© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

CONTENTS

CHAPTER I

STATIC MAXIMAL FLOW

CHAPTER II

FEASIBILITY THEOREMS AND COMBINATORIAL APPLICATIONS

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

CHAPTER III

MINIMAL COST FLOW PROBLEMS

CHAPTER IV

MULTI-TERMINAL MAXIMAL FLOWS

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

CHAPTER I

STATIC MAXIMAL FLOW

Introduction

The mathematical problem which forms the subject matter of this chapter, that of determining a maximal steady state flow from one point to another in a network subject to capacity limitations on arcs, comes up naturally in the study of transportation or communication networks. It was posed to the authors in the spring of 1955 by T.E.Harris, who, in conjunction with General F.S.Ross (Ret.), had formulated a simplified model of railway traffic flow, and pinpointed this particular problem as the central one suggested by the model [11]. It was not long after this until the main result, Theorem 5.1, which we call the max-flow min-cut theorem, was conjectured and established [4]. A number of proofs of this theorem have since appeared [2, 3, 5, 14]. The constructive proof given in § 5 is the simplest and most revealing of the several known to us.

Sections 1 and 2 discuss networks and flows in networks. There are many alternative ways of formulating the concept of a flow through a network; two of these are described in § 2. After introducing some notation in § 3, and defining the notion of a cut in § 4, we proceed to a statement and proof of the max-flow min-cut theorem in § 5. Sections 6, 7, 9, 10, and 11 amplify and extend this theorem. In § 8, the construction implicit in its proof is detailed and illustrated. This construction, which we call the "labeling process," forms the basis for almost all the algorithms presented later in the book. A consequence of the construction is the integrity theorem (Theorem 8.1), which has been known in connection with similar problems since G.B.Dantzig pointed it out in 1951 [1]. It is this theorem that makes network flow theory applicable in certain combinatorial investigations.

Section 12 provides a brief presentation of duality principles for linear programs. Since no proofs are included, the reader not familiar with linear inequality theory may find this section too brief to be very illuminating. But excellent discussions are available [8, 9, 10]. We include § 12 mainly to note that the max-flow min-cut theorem is a kind of combinatorial counterpart, for the special case of the maximal flow problem, of the more general duality theorem for linear programs.

Section 13 uses the max-flow min-cut theorem to examine maximal flow through a network as a function of a pair of individual arc capacities. The

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

main conclusion here, which may sound empty but is not, is that any two arcs either always reinforce each other or always interfere with each other.

1. Networks

A *directed network* or *directed linear graph* $G = [N; \mathscr{A}]$ consists of a collection N of elements $x, y, \ldots$, together with a subset $\mathscr{A}$ of the ordered pairs (x, y) of elements taken from N. It is assumed throughout that N is a finite set, since our interest lies mainly in the construction of computational procedures. The elements of N are variously called *nodes*, *vertices*, *junction points*, or *points*; members of $\mathscr{A}$ are referred to as *arcs*, *links*, *branches*, or *edges*. We shall use the node-arc terminology throughout.

A network may be pictured by selecting a point corresponding to each node x of N and directing an arrow from x to y if the ordered pair (x, y) is in $\mathscr{A}$. For example, the network shown in Fig. 1.1 consists of four nodes s, x, y, t, and six arcs (s, x), (s, y), (x, y), (y, x), (x, t) and (y, t).

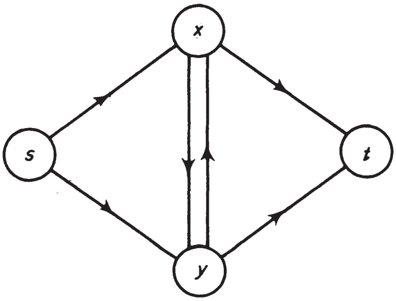

Figure 1.1

Such a network is said to be directed, since each arc carries a specific orientation or direction. Occasionally we shall also consider *undirected networks*, for which the set $\mathscr{A}$ consists of unordered pairs of nodes, or *mixed networks*, in which some arcs are directed, others are not. We can of course picture these in the same way, omitting arrowheads on arcs having no orientation. Until something is said to the contrary, however, each arc of the network will be assumed to have an orientation.

We have not as yet ruled out the possibility of arcs (x, x) leading from a node x to itself, but for our purposes we may as well do so. Thus, all arcs will be supposed to be of the form (x, y) with $x \neq y$. Also, while the existence of at most one arc (x, y) has been postulated, the notion of a network frequently allows multiple arcs joining x to y. Again, for most of the problems we shall consider, this added generality gains nothing, and so we shall continue to think of at most one arc leading from any node to another, unless an explicit statement is made to the contrary.

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

§1. NETWORKS

Let $x_1, x_2, \ldots, x_n$ $(n \geqslant 2)$ be a sequence of distinct nodes of a network such that (x_i, x_{i+1}) is an arc, for each $i = 1, \ldots, n - 1$. Then the sequence of nodes and arcs

$$x_1, (x_1, x_2), x_2, \ldots, (x_{n-1}, x_n), x_n \tag{1.1}$$

is called a *chain*; it leads from x_1 to x_n. Sometimes, for emphasis, we call (1.1) a *directed chain.* If the definition of a chain is altered by stipulating that $x_n = x_1$, then the displayed sequence is a *directed cycle.* For example, in the network of Fig. 1.1, the chain s, (s, x), x, (x, t), t leads from s to t; this network contains just one directed cycle, namely, x, (x, y), y, (y, x), x.

Let $x_1, x_2, \ldots, x_n$ be a sequence of distinct nodes having the property that either (x_i, x_{i+1}) or (x_{i+1}, x_i) is an arc, for each $i = 1, \ldots, n - 1$. Singling out, for each i, one of these two possibilities, we call the resulting sequence of nodes and arcs a *path from x_1 to x_n.* Thus a path differs from a chain by allowing the possibility of traversing an arc in a direction opposite to its orientation in going from x_1 to x_n. (For undirected networks, the two notions coincide.) Arcs (x_i, x_{i+1}) that belong to the path are *forward* arcs of the path; the others are *reverse* arcs. For example, the sequence s, (s, y), y, (x, y), x, (x, t), t is a path from s to t in Fig. 1.1; it contains the forward arcs (s, y), (x, t) and the reverse arc (x, y). If, in the definition of path, we stipulate that $x_1 = x_n$, then the resulting sequence of nodes and arcs is a *cycle.*

We may shorten the notation and refer unambiguously to the chain $x_1, x_2, \ldots, x_n$. Occasionally we shall also refer to the path $x_1, x_2, \ldots, x_n$; then it is to be understood that some selection of arcs has tacitly been made.

Given a network $[N; \mathscr{A}]$, one can form a *node-arc incidence matrix* as follows. List the nodes of the network vertically, say, the arcs horizontally, and record, in the column corresponding to arc (x, y), a 1 in the row corresponding to node x, a -1 in the row corresponding to y, and zeros elsewhere. For example, the network of Fig. 1.1 has incidence matrix

$$\begin{array}{c|cccccc|} & (s,x) & (s,y) & (x,y) & (y,x) & (x,t) & (y,t) \\ s & 1 & 1 & 0 & 0 & 0 & 0 \\ x & -1 & 0 & 1 & -1 & 1 & 0 \\ y & 0 & -1 & -1 & 1 & 0 & 1 \\ t & 0 & 0 & 0 & 0 & -1 & -1 \end{array}.$$

Clearly, all information about the structure of a network is embodied in its node-arc incidence matrix.

If $x \in N$, we let $A(x)$ ("after x") denote the set of all $y \in N$ such that $(x, y) \in \mathscr{A}$:

$$A(x) = \{y \in N | (x, y) \in \mathscr{A}\}. \tag{1.2}$$

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

Similarly, we let $B(x)$ ("before x") denote the set of all $y \in N$ such that $(y, x) \in \mathscr{A}$:

$$B(x) = \{y \in N \mid (y, x) \in \mathscr{A}\}. \tag{1.3}$$

For example, in the network of Fig. 1.1,

$$A(s) = \{x, y\}, \qquad B(s) = \varnothing \text{ (the empty set)}.$$

We shall on occasion require some other notions concerning networks. These will be introduced as needed.

2. Flows in networks

Given a network $G = [N; \mathscr{A}]$, suppose that each arc $(x, y) \in \mathscr{A}$ has associated with it a non-negative real number $c(x, y)$. We call $c(x, y)$ the *capacity* of the arc (x, y); it may be thought of intuitively as representing the maximal amount of some commodity that can arrive at y from x per unit time. The function c from $\mathscr{A}$ to non-negative reals is the *capacity function.* (Sometimes it will be convenient to allow infinite arc capacities also.)

The fundamental notion underlying most of the topics treated subsequently is that of a static or steady state flow through a network, which we now proceed to formulate. (Since dynamic flows will not be discussed until Chapter III, the qualifying phrase "static" or "steady state" will usually be omitted.)

Let s and t be two distinguished nodes of N. A *static flow of value v from s to t* in $[N; \mathscr{A}]$ is a function f from $\mathscr{A}$ to non-negative reals that satisfies the linear equations and inequalities

$$\sum_{y \in A(x)} f(x, y) - \sum_{y \in B(x)} f(y, x) = \begin{cases} v, & x = s, \\ 0, & x \neq s, t, \\ -v, & x = t, \end{cases} \tag{2.1}$$

$$f(x, y) \leqslant c(x, y) \qquad \text{all } (x, y) \in \mathscr{A}. \tag{2.2}$$

We call s the *source,* t the *sink,* and other nodes *intermediate.* Thus if the *net flow out of x* is defined to be

$$\sum_{y \in A(x)} f(x, y) - \sum_{y \in B(x)} f(y, x),$$

then the equations (2.1) may be verbalized by saying that the net flow out of the source is v, the net flow out of the sink is $-v$ (or the net flow into the sink is v), whereas the net flow out of an intermediate node is zero. An equation of the latter kind will be called a *conservation equation.*

When necessary to avoid ambiguity, we shall denote the value of a flow f by $v(f)$. Notice that a flow f from s to t of value v is a flow from t to s of value $-v$.

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

An example of a flow from s to t is shown in Fig. 2.1, where it is assumed that arc capacities are sufficiently large so that none are violated. The value of this flow is 3.

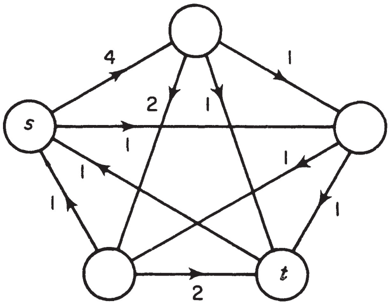

Figure 2.1

Given a flow f, we refer to $f(x, y)$ as the *arc flow* $f(x, y)$ or the *flow in arc* (x, y). Each arc flow $f(x, y)$ occurs in precisely two equations of (2.1), and has a coefficient 1 in the equation corresponding to node x, a coefficient -1 in the equation corresponding to node y. In other words, the coefficient matrix of equations (2.1), apart from the column corresponding to v, is the node-arc incidence matrix of the network. (By adding the special arc (t, s) to the network, allowing multiple arcs if necessary, a non-negative flow value v can be thought of as the "return flow" in (t, s), and all equations taken as conservation equations.)

A few observations. There is no question concerning the existence of flows, since $f = 0$, $v = 0$ satisfy (2.1) and (2.2). Also, while we have assumed that $\mathscr{A}$ may be a subset of the ordered pairs (x, y), $x \neq y$, with the capacity function c non-negative on $\mathscr{A}$, we could extend $\mathscr{A}$ to all ordered pairs by taking $c = 0$ outside of $\mathscr{A}$, or we could assume strict positivity of c by deleting from $\mathscr{A}$ arcs having zero capacity. Finally, the set of equations (2.1) is redundant, since adding the rows of its coefficient matrix produces the zero vector. Thus we could omit any one of the equations without loss of generality. We prefer, however, to retain the one-one correspondence between equations and nodes.

The static maximal flow problem is that of maximizing the variable v subject to the flow constraints (2.1) and (2.2). Before proceeding to this problem, it is worth while to point out an alternative formulation that is informative and will be useful in later contexts. This might be termed the arc-chain notion of a flow from s to t.

Suppose that $A_1, \ldots, A_m$ is an enumeration of the arcs of a network, the arc A_i having capacity $c(A_i)$; and let $C_1, \ldots, C_n$ be a list of all directed

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

chains from s to t. Form the m by n incidence matrix (a_{ij}) of arcs versus chains by defining

$$a_{ij} = \begin{cases} 1, & \text{if } A_i \in C_j, \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

Now let h be a function from the set of chains $C_1, \ldots, C_n$ to non-negative reals that satisfies the inequalities

$$\sum_{j=1}^{n} a_{ij} h(C_j) \leqslant c(A_i), \qquad i = 1, \ldots, m. \tag{2.4}$$

We refer to h as a *flow from s to t in arc-chain form*, and call $h(C_j)$ a *chain flow* or the *flow in chain C_j*. The *value* of h is

$$v(h) = \sum_{j=1}^{n} h(C_j). \tag{2.5}$$

When we need to distinguish the two notions of a flow from s to t thus far introduced, we shall call a function f from the set of arcs to non-negative reals which satisfies (2.1) and (2.2) for some v, a *flow from s to t in node-arc form*. There will usually be no need for the distinction, since we shall work almost exclusively with node-arc flows after this section.

Let us explore the relationship between these two formulations of the intuitive notion of a flow. Suppose that $x_1, \ldots, x_l$ is a list of the nodes, and let (b_{ki}), $k = 1, \ldots, l$, $i = 1, \ldots, m$, be the node-arc incidence matrix introduced earlier. Thus

$$b_{ki} = \begin{cases} 1, & \text{if } A_i = (x_k, y), \\ -1, & \text{if } A_i = (y, x_k), \\ 0, & \text{otherwise.} \end{cases} \tag{2.6}$$

Then

$$b_{ki} a_{ij} = \begin{cases} 1, & \text{if } A_i = (x_k, y) \text{ and } A_i \in C_j, \\ -1, & \text{if } A_i = (y, x_k) \text{ and } A_i \in C_j, \\ 0, & \text{otherwise,} \end{cases}$$

and it follows that

$$\sum_{i=1}^{m} b_{ki} a_{ij} = \begin{cases} 1, & \text{if } x_k = s, \\ -1, & \text{if } x_k = t, \\ 0, & \text{otherwise.} \end{cases} \tag{2.7}$$

If h is a flow from s to t in arc-chain form, and if we define

$$f(A_i) = \sum_{j=1}^{n} a_{ij} h(C_j), \qquad i = 1, \ldots, m, \tag{2.8}$$

then f is a flow from s to t in node-arc form, and $v(f) = v(h)$. For, by (2.4) and (2.8),

$$f(A_i) \leqslant c(A_i),$$

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

and by (2.7),

$$\begin{aligned}\sum_{i=1}^{m} b_{ki} f(A_i) &= \sum_{i=1}^{m} \sum_{j=1}^{n} b_{ki} a_{ij} h(C_j) \\ &= \sum_{j=1}^{n} \left(\sum_{i=1}^{m} b_{ki} a_{ij} \right) h(C_j) \\ &= \begin{cases} \sum_{j=1}^{n} h(C_j), & \text{if } x_k = s, \\ -\sum_{j=1}^{n} h(C_j), & \text{if } x_k = t, \\ 0, & \text{otherwise.} \end{cases}\end{aligned}$$

But these are precisely equations (2.1) for the function f and $v = \sum_{j=1}^{n} h(C_j)$.

On the other hand, we can start with a flow f in node-arc form having value v, and obtain from it a flow h in arc-chain form having value $v(h) \geqslant v$. Intuitively, the reason the inequality now appears is that the node-arc formulation permits flow along chains from t to s.

There are various ways of obtaining such an arc-chain flow h from a given node-arc flow f. One way is as follows. Define

$$(2.9) \qquad h(C_j) = \min_{A_i \in C_j} f_j(A_i), \qquad j = 1, \ldots, n,$$

where

$$(2.10) \qquad f_j(A_i) = f(A_i) - \sum_{p=1}^{j-1} a_{ip} h(C_p), \qquad j = 1, \ldots, n+1.$$

In words, look at the first chain C_1, reduce $f_1 = f$ by as much as possible (retaining non-negativity of arc flows) on arcs of C_1; this yields f_2. The process is then repeated with C_2 and f_2, and so on until all chains have been examined. It follows that f_{j+1} is a node-arc flow from s to t having value $v(f_{j+1}) = v - \sum_{p=1}^{j} h(C_p)$, since

$$\begin{aligned}\sum_{i=1}^{m} b_{ki} f_{j+1}(A_i) &= \sum_{i=1}^{m} b_{ki} f(A_i) - \sum_{i=1}^{m} \sum_{p=1}^{j} b_{ki} a_{ip} h(C_p), \\ &= \begin{cases} v - \sum_{p=1}^{j} h(C_p), & \text{if } x_k = s, \\ -v + \sum_{p=1}^{j} h(C_p), & \text{if } x_k = t, \\ 0, & \text{otherwise.} \end{cases}\end{aligned}$$

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

Moreover, $f_{j+1}(A_i) \leqslant f_j(A_i)$, all A_i, and $f_{j+1}(A_i) = 0$ for some $A_i \in C_j$. Hence the node-arc flow f_{n+1} vanishes on some arc of every chain from s to t. This implies that $v(f_{n+1}) \leqslant 0$, as the following lemma shows.

Lemma 2.1. *If f is a node-arc flow from s to t having value $v(f) > 0$, then there is a chain from s to t such that $f > 0$ on all arcs of this chain.*

Proof. Let X be the set of nodes defined recursively by the rules
(a) $s \in X$,
(b) if $x \in X$, and if $f(x, y) > 0$, then $y \in X$.
We assert that $t \in X$. For, suppose not. Then, summing the equations (2.1) over $x \in X$, and noting cancellations, we have

$$v(f) = \sum_{\substack{x \in X \\ y \notin X}} [f(x, y) - f(y, x)].$$

But by (b), if (x, y) is an arc with $x \in X$, $y \notin X$, then $f(x, y) = 0$. This and the last displayed equation contradict $v(f) > 0$. Thus $t \in X$. But for any $x \in X$, the definition of X shows that there is a chain from s to x such that $f > 0$ on arcs of this chain. Hence there is a chain from s to t with this property.

It follows from the lemma that the value of f_{n+1} is non-positive, that is

$$v(f_{n+1}) = v - \sum_{p=1}^{n} h(C_p) \leqslant 0.$$

Consequently $v(h) \geqslant v$. This proves

Theorem 2.2. *If h is an arc-chain flow from s to t, then f defined by* (2.8) *is a node-arc flow from s to t and $v(f) = v(h)$. On the other hand, if f is a node-arc flow from s to t, then h defined by* (2.9) *and* (2.10) *is an arc-chain flow from s to t, and $v(h) \geqslant v(f)$.*

A consequence of Theorem 2.2 is that it is immaterial whether the maximal flow problem is formulated in terms of the node-arc incidence matrix or the arc-chain incidence matrix. Thus, for example, since arcs of the form (x, s) or (t, x) can be deleted from $\mathscr{A}$ without changing the list of chains from s to t, we may always suppose in either formulation of the maximal flow problem that all source arcs point out from the source, and all sink arcs point into the sink. (For such networks, one has $v(h) = v(f)$ in the second part of Theorem 2.2 as well as the first part.)

A function h defined from f as in (2.9) and (2.10) will be termed a *chain decomposition* of f. A chain decomposition of f will, in general, depend on the ordering of the chains. For example, if in Fig. 1.1 we take $f = 1$ on all arcs, and take $C_1 = (s, x, t)$, $C_2 = (s, y, t)$, $C_3 = (s, x, y, t)$, $C_4 = (s, y, x, t)$, then $h(C_1) = h(C_2) = 1$, $h(C_3) = h(C_4) = 0$. But, examining the chains in reverse order would lead to $h(C_4) = h(C_3) = 1$, $h(C_2) = h(C_1) = 0$.

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

From the computational point of view, one would certainly suppose the node-arc formulation of the maximal flow problem to be preferable for most networks, since the number of chains from s to t is likely to be large compared to the number of nodes or the number of arcs. A computing procedure that required as a first step the enumeration of all chains from s to t would be of little value. There are less obvious reasons why the node-arc formulation is to be preferred from the theoretical point of view as well.†

3. Notation

To simplify the notation, we adopt the following conventions. If X and Y are subsets of N, let (X, Y) denote the set of all arcs that lead from $x \in X$ to $y \in Y$; and, for any function g from $\mathscr{A}$ to reals, let

$$\sum_{(x,y) \in (X,Y)} g(x, y) = g(X, Y). \tag{3.1}$$

Similarly, when dealing with a function h defined on the nodes of N, we put

$$\sum_{x \in X} h(x) = h(X). \tag{3.2}$$

We customarily denote a set consisting of one element by its single element. Thus if X contains the single node x, we write (x, Y), $g(x, Y)$, and so on.

Set unions, intersections, and differences will be denoted by $\cup$, $\cap$, and $-$, respectively. Thus $X \cup Y$ is the set of nodes in X or in Y, $X \cap Y$ the set of nodes in both X and Y, and $X - Y$ the set of nodes in X but not in Y. We use $\subseteq$ for set inclusion, and $\subset$ for proper inclusion. Complements of sets will be denoted by barring the appropriate symbol. For instance, the complement of X in N is $\bar{X} = N - X$.

Thus, if $X, Y, Z \subseteq N$, then

$$g(X, Y \cup Z) = g(X, Y) + g(X, Z) - g(X, Y \cap Z), \tag{3.3}$$

$$g(Y \cup Z, X) = g(Y, X) + g(Z, X) - g(Y \cap Z, X). \tag{3.4}$$

Hence if Y and Z are disjoint,

$$g(X, Y \cup Z) = g(X, Y) + g(X, Z),$$

$$g(Y \cup Z, X) = g(Y, X) + g(Z, X).$$

† Two comments are in order here. First, one can describe a computing procedure for the arc-chain formulation of the maximal flow problem that does not require an explicit enumeration of all chains [6]. Second, a strong theoretical reason for adopting the node-arc formulation, nonetheless, is that the node-arc incidence matrix has a desirable property not shared by the arc-chain incidence matrix. This is the unimodularity property, that is, every submatrix has determinant ± 1 or 0. See [12] for a full discussion of this property and its implications for linear programming problems.

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

Notice that

$$(B(x), x) = (N, x),$$
$$(x, A(x)) = (x, N),$$

and

$$g(N, X) = \sum_{x \in X} g(N, x) = \sum_{x \in X} g(B(x), x),$$

$$g(X, N) = \sum_{x \in X} g(x, N) = \sum_{x \in X} g(x, A(x)).$$

Later on (Chapter II) we shall use the notation $|X|$ to denote the number of elements in an arbitrary set X.

4. Cuts

Progress toward a solution of the maximal network flow problem is made with the recognition of the importance of certain subsets of arcs, which we shall call cuts. A *cut* $\mathscr{C}$ in $[N; \mathscr{A}]$ *separating* s *and* t is a set of arcs $(X, \bar{X})$ where $s \in X$, $t \in \bar{X}$. The *capacity* of the cut $(X, \bar{X})$ is $c(X, \bar{X})$.

For example, the set of arcs $\mathscr{C} = \{(s, y), (x, y), (x, t)\}$ with $X = \{s, x\}$, is a cut in the network of Fig. 1.1 separating s and t.

Notice that any chain from s to t must contain some arc of every cut $(X, \bar{X})$. For let $x_1, x_2, \ldots, x_n$ be a chain with $x_1 = s$, $x_n = t$. Since $x_1 \in X$, $x_n \in \bar{X}$, there is an x_i $(1 \leqslant i < n)$ with $x_i \in X$, $x_{i+1} \in \bar{X}$. Hence the arc (x_i, x_{i+1}) is a member of the cut $(X, \bar{X})$. It follows that if all arcs of a cut were deleted from the network, there would be no chain from s to t and the maximal flow value for the new network would be zero.

Since a cut blocks all chains from s to t, it is intuitively clear (and indeed obvious in the arc-chain version of the problem) that the value v of a flow f cannot exceed the capacity of any cut, a fact that we now prove from (2.1) and (2.2).

LEMMA 4.1. *Let f be a flow from s to t in a network $[N; \mathscr{A}]$, and let f have value v. If $(X, \bar{X})$ is a cut separating s and t, then*

$$v = f(X, \bar{X}) - f(\bar{X}, X) \leqslant c(X, \bar{X}). \tag{4.1}$$

PROOF. The equality of (4.1) was actually proved in Lemma 2.1. We re-prove it here, using the notation introduced in the preceding section.

Since f is a flow, f satisfies the equations

$$\begin{aligned} f(s, N) - f(N, s) &= v, & \\ f(x, N) - f(N, x) &= 0, & x \neq s, t, \\ f(t, N) - f(N, t) &= -v. & \end{aligned}$$

Now sum these equations over $x \in X$. Since $s \in X$ and $t \in \bar{X}$, the result is

$$v = \sum_{x \in X} (f(x, N) - f(N, x)) = f(X, N) - f(N, X).$$

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

Writing $N = X \cup \bar{X}$ in this equality yields

$$v = f(X, X \cup \bar{X}) - f(X \cup \bar{X}, X)$$
$$= f(X, X) + f(X, \bar{X}) - f(X, X) - f(\bar{X}, X),$$

thus verifying the equality in (4.1). Since $f(\bar{X}, X) \geqslant 0$ and $f(X, \bar{X}) \leqslant c(X, \bar{X})$ by virtue of (2.2), the inequality of (4.1) follows immediately.

In words, the equality of (4.1) states that the value of a flow from s to t is equal to the net flow across any cut separating s and t.

5. Maximal flow

We are now in a position to state and prove the fundamental result concerning maximal network flow [4, 5].

THEOREM 5.1. (Max-flow min-cut theorem.) *For any network the maximal flow value from s to t is equal to the minimal cut capacity of all cuts separating s and t.*

Before proving Theorem 5.1, we illustrate it with an example. Consider the network of Fig. 1.1 with capacity function c and flow f as indicated in Fig. 5.1, $c(x, y)$ being the first member of the pair of numbers written adjacent to arc (x, y), and $f(x, y)$ the second. Here the flow value is 3. Since the cut composed of arcs (s, x), (y, x), and (y, t) also has capacity 3, it follows from Lemma 4.1 that the flow is maximal and the cut minimal.

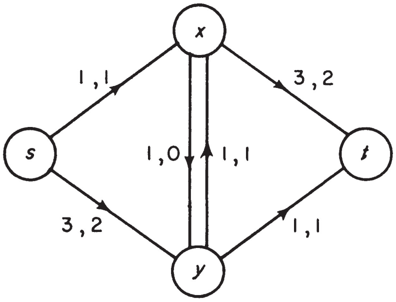

Figure 5.1

PROOF OF THEOREM 5.1. By Lemma 4.1, it suffices to establish the existence of a flow f and a cut $(X, \bar{X})$ for which equality of flow value and cut capacity holds. We do this by taking a maximal flow f (clearly such exists) and defining, in terms of f, a cut $(X, \bar{X})$ such that

$$f(X, \bar{X}) = c(X, \bar{X}),$$
$$f(\bar{X}, X) = 0,$$

so that equality holds throughout (4.1).

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

Thus, let f be a maximal flow. Using f, define the set X recursively as follows:

(a) $s \in X$;
(b) if $x \in X$ and $f(x, y) < c(x, y)$, then $y \in X$;
if $x \in X$ and $f(y, x) > 0$, then $y \in X$.

We assert that $t \in \overline{X}$. For, suppose not. It then follows from the definition of X that there is a path from s to t, say

$$s = x_1, x_2, \ldots, x_n = t,$$

having the property that for all forward arcs (x_1, x_{i+1}) of the path,

$$f(x_i, x_{i+1}) < c(x_i, x_{i+1}),$$

whereas for all reverse arcs (x_{i+1}, x_i) of the path,

$$f(x_{i+1}, x_i) > 0.$$

Let ε_1 be the minimum of $c - f$ taken over all forward arcs of the path, ε_2 the minimum of f taken over all reverse arcs, and let $\varepsilon = \min(\varepsilon_1, \varepsilon_2) > 0$. Now alter the flow f as follows: increase f by ε on all forward arcs of the path, and decrease f by ε on all reverse arcs. It is easily checked that the new function thus defined is a flow from s to t having value $v + \varepsilon$. But then f is not maximal, contrary to our assumption, and thus $t \in \overline{X}$.

Consequently $(X, \overline{X})$ is a cut separating s and t. Moreover, from the definition of X, it follows that

$$\begin{aligned} f(x, \bar{x}) &= c(x, \bar{x}) \qquad && \text{for } (x, \bar{x}) \in (X, \overline{X}), \\ f(\bar{x}, x) &= 0 && \text{for } (\bar{x}, x) \in (\overline{X}, X), \end{aligned}$$

since otherwise $\bar{x}$ would be in X. Thus

$$f(X, \overline{X}) = c(X, \overline{X}), \qquad f(\overline{X}, X) = 0,$$

so that equality holds in 4.1.

Several corollaries can be gleaned from Lemma 4.1, Theorem 5.1, and its proof.

We shall call a path from s to t a *flow augmenting path* with respect to a flow f provided that $f < c$ on forward arcs of the path, and $f > 0$ on reverse arcs of the path. Then we have

COROLLARY 5.2. *A flow f is maximal if and only if there is no flow augmenting path with respect to f.*

PROOF. If f is maximal, then clearly no flow augmenting path exists. Suppose, conversely, that no flow augmenting path exists. Then the set X defined recursively using f as in the proof of Theorem 5.1 cannot contain the sink t. Hence, as in the proof of Theorem 5.1, $(X, \overline{X})$ is a cut separating s and t having capacity equal to the value of f. Consequently f is maximal.

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

Corollary 5.2 is of fundamental importance in the study of network flows. It says, in essence, that in order to increase the value of a flow, it suffices to look for improvements of a very restricted kind.

We say that an arc (x, y) is *saturated* with respect to a flow f if $f(x, y) = c(x, y)$ and is *flowless* with respect to f if $f(x, y) = 0$. Thus an arc that is both saturated and flowless has zero capacity. Corollary 5.3 characterizes a minimal cut in terms of these notions.

COROLLARY 5.3. *A cut $(X, \overline{X})$ is minimal if and only if every maximal flow f saturates all arcs of $(X, \overline{X})$ whereas all arcs of $(\overline{X}, X)$ are flowless with respect to f.*

Using Corollary 5.3 it is easy to prove

COROLLARY 5.4. *Let $(X, \overline{X})$ and $(Y, \overline{Y})$ be minimal cuts. Then $(X \cup Y, \overline{X \cup Y})$ and $(X \cap Y, \overline{X \cap Y})$ are also minimal cuts.*

The following theorem shows that the minimal cut $(X, \overline{X})$ singled out in the proof of Theorem 5.1 does not, in actuality, depend on the maximal flow f.

THEOREM 5.5. *Let $(Y, \overline{Y})$ be any minimal cut, let f be a maximal flow, and let $(X, \overline{X})$ be the minimal cut defined relative to f in the proof of Theorem 5.1. Then $X \subseteq Y$.*

PROOF. Suppose that X is not included in Y. Then $X \cap Y \subset X$, and $(X \cap Y, \overline{X \cap Y})$ is a minimal cut by Corollary 5.4. Let x be a node in X that is not in $X \cap Y$. Since $x \in X$ and $x \neq s$, there is a path from s to x, say $s = x_1, x_2, \ldots, x_k = x$, such that each forward arc of the path is unsaturated with respect to f, while each reverse arc carries positive flow. But since $s \in X \cap Y$ and $x \in \overline{X \cap Y}$, there is a pair x_i, x_{i+1} $(1 \leqslant i < k)$ such that $x_i \in X \cap Y$, $x_{i+1} \in \overline{X \cap Y}$. If (x_i, x_{i+1}) is a forward arc of the path, then $f(x_i, x_{i+1}) < c(x_i, x_{i+1})$, contradicting Corollary 5.3. Similarly if (x_{i+1}, x_i) is a reverse arc of the path, Corollary 5.3 is contradicted. Hence $X \subseteq Y$.

Thus if $(X_i, \overline{X}_i)$, $i = 1, \ldots, m$, are all the minimal cuts separating source and sink, the set X defined relative to a particular maximal flow in the proof of Theorem 5.1 is the intersection of all X_i and hence does not depend on the selection of the flow.

Although the minimal cut $(X, \overline{X})$ was picked out in the proof of Theorem 5.1 by a recursive definition of the source set X, symmetrically we could have generated a minimal cut $(Y, \overline{Y})$ by defining its sink set $\overline{Y}$ in terms of a maximal flow f as follows:

(a′) $t \in \overline{Y}$;
(b′) if $y \in \overline{Y}$ and $f(x, y) < c(x, y)$, then $x \in \overline{Y}$;
if $y \in \overline{Y}$ and $f(y, x) > 0$, then $x \in \overline{Y}$.

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

Equivalently, one can think of reversing all arc orientations and arc flows, interchanging source and sink so that t becomes the source, s the sink, and then use the definition given in the proof of Theorem 5.1 to construct $\bar{Y}$. Again, although its definition is made relative to a particular maximal flow, the set $\bar{Y}$ does not actually depend on the selection, since $\bar{Y}$ is the intersection of the sink sets $\bar{X}_i$ of all minimal cuts $(X_i, \bar{X}_i)$.

Using both definitions, we can state a criterion for uniqueness of a minimal cut.

THEOREM 5.6. *Let X be the set of nodes defined in the proof of Theorem* 5.1, *let $\bar{Y}$ be the set defined above, and assume that c is strictly positive. The minimal cut $(X, \bar{X})$ is unique if and only if $(X, \bar{X}) = (Y, \bar{Y})$.*

PROOF. We must show that if $(X, \bar{X}) = (Y, \bar{Y})$, and if $(Z, \bar{Z})$ is any minimal cut, then $(X, \bar{X}) = (Z, \bar{Z})$.

First note that if $(X, \bar{X}) = (Y, \bar{Y})$, then both equal $(X, \bar{Y})$. For, $X \subseteq Y$ by Theorem 5.5, hence $(X, \bar{Y}) \subseteq (Y, \bar{Y})$. On the other hand, if $(u, v) \in (X, \bar{X}) = (Y, \bar{Y})$, then $u \in X$ and $v \in \bar{Y}$, so $(u, v) \in (X, \bar{Y})$.

For the arbitrary minimal cut $(Z, \bar{Z})$, we have, again by Theorem 5.5 and its analogue for $(Y, \bar{Y})$, that $X \subseteq Z$, $\bar{Y} \subseteq \bar{Z}$. Thus $(X, \bar{Y}) \subseteq (Z, \bar{Y}) \subseteq (Z, \bar{Z})$. Hence $c(X, \bar{Y}) \leqslant c(Z, \bar{Z})$. Now if $(X, \bar{Y}) \subset (Z, \bar{Z})$, then either some arcs of $(Z, \bar{Z})$ have zero capacity, contradicting our assumption $c > 0$, or $c(X, \bar{Y}) < c(Z, \bar{Z})$, contradicting the minimality of $(Z, \bar{Z})$. Thus $(X, \bar{X}) = (X, \bar{Y}) = (Z, \bar{Z})$.

Notice that Theorem 5.6 is not valid if the assumption $c > 0$ is relaxed to $c \geqslant 0$. For instance, in the network shown in Fig. 5.2, $X = \{s\}$, $\bar{Y} = \{t\}$,

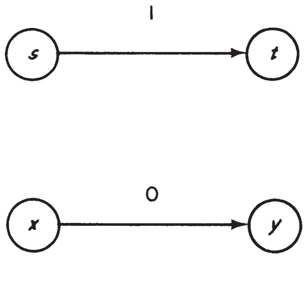

Figure 5.2

and $(X, \bar{X}) = (Y, \bar{Y}) = (s, t)$. However, $(Z, \bar{Z})$ with $Z = \{s, x\}$ is another minimal cut that contains both arcs.

6. Disconnecting sets and cuts

We have characterized cuts as sets of arcs of the form $(X, \bar{X})$ with $s \in X$, $t \in \bar{X}$, and have noted that a cut blocks all chains from s to t. Thus if we call a set of arcs a *disconnecting set* if it has the chain blocking property,

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

then a cut is a disconnecting set. The converse, however, is not necessarily true. For example, the set of all arcs in a network is a disconnecting set, but may not be a cut.

That every disconnecting set contains a cut can be seen easily as follows. Let $\mathscr{D}$ denote the disconnecting set, and define a subset X of nodes by the rule

(a) $s \in X$;

(b) if $x \in X$ and $(x, y) \in \mathscr{A} - \mathscr{D}$, then $y \in X$.

It is clear that $t \in \bar{X}$ and $(X, \bar{X}) \subseteq \mathscr{D}$. Notice that if $\mathscr{D}$ is a proper disconnecting set, that is, a disconnecting set whose proper subsets are not disconnecting, then $(X, \bar{X}) = \mathscr{D}$. Thus every proper disconnecting set is a cut. The converse may not hold, though. For example, in Fig. 5.2, the cut $(X, \bar{X})$ with $X = \{s, x\}$ is not a proper disconnecting set.

We may summarize the discussion thus far by saying:

(1) the class of proper disconnecting sets is included in the class of cuts, which, in turn, is included in the class of disconnecting sets, and that each of these inclusions may be proper;

(2) every disconnecting set contains a cut.

It follows that the notion of a cut could be replaced by either that of disconnecting set or proper disconnecting set in the statement of the max-flow min-cut theorem.

We have chosen to focus attention on cuts rather than disconnecting sets because the former are more convenient to work with when dealing with flows in node-arc form; the latter are convenient for an arc-chain formulation of the maximal flow problem. (See [4], where a proof of Theorem 5.1 which uses the arc-chain formulation is given.)

Notice that, in any case, restricting attention to proper disconnecting sets is as far as one can go in narrowing the class of sets of arcs that require consideration, since every proper disconnecting set of a network has minimal capacity for some capacity function: for instance, $c(x, y) = 1$ if $(x, y) \in \mathscr{D}$, $c(x, y) = \infty$ otherwise, singles out the proper disconnecting set $\mathscr{D}$ as the unique minimal cut.

7. Multiple sources and sinks

Although the assumption has been that the network has a single source and single sink, it is easy to see that the situation in which there are multiple sources and sinks, with flow permitted from any source to any sink, presents nothing new, since the adjunction of two new nodes and several arcs to the multiple source, multiple sink network reduces the problem to the case of a single source and sink.

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

In more detail, suppose that the nodes N of a network $[N; \mathscr{A}]$ are partitioned into three sets:

S (the set of sources),
T (the set of sinks),
R (the set of intermediate nodes),

and consider the problem of finding a maximal flow from S to T.

A flow from S to T may be thought of as a real valued function f defined on $\mathscr{A}$ that satisfies

$$f(x, N) - f(N, x) = 0 \qquad \text{for } x \in R, \tag{7.1}$$

$$0 \leqslant f(x, y) \leqslant c(x, y) \qquad \text{for } (x, y) \in \mathscr{A}, \tag{7.2}$$

the flow value being

$$v = f(S, N) - f(N, S). \tag{7.3}$$

Extend $[N; \mathscr{A}]$ to a network $[N^*; \mathscr{A}^*]$ by adjoining two nodes u, v and all arcs (u, S), (T, v), and extend the capacity function c defined on $\mathscr{A}$ to c^* defined on $\mathscr{A}^*$ by

$$\begin{aligned} c^*(u, x) &= \infty, && x \in S, \\ c^*(x, v) &= \infty, && x \in T, \\ c^*(x, y) &= c(x, y), && (x, y) \in \mathscr{A}. \end{aligned}$$

Thus the restriction f of a flow f^* from u to v in $[N^*; \mathscr{A}^*]$ is a flow from S to T in $[N; \mathscr{A}]$. *Vice versa,* a flow f from S to T in $[N; \mathscr{A}]$ can be extended to a flow f^* from u to v in $[N^*; \mathscr{A}^*]$ by defining

$$\begin{aligned} f^*(u, x) &= f(x, N) - f(N, x), && x \in S, \\ f^*(x, v) &= f(N, x) - f(x, N), && x \in T, \\ f^*(x, y) &= f(x, y), && \text{otherwise.} \end{aligned}$$

Consequently the maximal flow problem from S to T in $[N; \mathscr{A}]$ is equivalent to a single source, single sink problem in the extended network.

Relevant cuts for the case of many sources S and sinks T are those separating S and T: that is, a set of arcs $(X, \bar{X})$ with $S \subseteq X$, $T \subseteq \bar{X}$. Or, in terms of disconnecting sets, the appropriate notion would be a set of arcs that blocks all chains from S to T. The max-flow min-cut theorem and its corollaries, as well as the other theorems of § 5, remain valid, *mutatis mutandis,* as can be seen either from the equivalent extended problem or by making slight changes in the proofs throughout.

The situation in which there are several sources and sinks, but in which certain sources can "ship" only to certain sinks, is distinctly different. For such a problem, which might be thought of in terms of the simultaneous

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

flow of several commodities, the maximal flow value can be less than the minimal disconnecting set capacity. Here a disconnecting set means a collection of arcs that blocks all chains from sources to corresponding sinks. For example, consider the network shown in Fig. 7.1 with sources s_1, s_2, s_3, and sinks t_1, t_2, t_3. Each arc has unit capacity. Assume that s_i, t_i $(i = 1, 2, 3)$ are the source and sink for commodity i. Then the maximal flow value is 3/2, obtained by sending a half unit of commodity i along the unique chain from s_i to t_i. However, the arcs (x, y) and (y, z) are a minimal disconnecting set having capacity 2.

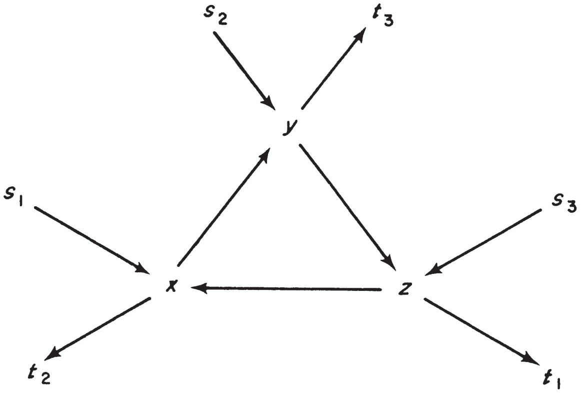

Figure 7.1

8. The labeling method for solving maximal flow problems

Under mild restrictions on the capacity function, the proof of the max-flow min-cut theorem given in § 5 provides a simple and efficient algorithm for constructing a maximal flow and minimal cut in a network [5].

The algorithm may be started with the zero flow. The computation then progresses by a sequence of "labelings" (*Routine* A below), each of which either results in a flow of higher value (*Routine* B below) or terminates with the conclusion that the present flow is maximal.

To ensure termination, it will be assumed that the capacity function c is integral valued. This is not an important restriction computationally, since a problem with rational arc capacities can be reduced to the case of integral capacities by clearing fractions, and of course, for computational purposes, confining attention to rational numbers is really no restriction.

Given an integral flow f, we proceed to assign labels to nodes of the network, a label having one of the forms (x^+, ε) or (x^-, ε), where $x \in N$ and ε is a positive integer or ∞, according to the rules delineated in *Routine* A.

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

During *Routine* A, a node is considered to be in one of three states: unlabeled, labeled and scanned, or labeled and unscanned. Initially all nodes are unlabeled.

Routine A (labeling process). First the source s receives the label $(-, \varepsilon(s) = \infty)$. (The source is now labeled and unscanned; all other nodes are unlabeled.) In general, select any labeled, unscanned node x. Suppose it is labeled $(z^{\pm}, \varepsilon(x))$. To all nodes y that are unlabeled, and such that $f(x, y) < c(x, y)$, assign the label $(x^{+}, \varepsilon(y))$, where

$$\varepsilon(y) = \min\,[\varepsilon(x), c(x, y) - f(x, y)]. \tag{8.1}$$

(Such y are now labeled and unscanned.) To all nodes y that are now unlabeled, and such that $f(y, x) > 0$, assign the label $(x^{-}, \varepsilon(y))$, where

$$\varepsilon(y) = \min\,[\varepsilon(x), f(y, x)]. \tag{8.2}$$

(Such y are now labeled and unscanned and x is now labeled and scanned.) Repeat the general step until either the sink t is labeled and unscanned, or until no more labels can be assigned and the sink is unlabeled. In the former case, go to *Routine* B; in the latter case, terminate.

Routine B (flow change). The sink t has been labeled $(y^{\pm}, \varepsilon(t))$. If t is labeled $(y^{+}, \varepsilon(t))$, replace $f(y, t)$ by $f(y, t) + \varepsilon(t)$; if t is labeled $(y^{-}, \varepsilon(t))$, replace $f(t, y)$ by $f(t, y) - \varepsilon(t)$. In either case, next turn attention to node y. In general, if y is labeled $(x^{+}, \varepsilon(y))$, replace $f(x, y)$ by $f(x, y) + \varepsilon(t)$, and if labeled $(x^{-}, \varepsilon(y))$, replace $f(y, x)$ by $f(y, x) - \varepsilon(t)$, and go on to node x. Stop the flow change when the source s is reached, discard the old labels, and go back to *Routine* A.

The labeling process is a systematic search for a flow augmenting path from s to t (Corollary 5.2). Enough information is carried along in the labels so that if the sink is labeled (henceforth we term this case *breakthrough*), the resulting flow change along the path can be made readily. If, on the other hand, *Routine* A ends and the sink has not been labeled (*nonbreakthrough*), the flow is maximal and the set of arcs leading from labeled to unlabeled nodes is a minimal cut, since the labeled nodes correspond to the set X defined in the proof of Theorem 5.1.

A main reason underlying the computational efficiency of the labeling process is that once a node is labeled and scanned it can be ignored for the remainder of the process. Labeling a node x corresponds to locating a path from s to x that can be the initial segment of a flow augmenting path. While there may be many such paths from s to x, finding one suffices.

If the flow f is integral and *Routine* A results in breakthrough, then the flow change $\varepsilon(t)$ of *Routine* B, being the minimum of positive integers, is a positive integer. Hence if the computation is initiated with an integral flow, each successive flow is integral. Consequently the algorithm is finite, since the flow value increases by at least one unit with each occurrence of

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

breakthrough; upon termination, a maximal flow has been constructed that is integral. Although this fact is a trivial consequence of the construction, the fact itself is important and will be used time and again in the solution of combinatorial problems. We therefore state it as a theorem.

THEOREM 8.1 (Integrity theorem). *If the capacity function c is integral valued, there exists a maximal flow f that is also integral valued.*

The following numerical example illustrates the use of the labeling method in constructing a maximal flow.

EXAMPLE. Let the given network be that of Fig. 1.1 with arc capacities and initial flow as indicated in Fig. 8.1, the pair $c(x, y), f(x, y)$ being written in that order adjacent to arc (x, y).

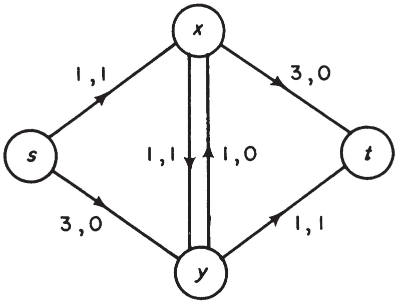

Figure 8.1

Start *Routine* A by assigning s the label $(-, \infty)$, see Fig. 8.2. From s,

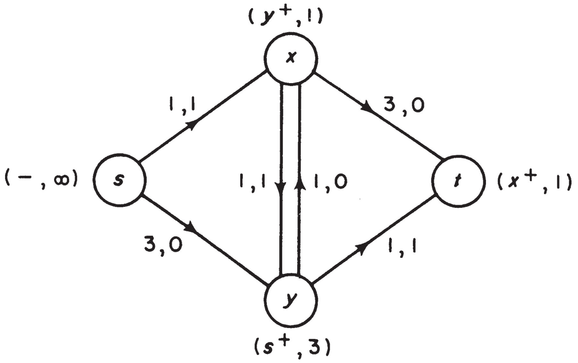

Figure 8.2

label y with $(s^+, \min(3, \infty)) = (s^+, 3)$, thus completing the labeling from s. From y, x can be labeled $(y^+, 1)$ (or $(y^-, 1)$), and is the only unlabeled node

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

that can be labeled from y. Again select a labeled, unscanned node (x is the only such), and continue assigning labels. This time breakthrough occurs: the sink t can be labeled (x^+, 1). This locates a flow augmenting path, found by backtracking from the sink according to the directions given in the labels, along which a flow change of $\varepsilon(t) = 1$ can be made. Here the path is the chain s, y, x, t. The new flow of value 2 is shown in Fig. 8.3.

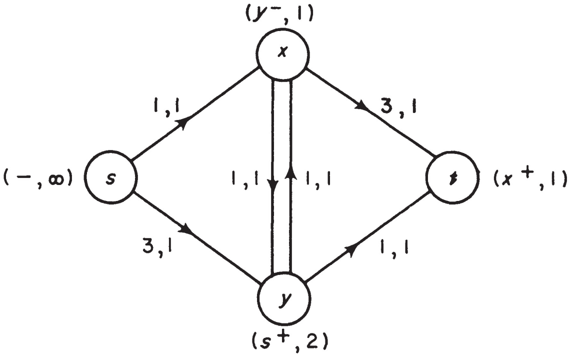

Figure 8.3

Now discard the old labels and repeat the labeling process. This time the labels shown in Fig. 8.3 are obtained. Again breakthrough has resulted and a flow improvement of $\varepsilon(t) = 1$ can be made along the path s, (s, y), y, (x, y), x, (x, t), t, yielding the flow shown in Fig. 8.4.

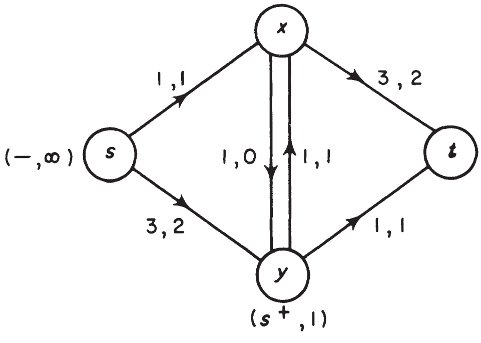

Figure 8.4

Repetition of *Routine* A now results in non-breakthrough, the labeled set of nodes being those shown in Fig. 8.4. Thus the flow of Fig. 8.4 is maximal and a minimal cut consists of the arcs (s, x), (y, x), and (y, t).

Labeling backward from the sink by rules corresponding to (a′), (b′) of

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

§ 5 locates the same cut, and hence by Theorem 5.6 this is the unique minimal cut separating s and t.

We conclude this section with an example indicating that the labeling process might fail to terminate if arc capacities are irrational. Specifically, the example shows that if the process is interpreted broadly enough to permit the selection of any flow augmenting path at each stage of the computation, then finite termination may not occur when arc capacities are irrational.

Before describing this example, we make one definition which will be helpful in the description. If $[N; \mathscr{A}]$ is a network with capacity function c, and if f is a flow from s to t in $[N; \mathscr{A}]$, then $c(x, y) - f(x, y)$ is the *residual capacity* of arc (x, y) with respect to f.

Now consider the recursion

$$a_{n+2} = a_n - a_{n+1}.$$

This recursion has a solution $a_n = r^n$, where $r = (-1 + \sqrt{5})/2 < 1$. Thus the series $\sum_{n=1}^{\infty} a_n$ converges to some sum S. We construct a directed network with four "special arcs"

$$\begin{aligned} A_1 &= (x_1, y_1), \\ A_2 &= (x_2, y_2), \\ A_3 &= (x_3, y_3), \\ A_4 &= (x_4, y_4), \end{aligned}$$

and the additional arcs (y_i, y_j), (x_i, y_j), (y_i, x_j), for $i \neq j$, together with source arcs (s, x_i) and sink arcs (y_i, t). The four special arcs have capacities a_0, a_1, a_2, a_2, respectively; all other arcs have capacity S.

Step 1. Find a chain from s to t that includes, from among the special arcs, only A_1, and impose a_0 units of flow in this chain. For example, take the chain s, x_1, y_1, t. (The special arcs now have residual capacities 0, a_1, a_2, a_2, respectively.)

Inductive step. Suppose the special arcs A_1', A_2', A_3', A_4' (some rearrangement of A_1, A_2, A_3, A_4) have residual capacities 0, a_n, a_{n+1}, a_{n+1}. Find a chain from s to t that includes, from among the special arcs, only A_2' and A_3', and impose a_{n+1} additional units of flow along this chain. For example, the chain s, x_2', y_2', x_3', y_3', t will do. (The special arcs now have residual capacities 0, $a_n - a_{n+1} = a_{n+2}$, 0, a_{n+1}.) Next find a path from s to t that contains A_2' as a forward arc, A_1' and A_3' as reverse arcs, the latter being the only reverse arcs of the path, and impose an additional flow of a_{n+2} units along this path. For example, the path s, x_2', y_2', y_1', x_1', y_3', x_3', y_4', t containing the reverse arcs (y_1', x_1'), (y_3', x_3') will do. (The special arcs now have residual capacities a_{n+2}, 0, a_{n+2}, a_{n+1}.)

The inductive step increases the flow value by $a_{n+1} + a_{n+2} = a_n$. Hence no non-special arc is ever required to carry more than $\sum_{n=1}^{\infty} a_n = S$ units

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

of flow in repeating the inductive step. The process converges to a flow having value S, whereas the maximal flow value for this network is $4S$.

9. Lower bounds on arc flows

Although lower bounds of zero have been assumed on all arc flows, there is no real necessity for this assumption in constructing maximal flows. If the conditions

$$0 \leqslant f(x, y) \leqslant c(x, y) \tag{9.1}$$

are replaced by

$$l(x, y) \leqslant f(x, y) \leqslant c(x, y), \tag{9.2}$$

where l is a given real valued function defined on arcs of $\mathscr{A}$ that satisfies

$$0 \leqslant l(x, y) \leqslant c(x, y), \tag{9.3}$$

the labeling process can be varied to handle this situation provided one has an initial flow to start the computation. There may be no function f satisfying the equations (2.1) and the inequalities (9.2) (e.g., take $l = c$ in the example of the preceding section), but assuming that these constraints are compatible for a given integral valued l and c, and that an initial f satisfying them has been found, the only change in the labeling rules for constructing a maximal flow is the following. If x has been labeled $(z^\pm, \varepsilon)$, then y may be labeled $[x^-, \min(\varepsilon, f(y, x) - l(y, x))]$ provided $f(y, x) > l(y, x)$.

It is also easy to see that the analogue of Theorem 5.1 becomes

THEOREM 9.1. *If there is a function f satisfying* (2.1) *and* (9.2) *for some number v, then the maximal value of v subject to these constraints is equal to the minimum of* $c(X, \bar{X}) - l(\bar{X}, X)$ *taken over all $X \subseteq N$ with $s \in X$, $t \in \bar{X}$.*

On the other hand, still assuming the existence of a function f satisfying (2.1) and (9.2) for some v, the minimal value of v may be found in a similar way: if x is labeled $(z^\pm, \varepsilon)$ and if $f(x, y) > l(x, y)$, attach the label $[x^-, \min(\varepsilon, f(x, y) - l(x, y))]$ to y; or if $f(y, x) < c(y, x)$, assign y the label $[x^+, \min(\varepsilon, c(y, x) - f(y, x))]$.

Here the analogue of Theorem 5.1 is

THEOREM 9.2. *If there is a function f satisfying* (2.1) *and* (9.2) *for some number v, the minimal value of v subject to these constraints is equal to the maximum of $l(X, \bar{X}) - c(\bar{X}, X)$ taken over all $X \subseteq N$ with $s \in X$, $t \in \bar{X}$.*

The questions that still remain are those of determining conditions under which the constraints (2.1) and (9.2) are compatible, and of constructing a function f satisfying them when these conditions hold. We postpone these questions for the moment. They, and similar questions, will be taken up in Chapter II.

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

10. Flows in undirected and mixed networks

Let us suppose that the network is undirected or mixed, and that each arc has a non-negative flow capacity. If the arc (x, y) is undirected with capacity $c(x, y)$, we intepret this to mean that

$$\begin{aligned} f(x, y) &\leqslant c(x, y), \\ f(y, x) &\leqslant c(x, y), \\ f(x, y) \cdot f(y, x) &= 0. \end{aligned} \tag{10.1}$$

That is, $f(x, y)$ is the flow from x to y in (x, y), and the arc (x, y) has a flow capacity $c(x, y)$ in either direction, but flow is permitted in only one of the two directions.

For example, one might think of a network of city streets, each street having a traffic flow capacity, and ask the question: how should one-way signs be put up on streets not already oriented in order to permit the largest traffic flow from some set of points to another?

At first glance, it might appear that this problem would involve examination of a large number of maximal flow problems obtained by orienting the network in various ways. But a moment's thought shows that the problem can be solved by considering only one directed network: namely, that obtained by replacing each undirected arc with a pair of oppositely directed arcs, each having capacity equal to the old arc. The reason for this is that, given any solution f, v of the flow constraints (2.1) and (2.2), one can produce a solution f', v in which

$$f'(x, y) \cdot f'(y, x) = 0$$

by taking

$$f'(x, y) = \max\,(0, f(x, y) - f(y, x)). \tag{10.2}$$

In words, we can cancel arc flows in opposite directions.

Thus, since it is clear that the maximal flow value for any specific orientation of the given network is no greater than the maximal flow value obtained by replacing each undirected arc by a pair of directed arcs, allowing both orientations for each undirected arc solves the original problem of maximizing v subject to the flow equations (2.1), capacity constraints (2.2) for directed arcs, and constraints (10.1) for undirected arcs.

11. Node capacities and other extensions

Other kinds of inequality constraints in addition to bounds on arc flows can be imposed without altering the character of the maximal flow problem. For instance, suppose that each node x has a flow capacity $k(x) \geqslant 0$, and that it is desired to find a maximal flow from s to t subject to both arc and node capacities.

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

More explicitly, let us assume that all source arcs are directed from the source and all sink arcs into the sink, and that it is desired to maximize $f(s, N)$ subject to

$$f(x, N) - f(N, x) = 0, \qquad x \neq s, t, \tag{11.1}$$

$$0 \leqslant f(x, y) \leqslant c(x, y), \qquad (x, y) \in \mathscr{A}, \tag{11.2}$$

$$f(x, N) \leqslant k(x), \qquad x \neq t, \tag{11.3}$$

$$f(N, t) \leqslant k(t). \tag{11.4}$$

This problem can be reduced to the arc capacity case by a simple device. Define a new network $[N^*; \mathscr{A}^*]$ from $[N; \mathscr{A}]$ as follows. To each $x \in N$ we make correspond two nodes $x', x'' \in N^*$; if $(x, y) \in \mathscr{A}$, then $(x', y'') \in \mathscr{A}^*$; in addition, $(x'', x') \in \mathscr{A}^*$ for each $x \in N$. The (arc) capacity function defined on $\mathscr{A}^*$ is

$$c^*(x', y'') = c(x, y), \qquad (x, y) \in \mathscr{A}, \tag{11.5}$$

$$c^*(x'', x') = k(x), \qquad x \in N. \tag{11.6}$$

Thus, for example, if the given network $[N; \mathscr{A}]$ is that of Fig. 11.1, the network $[N^*; \mathscr{A}^*]$ is shown in Fig. 11.2.

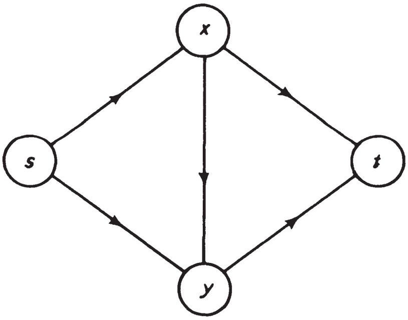

Figure 11.1

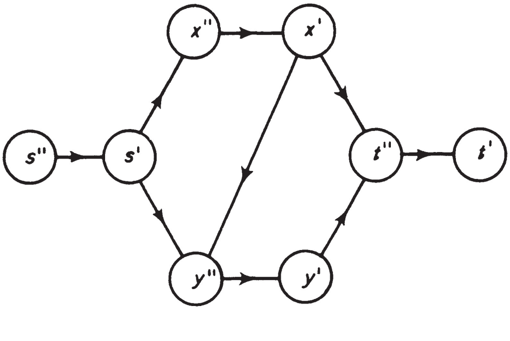

Figure 11.2

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

In effect, each node x has been split into two parts, a "left" part x'' and a "right" part x', so that all arcs entering x now enter its left part, whereas all arcs leaving x now leave its right part. The capacity $k(x)$ is then imposed as an arc capacity on the new arc leading from the left part of x to its right part.

Thus any function f satisfying (11.1)–(11.4), that is, any flow from s to t in $[N; \mathscr{A}]$ that does not exceed the node capacities, yields an equivalent flow f^* from s'' to t' in $[N^*; \mathscr{A}^*]$ by defining

$$f^*(x', y'') = f(x, y), \qquad (x, y) \in \mathscr{A}, \tag{11.7}$$

$$f^*(x'', x') = f(x, N), \qquad x \neq t, \tag{11.8}$$

$$f^*(t'', t') = f(N, t), \tag{11.9}$$

and conversely.

If the notion of a disconnecting set is extended to include nodes as well as arcs, the analogue of the max-flow min-cut theorem asserts that the maximal flow value is equal to the capacity of a disconnecting set of nodes and arcs having minimal capacity.

In a similar way, more general kinds of constraints on the flow out of or into node x can be reduced to the case of arc capacities by enlarging the network. For example, suppose that the nodes of the set $A(x)$ are put into subsets

$$A_1(x), \ldots, A_{m(x)}(x) \tag{11.10}$$

with the proviso that

$$A_i(x) \cap A_j(x) \neq \varnothing \Rightarrow A_i(x) \subseteq A_j(x) \quad \text{or} \quad A_j(x) \subseteq A_i(x), \tag{11.11}$$

and assume, in addition to the flow equations,

$$f(x, A_i(x)) \leqslant k_i(x), \qquad i = 1, \ldots, m(x). \tag{11.12}$$

Constraints of the form (11.12), under the assumption (11.11), can be handled as indicated schematically in Fig. 11.3 and Fig. 11.4 for a single node x.

Constraints of a similar kind on flow into x can be reduced to arc constraints by enlarging the network in an analogous fashion.

Notice that inequality constraints (11.2), (11.3), (11.4) are a special case of (11.12) and similar constraints on flow into x:

$$f(B_j(x), x) \leqslant h_j(x), \qquad j = 1, \ldots, n(x). \tag{11.13}$$

If we refer to each set $(x, A_i(x))$ and $(B_j(x), x)$ as an elementary set of arcs, and extend the notion of a disconnecting set of arcs to say that a collection $\mathscr{B}$ of elementary sets is a disconnecting collection if each chain from s to t has an arc in common with some elementary set contained in $\mathscr{B}$, it can be shown that the maximal flow value from s to t is equal to the

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

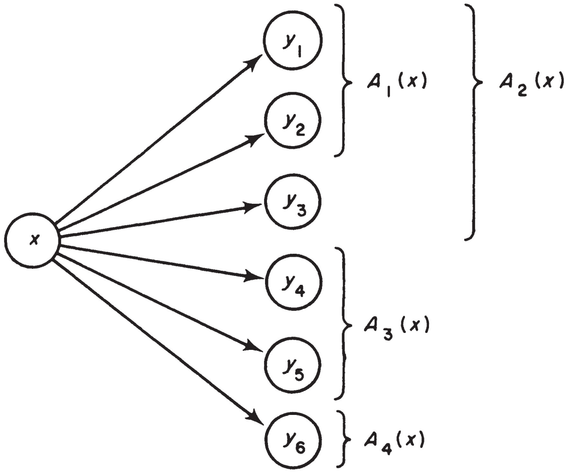

Figure 11.3

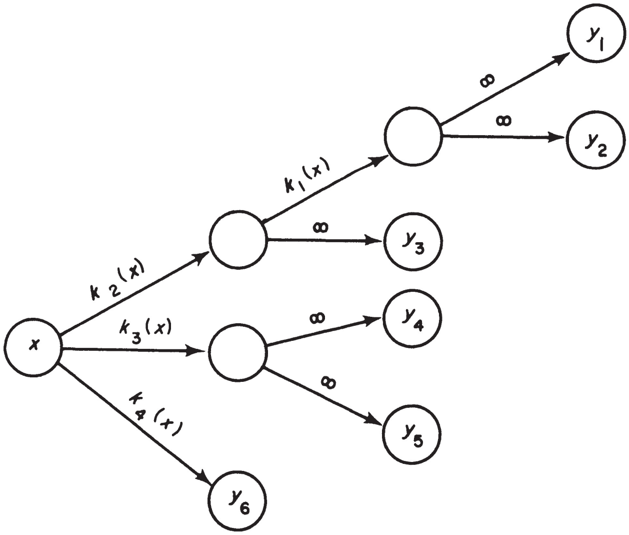

Figure 11.4

minimal blocking capacity (under the assumption (11.11) and a similar assumption on $B_j(x)$).

12. Linear programming and duality principles

The problem of finding a maximal flow through a network, whether stated in node-arc or in arc-chain form, is one of extremizing a linear

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

function subject to linear equations and linear inequalities. Such a problem is called a linear programming problem. There are various known methods of computing answers to linear programs. The method that is in general use is G. B. Dantzig's simplex algorithm, around which a sizeable literature has already grown up. It is not our purpose here to discuss the theory of linear inequalities or algorithms for solving general linear programs, since this book is devoted, for the most part, to special kinds of linear programs that arise in transportation, communication, or certain kinds of combinatorial problems, and to a presentation of special algorithms for solving these linear programs. We would be negligent, however, if some mention were not made of linear programming duality principles in connection with these problems.

Associated with every linear programming problem in variables $w_1, \ldots, w_n$:

$$(12.1)\qquad \begin{aligned} &a_{11}w_1 + \ldots + a_{1l}w_l + a_{1,l+1}w_{l+1} + \ldots + a_{1n}w_n = b_1 \\ &\qquad\qquad\qquad\ldots \\ &a_{k1}w_1 + \ldots + a_{kl}w_l + a_{k,l+1}w_{l+1} + \ldots + a_{kn}w_n = b_k \\ &a_{k+1,1}w_1 + \ldots + a_{k+1,l}w_l + a_{k+1,l+1}w_{l+1} + \ldots + a_{k+1,n}w_n \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \leqslant b_{k+1} \\ &\qquad\qquad\qquad\ldots \\ &a_{m1}w_1 + \ldots + a_{ml}w_l + a_{m,l+1}w_{l+1} + \ldots + a_{mn}w_n \leqslant b_m \end{aligned}$$

$$(12.2)\qquad w_1, \ldots, w_l \text{ unrestricted in sign; } w_{l+1}, \ldots, w_n \geqslant 0$$

$$(12.3)\qquad \text{maximize } c_1w_1 + \ldots + c_nw_n$$

is a *dual program* obtained by assigning multipliers $\lambda_1, \ldots, \lambda_m$ to the individual constraints of (12.1) and forming the program

$$(12.4)\qquad \begin{aligned} &a_{11}\lambda_1 + \ldots + a_{k1}\lambda_k + a_{k+1,\,1}\lambda_{k+1} + \ldots + a_{m1}\lambda_m = c_1 \\ &\qquad\qquad\qquad\ldots \\ &a_{1l}\lambda_1 + \ldots + a_{kl}\lambda_k + a_{k+1,\,l}\lambda_{k+1} + \ldots + a_{ml}\lambda_m = c_l \\ &a_{1,\,l+1}\lambda_1 + \ldots + a_{k,\,l+1}\lambda_k + a_{k+1,\,l+1}\lambda_{k+1} + \ldots + a_{m,\,l+1}\lambda_m \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \geqslant c_{l+1} \\ &\qquad\qquad\qquad\ldots \\ &a_{1n}\lambda_1 + \ldots + a_{kn}\lambda_k + a_{k+1,\,n}\lambda_{k+1} + \ldots + a_{mn}\lambda_m \geqslant c_m \end{aligned}$$

$$(12.5)\qquad \lambda_1, \ldots, \lambda_k \text{ unrestricted in sign; } \lambda_{k+1}, \ldots, \lambda_m \geqslant 0$$

$$(12.6)\qquad \text{minimize } b_1\lambda_1 + \ldots + b_m\lambda_m.$$

Here the a_{ij}, b_i, and c_j are given real numbers.

The matrix of the constraints (12.4) is the transpose of that of (12.1). Equalities of (12.4) correspond to unrestricted variables $w_1, \ldots, w_l$, and

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

inequalities to non-negative variables $w_{l+1}, \ldots, w_n$. The multipliers or dual variables $\lambda_1, \ldots, \lambda_k$ that correspond to equations of (12.1) are unrestricted in sign, whereas $\lambda_{k+1}, \ldots, \lambda_m$, corresponding to inequalities of (12.1), are non-negative.

Observe that if the dual problem (12.4), (12.5), and (12.6) is written in the form of the *primal problem* (12.1), (12.2), and (12.3), by multiplying each of the constraints of (12.4) by -1 and maximizing $-\sum b_i\lambda_i$, then the dual of (12.4), (12.5), (12.6) is (12.1), (12.2), (12.3). In other words, the dual of the dual is the primal.

The constraints of the primal problem are said to be *feasible* if there is a vector $w = (w_1, \ldots, w_n)$ satisfying them; w is then called a *feasible vector*, and the primal problem is termed *feasible*. A feasible vector w that maximizes the linear form $\sum c_j w_j$ is called *optimal*. Analogous language is used for the dual problem.

Thus a linear programming problem either has

(a) optimal (and hence feasible) vectors;

(b) feasible vectors, but no optimal vector;

(c) no feasible vectors.

The fundamental duality theorem of linear programming [9] relates the way these situations can occur in a pair of dual programs, and asserts equality between the maximum in the primal and the minimum in the dual: if case (a) holds for the primal, then (a) holds for its dual and the maximum value of $\sum c_j w_j$ is equal to the minimum value of $\sum b_i\lambda_i$; if (b) holds for the primal, then (c) holds for the dual; if (c) holds for the primal, either (b) or (c) is valid for the dual.

That the maximum value of $\sum c_j w_j$ is no greater than the minimum of $\sum b_i\lambda_i$ if both primal and dual have feasible vectors is easily seen. Letting w and λ be feasible in their respective programs, it follows that

$$\sum_j c_j w_j \leqslant \sum_j \sum_i \lambda_i a_{ij} w_j, \tag{12.7}$$

since unrestricted variables w_j correspond to equations $\sum_i \lambda_i a_{ij} = c_j$ and non-negative variables w_j to inequalities $\sum_i \lambda_i a_{ij} \geqslant c_j$.

Thus equality holds in (12.7) if and only if

$$\sum_i \lambda_i a_{ij} > c_j \Rightarrow w_j = 0. \tag{12.8}$$

Similarly,

$$\sum_j \sum_i \lambda_i a_{ij} w_j \leqslant \sum_i b_i \lambda_i, \tag{12.9}$$

since the λ_i that are unrestricted in sign correspond to equations $\sum_j a_{ij} w_j = b_i$, whereas non-negative λ_i correspond to inequalities $\sum_j a_{ij} w_j \leqslant b_i$.

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

Thus, equality holds in (12.9) if and only if

$$\lambda_i > 0 \Rightarrow \sum_j a_{ij} w_j = b_i. \tag{12.10}$$

Consequently

$$\sum c_j w_j \leqslant \sum b_i \lambda_i, \tag{12.11}$$

equality holding if and only if (12.8) and (12.10) are valid. The major content of the duality theorem is the assertion that if case (a) holds for the primal, it also holds for the dual, and that there are then feasible solutions to primal and dual problems that satisfy the optimality criteria (12.8) and (12.10).

Our purpose in giving this sketchy résumé of linear programming duality theory is twofold. First, we shall note that the max-flow min-cut theorem provides a proof of the duality theorem for the special case of maximal flow problems. Second, although the algorithms to be presented subsequently do not require appeal to the duality theorem, they were motivated by duality considerations, and we want to feel free to invoke such considerations where convenient.

If we take the constraints of the maximal flow problem in the node-arc form and assign multipliers $\pi(x)$ to the equations (2.1), multipliers $\gamma(x, y)$ to the capacity inequalities (2.2), then, recalling that the coefficient matrix of the equations is (apart from the column corresponding to the variable v) the node-arc incidence matrix of the network, it follows that the dual has constraints

$$\begin{aligned} -\pi(s) + \pi(t) &\geqslant 1, & \\ \pi(x) - \pi(y) + \gamma(x, y) &\geqslant 0, & \text{all } (x, y), \\ \gamma(x, y) &\geqslant 0, & \text{all } (x, y), \end{aligned} \tag{12.12}$$

subject to which the form

$$\sum_{\mathscr{A}} c(x, y)\gamma(x, y) \tag{12.13}$$

is to be minimized. In (12.12), the first constraint comes from the v-column of the primal problem, the second from the (x, y)-column. The dual variables $\pi(x)$ are unrestricted in sign since they correspond to equations, whereas the variables $\gamma(x, y)$ correspond to inequalities and are consequently non-negative.

If $(X, \bar{X})$ is a minimal cut separating s and t, it can be checked that an optimal solution to the dual problem is provided by taking

$$\pi(x) = \begin{cases} 0 & \text{for } x \in X, \\ 1 & \text{for } x \in \bar{X}, \end{cases} \tag{12.14}$$

$$\gamma(x, y) = \begin{cases} 1 & \text{for } (x, y) \in (X, \bar{X}), \\ 0 & \text{otherwise.} \end{cases} \tag{12.15}$$

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

This follows since (12.14) and (12.15) define a feasible solution to the dual that produces equality between the primal form v and dual form (12.13). Or one can check the optimality properties (12.8) and (12.10).

In particular, the dual of the maximal flow problem always has an integral solution. It can be shown, in fact, that all extreme points of the convex polyhedral set defined by setting $\pi(s) = 0$ in (12.12), which corresponds to dropping the (redundant) source equation in the primal problem, are of the form given in (12.14) and (12.15) for some X with $s \in X$. Using this fact, the max-flow min-cut theorem can be deduced from the duality theorem [2].

13. Maximal flow value as a function of two arc capacities

For a given network $[N; \mathscr{A}]$ with specified sources S and sinks T, the value $\bar{v}$ of a maximal flow from S to T is solely a function of the individual arc capacities. Indeed, if $\mathscr{A} = \{A_1, A_2, \ldots, A_m\}$ and A_i has capacity $c(A_i)$, we know that

$$(13.1) \qquad \bar{v} = \min_{\mathscr{C} \subseteq \mathscr{A}} \sum_{A_i \in \mathscr{C}} c(A_i),$$

the minimum being taken over all cuts $\mathscr{C}$ separating S and T. The theorems and proofs of this section provide insight into the behavior of $\bar{v}$ considered as a function of two arc capacities, everything else being held fixed. Both theorems and proofs are due to Shapley [16].

It will be convenient to allow infinite capacities for the two arcs in question, and hence infinite $\bar{v}$. However, the capacities of other arcs are assumed finite.

Let $\bar{v}_i(\xi)$ denote the maximal flow value when the capacity $c(A_i)$ has been replaced by the non-negative variable ξ. Similarly, $\bar{v}_{ij}(\xi, \eta)$ denotes the maximal flow value when $c(A_i)$ and $c(A_j)$ have been replaced by non-negative variables ξ and η. It is a consequence of (13.1) that

$$(13.2) \qquad \bar{v}_i(\xi) = \min\,[\bar{v}_i(0) + \xi, \bar{v}_i(\infty)].$$

In more detail, if ξ is less than the *critical capacity*

$$(13.3) \qquad \xi^* = \bar{v}_i(\infty) - \bar{v}_i(0),$$

the arc A_i is a member of every minimal cut, whereas for $\xi > \xi^*$, the arc A_i is in no minimal cut. Here ξ^* may be either zero or infinite. If the critical capacity ξ^* is strictly positive, and if $c(A_i) = \xi^*$, there is a minimal cut containing A_i and a minimal cut not containing A_i.

Two applications of (13.2) yield

$$(13.4) \qquad \bar{v}_{ij}(\xi, \eta) = \min\,[\bar{v}_{ij}(0, 0) + \xi + \eta,\ \bar{v}_{ij}(0, \infty) + \xi,\ \bar{v}_{ij}(\infty, 0) + \eta,\ \bar{v}_{ij}(\infty, \infty)].$$

(continued...)

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

INDEX

For general queries, contact info@press.princeton.edu

© Copyright, Princeton University Press. No part of this book may be distributed, posted, or reproduced in any form by digital or mechanical means without prior written permission of the publisher.

INDEX

For general queries, contact info@press.princeton.edu