DISCRETE MATHEMATICS

W W L CHEN

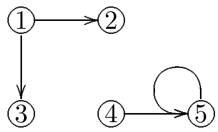
Chapter 20

DIGRAPHS

20.1. Introduction

A digraph (or directed graph) is simply a collection of vertices, together with some arcs joining some of these vertices.

Example 20.1.1. The digraph



has vertices $1, 2, 3, 4, 5$, while the arcs may be described by $(1, 2), (1, 3), (4, 5), (5, 5)$. In particular, any arc can be described as an ordered pair of vertices.

Definition. A digraph is an object $D = (V, A)$, where V is a finite set and A is a subset of the cartesian product $V \times V$. The elements of V are known as vertices and the elements of A are known as arcs.

Remark. Note that our definition permits an arc to start and end at the same vertex, so that there may be "loops". Note also that the arcs have directions, so that $(x, y) \neq (y, x)$ unless $x = y$.

Example 20.1.2. In Example 20.1.1, we have $V = \{1, 2, 3, 4, 5\}$ and $A = \{(1, 2), (1, 3), (4, 5), (5, 5)\}$. We can also represent this digraph by an adjacency list

1	2	3	4	5
2			5	5
3				

where, for example, the first column indicates that $(1, 2), (1, 3) \in A$ and the second column indicates that no arc in A starts at the vertex 2.

† This chapter was written at Macquarie University in 1992.

A digraph $D = (V, A)$ can be interpreted as a relation $\mathcal{R}$ on the set V in the following way. For every $x, y \in V$, we write $x\mathcal{R}y$ if and only if $(x, y) \in A$.

Example 20.1.3. The digraph in Example 20.1.2 represents the relation $\mathcal{R}$ on $\{1, 2, 3, 4, 5\}$, where

$$1\mathcal{R}2, \qquad 1\mathcal{R}3, \qquad 4\mathcal{R}5, \qquad 5\mathcal{R}5$$

and no other pair of elements are related.

The definitions of walks, paths and cycles carry over from graph theory in the obvious way.

Definition. A directed walk in a digraph $D = (V, A)$ is a sequence of vertices

$$v_0, v_1, \ldots, v_k \in V$$

such that for every $i = 1, \ldots, k$, $(v_{i-1}, v_i) \in A$. Furthermore, if all the vertices are distinct, then the directed walk is called a directed path. On the other hand, if all the vertices are distinct except that $v_0 = v_k$, then the directed walk is called a directed cycle.

Definition. A vertex y in a digraph $D = (V, A)$ is said to be reachable from a vertex x if there is a directed walk from x to y.

Remark. It is possible for a vertex to be not reachable from itself.

Example 20.1.4. In Example 20.1.3, the vertices 2 and 3 are reachable from the vertex 1, while the vertex 5 is reachable from the vertices 4 and 5.

It is sometimes useful to use square matrices to represent adjacency and reachability.

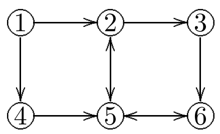
Definition. Let $D = (V, A)$ be a digraph, where $V = \{1, 2, \ldots, n\}$. The adjacency matrix of the digraph D is an $n \times n$ matrix $\mathcal{A}$ where a_{ij}, the entry on the i-th row and j-th column, is defined by

$$a_{ij} = \begin{cases} 1 & ((i, j) \in A), \\ 0 & ((i, j) \notin A). \end{cases}$$

The reachability matrix of the digraph D is an $n \times n$ matrix $\mathcal{R}$ where r_{ij}, the entry on the i-th row and j-th column, is defined by

$$r_{ij} = \begin{cases} 1 & (j \text{ is reachable from } i), \\ 0 & (j \text{ is not reachable from } i). \end{cases}$$

Example 20.1.5. Consider the digraph described by the following picture.



Then

$$\mathcal{A} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \qquad \text{and} \qquad \mathcal{Q} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

Remark. Reachability may be determined by a breadth-first search in the following way. Let us consider Example 20.1.5. Start with the vertex 1. We first search for all vertices y such that $(1, y) \in A$.

These are vertices 2 and 4, so that we have a list $2, 4$. Note that here we have the information that vertices 2 and 4 are reachable from vertex 1. However, we do not include 1 on the list, since we do not know yet whether vertex 1 is reachable from itself. We then search for all vertices y such that $(2, y) \in A$. These are vertices 3 and 5, so that we have a list $2, 4, 3, 5$. We next search for all vertices y such that $(4, y) \in A$. Vertex 5 is the only such vertex, so that our list remains $2, 4, 3, 5$. We next search for all vertices y such that $(3, y) \in A$. Vertex 6 is the only such vertex, so that our list becomes $2, 4, 3, 5, 6$. We next search for all vertices y such that $(5, y) \in A$. These are vertices 2 and 6, so that our list remains $2, 4, 3, 5, 6$. We next search for all vertices y such that $(6, y) \in A$. Vertex 5 is the only such vertex, so that our list remains $2, 4, 3, 5, 6$, and the process ends. We conclude that vertices $2, 3, 4, 5, 6$ are reachable from vertex 1, but that vertex 1 is not reachable from itself. We now repeat the whole process starting with each of the other five vertices.

Clearly we must try to find a better method. This is provided by Warshall's algorithm. However, before we state the algorithm in full, let us consider the central idea of the algorithm. The following simple idea is clear. Suppose that i, j, k are three vertices of a digraph. If it is known that j is reachable from i and that k is reachable from j, then k is reachable from i.

Let $V = \{1, \ldots, n\}$. Suppose that $\mathcal{Q}$ is an $n \times n$ matrix where q_{ij}, the entry on the i-th row and j-th column, is defined by

$$q_{ij} = \begin{cases} 1 & \text{(it is already established that } j \text{ is reachable from } i\text{)}, \\ 0 & \text{(it is not yet known whether } j \text{ is reachable from } i\text{)}. \end{cases}$$

Take a vertex j, and keep it fixed.

(1) Investigate the j-th row of $\mathcal{Q}$. If it is already established that k is reachable from j, then $q_{jk} = 1$. If it is not yet known whether k is reachable from j, then $q_{jk} = 0$.

(2) Investigate the j-th column of $\mathcal{Q}$. If it is already established that j is reachable from i, then $q_{ij} = 1$. If it is not yet known whether j is reachable from i, then $q_{ij} = 0$.

(3) It follows that if $q_{ij} = 1$ and $q_{jk} = 1$, then j is reachable from i and k is reachable from j. Hence we have established that k is reachable from i. We can therefore replace the value of q_{ik} by 1 if it is not already so.

(4) Doing a few of these manipulations simultaneously, we are essentially "adding" the j-th row of $\mathcal{Q}$ to the i-th row of $\mathcal{Q}$ provided that $q_{ij} = 1$. By "addition", we mean Boolean addition; in other words, $0 + 0 = 0$ and $1 + 0 = 0 + 1 = 1 + 1 = 1$. To understand this addition, suppose that $q_{ik} = 1$, so that it is already established that k is reachable from i. Then replacing q_{ik} by $q_{ik} + q_{jk}$ (the result of adding the j-th row to the i-th row) will not alter the value 1. Suppose, on the other hand, that $q_{ik} = 0$. Then we are replacing q_{ik} by $q_{ik} + q_{jk} = q_{jk}$. This will have value 1 if $q_{jk} = 1$, *i.e.* if it is already established that k is reachable from j. But then $q_{ij} = 1$, so that it is already established that j is reachable from i. This justifies our replacement of $q_{ik} = 0$ by $q_{ik} + q_{jk} = 1$.

WARSHALL'S ALGORITHM. *Consider a digraph $D = (V, A)$, where $V = \{1, \ldots, n\}$.*

(1) Let $\mathcal{Q}_0 = \mathcal{A}$.
(2) Consider the entries in $\mathcal{Q}_0$. Add row 1 of $\mathcal{Q}_0$ to every row of $\mathcal{Q}_0$ which has entry 1 on the first column. We obtain the new matrix $\mathcal{Q}_1$.
(3) Consider the entries in $\mathcal{Q}_1$. Add row 2 of $\mathcal{Q}_1$ to every row of $\mathcal{Q}_1$ which has entry 1 on the second column. We obtain the new matrix $\mathcal{Q}_2$.
(4) For every $j = 3, \ldots, n$, consider the entries in $\mathcal{Q}_{j-1}$. Add row j of $\mathcal{Q}_{j-1}$ to every row of $\mathcal{Q}_{j-1}$ which has entry 1 on the j-th column. We obtain the new matrix $\mathcal{Q}_j$.
(5) Write $\mathcal{R} = \mathcal{Q}_n$.

EXAMPLE 20.1.6. Consider Example 20.1.5, where

$$\mathcal{A} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

We write $\mathcal{Q}_0 = \mathcal{A}$. Since no row has entry 1 on the first column of $\mathcal{Q}_0$, we conclude that $\mathcal{Q}_1 = \mathcal{Q}_0 = \mathcal{A}$. Next we add row 2 to rows $1, 5$ to obtain

$$\mathcal{Q}_2 = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Next we add row 3 to rows $1, 2, 5$ to obtain

$$\mathcal{Q}_3 = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Next we add row 4 to row 1 to obtain $\mathcal{Q}_4 = \mathcal{Q}_3$. Next we add row 5 to rows $1, 2, 4, 5, 6$ to obtain

$$\mathcal{Q}_5 = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

Finally we add row 6 to every row to obtain

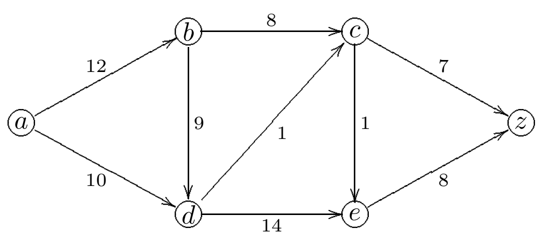
$$\mathcal{R} = \mathcal{Q}_6 = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

20.2. Networks and Flows

Imagine a digraph and think of the arcs as pipes, along which some commodity (water, traffic, *etc.*) will flow. There will be weights attached to each of these arcs, to be interpreted as capacities, giving limits to the amount of commodity which can flow along the pipe. We also have a source a and a sink z; in other words, all arcs containing a are directed away from a and all arcs containing z are directed to z.

DEFINITION. By a network, we mean a digraph $D = (V, A)$, together with a capacity function $c : A \to \mathbb{N}$, a source vertex $a \in V$ and a sink vertex $z \in V$.

EXAMPLE 20.2.1. Consider the network described by the picture below.



This has vertex a as source and vertex z as sink.

Suppose now that a commodity is flowing along the arcs of a network. For every $(x,y) \in A$, let $f(x,y)$ denote the amount which is flowing along the arc (x,y). We shall use the convention that, with the exception of the source vertex a and the sink vertex z, the amount flowing into a vertex v is equal to the amount flowing out of the vertex v. Also the amount flowing along any arc cannot exceed the capacity of that arc.

DEFINITION. A flow from a source vertex a to a sink vertex z in a network $D = (V, A)$ is a function $f : A \to \mathbb{N} \cup \{0\}$ which satisfies the following two conditions:

(F1) (CONSERVATION) For every vertex $v \neq a, z$, we have $I(v) = O(v)$, where the inflow $I(v)$ and the outflow $O(v)$ at the vertex v are defined by

$$I(v) = \sum_{(x,v)\in A} f(x,v) \qquad \text{and} \qquad O(v) = \sum_{(v,y)\in A} f(v,y).$$

(F2) (LIMIT) For every $(x,y) \in A$, we have $f(x,y) \leq c(x,y)$.

REMARK. We have restricted the function f to have non-negative integer values. In general, neither the capacity function c nor the flow function f needs to be restricted to integer values. We have made our restrictions in order to avoid complications about the existence of optimal solutions.

It is not hard to realize that the following result must hold.

THEOREM 20A. *In any network with source vertex a and sink vertex z, we must have $O(a) = I(z)$.*

DEFINITION. The common value of $O(a) = I(z)$ of a network is called the value of the flow f, and is denoted by $\mathcal{V}(f)$.

Consider a network $D = (V, A)$ with source vertex a and sink vertex z. Let us partition the vertex set V into a disjoint union $S \cup T$ such that $a \in S$ and $z \in T$. Then the net flow from S to T, in view of (F1), is the same as the flow from a to z. In other words, we have

$$\mathcal{V}(f) = \sum_{\substack{x\in S\\ y\in T\\ (x,y)\in A}} f(x,y) - \sum_{\substack{x\in T\\ y\in S\\ (x,y)\in A}} f(x,y).$$

Clearly both sums on the right-hand side are non-negative. It follows that

$$\mathcal{V}(f) \leq \sum_{\substack{x\in S\\ y\in T\\ (x,y)\in A}} f(x,y) \leq \sum_{\substack{x\in S\\ y\in T\\ (x,y)\in A}} c(x,y),$$

in view of (F2).

DEFINITION. If $V = S \cup T$, where S and T are disjoint and $a \in S$ and $z \in T$, then we say that (S,T) is a cut of the network $D = (V, A)$. The value

$$c(S,T) = \sum_{\substack{x\in S\\ y\in T\\ (x,y)\in A}} c(x,y)$$

is called the capacity of the cut (S,T).

We have proved the following theorem.

THEOREM 20B. *Suppose that $D = (V, A)$ is a network with capacity function $c : A \to \mathbb{N}$. Then for every flow $f : A \to \mathbb{N} \cup \{0\}$ and every cut (S, T) of the network, we have*

$$\mathcal{V}(f) \leq c(S, T).$$

Suppose that f_0 is a flow where $\mathcal{V}(f_0) \geq \mathcal{V}(f)$ for every flow f, and suppose that (S_0, T_0) is a cut such that $c(S_0, T_0) \leq c(S, T)$ for every cut (S, T). Then we clearly have $\mathcal{V}(f_0) \leq c(S_0, T_0)$. In other words, the maximum flow never exceeds the minimum cut.

20.3. The Max-Flow-Min-Cut Theorem

In this section, we shall describe a practical algorithm which will enable us to increase the value of a flow in a given network, provided that the flow has not yet achieved maximum value. This method also leads to a proof of the important result that the maximum flow is equal to the minimum cut.

We shall use the following notation. Suppose that $(x, y) \in A$. Suppose further that $c(x, y) = \alpha$ and $f(x, y) = \beta$. Then we shall describe this information by the following picture.

$$x \xrightarrow{\alpha(\beta)} y \tag{1}$$

Naturally $\alpha \geq \beta$ always.

Definition. In the notation of the picture (1), we say that we can label forwards from the vertex x to the vertex y if $\beta < \alpha$, *i.e.* if $f(x, y) < c(x, y)$.

Definition. In the notation of the picture (1), we say that we can label backwards from the vertex y to the vertex x if $\beta > 0$, *i.e.* if $f(x, y) > 0$.

Definition. Suppose that the sequence of vertices

$$v_0(= a), v_1, \ldots, v_k(= z) \tag{2}$$

satisfies the property that for each $i = 1, \ldots, k$, we can label forwards or backwards from v_{i-1} to v_i. Then we say that the sequence of vertices in (2) is a flow-augmenting path.

Let us consider two examples.

Example 20.3.1. Consider the flow-augmenting path given in the picture below (note that we have not shown the whole network).

$$a \xrightarrow{9(7)} b \xrightarrow{8(5)} c \xrightarrow{4(0)} f \xrightarrow{3(1)} z$$

Here the labelling is forwards everywhere. Note that the arcs $(a, b), (b, c), (c, f), (f, z)$ have capacities $9, 8, 4, 3$ respectively, whereas the flows along these arcs are $7, 5, 0, 1$ respectively. Hence we can increase the flow along each of these arcs by $2 = \min\{9 - 7, 8 - 5, 4 - 0, 3 - 1\}$ without violating (F2). We then have the following situation.

$$a \xrightarrow{9(9)} b \xrightarrow{8(7)} c \xrightarrow{4(2)} f \xrightarrow{3(3)} z$$

We have increased the flow from a to z by 2.

Example 20.3.2. Consider the flow-augmenting path given in the picture below (note again that we have not shown the whole network).

$$a \xrightarrow{9(7)} b \xrightarrow{8(5)} c \xleftarrow{4(2)} g \xrightarrow{10(8)} z$$

Here the labelling is forwards everywhere, with the exception that the vertex g is labelled backwards from the vertex c. Note now that $2 = \min\{9-7, 8-5, 2, 10-8\}$. Suppose that we increase the flow along each of the arcs $(a, b), (b, c), (g, z)$ by 2 and decrease the flow along the arc (g, c) by 2. We then have the following situation.

$$\begin{tikzcd} a \arrow[r, "9(9)"] & b \arrow[r, "8(7)"] & c & g \arrow[l, "4(0)"'] \arrow[r, "10(10)"] & z \end{tikzcd}$$

Note that the sum of the flow from b and g to c remains unchanged, and that the sum of the flow from g to c and z remains unchanged. We have increased the flow from a to z by 2.

FLOW-AUGMENTING ALGORITHM. *Consider a flow-augmenting path of the type* (2). *For each $i = 1, \ldots, k$, write*

$$\delta_i = \begin{cases} c(v_{i-1}, v_i) - f(v_{i-1}, v_i) & ((v_{i-1}, v_i) \in A \text{ (forward labelling)}), \\ f(v_i, v_{i-1}) & ((v_i, v_{i-1}) \in A \text{ (backward labelling)}), \end{cases}$$

and let

$$\delta = \min\{\delta_1, \ldots, \delta_k\}.$$

We increase the flow along any arc of the form (v_{i-1}, v_i) (forward labelling) by δ and decrease the flow along any arc of the form (v_i, v_{i-1}) (backward labelling) by δ.

Definition. Suppose that the sequence of vertices

$$v_0(= a), v_1, \ldots, v_k \tag{3}$$

satisfies the property that for each $i = 1, \ldots, k$, we can label forwards or backwards from v_{i-1} to v_i. Then we say that the sequence of vertices in (3) is an incomplete flow-augmenting path.

Remark. The only difference here is that the last vertex is not necessarily the sink vertex z.

We are now in a position to prove the following important result.

THEOREM 20C. *In any network with source vertex a and sink vertex z, the maximum value of a flow from a to z is equal to the minimum value of a cut of the network.*

Proof. Consider a network $D = (V, A)$ with capacity function $c : A \to \mathbb{N}$. Let $f : A \to \mathbb{N} \cup \{0\}$ be a maximum flow. Let

$$S = \{x \in V : x = a \text{ or there is an incomplete flow-augmenting path from } a \text{ to } x\},$$

and let $T = V \setminus S$. Clearly $z \in T$, otherwise there would be a flow-augmenting path from a to z, and so the flow f could be increased, contrary to our hypothesis that f is a maximum flow. Suppose that $(x, y) \in A$ with $x \in S$ and $y \in T$. Then there is an incomplete flow-augmenting path from a to x. If $c(x, y) > f(x, y)$, then we could label forwards from x to y, so that there would be an incomplete flow-augmenting path from a to y, a contradiction. Hence $c(x, y) = f(x, y)$ for every $(x, y) \in A$ with $x \in S$ and $y \in T$. On the other hand, suppose that $(x, y) \in A$ with $x \in T$ and $y \in S$. Then there is an incomplete flow-augmenting path from a to y. If $f(x, y) > 0$, then we could label backwards from y to x, so that there would be an incomplete flow-augmenting path from a to x, a contradiction. Hence $f(x, y) = 0$ for every $(x, y) \in A$ with $x \in T$ and $y \in S$. It follows that

$$\mathcal{V}(f) = \sum_{\substack{x \in S \\ y \in T \\ (x,y) \in A}} f(x, y) - \sum_{\substack{x \in T \\ y \in S \\ (x,y) \in A}} f(x, y) = \sum_{\substack{x \in S \\ y \in T \\ (x,y) \in A}} c(x, y) = c(S, T).$$

For any other cut (S', T'), it follows from Theorem 20B that

$$c(S', T') \geq \mathcal{V}(f) = c(S, T).$$

Hence (S, T) is a minimum cut. ♣

MAXIMUM-FLOW ALGORITHM. *Consider a network $D = (V, A)$ with capacity function $c : A \to \mathbb{N}$.*

(1) Start with any flow, usually $f(x, y) = 0$ for every $(x, y) \in A$.

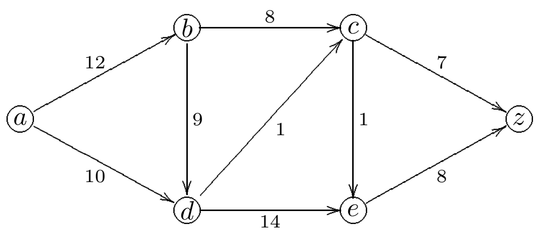
(2) Use a Breadth-first search algorithm to construct a tree of incomplete flow-augmenting paths, starting from the source vertex a.

(3) If the tree reaches the sink vertex z, pick out a flow-augmenting path and apply the Flow-augmenting algorithm to this path. The flow is now increased by δ. Then return to (2) with this new flow.

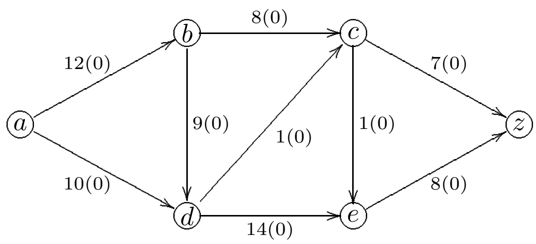
(4) If the tree does not reach the sink vertex z, let S be the set of vertices of the tree, and let $T = V \setminus S$. The flow is a maximum flow and (S, T) is a minimum cut.

Remark. It is clear from steps (2) and (3) that all we want is a flow-augmenting path. If one such path is readily recognizable, then we do not need to carry out the Breadth-first search algorithm. This is particularly important in the early part of the process.

Example 20.3.3. We wish to find the maximum flow of the network described by the picture below.



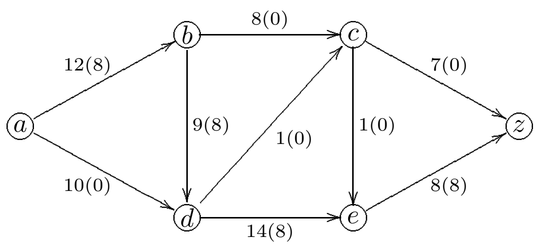
We start with a flow $f : A \to \mathbb{N} \cup \{0\}$ defined by $f(x, y) = 0$ for every $(x, y) \in A$. Then we have the situation below.



By inspection, we have the following flow-augmenting path.

$$a \xrightarrow{12(0)} b \xrightarrow{9(0)} d \xrightarrow{14(0)} e \xrightarrow{8(0)} z$$

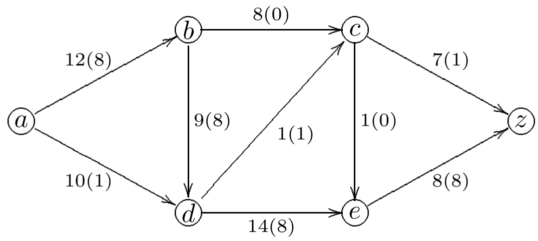
We can therefore increase the flow from a to z by 8, so that we have the situation below.



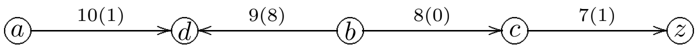
By inspection again, we have the following flow-augmenting path.

$$a \xrightarrow{10(0)} d \xrightarrow{1(0)} c \xrightarrow{7(0)} z$$

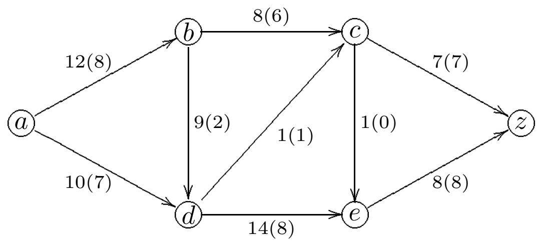
We can therefore increase the flow from a to z by 1, so that we have the situation below.



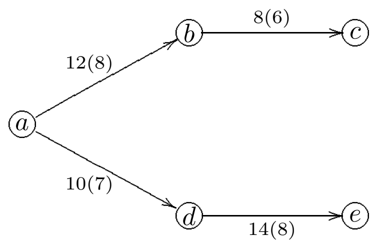
By inspection again, we have the following flow-augmenting path.



We can therefore increase the flow from a to z by $6 = \min\{10 - 1, 8, 8 - 0, 7 - 1\}$, so that we have the situation below.



Next, we use a Breadth-first search algorithm to construct a tree of incomplete flow-augmenting paths. This may be in the form



This tree does not reach the sink vertex z. Let $S = \{a, b, c, d, e\}$ and $T = \{z\}$. Then (S, T) is a minimum cut. Furthermore, the maximum flow is given by

$$\sum_{(v,z)\in A} f(v, z) = 7 + 8 = 15.$$

Problems for Chapter 20

1. Consider the digraph described by the following adjacency list.

1	2	3	4	5	6
4	1	2	2	6	1
5			3		
			5		

   a) Find a directed path from vertex 3 to vertex 6.
   b) Find a directed cycle starting from and ending at vertex 4.
   c) Find the adjacency matrix of the digraph.
   d) Apply Warshall's algorithm to find the reachability matrix of the digraph.

2. Consider the digraph described by the following adjacency list.

1	2	3	4	5	6	7	8
2	1	5	6	4	5	6	4
3	4		8	6		8	

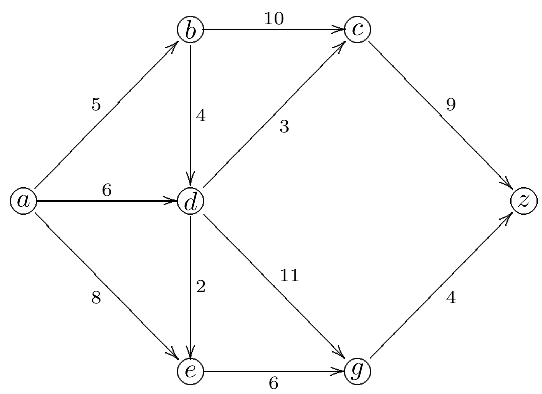
a) Does there exist a directed path from vertex 3 to vertex 2?
b) Find all the directed cycles of the digraph.
c) Find the adjacency matrix of the digraph.
d) Apply Warshall's algorithm to find the reachability matrix of the digraph.

3. Consider the digraph described by the following adjacency list.

1	2	3	4	5	6	7	8
2	4	1	3	6	7	5	7
4					8		

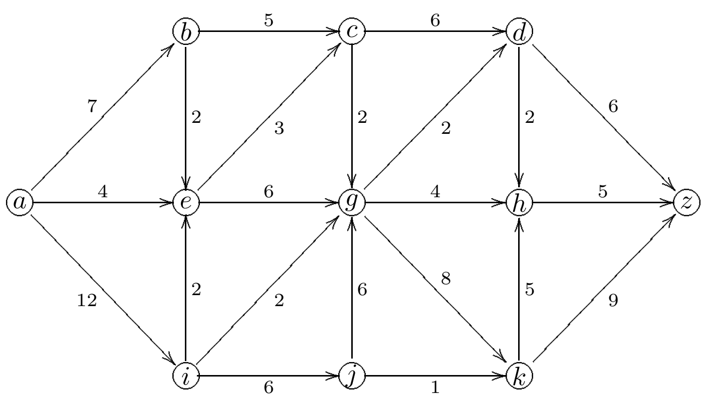
a) Does there exist a directed path from vertex 1 to vertex 8?
b) Find all the directed cycles of the digraph starting from vertex 1.
c) Find the adjacency matrix of the digraph.
d) Apply Warshall's algorithm to find the reachability matrix of the digraph.

4. Consider the network $D = (V, A)$ described by the following diagram.



a) A flow $f : A \to \mathbb{N} \cup \{0\}$ is defined by $f(a, b) = f(b, c) = f(c, z) = 3$ and $f(x, y) = 0$ for any arc $(x, y) \in A \setminus \{(a, b), (b, c), (c, z)\}$. What is the value $\mathcal{V}(f)$ of this flow?
b) Find a maximum flow of the network, starting with the flow in part (a).
c) Find a corresponding minimum cut.

5. Consider the network $D = (V, A)$ described by the following diagram.

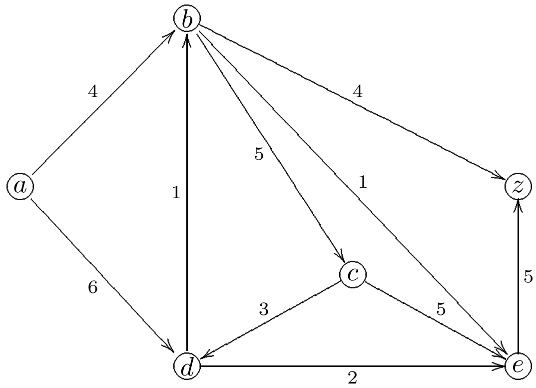


a) A flow $f : A \to \mathbb{N} \cup \{0\}$ is defined by $f(a,i) = f(i,j) = f(j,g) = f(g,k) = f(k,h) = f(h,z) = 5$ and $f(x,y) = 0$ for any arc $(x,y) \in A \setminus \{(a,i),(i,j),(j,g),(g,k),(k,h),(h,z)\}$. What is the value $\mathcal{V}(f)$ of this flow?

b) Find a maximum flow of the network, starting with the flow in part (a).

c) Find a corresponding minimum cut.

6. Find a maximum flow and a corresponding minimum cut of the following network.



− ∗ − ∗ − ∗ − ∗ − ∗ −