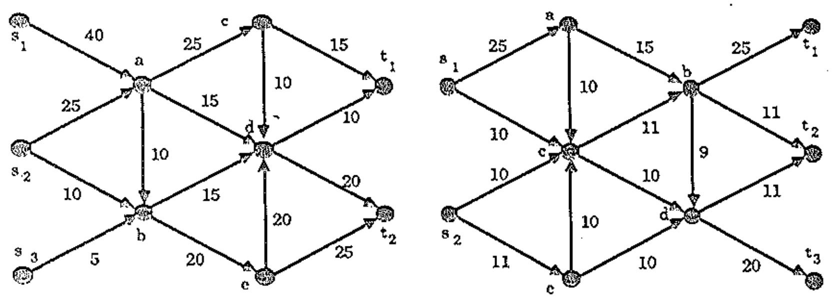


Figure 8.18: Two networks with several sources and sinks.

8.3 Separating Sets

In this section we shall use the Max-Flow Min-Cut Theorem to prove some results related to the connectivity of graphs.

> Let u and v be two distinct vertices of a graph G. A set S of vertices of G, containing neither u nor v, is said to be $u-v$ **separating** if the vertex deleted subgraph $G-S$ is disconnected with u and v lying in different components. In this case, S is also said to **separate** u and v.
>
> Similarly, a set F of edges of G is said to be $u-v$ **separating** if the edge deleted subgraph $G-F$ is disconnected with u and v lying in different components. In this case, F is said to **separate** u and v.

For example, in the graph G of Figure 8.10, the set of vertices $S = \{t, w, y\}$ is $u-v$ separating while the set of edges $F = \{tv, tx, wx, wz, uy\}$ is $u-v$ separating.

Our first result gives a simple characterisation of separating sets.

Theorem 8.3 *Let u and v be two distinct vertices of the graph G.*

(a) A set S of vertices of G is $u-v$ separating if and only if every $u-v$ path has at least one internal vertex belonging to S.

(b) A set F of edges of G is $u-v$ separating if and only if every $u-v$ path has at least one edge belonging to F.

Proof (*a*) Let S be a $u-v$ separating set of vertices in G and let P be a $u-v$ path. If P has no internal vertices belonging to S then the deletion of S from G leaves P intact. But then u and v will be in the same connected component of $G-S$, which is a contradiction since S is $u-v$ separating. Thus every $u-v$ path must have at least one internal vertex belonging to S.

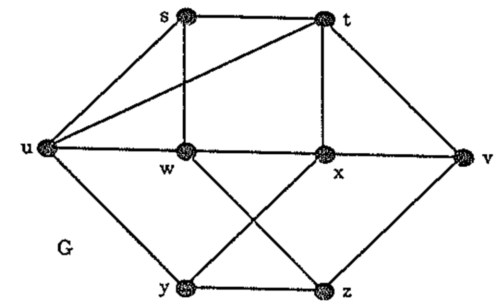


Figure 8.19

Conversely, suppose that every $u-v$ path has an internal vertex belonging to S. Then if we delete S from G there will be no path from u to v in the resulting subgraph $G-S$. In other words, u and v are in different connected components of $G-S$ and so S is $u-v$ separating, as required.

(*b*) The proof of this is similar to part (*a*)'s and is left as Exercise 8.3.1. □

Our goal now is to show that, for two nonadjacent vertices u and v, the size of a smallest possible $u-v$ separating set of vertices (or edges) is equal to the largest possible number of internally disjoint (respectively edge disjoint) $u-v$ paths. Since this equates minimums with maximums, perhaps it is not too surprising that we will prove these results using the max-flow min-cut theorem. The vertex version was first proved by K. Menger in 1927 [44], but the proof given here is from a 1956 article by G. B. Dantzig and D. R. Fulkerson [14].

Theorem 8.4 (Menger's Theorem) *Let u and v be two nonadjacent vertices of a graph G. Then the maximum number of internally disjoint $u-v$ paths in G equals the minimum number of vertices in a $u-v$ separating set.*

Proof Let m be the maximum number of internally disjoint $u-v$ paths in G and let n be the minimum number of vertices in a $u-v$ separating set. Let $P_{(1)}, \ldots, P_{(m)}$ be m internally disjoint $u-v$ paths and let S be a $u-v$ separating set of vertices. Then, by Theorem 8.3 (a), each of the paths $P_{(1)}, \ldots, P_{(m)}$ contains at least one member of S. On the other hand, since the paths are internally disjoint, no member of S can occur as an internal vertex in more than one of the paths. From this we see that we have at least as many vertices in S as we have paths, i.e., $|S| \geq m$. Since S was any separating set, it follows that $m \leq n$.

To prove that $m \geq n$ we construct a network N from G, having u as source and v as sink, as follows. The vertex set of N consists of u and v together with a pair of vertices, denoted by $x^{(1)}$ and $x^{(2)}$, for each vertex x in G different from both u and v. Each such pair is joined by an arc from $x^{(1)}$ to $x^{(2)}$. Such an arc will be called an **internal** arc of N while all other arcs (still to be defined) will be called **external**. Now to each edge of the form ux in G we associate an arc from u to $x^{(1)}$ in N and to each edge of the form yv we associate an arc from $y^{(2)}$ to v. Finally, to each edge xy

in G, having neither u nor v as an end vertex, we associate two arcs, one from $x^{(2)}$ to $y^{(1)}$ and the other from $y^{(2)}$ to $x^{(1)}$. Figure 8.20 illustrates this construction.

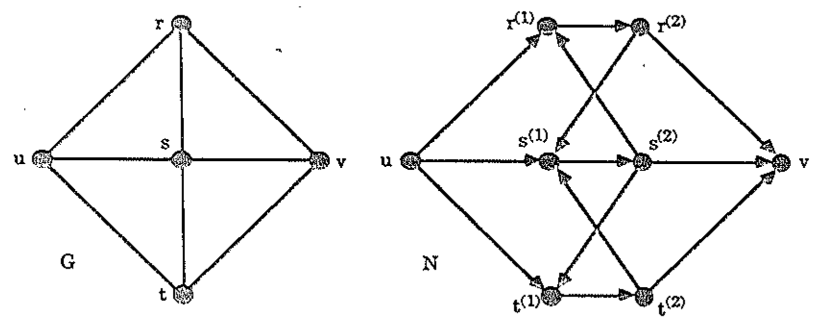


Figure 8.20

Now assign a capacity of one to each internal edge and infinite capacity to each of the external edges. (If the reader is unhappy with infinite capacities then he can change these to a large number, bigger than the total number of arcs in N, say.)

We now prove that m, the maximum number of internally disjoint $u-v$ paths in G, is equal to the value d of a maximum flow f in the network N. First note that in N any vertex of the form $x^{(1)}$ is the tail of only one arc, the associated internal arc, and this has capacity one. Similarly, any vertex of the form $x^{(2)}$ is the head of only one arc, again the associated internal arc with capacity one. It follows from this that the value of a flow into and out of any vertex in N is either one or zero. In particular, the maximum flow f of value d must be obtained using d internally disjoint directed $u-v$ paths in N, each contributing the value one to the total flow value.

Now any $u-v$ path $P = uu_1u_2\ldots u_kv$ in G uniquely induces the $u-v$ directed path

$$Q = uu_1^{(1)}u_1^{(2)}u_2^{(1)}u_2^{(2)}\ldots u_k^{(1)}u_k^{(2)}v$$

in N and this path Q permits a flow of value one along it. On the other hand, any directed $u-v$ path Q permitting such a flow must be induced in this way from a $u-v$ path P in G since the internal arcs are the only arcs of capacity one in N. Moreover, if the $u-v$ paths $P_{(1)},\ldots,P_{(t)}$ induce the directed paths $Q_{(1)},\ldots,Q_{(m)}$ in N then it is easy to see that $P_{(1)},\ldots,P_{(t)}$ are internally disjoint in G if and only if $Q_{(1)},\ldots,Q_{(m)}$ are internally disjoint in N. It now follows that the maximal flow value d corresponds to d internally disjoint $u-v$ paths in G and so $d \leq m$. Similarly, since m internally disjoint paths in G can induce a flow of value m in N and so $m \leq d$. Hence $m = d$, as required.

Now the Max-Flow Min-Cut Theorem tells us that there is some cut $A(X,\overline{X})$ in N such that $c(X,\overline{X}) = d$. Since d is finite, each arc from X to $\overline{X}$ must be an internal arc. Moreover, since $A(X,\overline{X})$ is a cut, each directed path from u to v must use at least one of these arcs. It follows that deleting the tails of all the arcs in $A(X,\overline{X})$ removes all possible $u-v$ directed paths in N. This in turn implies that the set of d vertices in G corresponding to these d tails must be a $u-v$ separating set. Since this has produced a $u-v$ separating set of size d we must have $d \geq n$. Thus, since $d = m$ and $m \leq n$, we have $m = n$, as required. □

As a simple consequence of Theorem 8.3 we have the following characterisation of n-connected graphs. Note that it generalises Whitney's Theorem (Theorem 2.21).

Theorem 8.5 *A simple graph G is n-connected if and only if, given any pair of distinct vertices u and v of G, there are at least n internally disjoint paths from u to v.*

Proof Suppose that G is n-connected and u and v are two distinct vertices of G. Then any $u-v$ separating set must have at least n vertices. Thus, by Theorem 8.4, there must be at least n internally disjoint $u-v$ paths, as required.

Conversely, suppose that, given any pair of distinct vertices u and v of G, there are at least n internally disjoint paths from u to v. Then, by Theorem 8.4, every $u-v$ separating set must have at least n vertices for each pair of vertices u and v. Thus it requires the deletion of at least n vertices from G in order to produce a disconnected graph or K_1. In other words, G is n-connected, as required. □

We now turn our attention to the edge analogues of Theorems 8.4 and 8.5. Their proofs are similar but a little easier. They are due to Ford and Fulkerson.

Theorem 8.6 (The edge version of Menger's Theorem) *Let u and v be two vertices of a graph G. Then the maximum number of edge disjoint $u-v$ paths in G equals the minimum number of edges in a $u-v$ separating set.*

Proof Let m be the maximum number of edge disjoint $u-v$ paths in G and let n be the minimum number of edges in a $u-v$ separating set. Let $P_{(1)},\ldots,P_{(m)}$ be m edge disjoint $u-v$ paths and let F be a $u-v$ separating set of edges. Then, by Theorem 8.3 (b), each of the paths $P_{(1)},\ldots,P_{(m)}$ contains at least one member of F. On the other hand, since the paths are edge disjoint, no member of F can occur as an edge in more than one of the paths. From this we see that we have at least as many edges in F as we have paths, i.e., $|F| \geq m$. Since F was any separating set, it follows that$m \leq n$.

To prove that $m \geq n$ we construct a network N from G, having u as source and v as sink, as follows. The vertex set of N is defined to be $V(G)$. To each edge of the form ux in G we associate an arc from u to x in N and to each edge of the form yv we associate an arc from y to v. To complete the arc set of N, for each edge uv in G, having neither u nor v as an end vertex, there are two associated arcs in N, namely one from u to v and one from v to u. Figure 8.21 illustrates this construction. Now assign a capacity of one to each arc in N.

We now prove that m, the maximum number of edge disjoint $u-v$ paths in G, is equal to the value d of a maximum flow f in the network N. First note that any $u-v$ path $P = uu_1u_2\ldots u_kv$ in G translates simply over to a directed $pu-v$ path P in N and, under this translation, any set of edge disjoint $u-v$ paths retains the property of being edge disjoint. Thus we can choose a set of m edge disjoint directed

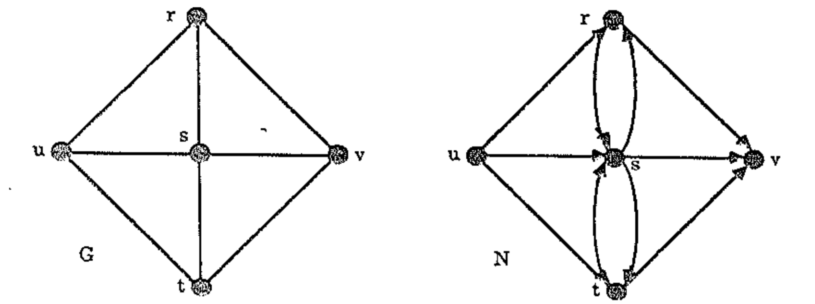


Figure 8.21

$u - v$ paths in N. Clearly each path in this set can contribute one to a flow in N and so there is a flow from u to v with value m. Thus $d \leq m$.

To see the opposite inequality, let f be a maximal flow with value d. Then since the capacity of each arc in N is just one, each arc can only contribute at most once to the flow's value d and so the flow is achieved by d edge disjoint directed $u - v$ trails, (each having flow component of value one). Moreover, given one of these directed trails, if it involves a pair of arcs (x, y) and (y, x), both having flow value one, then, by omitting these arcs, we may prune the trail to obtain a new trail still with flow value one. Indeed, doing this for all such pairs of arcs in each of the d trails, we produce d edge disjoint $u - v$ paths in N. We illustrate this pruning procedure in Figure 8.22. From this we get that $d \leq m$ and so $d = m$.

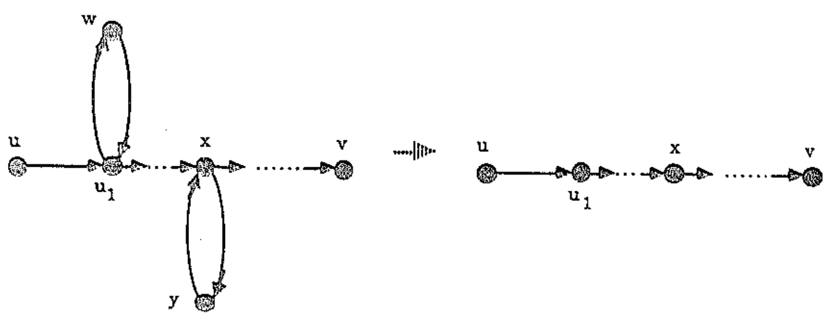


Figure 8.22

The max-flow min-cut theorem now tells us that there is some cut $C = A(X, \overline{X})$ in N such that $c(X, \overline{X}) = d$. Since C is a cut, every directed $u - v$ path in N uses at least one arc of C. Moreover, such a path is induced from a $u - v$ path in G and so each $u - v$ path in G uses at least one edge of the form xy such that the induced arc xy belongs to C. It follows that the set of all such edges in G is a $u - v$ separating set of edges and this set has at most $|C|$ we get $n \leq |C|$. However, $d = c(X, \overline{X}) = |C|$, since

the capacity of each arc in C is one, and so $n \leq d$. Thus, since $d = m$ and $m \leq n$, we have $m = n$, as required. □

We finish this chapter with a quick look at edge connectivity.

> Let G be a simple graph. The **edge connectivity** of G, denoted by $\kappa_e(G)$, is the smallest number of edges in G whose deletion from G either leaves a disconnected graph or an empty graph.

For example, any nonempty simple graph G with a bridge has $\kappa_e(G) = 1$. Clearly $\kappa_e(G) = 0$ if and only if either G is disconnected or an empty graph. The graph G of Figure 8.23 has $\kappa_e(G) = 2$.

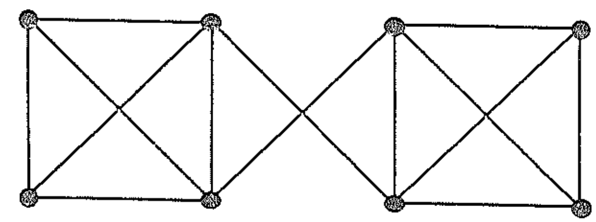


Figure 8.23

> A simple graph G is called n-**edge-connected** (where $n \geq 1$) if $\kappa_e(G) \geq n$.

As a simple consequence of Theorem 8.5 we have the following characterisation of n-connected graphs.

Theorem 8.7 *A simple graph G is n-edge-connected if and only if, given any pair of distinct vertices u and v of G, there are at least n edge disjoint paths from u to v.*

Proof Suppose that G is n-edge-connected and u and v are two distinct vertices of G. Then any $u - v$ separating set of edges must have at least n members. Thus, by Theorem 8.6, there must be at least n edge disjoint $u - v$ paths, as required.

Conversely, suppose that, given any pair of distinct vertices u and v of G, there are at least n edge disjoint paths from u to v. Then, by Theorem 8.6, for each pair of vertices u and v, every $u - v$ separating set of edges must have at least n members. Thus it requires the deletion of at least n edges from G in order to produce a disconnected graph or an empty graph. In other words, G is n-edge-connected, as required. □

Finally we give a result showing how the two connectivities are related. We let $\delta(G)$ denotes the smallest of all the vertex degrees in G.

Theorem 8.8 *Let G be a simple graph. Then*

$$\kappa(G) \leq \kappa_e(G) \leq \delta(G).$$

Proof Let v be a vertex with degree $\delta(G)$. Then deleting the $\delta(G)$ edges incident with v creates either the empty graph K_1 (if v is the only vertex in G) or a disconnected subgraph H, since v is an isolated vertex in H. Hence $\kappa_e(G) \leq \delta(G)$.

Now if $\kappa_e(G) = 0$ then either G is disconnected or empty and so $\kappa(G) = 0$. If $\kappa_e(G) = 1$ then G is connected and has a bridge, e say. In this case either $G = K_2$ or G has a cut vertex (namely one of e's ends). In either case $\kappa(G) = 1$. Thus we may now assume that $\kappa_e(G) \geq 2$. To simplify the notation we set $k = \kappa_e(G)$.

Then G has a set $\{e_1, \ldots, e_k\}$ of edges whose deletion disconnects G, but no set with less edges has this effect. Thus, deleting the first $k-1$ of these edges, $e_1, \ldots, e_{k-1}$, produces a connected subgraph H having the undeleted edge, e_k, as a bridge. Let $e_k = uv$. For $i = 1, \ldots, k-1$, it is now possible, since G is simple, to choose an end vertex u_i of the edge e_i, different from both u and v, such that $u_i \neq u_j$ for $i \neq j$. Let H' denote the subgraph obtained by deleting these $k-1$ vertices from G. If H' is disconnected then $\kappa(G) \leq k-1 < k = \kappa_e(G)$ and we have our required result. If, however, H' is connected then, since it is a subgraph of H containing the edge $e_k = uv$ as a bridge, it is either isomorphic to K_2 or it has either u or v as a cut vertex. Thus a deletion of one further vertex will produce from H' either a disconnected graph or a K_1. This shows that $\kappa(G) \leq k = \kappa_e(G)$. Hence $\kappa(G) \leq \kappa_e(G) \leq \delta(G)$. □

Exercises for Section 8.3

8.3.1 Let u and v be two distinct vertices of the graph G. Prove that a set F of edges of G is $u-v$ separating if and only if every $u-v$ path has at least one edge belonging to F. (This is Theorem 8.3 (b).)

8.3.2 Verify that Theorems 8.4 and 8.6 are true for the vertices u and v of the graphs G of Figure 8.24.

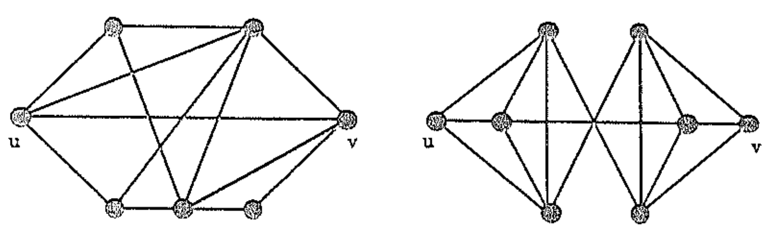


Figure 8.24

8.3.3 Verify that Theorems 8.5 and 8.7 are true for the complete bipartite graphs $K_{2,3}$ and $K_{3,4}$, the wheel W_5 and the 4-cube Q_4.

8.3.4 (In this exercise we show how matchings in a bipartite graph correspond to flows in an associated network.) Let G be a bipartite graph with bipartition $V(G) = X \cup Y$. Construct a network N from G by introducing two new vertices s and t, the source and sink of N, an arc from s to each of the vertices in X, an arc from each of these vertices in Y to t, and orienting the edges of G from vertices in X to those in Y. Assign capacity 1 to the arcs from s and also to the arcs to t and let all other arcs have infinite capacity (or a very large capacity).

(a) Let f be a flow from s to t in N and let M consist of all the edges xy in G such that the flow value $f(x,y)$ (on the corresponding arc) is positive. Prove that this defines a matching M in G.

(b) Conversely, let M be a matching in G. For each arc (u,v) in N, define $f(u,v) = 1$ if M saturates either u or v and $f(u,v) = 0$ otherwise. Prove that this defines a flow f on N with value equal to the number of edges in the matching M.

(c) Prove that the definitions of (a) and (b) establish a one-to-one correspondence between the set of matchings in G and the set of flows on N in such a way that maximum flows correspond to maximum matchings.

8.3.5 (In this exercise we sketch a proof of how Menger's Theorem can be used to prove Hall's Marriage Theorem (Theorem 4.3).) Let G be a bipartite graph with bipartition $V(G) = X \cup Y$ and suppose that $|N(S)| \geq |S|$ for every subset S of X. Adjoin to G two new vertices s and t, an edge joining s to each vertex of X and an edge joining t to each vertex of Y. (Compare this with the construction of the network in the previous exercise.) Denote this supergraph of G by G_1.

(a) Let $X = \{x_1, \ldots, x_n\}$. Prove that G has a matching M which saturates X if and only if G_1 has a set of n internally disjoint $s-t$ paths $P_{(1)}, \ldots, P_{(n)}$ (so that, after reordering the paths if necessary, $P_{(i)}$ must go through x_i for each i).

(b) Using Hall's condition that $|N(S)| \geq |S|$ for every subset S of X, prove that any $s-t$ separating set of vertices in G_1 has at least n vertices and hence that G has a matching which saturates X.

8.3.6 Let G be a plane graph and let G^* be its dual (as defined in Section 5.6). For each set of edges F in G let F^* denote the set of edges in G^* corresponding to those in F.

(a) Prove that a set of edges F in G forms a cycle in G if and only if the edge-deleted subgraph $G^* - F^*$ is disconnected but $G^* - F_1^*$ is connected for any proper subset F_1 of F.

(b) Prove that for any set F of edges in G, the edges of F^* form a cycle in G^* if and only if $G - F$ is disconnected but $G - F_1$ is connected for any proper subset F_1 of F.