CHAPTER 2

Connectivity and Network Flows

András FRANK

Department of Computer Science, Eötvös University, Múzeum krt. 6–8, Budapest H-1088, Hungary

Contents

HANDBOOK OF COMBINATORICS
Edited by R. Graham, M. Grötschel and L. Lovász

1. Introduction, preliminaries

Intuitively, a graph is felt to be connected if there is no way to separate it into two parts with no connection between the two parts. Or, equivalently, for any two nodes of the graph there is a path connecting them.

If one is interested in various properties of graphs, it is often useful to dismantle the graph into connected components and then investigate those components separately. For example, to decide whether a graph is k-colorable it suffices to deal with the connected components. A similar idea works for higher connectivity, as well. Namely, we separate first the graph into highly connected parts, then establish properties of these parts, and finally, using these properties we try to obtain information about the whole graph. For example, to prove Kuratowski's theorem on plane representation of graphs one can assume first that the graph is 3-connected since otherwise the graph can be decomposed along a 2-separation and the planar representations of the smaller parts can be pieced together. Second, by exploiting stronger properties of 3-connected graphs one can more easily deduce Kuratowski's theorem for 3-connected graphs. (See Thomassen 1980b.) (We remark that the same program proved extremely useful for matroids, as well.)

But what does "higher connectivity" mean? Intuitively, one may have at least two possible definitions for k-connectivity. First, the graph is not only connected but remains so after deleting any set of at most $k-1$ nodes. Second, a graph is k-connected if there are k (openly) disjoint paths between any two of its nodes. Fortunately, by a theorem of Whitney, these seemingly different concepts coincide. Whitney's (1932) theorem is an easy consequence of what can be considered the fundamental result of this whole chapter: Menger's theorem. It states that either there are k openly disjoint paths between two specified nodes or there is a set of less than k elements separating these two nodes.

Another source of the theory to be surveyed here is network flows. Its basic result, the max-flow min-cut theorem, can be considered as a capacitated (and directed) counterpart of Menger's theorem. Network flow theory is a systematic treatment of combinatorial optimization problems. It found a great number of applications in practice as well as in other branches of mathematics.

In this chapter we try to provide a rather comprehensive overview of results belonging to this area. Certain (mostly easier) parts are discussed in greater detail in order to give some hints on the general techniques used. Other parts are more difficult so we confine ourselves to give a general framework filled with various results but no proofs. In some cases, however, when it did not need too much space, proofs of some deep theorems (e.g., Tutte's wheel theorem or Nash-Williams' theorem on covering trees) have been included.

Throughout the chapter we use the following notation. Let V be a finite set and s, t elements of V. A subset X of V is called an $s\bar{t}$-set if $s \in X \subseteq V - t$. We often do not distinguish between a one-element set and its single element. Let $D = (V, A)$ be a digraph. The elements of A are called directed edges or arcs. For $S \subseteq V$ let $\Delta^+(S)$ denote the set of arcs with tail in S and head in $V - S$. We use the notation

$\Delta^-(S) := \Delta^+(V-S)$, $\delta^+(S) =: |\Delta^+(S)|$, $\delta^-(S) := \delta^+(V-S)$. For a graph or digraph G and vector $x: A \to \mathbb{R}$, let $d_x(X,Y)$ denote the sum $\sum (x(a)$: $a \in A$, one end of a is in $X - Y$, the other end is in $Y - X)$. When $x \equiv 1$ we use $d(X, Y)$ for $d_x(X,Y)$.

If it is not ambiguous we will use the notation $e = uv$ for an edge with endpoints u and v. Similarly $e = (u, v)$ stands for an arc with tail u and head v. An edge uv or an arc (u, v) is said to leave (enter) S if S is a $u\bar{v}$-set ($v\bar{u}$-set).

To complete this introductory section let us draw attention to some other survey papers and books. Concerning connectivity two books of Tutte (1966, 1984) deserve special mentioning. Mader's survey paper (1979) includes a long list of connectivity results and their relationship. Bollobás' (1978) book is also an excellent reference that includes the proof of many difficult theorems.

As far as network flows are concerned the classical book of Ford and Fulkerson (1962) is even today a refreshing reading. Recently, Ahuja et al. (1993) provided a comprehensive book on network flow techniques. Another useful survey on network flow theory, given by Goldberg et al. (1990), appeared in a book entitled "Paths, Flows, and VLSI-Layout" (B. Korte et al., eds., Springer, 1990). This book includes other important surveys concerning connectivity results. One of them, due to A. Schrijver, is concerned with the homotopic paths packing problem. The survey of N. Robertson and P.D. Seymour outlines the authors' very complex disjoint paths method. A third survey paper from the same book, written by the present author, provides an overview on packing paths, circuits, and cuts.

2. Reachability

2.1. Paths and walks

In a graph by a *walk* W we mean an alternating sequence $(v_0, e_1, v_1, e_2, \ldots, e_n, v_n)$ consisting of nodes and edges where e_i is an edge between v_{i-1} to v_i. The nodes v_0 and v_n are called the endpoints of W. Sometimes we say that W is a walk between v_0 and v_n or that W is a walk from v_0 to v_n (or from v_n to v_0). In a digraph by a (*directed*) *walk* W we mean an alternating sequence $(v_0, e_1, v_1, e_2, \ldots, e_k, v_k)$ where e_i is an arc from v_{i-1} to v_i. We say that W is a walk from v_0 to v_n or that v_n is reachable from v_0 by W.

The following definitions concern both graphs and digraphs. The number k of edges of a walk is called the *length* of the walk. The *distance* of t from s is the minimum length of a path from s to t. Obviously, if W_1 is a walk (directed or undirected) from u to v and W_2 is a walk from v to w, then the concatenation $W = W_1W_2$ is a walk from u to w.

We say that a walk is *simple* if all its defining terms are distinct. A simple walk is called a *path*. A walk is called *closed* if its endpoints coincide. If otherwise the terms of a closed walk are distinct, it is called a *circuit*.

Let $W = (v_0, e_1, \ldots, e_k, v_k)$ be a walk. Suppose that $v_i = v_j$ for some $i, j, 0 \leq i < j \leq k$, and the subsequence $C = (v_i, e_{i+1}, \ldots, e_{j-1}, v_j)$ is a circuit. *Reducing* W by circuit C means that we define a new walk $W' := (v_0, e_1, \ldots, v_i, e_{j+1}, \ldots, e_k, v_k)$. *Simplifying* W means that one reduces W as long as possible. The final walk is a path from v_0 to v_k (that may depend on the order of reductions). Thus we have the following.

Proposition 2.1. (a) *In a graph if there is a walk between two nodes u and v, there is a path between u and v.*

(b) *In a digraph if there is a directed walk from u to v, there is a directed path from u to v.*

Let us call two nodes of a graph $G = (V, E)$ equivalent if there is a path connecting them. This is an equivalence relation: from the definition of path it is symmetric and reflexive, by Proposition 2.1 it is transitive. An equivalence class is called a *component* of G. If G has exactly one equivalence class, G is called *connected*. Equivalently, G is *connected* if there is a path between any two of its nodes.

We call two nodes u and v of a digraph $D = (V, A)$ equivalent if there is a directed path from u to v and one from v to u. This is again an equivalence relation. An equivalence class is called a *strong component*. If D has exactly one equivalence class, D is called *strongly connected*. Equivalently, a digraph D is *strongly connected* if there is a directed path from every node to any other.

Proposition 2.2. *Let s and t be two specified nodes of a digraph $D = (V, A)$. There is a directed path from s to t if and only if for all $s\bar{t}$-sets S there is an edge leaving S.*

Proof. The necessity is straightforward. The sufficiency follows by observing that, if there is no path from s to t, the set S of nodes reachable from s has no leaving edges. □

Let us introduce some further notions. An undirected graph is called a *tree* if it is connected but deleting any of its edges disconnects the graph.

Proposition 2.3. *For a graph $G = (V, E)$ the following are equivalent:*

(a) *G is a tree.*

(b) *G is a connected graph containing no circuit.*

(c) *In G there is a unique path between any pair of nodes.*

(d) *G is connected and $|E| = |V| - 1$.*

(e) *G can be built up from any of its nodes by consecutively adjoining edges so that one end of the currently added edge belongs to the graph having already been constructed while the other endpoint does not.*

A graph is called a *forest* if each of its components is a tree. A digraph $D = (V, A)$ is called an *arborescence* if D arises from a tree by orienting the edges in such a way that every node but one has one entering arc. The exceptional node, called the *root*, has no entering arc. The union of node-disjoint arborescences is called a *branching*. (Equivalently, a branching is a directed forest such that the in-degree of each node is at most one.)

Proposition 2.4. *For a digraph D the following are equivalent:*
(a) *D is an arborescence.*
(b) *D contains a node r such that every node can be reached from r by a unique path.*
(c) *D contains a node r such that every node can be reached from r and deleting any edge yields a node that is not reachable from r.*
(d) *D can be built from a node r by adjoining sequentially arcs so that the tail of the currently added new arc belongs to the digraph having already been constructed while the head is a new node.*

Let $D=(V, A)$ be a digraph with a specified node s. We describe now a simple device, the labeling technique, to determine the set S of nodes reachable from s along with a sub-arborescence of D rooted at s that spans S.

We use a label called R-label for every node v showing if v has already been reached or not. If not, the label has entry "NON-REACHED". If v is reached its R-label says REACHED and contains the arc $(u, v) \in A$ along which v has been reached. The only exception is the source node s: the entry of its label is always REACHED. At the beginning every node but s has NON-REACHED in its R-label.

We also use another label called S-label for every node to indicate whether v is SCANNED or UNSCANNED.

At the beginning of the algorithm for every node the entry of every S-label is UNSCANNED. In a general step we pick up an unscanned node u that has already been reached (at the beginning only the source is such) and decide if there is a non-reached node v such that (u, v) is an arc of D. If there is none, declare u SCANNED and repeat. Otherwise declare v REACHED and put (u, v) into its R-label and repeat.

The algorithm terminates if there is no more unscanned node which is reached.

Proposition 2.5. *The set S of nodes that have REACHED in their R-label has no leaving arcs and consists precisely of nodes reachable from s. The set of arcs occurring in their R-labels forms an arborescence rooted at s with node set S.*

Note that the procedure can be applied to undirected graphs as well. In the algorithm there is much freedom in choosing a reached and unscanned node. One possible strategy is to choose each time an unscanned node u which has been reached earliest. In this case the procedure is called *breadth first search* (BFS).

An application of BFS is to compute the distance of the nodes in S from s. The only modification in the above algorithm is that we need a third variable $\mathrm{dist}(v)$ at every node v to store the distance of v from s. At the beginning this is 0 at s and ∞ at all other nodes. When a node v is reached from u we define $\mathrm{dist}(v)$ to be $\mathrm{dist}(u)+1$.

Another natural strategy is to choose each time an unscanned node that has been reached latest. In this case the procedure is called *depth first search* (DFS). Depth first search has a great number of important applications and we will mention three of them in the next section.

Finally, we mention a third kind of search, the so-called maximum cardinality search. Here each time an unscanned node is chosen that has a maximum number of already reached neighbours. This search was introduced by Tarjan and Yannakakis (1984) in order to find a simplicial ordering of the nodes of a chordal graph. Nagamochi and Ibaraki (1992) showed how maximum cardinality search can be used to find a sparse k-connected subgraph of a k-connected graph.

We remark that it is not difficult to implement these search procedures so as to run in linear-time. (For details, see Tarjan 1983.)

2.2. 2-Connectivity and strong connectivity

Given a graph or digraph $G=(V, E)$, a node $v \in V$ is called a *cut node* if E can be partitioned into two non-empty subsets E_1 and E_2 such that $V(E_1)$ and $V(E_2)$ have just the node v in common. (For $F \subseteq E$, $V(F)$ denotes the set of nodes incident to at least one element of F.) In particular, a node incident to a loop and to another edge is a cut node. If G is loopless, then v is a cut node if and only if its deletion increases the number of components: $c(G-v)>c(G)$.

A connected graph is called a *block* if it has no cut node. A graph is called *2-connected* if it is a block and has at least three nodes.

Proposition 2.6. *For a loopless graph $G=(V, E)$ with $|V| \geqslant 3$ the following are equivalent:*
(a) *G is 2-connected.*
(b) *For any two nodes there is a circuit containing them.*
(c) *For any two sets A, $B \subseteq V$ with $|A|$, $|B| \geqslant 2$ there are two disjoint paths connecting nodes of A and B.*
(d) *Any pair of edges is contained in a circuit.*
(e) *G can be built up from a circuit by sequentially adjoining edges (loops are not allowed) and subdividing edges (in any order).*

The following is a useful reduction property of 2-connected graphs.

Proposition 2.7. *For every edge e of a 2-connected graph with at least four nodes either the deletion or the contraction of e results in a 2-connected graph.*

(*Contracting* an edge uv means that we identify u and v into a new node z and for each edge uw or vw of G we introduce an edge zw. *Deleting* an edge $e=uv$ means that we leave out e from E.)

A strongly connected digraph with at least three nodes is called a *strong block* if it has no cut node.

Proposition 2.8. *For a digraph $D=(V,A)$ with $|V|\geq 3$ the following are equivalent*:

(a) *D is a strong block.*

(b) *D can be built up from a directed circuit by sequentially adding arcs (no loops allowed) and subdividing arcs.*

Subdividing an arc (u,v) means that we replace (u,v) by a path P from u to v where the inner nodes of P are new nodes of the graph.

(Note that it is not true that every pair of nodes of a strong block lies on a directed circuit.)

Let $G=(V,E)$ be a connected but not necessarily 2-connected graph. A *block of graph* G is a subgraph that is a block and is maximal with respect to this property. The blocks of G form a tree-like structure in the following sense. Let $B_1, B_2, \ldots, B_k$ be the blocks of G. Form a bipartite graph $T=(V,B;F)$ where the elements of $B=\{b_1, b_2, \ldots, b_k\}$ correspond to the blocks of G. In T let nodes v_i and b_j be connected by an edge if $v_i \in B_j$.

Proposition 2.9. *The blocks of a graph $G=(V,E)$ partition the set E of edges. Two edges belong to the same block if and only if there is a circuit containing both. Any two blocks have at most one node in common and the nodes belonging to more than one block are cut nodes. The graph $T=(V,B;F)$ is a tree.*

We call an edge e of a graph $G=(V,E)$ a *cut edge* or an *isthmus* if $G-e$ has more components than G. A connected graph is called 2-*edge connected* if it contains no cut edges.

Proposition 2.10. *For a connected graph $G=(V,E)$ the following are equivalent*:

(a) *G is 2-edge-connected.*

(b) *For any pair of nodes there are two edge-disjoint paths connecting them.*

(c) *Any edge is contained in a circuit.*

(d) *G can be built up from a node by sequentially adjoining edges (loops are allowed) and subdividing edges.*

Property (d) is sometimes formulated in another way. By an *ear-decomposition* of G we mean a sequence $G_0, G_1, \ldots, G_t=G$ of subgraphs of G where G_0 consists of one node and no edge, and each G_i arises from G_{i-1} by adding a path P_i for which the two (not necessarily distinct) end-nodes belong to G_{i-1} while the inner nodes of P_i do not. The paths P_i are called *ears*. (P_i may consist of a single edge). Now (d) is equivalent to saying that a graph is 2-edge-connected if and only if it has an ear-decomposition. There are several other ear-decomposition theorems. One is mentioned in the next proposition. Another asserts that a graph is 2-connected if and only if there is an ear-decomposition using only open ears. Yet another (due to L. Lovász) says that a graph is factor-critical if and only if there is an ear-decomposition using only ears of odd length. (See the chapter on matchings by W. Pulleyblank.)

Proposition 2.11. *Let $D=(V,A)$ be a digraph whose underlying graph is connected. The following are equivalent*:

(a) *D is strongly connected.*

(b) *There is at least one arc leaving each set $X \subset V(D)$, $X \neq \emptyset$, that is, there is no directed cut.*

(c) *Every arc is in a directed circuit.*

(d) *D can be built up from a node by sequentially adding arcs (loops are allowed) and subdividing arcs.*

Let $D=(V,A)$ be a digraph whose underlying graph is connected. Let $C_1, C_2, \ldots, C_k$ be maximal strongly connected subgraphs of D. These subgraphs are called the *strong components* of G. The name is justified by the following proposition.

Proposition 2.12. *The node sets $V(C_i)$ $(i=1,\ldots,k)$ form a partition of V. By contracting each C_i into a node one obtains an acyclic digraph.*

Using depth first search, both the blocks of an undirected graph and the strong components of a directed graph can be found in linear time (Tarjan 1972).

Propositions 2.10 and 2.11 indicate that the analogous concepts for graphs and digraphs are 2-edge-connectivity and strong connectivity. Actually, parts (d) of these propositions immediately imply a theorem of Robbins (1939).

Corollary 2.13. *A graph G has a strongly connected orientation if and only G is 2-edge-connected.*

Here we provide another proof of this result that gives rise to a linear time algorithm (Tarjan 1972). Let s be an arbitrary node of G. Let T be a spanning tree determined by depth first search. Define an arborescence F by orienting the edges of T away from s.

Claim. *The unique path in T connecting the endpoints of any edge $e=uv \in E-T$ is a directed path P_e in F.*

The orientation of G obtained by orienting each edge $e \in E-T$ so as to form a directed circuit with P_e is strongly connected.

The following slight extension of Corollary 2.13 also holds (Boesch and Tindell 1980).

Theorem 2.13a. *The undirected edges of a mixed graph G (i.e., a graph having directed and undirected edges) can be oriented in such a way that the resulting digraph is strongly connected if and only if there is no cut edge in G and there is no directed cut.*

Corollary 2.13 naturally gives rise to the following question. Given a digraph

$D = (V, A)$, what is the minimum number of arcs the reversal of which makes D strongly connected? We must require that the underlying graph of D is 2-edge-connected. Obviously, the required subset of arcs meets all the directed cuts. Conversely, from Theorem 2.13a one can derive the following.

Proposition 2.14. *If F is a minimal set of edges covering all the directed cuts, the reversal of the elements of F leaves a strongly connected digraph.*

Therefore the following deep theorem of Lucchesi and Younger (1978) answers the question above. (For a relatively simple proof, see Lovász 1976b, for a constructive proof yielding a polynomial-time algorithm, see Frank 1981.)

Theorem 2.15 (Lucchesi and Younger 1978). *Let $D = (V, A)$ be an arbitrary digraph. The minimum number of arcs covering all the directed cuts is equal to the maximum number of pairwise disjoint directed cuts.*

3. Directed walks and paths of minimum cost

3.1. Walks and paths

Throughout this section we use the terms walk, path, circuit to mean diwalk, dipath, dicircuit, respectively. The *length* of a path is the number of edges of the path. Let $D = (V, A)$ be a loop-free digraph on n nodes and s and t two specified nodes called *source* and *sink*, respectively. Given a cost function $c: A \to \mathbb{R}$, find a path from s to t of minimum cost. Here the *cost* $c(P)$ of a path P is the sum of the cost of its arcs. A path is called a *min-cost path* if it has minimum cost among the paths from its origin to its terminus.

In this section we survey this problem and its variants. (For more detailed analysis, see Lawler 1976 and Tarjan 1983.) It turns out that computing one min-cost path from s to t is not simpler than computing a min-cost path from s to every other node reachable from s. So we focus mainly on that problem. We are going to describe several algorithms but the emphasis will be put on ideas and we do not seek for finding the most efficient procedures. The complexity of an algorithm heavily depends on the representation of the problem and the data structure used.

Let $w_k(v)$ denote the minimum cost of a walk of length at most k from s to v. If there is no such a walk, let $w_k(v) := \infty$. We will assume that each node of D is reachable from s. The following recursion is straightforward.

Proposition 3.1. $w_{k+1}(v) = \min(w_k(v), \min(w_k(u) + c(u, v): (u, v) \in A))$ *for* $v \in V$.

(The minimum taken over the empty set is defined to be ∞.)

Relying on this proposition one can easily design an $O(k|A|)$ algorithm to compute $w_k(v)$ for $v \in V$ as well as a walk $W_k(v)$ from s to v of length at most k with cost $w_k(v)$.

To outline this let us assume for convenience that for every node $v \in V - s$ there is an arc (s, v). If this is not the case, add a new arc $e = (s, v)$ with $c(s, v) = \infty$. At the beginning let $w_1(v) := c(s, v)$ for $v \in V - s$ and $w_1(s) = 0$. Furthermore, let $W_1(v) = \{s, (s, v), v\}$ for $v \in V - s$ and $W_1(s) := \{s\}$. In the $(k+1)$th phase of the algorithm $w_{k+1}(v)$ and $W_{k+1}(v)$ is computed with the help of the formula in the proposition.

What about minimum cost paths? If the cost function c is arbitrary, there is no hope for a good algorithm since the problem of finding a longest path from s to t is NP-complete and that problem can be formulated as a minimum cost path problem by choosing $c(e) = -1$ for every arc e.

Therefore it would be natural to suppose that $c \geq 0$. However, not everything is lost if we do not require that much. The minimum cost path problem is tractable under the weaker assumption that there are no circuits of negative total cost (or, shortly, negative circuit). In particular, in acyclic digraphs the min-cost paths can be computed in polynomial time for arbitrary cost functions.

We call a cost-function c *conservative* if there is no directed circuit with negative total weight. A function $\pi: V \to \mathbb{R}$ is called a *feasible potential* (subject to c) if $\pi(v) - \pi(u) \leq c(u, v)$ for every arc $(u, v) \in A$.

Theorem 3.2. *Given a digraph $D = (V, A)$ and a cost-function $c: A \to \mathbb{R}$, c is conservative if and only if there is a feasible potential. The potential can be chosen integer-valued if c is integer-valued.*

Proof. Suppose first that π is a feasible potential and $C = (v_0, E_1, v_1, e_2, \ldots, e_n, v_n = v_0)$ is an arbitrary circuit. Then we have $c(C) = \sum c(e_i) \geq \sum [\pi(v_i) - \pi(v_{i-1})] = 0$.

Conversely, suppose that there is no negative circuit. We can assume that there is a node s from which every other node is reachable. For otherwise adjoin a new node s to D and an arc (s, v) for each $v \in V$. Define the cost of the new arcs to be 0. Since there is no circuit containing the new node, the extended digraph contains no negative circuit.

We claim that $\pi(v) := w_n(v)$ (where $n = |V|$) is a feasible potential. Indeed, if there is no negative circuit, then $w_n(v)$ can be realized by a path P_v which has at most $n - 1$ arcs. Then $\pi(u) + c(u, v) = c(P_u) + c(u, v) \geq \pi(v)$. $\square$

Let $p_k(v)$ denote the minimum cost of a path of length at most k from s to v. Obviously, $p_n(v)$ is the minimum cost of a path from s to v. The preceding proof also shows the following.

Proposition 3.3. *If none of the walks $W_n(v)$ $(v \in V)$ induces a negative circuit, then there is no negative circuit in D, i.e., c is conservative.*

Since for a node $v \in V$ simplifying $W_n(v)$ can easily be carried out (in linear time) we have obtained an algorithm, due to Bellman (1958) and Ford (1956), that either finds a negative circuit or computes a min-cost path from s to every other node v. The complexity of the algorithm is $O(|A||V|)$.

In section 5 we will make use of the following problem, also interesting for its own sake. Suppose that there are negative circuits in a digraph $D=(V,A)$ and we want to eliminate all of the negative circuits by increasing the cost of every edge by the same value ε so that ε is as small as possible. It is an easy exercise to see that ε has the following interpretation: $-\varepsilon$ is the minimum circuit mean. (The *mean* of a circuit C is $c(C)/|C|$.)

To compute ε revise the algorithm mentioned after Proposition 3.1 as follows. Whenever a negative circuit C is detected by the algorithm, compute the current mean cost ε' of C and update the cost function by increasing the cost of each arc by $|\varepsilon'|$. Obviously C becomes a circuit of zero cost. The algorithm halts at the nth stage when there is no more circuit of negative (current) cost. One can see that the minimal ε is the sum of increments of costs and that a circuit C that became of zero cost last has the minimum mean cost. (See Karp 1978.)

Next we list some basic properties of minimum cost paths. For convenience, let us assume that every node of D can be reached from a specified node s.

Proposition 3.4. *Suppose that c is conservative. If $P=(s=v_0,e_1,\ldots,e_i,v_i,\ldots,e_j,v_j,\ldots,t)$ is a min-cost path from s to t, then a subpath $R=(v_i,\ldots,e_j,v_j)$ is a min-cost path from v_i to v_j. (Note that the corresponding statement for undirected graphs does not hold.)*

Proof. Let R' be a path from v_i to v_j for which $c(R')<c(R)$. Construct a walk W by replacing the segment R in P by R' and let P' be a path obtained by simplifying W. Since there is no negative circuit we have $c(P')\leqslant c(W)<c(P)$. □

Theorem 3.5. *If c is conservative, the minimum cost of a path from s to t is equal to $\max(\pi(t)-\pi(s)$: π a feasible potential).*

Proof. Let $P=(s=v_0,\ldots,v_k=t)$ be a path from s to t and π a feasible potential. Then $(*)$ $c(P)=\sum(c(v_{i-1},v_i): i=1,\ldots,k)\geqslant\sum(\pi(v_i)-\pi(v_{i-1}): i=1,\ldots,k)=\pi(t)-\pi(s)$ and so $\max\leqslant\min$.

To see the other direction let $p_n(v)$ denote the minimum cost of a path from s to v. We have seen that p_n is a feasible potential. Since $p_n(t)-p_n(s)=p_n(t)=c(P)$ we have equality in $(*)$. □

For a feasible potential π we say an arc (u,v) to be *tight* if $\pi(v)-\pi(u)=c(u,v)$. By Theorem 3.5 a path P is a min-cost path if and only if there is a feasible potential π such that P consists of arcs which are tight with respect to π.

A *min-cost-path s-arborescence F* is an arborescence of D rooted at s for which the unique path in F from s to any other node v is a minimum cost path in D.

Proposition 3.6. *If there is no negative circuit, there is a spanning min-cost-path s-arborescence.*

Proof. Let L be the union of arcs belonging to any min-cost path starting from s. The arcs in L are tight with respect to p_n and L contains a spanning s-arborescence. □

Proposition 3.7. *A spanning s-arborescence F is a min-cost-path s-arborescence if and only if $c_F(v)-c_F(u)\leqslant c(u,v)$ for every $(u,v)\in A$ ($c_F(v)$ denotes the cost of the unique path in F from s to v).*

Above we outlined a method of complexity $O(|A||V|)$ to decide if a cost function c is conservative and, if so, to compute a min-cost-path s-arborescence. It is not known whether there is a method of complexity $O(|V|^2)$. There are, however, two special cases when such an algorithm exists.

3.2. *Non-negative costs and acyclic digraphs*

Assume the cost function c is non-negative. We briefly summarize Dijkstra's (1959) method. The basic observation is the following.

Lemma 3.8. *If T is a min-cost-path s-arborescence (not necessarily spanning) and $m_T:=\min(p_n(u)+c(u,v): (u,v)\in\Delta^+(V(T))$ is attained at an arc $a=(u_a,v_a)$, then $T+a$ is a min-cost-path s-arborescence.*

Proof. Let P_1 denote the path obtained from the path in T from s to u_a by adding a. Let P be any path from s to v_a and $e=u_ev_e$ the first arc on P that leaves $V(T)$. Since $c\geqslant 0$, $c(P')\leqslant c(P)$ where P' is the subpath of P from s to v_e. By the choice of a, $c(P_1)\leqslant c(P')$ so P_1 is a min-cost path. □

Dijkstra's method 3.9 consists of $n-1$ phases. Starting at s we build up, arc by arc, a min-cost-path s-arborescence T. In order to compute m_T we maintain a label $l(v)=\min(p_n(u)+c(u,v): (u,v)\in\Delta^+(V_T))$ for $v\in V-V(T)$. This label tells us which arc $a=(u_a,v_a)$ has to be added to the current T. When an arc $a=(u_a,v_a)$ has been added to T, label $l(v)$ is updated by $l(v):=\min(l(v),l(v_a)+c(v_a,v))$. Thus updating all $l(v)$ needs $O(n)$ time and the overall complexity of the algorithm is $O(n^2)$.

As an application, let us return for a moment to the general case when c may not be non-negative but there is no negative circuit. Dijkstra's algorithm can be used to show that there is an $O(n^2)$ algorithm for computing a min-cost-path s-arborescence provided that a feasible potential π is available. Indeed, define $c'(u,v):=c(u,v)-\pi(v)+\pi(u)$.

Claim 3.10. *A path P from s to t is a min-cost path with respect to c if and only if P is a min-cost path with respect to c'.*

Proof. For any path R from s to t the difference $c(R)-c'(R)=\pi(t)-\pi(s)$ does not depend on R whence the claim follows. □

Therefore one can apply Dijkstra's algorithm to c'.

Another important special case when an $O(n^2)$ algorithm is available is the one of acyclic digraphs. We can suppose again that every node is reachable from s. The following slight modification of Dijkstra's algorithm works.

Lemma 3.11. *Let T be a min-cost-path s-arborescence and $v \in V - V(T)$ such that $\Delta^-(v) \subseteq \Delta^+(V(T))$. If $m_T(v) := \min(p_n(u) + c(u,v)$: $(u,v) \in \Delta^+(V(T))$ is attained at an arc e, then $T + e$ is a min-cost-path s-arborescence.*

The proof is straightforward. To implement the algorithm one has to find an ordering $v_1 = s, v_2, \dots, v_n$ of the nodes such that (v_i, v_j) can be an arc only if $i < j$ and then build the s-arborescence along this ordering. (Using depth first search such an ordering can easily be found in $O(|E|)$ time.)

This algorithm enables us to find a maximum path in an acyclic digraph, in particular, a maximum weight chain in a weighted poset.

3.3. *Shortest paths in undirected graphs*

Finally, we are interested in finding a min-cost path between two nodes s and t of an undirected graph $G = (V, E)$. If the cost function c is non-negative, the min-cost path problem can easily be reduced to the directed case by replacing each edge by a pair of oppositely directed arcs. This reduction, however, does not work in the general case, even if there is no negative circuit, since then we would introduce a negative (2-element) circuit. To overcome this difficulty one has to invoke matching theory, in particular, the theory of T-joins.

Let $E^- := \{e \in E: w(e) < 0\}$. Let $T_w := \{v \in V$: an odd number of edges from E^- is incident to $v\}$ and $T := \{s, t\} \oplus T_w$ where $\oplus$ denotes the symmetric difference. Define $w'(e) := |w(e)|$ for each $e \in E$. If there is no circuit of negative w-cost, then, for any T-join F of minimum w'-cost, $F \oplus E^-$ consists of an st-path of minimum w-cost and some disjoint circuit of zero w-cost. Therefore in order to compute a min-cost st-path it suffices to compute F. This, in turn, can be done with the help of a weighted matching algorithm. (While solving the min-cost path problem for directed graphs was not too difficult, one may be wondering if it is indeed necessary to invoke such a sophisticated tool, the matching algorithm, for solving the minimum cost path problem in undirected graphs. However this is not surprising anymore once one observes that an algorithm solving the latter problem can easily be used to compute a minimum weight perfect matching.)

For the structure of distances, see Sebő (1993).

4. Circulations and flows

4.1. *Feasible circulations and maximum flows*

In the theory of network flows there are several models which are, on one hand, equivalent via elementary constructions. On the other hand, for different type of applications it is convenient to have various models. We are going to survey the two basic ones: flows from a source to a sink and circulations. Although historically flows came earlier here we start with circulations. For more detailed network flow theory, see Ford and Fulkerson (1962), Lawler (1976), Phillips and Garcia-Diaz (1981), Lovász and Plummer (1986), Ahuja et al. (1993), Goldberg et al. (1990).

Throughout this section we work with a digraph $D = (V, A)$. Let $f: A \to \mathbb{R} \cup \{-\infty\}$ be a lower capacity, $g: A \to \mathbb{R} \cup \{+\infty\}$ an upper capacity such that $f \leq g$. For a vector $x: A \to \mathbb{R}$ and a subset $S \subseteq V$ let $\delta_x^-(S) := \sum (x(u,v)$: $(u,v) \in A$, (u,v) enters $S)$ and let $\delta_x^+(S) := \delta_x^-(V - S)$. Vector x is called a *circulation* if the *conservation rule* $\delta_x^-(v) = \delta_x^+(v)$ holds at every node v. It is easily seen that, given a circulation x, $\delta_x^-(X) = \delta_x^+(X)$ for all $X \subseteq V$. A circulation x is *feasible* if $f \leq x \leq g$.

Theorem 4.1 (Hoffman 1960). *There exists a feasible circulation if and only if*

$$\delta_f^-(X) \leq \delta_g^+(X) \quad \textit{for every } X \subseteq V . \tag{4.1}$$

If f and g are integer-valued and (4.1) *holds, there is an integer-valued feasible circulation.*

Proof. *Necessity.* If x is a feasible circulation, then $\delta_g^+(X) - \delta_f^-(X) \geq \delta_x^+(X) - \delta_x^-(X) = 0$ and (4.1) follows.

Sufficiency. Let $\gamma(X) := \delta_g^+(X) - \delta_f^-(X)$. Then (4.1) is equivalent to $\gamma(X) \geq 0$.

Lemma 4.2. $\gamma(X) + \gamma(Y) = \gamma(X \cap Y) + \gamma(X \cup Y) + d_{g-f}(X, Y)$.

Proof. The contribution of any arc to the two sides is the same. $\square$

Choose a counter-example for which the number q of arcs with $f(a) < g(a)$ is minimum. There is such an arc $a = (s, t)$ since otherwise $x := f(=g)$ is a feasible circulation (by (4.1)). Modify f by increasing $f(a)$ as much as possible without violating (4.1). By the minimal choice of q the modified $f(a)$ is still smaller than $g(a)$. Furthermore, there is a $t\bar{s}$-set T for which $\delta_f^-(T) = \delta_g^+(T)$, that is, $\gamma(T) = 0$. Similarly, reduce $g(a)$ as much as possible without violating (4.1). Again, the modified $g(a)$ is bigger than (the modified) $f(a)$ and there is an $s\bar{t}$-set S for which $\gamma(S) = 0$.

Because of arc a the value $d_{f-g}(S, T)$ is strictly positive. Thus, by Lemma 4.2 and by (4.1) we have $0 + 0 = \gamma(S) + \gamma(T) > \gamma(A \cap T) + \gamma(S \cup T) \geq 0 + 0$, a contradiction. The same proof shows that if f and g are integer-valued, then there is an integer-valued feasible circulation. $\square$

Let $D = (V, A)$ be a digraph with a specified source s and sink t. We assume, without restricting generality, that no arcs enter s and no arcs leave t. Let $g: A \to \mathbb{R}_+$ be a capacity function that is positive everywhere. A vector $x: A \to \mathbb{R}_+$

is called a *flow* from s to t, or an *st-flow*, if $\delta_x^-(v)=\delta_x^+(v)$ holds for every $v\in V-\{s,t\}$. A flow x is called *feasible* if $0\leqslant x\leqslant g$.

It can easily be seen that for a flow x and for any $s\bar{t}$-set S the *netflow* leaving S defined by $\delta_x^+(S)-\delta_x^-(S)$ does not depend on the choice of S. This common value $\delta_x^+(s)$ is called the *flow value* of x and denoted by val(x). The value $x(u,v)$, $(u,v)\in A$, is called the *arc-flow*. A flow x is called a *path-flow* if x is positive only along a path from s to t.

The fundamental theorem of network flows, called the max-flow min-cut (MFMC) theorem, is due to Ford and Fulkerson (1956) and to Elias et al. (1956).

Theorem 4.3. *The maximum value of a feasible st-flow is the minimum of $\delta_g^+(S)$ over all $s\bar{t}$-sets S. If g is integer-valued, there is an integer-valued maximum flow.*

Proof. Let x be a feasible st-flow and S an $s\bar{t}$-set. Then we have $\text{val}(x)=\delta_x^+(S)-\delta_x^-(S)\leqslant\delta_g^+(S)$ from which max $\leqslant$ min follows.

To see the other direction let m denote the minimum in question. Adjoin a new arc $e=(t,s)$ to D and define $f(e):=g(e):=m$. Let f be zero on all old arcs. It is easy to see that (4.1) holds for that choice of f and g so by Theorem (4.1) there is a feasible circulation (integral if f and g are integral). This circulation, without the new arc e, is an st-flow of value m. □

From the first part of the proof we see that an x is a maximum flow and $\delta_g^+(S)$ is of minimum if and only if

(a) $x(a)=g(a)$ for every arc a leaving S and

(b) $x(a)=0$ for every arc a entering S.

These optimality criteria are crucial for the next algorithm due to Ford and Fulkerson. This provides an algorithmic proof of Theorem 4.3 for the case when the capacity function g is integer-valued (and thus, when g is rational). A refinement of that method, due to J. Edmonds and R. Karp and to E.A. Dinits, will provide a strongly polynomial algorithm and an algorithmic proof for arbitrary capacities.

4.2. *Augmenting paths method*

The algorithm of Ford and Fulkerson starts with an arbitrary feasible st-flow x (for example $x\equiv 0$) and iteratively improves it. To describe one iteration let x be a feasible st-flow. Construct an auxiliary digraph $D_x=(V,A_x)$ as follows. An arc (u,v) belongs to A_x if either (i) $(u,v)\in A$ and $x(u,v)<g(u,v)$ and then this arc of D_x is called a *forward* arc, or (ii) $(v,u)\in A$ and $x(v,u)>0$ and then (u,v) is a *backward* arc.

Let S denote the set of nodes reachable from s in D_x.

Case 1: $t\notin S$. Since no arc of D_x leaves S, optimality criteria (a) and (b) hold and the algorithm terminates.

Case 2: $t\in S$. Let P be any path in D_x from s to t.

Let $\Delta_1:=\min(g(u,v)-x(u,v)$: (u,v) is a forward arc of $P)$ and $\Delta_2=\min(x(v,u)$: (u,v) is a backward arc of $P)$. Let $\Delta=\min(\Delta_1,\Delta_2)$. Then Δ is positive. Call an arc of P *critical* if Δ is attained at that arc.

Update x as follows. If (u,v) is a forward arc of P, increase $x(u,v)$ by Δ. If (u,v) is a backward arc of P, decrease $x(v,u)$ by Δ. An easy consideration shows that the revised x' is a feasible st-flow with $\text{val}(x')=\text{val}(x)+\Delta$. Consequently, if g is integer-valued case 2 can occur only finitely many times.

In case of rational g we can easily reduce the problem by multiplying through the components of g by the least common denominator. When g is irrational, the algorithm above may not terminate. An example to show this pathological situation is given in Ford and Fulkerson (1962) (a simpler one occurs in Lovász and Plummer 1986). Another drawback of the algorithm is that even for integer capacities the number of iterations may be proportional to the largest occurring capacity M, as the following example shows, see fig. 4.1.

If among the possible augmenting paths we always choose the one of three arcs, then the flow augmentation Δ is just one in every step. Therefore the complexity of the algorithm is exponential in log M, the size of the input.

4.3. *The method of Edmonds, Karp and Dinits*

To overcome these difficulties Edmonds and Karp (1972) and Dinits (1970) proposed to choose a shortest augmenting path at each iteration. This simple modification makes it possible to bound the number of iterations in the Ford–Fulkerson algorithm by a polynomial of $|V|$ and $|A|$, irrespective of the capacities.

Let $\sigma_x(v)$ denote the distance of v from s in D_x. (If there is no path from s to v, then $\sigma_x(v)=\infty$). Let P be a shortest path in D_x from s to t. Then for any arc (u,v) of P, $\sigma_x(v)=\sigma_x(u)+1$.

Lemma 4.4. *Performing an augmentation along P does not decrease $\sigma_x(v)$.*

Proof. Let us consider how an augmentation affects D_x. Since the flow has been changed only at the arcs of D corresponding to the arcs of P, D_x may be changed at the arcs of P. Namely, the (possibly) new arcs of D_x' are the arcs of P in reverse orientation (while the critical arcs of P disappear). The distance of a node v from s could decrease only if we adjoin an arc (u,w) for which $\sigma_x(w)>\sigma_x(u)+1$ and the lemma follows. □

The sequence of augmentations can be divided into phases. In one phase $\sigma_x(t)$ remains the same. By Lemma 4.4 there may be at most $|V|-1$ phases.

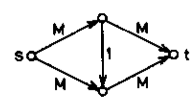

Figure 4.1.

Lemma 4.5. *Within one phase at most $|A|$ augmentations may occur.*

Proof. Let $\sigma_i(v)$ denote the distance of v from s in the auxiliary digraph at the beginning of a given phase i. Call an arc (u, v) i-tight if $\sigma_i(v) = \sigma_i(u) + 1$. Within phase i only i-tight arcs may be used. By Lemma 4.4 an augmentation eliminates at least one i-tight arc from the current auxiliary digraph and no new i-tight arc arises. Since an auxiliary digraph may have at most $|A|$ arcs, the lemma follows. □

By Lemma 4.5 the algorithm needs at most $|V||A|$ augmentations. One augmentation can be performed in $O(|V|)$ time so the overall complexity is $O(|V|^2|A|)$.

Remark. The original augmenting paths method leaves freedom in choosing augmenting paths. The modification above imposes certain restrictions but still the algorithm may have different runs and, as a result, it may end up with different maximum flows. The final minimum cut $\delta^+(S)$, however, provided by the augmenting paths method is independent of the run of the algorithm. It is easy to show that if both X and Y minimize $\delta_g^+(Z)$ over $s\bar{t}$-sets, then both $X \cap Y$ and $X \cup Y$ are minimizing $s\bar{t}$-sets and therefore there is a unique minimal minimizing set S. The augmenting path method ends up with this S.

Since the algorithm of Edmonds, Karp and Dinits quite a few improvements on the complexity of max-flow min-cut algorithms have been devised. See Cherkasskij (1977a), Malhotra et al. (1978), Galil (1980), Shiloach (1978), Sleator (1980).

Max-flow algorithms using augmenting paths have the characteristic feature that the current flow is always changed in a "big piece": along all the edges of an augmenting path. An important conceptional development was the introduction of preflows by Karzanov (1974). A *preflow* is a non-negative function x on the edges of a digraph so that $\delta_x^-(v) \geq \delta_x^+(v)$ holds for every node v distinct from the source. A *preflow-push algorithm* changes the current preflow each time along just one edge. This is the basis of the greater flexibility and efficiency of preflow-push algorithms. See Shiloach and Vishkin (1982), Goldberg and Tarjan (1986), Cheriyan and Maheshwari (1989). Alon (1990) described a deterministic version of a randomized algorithm of Cheriyan and Hagerup (1990) whose complexity is $O(nm \log n)$

The interested reader may find a much more detailed comparison of the max-flow algorithms in the survey paper of Goldberg et al. (1990) and in the textbook of Ahuja et al. (1993).

4.4. *Finding feasible circulations*

We have seen how the max-flow min-cut theorem could be derived from Hoffman's circulation theorem. We show now the reverse direction. This way we will have a tool by which a feasible circulation can be found (or a set violating

(4.1)) with the help of one max-flow min-cut computation (on a slightly larger digraph).

For simplicity we restrict ourselves to finite f and g (the general case is left to the reader). For $v \in V$ denote $\gamma(v) = \delta_f^-(v) - \delta_f^+(v)$. If γ is zero everywhere, then f is a feasible circulation. Otherwise, the sets $S = (v: \gamma(v) > 0)$ and $T = (v: \gamma(v) < 0)$ are non-empty. Let $D' = (V', A')$ where $V' = V \cup \{s, t\}$ and $A' = A \cup (s, v): v \in S \cup ((v, t): v \in T)$. Define a capacity function g' as follows. $g'(s, v) := \gamma(v)$ if $v \in S$, $g'(v, t) = -\gamma(v)$ if $v \in T$ and $g'(a) = g(a) - f(a)$ if $a \in A$. Let $M = \sum (\gamma(v): v \in S)$.

Lemma 4.6. (a) *x is an st-flow of value M in D' (with respect to g') if and only if $f + x$ (restricted circulation).*

(b) *A set $X \subseteq V$ violates* (4.1) *if and only if $\delta_{g'}^+(X + s) < M$.*

4.5. *Other models and applications*

Via elementary constructions flows and circulations are equivalent. There are other, more sophisticated variants. For example, one can impose lower bound on the arc-flows in the maximum flow problem. Or, lower and upper bounds can be requested on the in-flows $(\delta_x^-(v))$ at nodes. Moreover, instead of one source and one sink multiple sources and sinks can be specified. This generalization is sometimes called a *transshipment* problem. The *transportation* problem consists of finding a minimum cost degree-constrained subgraph of a bipartite graph. With relatively simple elementary constructions all these models go back to flows or circulations. We refer to the classical book of Ford and Fulkerson (1962).

Next, we are going to survey some of the combinatorial consequences of the flow theory.

Theorem 4.7 (Menger's theorem arc-version). *Let $D = (V, A)$ be a digraph with two specified nodes s and t. The maximum number of arc-disjoint paths from s to t is the minimum of $\delta^+(S)$ over all $s\bar{t}$-sets S.*

Proof. Apply Theorem 4.3 with $g \equiv 1$ and notice that every flow from s to t is the sum of path-flows from s to t and a non-negative circulation. □

In section 7 some other versions of Menger's theorem will be discussed.

Theorem 4.8 (König 1915). *The maximum number of disjoint edges of a bipartite graph $G = (V_1, V_2; E)$ is the minimum number of nodes covering all the edges.*

Proof. Orient the edges of G from V_1 to V_2. Then extend G by two new nodes s and t and new arcs (s, v) $(v \in V_1)$ and $(v, t)(v \in V_2)$. Let the capacities of all the new arcs be 1 and the other capacities M, a big number. A maximum st-flow of value k corresponds to k independent edges of G. By Theorem 4.7 there is an

$s\bar{t}$-set S for which $\delta_g^+(S)=k$. No arc of capacity M can leave S therefore $(V_1-S)\cup(V_2\cap S)$ is a covering of E with cardinality k. □

4.6. Gomory–Hu trees

In many applications the ability to compute the minimum cut separating two nodes is more important than finding a maximum flow. In undirected graphs the minimum cuts separating every pair of nodes has an especially attractive structure.

Let $G=(V,E)$ be a connected undirected graph and $g:E\to\mathbb{R}_+$ a non-negative capacity function. Let $\lambda(u,v)$ denote the maximum flow value between u and v. We say that a set Z separating u and v is *uv-minimal* if $d_g(Z)$ is minimal over all subsets separating u and v. Equivalently, $d_g(Z)=\lambda(u,v)$.

A set X is called *critical* if X is uv-minimal for some u and v. In order to have some insight into the structure of critical sets, one may be interested in a list of critical sets that contains a uv-minimal set for each pair u, v. How short can this list be? By choosing a separate uv-minimal set for each pair there is a list of $n(n-1)/2$ sets. But one can do much better.

Let $G_T=(V,F)$ be any tree on node set V (not necessarily a subgraph of G). For every edge e in F let $m(e):=d_g(X_e)$ where X_e and $V-X_e$ are the two components of G_T-e. G_T is called a *Gomory–Hu tree* of G (with respect to the given capacity function g) if (a) for every pair $\{s,t\}$ of nodes $\lambda(s,t)$ is the minimum of m-values over the edges of the unique path in G_T connecting s and t and (b) if e is an edge where the minimum is attained, then X_e is st-minimal.

For example, if $G=K_{3,3}$ and $g\equiv 1$, then a star of five edges forms a Gomory–Hu tree (and there is no other one showing that a Gomory–Hu tree cannot be chosen, in general, as a subgraph of G).

Theorem 4.9 (Gomory and Hu 1961). *Every graph possesses a Gomory–Hu tree.*

We are going to consider only the case $g\equiv 1$. For general g the proof goes along the same line. We need the following terminology. A family $\mathcal{F}$ of subsets of nodes is called *laminar* if for any two non-disjoint members of $\mathcal{F}$ one of them includes the other. We say that *$\mathcal{F}$ separates* nodes u and v if at least one member of $\mathcal{F}$ separates u and v.

Let $\mathcal{F}$ be a laminar family and $\{u,v\}$ a pair of nodes not separated by $\mathcal{F}$. We say that a subset X of nodes separating u and v is *uv-minimal with respect to $\mathcal{F}$* if $\mathcal{F}\cup\{X\}$ is laminar and $d(X)$ is as small as possible. Note that such an X can be computed by one MFMC computation in a graph obtained from G by contracting the complement of the smallest member X of $\mathcal{F}$ containing u and v and contracting the maximal members of $\mathcal{F}$ included in X.

Proof. Let us construct a laminar family $\mathcal{F}$ of $n-1$ sets as follows. Let $\mathcal{F}_0$ be empty. Suppose we have constructed a laminar family $\mathcal{F}_{k-1}=\{A_1,\dots,A_{k-1}\}$ for some $k=1,\dots,n-1$. Let $\{u_k,v_k\}$ be any pair of nodes not separated by $\mathcal{F}_{k-1}$. Determine a set A_k that is u_kv_k-minimal with respect to $\mathcal{F}_{k-1}$ and let $\mathcal{F}_k:=\mathcal{F}_{k-1}\cup\{A_k\}$.

Let $\mathcal{F}:=\mathcal{F}_{n-1}$. Let $E_1:=\{u_iv_i: i=1,\dots,n-1\}$. From the construction we see that $T_1:=(V,E_1)$ is a tree. (This is just an auxiliary tree for the proof.)

Claim 1. *Let X be an xx'-minimal set and Y a critical set. If Y does not contain x and x', then either $X-Y$ or $X\cup Y$ is xx'-minimal. If Y contains x and x', then either $X\cap Y$ or $Y-X$ is xx'-minimal.*

Proof. To prove the first statement suppose that Y is yy'-minimal. Then either $Y-X$ or $Y\cap X$ separates $\{y,y'\}$. In the first case one has $\lambda(x,x')+\lambda(y,y')=d(X)+d(Y)\geqslant d(X-Y)+d(Y-X)\geqslant\lambda(x,x')+\lambda(y,y')$. Hence equality follows everywhere showing that $X-Y$ is xx'-minimal.

In the second case $\lambda(x,x')+\lambda(y,y')=d(X)+d(Y)\geqslant d(X\cup Y)+d(X\cap Y)\geqslant\lambda(x,x')+\lambda(y,y')$. Hence equality follows everywhere showing that $X\cup Y$ is xx'-minimal, as required for the first statement.

The second statement follows from the first one if we replace Y by its complement. □

Claim 2. *$A_i\in\mathcal{F}$ is u_iv_i-minimal for each $i=1,\dots,n-1$.*

Proof. The claim is clear for $i=1$. Suppose we have already proved it for $1,2,\dots,i-1$ and let $x:=u_i$, $x':=v_i$. By induction on $j=0,1,\dots,i-1$ we are going to show that $(*)$ there is an xx'-minimal set X for which $\mathcal{F}_j\cup\{X\}$ is laminar. From this the claim will clearly follow.

$(*)$ obviously holds for $j=0$. Suppose we have already shown $(*)$ for some $0\leqslant j<i-1$. Let X' be the xx'-minimal set assured by Claim 1 when applied to X and $Y:=A_j$. Then $\mathcal{F}_j\cup\{X'\}$ is laminar, as required. □

Claim 3. *For every pair $\{s,t\}$ of nodes there is an st-minimal member of $\mathcal{F}$.*

Proof. Let P be the unique path in T_1 connecting s and t and let $M:=\min(\lambda(u_i,v_i): u_iv_i$ an edge of $P)$. By the MFMC theorem $\lambda(s,t)\geqslant M$.

Let j be the smallest subscript for which u_jv_j is an edge of P and $\lambda(u_j,v_j)=M$. We claim that A_j does not separate any other edge of P. Indeed, if A_j separates an edge u_iv_i of P, then $i<j$ by the construction of $\mathcal{F}$. Furthermore, $M\leqslant\lambda(u_i,v_i)\leqslant d(A_j)=M$ and hence $\lambda(u_i,v_i)=M$, contradicting the minimal choice of j.

Therefore A_j must separate $\{s,t\}$ and hence we have $M\geqslant\lambda(u_j,v_j)=d(A_j)\geqslant\lambda(s,t)\geqslant M$. We can conclude that A_j is an st-minimal set. □

Let $A_0:=V$ and $\mathcal{F}':=\mathcal{F}\cup\{V\}$. For each $A_i\in\mathcal{F}'$ the union of maximal sets of $\mathcal{F}$ included in A_i is precisely one element smaller than A_i. Let t_i denote this element and for $i\geqslant 1$ let $s_i:=t_j$ where A_j is the unique minimal element of $\mathcal{F}'$ including A_i. Let $F':=\{(s_i,t_i): i=1,\dots,n-1\}$. Then $G_T'=(V,F')$ is an arborescence such that each arc of it enters one member of $\mathcal{F}$. Let G_T denote the underlying (undirected) tree. By this construction and by Claim 3, G_T is a Gomory–Hu tree. □

The following corollary, due to Padberg and Rao (1982), found a nice application in a linear programming approach to matching problems. It basically asserts that a Gomory–Hu tree encodes not only a minimum cut separating any given pair of nodes but also a minimum T-cut for any even subset T of nodes. (For the definition of T-cut and T-join, see the chapter on matchings.)

Corollary 4.10 (Padberg and Rao 1982). *Let G_T be a Gomory–Hu tree and T an even subset of nodes. Then a minimum T-cut can be obtained by choosing an edge e of G_T for which the cut determined by the two components of $G_T - e$ is a T-cut and $m(e)$ is as small as possible.*

Proof. Let C be a minimum T-cut and let J be the set of edges e of G_T for which the cut C_e determined by $G_T - e$ is a T-cut. Clearly J is a T-join and therefore there is an edge $e = uv \in C \cap J$. We have $|C| \geq \lambda(u,v) = m(u,v) = |C_e| \geq |C|$ showing that C_e is also a minimum T-cut. □

Another corollary states that if the degree of each node of a graph is at least k, then there are two distinct nodes which are connected by at least k edge-disjoint paths. Indeed, if u is a node having degree one in the Gomory–Hu tree and v is its neighbour in the tree, then u and v will do.

5. Minimum cost circulations and flows

5.1. Min-cost circulations

In the previous section we got to know how to find feasible circulations. It is not a less striking problem to find a possible circulation that minimizes a specified linear cost function. To attack this problem let us consider the set Q of all feasible circulations. Q forms a polyhedron in $\mathbb{R}^E$ called a *circulation polyhedron* and denoted by $C(D; f, g)$. (That is, $C(D; f, g) = \{x \in \mathbb{R}^E : \delta_x^-(v) = \delta_x^+(v)$ for $v \in V$ and $f(u,v) \leq x(u,v) \leq g(u,v)$ for $(u,v) \in A\}$. Q is said to be *integral* if f and g are integer-valued.

Theorem 4.1 implies that a (non-empty) integral circulation polyhedron contains an integer point. Since the face of an (integral) circulation polyhedron is obviously an (integral) circulation polyhedron, we have proved the following.

Theorem 5.1. *Every face of an integral circulation polyhedron Q contains an integer point.*

Let $Q = C(D; f, g)$ be a non-empty circulation polyhedron and $c : A \to \mathbb{R}$ a cost function. The cost cx of a circulation x is defined by $\sum (x(a)c(a) : a \in A)$. What is the minimum cost of a feasible circulation and when does this minimum exist?

Define a digraph $D' = (V, A')$ and a cost function c' on A' as follows. An arc (u,v) belongs to A' if either (i) $(v,u) \in A$ and $f(v,u) = -\infty$ or (ii) $(u,v) \in A$ and $g(u,v) = \infty$. In case (i) let $c'(u,v) = -c(v,u)$, in case (ii) let $c'(u,v) = c(u,v)$.

Theorem 5.2. *The following are equivalent:*

(1a) *There is a feasible circulation of minimum cost.*

(1b) *There is no negative circuit in D' with respect to c'.*

(1c) *There is a potential $\pi : V \to \mathbb{R}$ such that*

$$\pi(v) - \pi(u) \leq c(u,v) \quad \textit{whenever } (u,v) \in A,\ g(u,v) = \infty \tag{5.1i}$$

and

$$\pi(v) - \pi(u) \geq c(u,v) \quad \textit{whenever } (u,v) \in A,\ f(u,v) = -\infty\,. \tag{5.1ii}$$

Note that a potential satisfying (1c) can be found (if one exists) by a min-cost path computation.

How can we characterize optimal feasible circulations? For $x \in Q$ define a digraph $D_x = (V, A_x)$ and a cost function c_x as follows. Let an arc (u,v) belong to A_x if either (i) $(u,v) \in A, x(u,v) < g(u,v)$ and then let $c_x(u,v) = c(u,v)$ (forward arc of A_x) or (ii) $(v,u) \in A, x(v,u) > f(v,u)$ and then let $c_x(u,v) = -c(v,u)$ (backward arc of A_x).

Theorem 5.3. *For a feasible circulation x the following are equivalent:*

(2a) *x is of minimum cost.*

(2b) *There is no negative circuit in D_x with respect to c_x.*

(2c) *There is a potential $\pi : V \to \mathbb{R}$ such that*

$$\pi(v) - \pi(u) \leq c(u,v) \quad \textit{if } x(u,v) < g(u,v)((u,v) \in A) \tag{5.2i}$$

and

$$\pi(v) - \pi(u) \geq c(u,v) \quad \textit{if } x(u,v) > f(u,v)((u,v) \in A)\,. \tag{5.2ii}$$

We call (5.2i) and (5.2ii) *optimality criteria*.

5.2. Min-cost circulation algorithm

The algorithm (Ford and Fulkerson 1962) starts with a feasible circulation x and a potential π satisfying (1c). If x and π satisfy (2c) as well, we are done. Otherwise, let $(t,s) \in A$ violate, say, (5.2i), that is $x(t,s) < g(t,s)$ and $\pi(s) - \pi(t) > c(t,s)$. (The case when an arc violates (5.2ii) is analogous.)

Define a digraph $D_x = (V, A_x)$ and a capacity function $g_x : A_x \to \mathbb{R}_+$ as follows. An arc (u,v) belongs to A_x if either

$$(u,v) \in A,\ \pi(v) - \pi(u) \geq c(u,v) \quad \text{and} \quad x(u,v) < g(u,v) \tag{3i}$$

or

$$(v,u) \in A,\ \pi(u) - \pi(v) \leq c(v,u) \quad \text{and} \quad x(v,u) > f(v,u)\,. \tag{3ii}$$

Let $g_x(u,v) = g(u,v) - x(u,v)$ in case (i) and $g_x(u,v) = x(v,u) - f(v,u)$ in case (ii). Let M denote the maximum value of an st-flow and let $\Delta = \min(M, g(t,s) - x(t,s))$. With the help of a max-flow min-cut computation determine an st-flow z

of value Δ. Revise x as follows.

$$x'(u,v) = x(u,v) + \Delta \quad \text{if } (u,v) \text{ is a forward arc of } D_x\,,$$

$$x'(u,v) = x(u,v) - \Delta \quad \text{if } (v,u) \text{ is a backward arc of } D_x\,.$$

Claim. *x' is a feasible circulation. An arc $(u,v)\in A$ satisfies (5.2i) and (5.2ii) with respect to x' if it satisfies with respect to x.*

There may be two cases.

Case 1: $\mathrm{val}(z) = g(t,s) - x(t,s)$. In this case arc (t,s) no longer violates the optimality criteria.

Case 2: $M(=\mathrm{val}(z)) < g(t,s) - x(t,s)$. Let S be the $s\bar{t}$-set determined by the max-flow min-cut computation. Let $\varepsilon_1 = \min(c(u,v) - \pi(v) + \pi(u)\colon (u,v)$ leaves S, $x(u,v) < g(u,v))$. Let $\varepsilon_2 = \min(\pi(u) - \pi(v) + c(u,v)\colon (u,v)$ enters S, $x(u,v) > f(u,v))$. Define $\varepsilon = \min(\varepsilon_1, \varepsilon_2)$ and revise π as follows: $\pi'(v) = \pi(v) + \varepsilon$ if $v \notin S$ and $= \pi(v)$ if $v \in S$.

Repeat the procedure with x' and π'.

It can be shown that this algorithm is finite for any cost and capacity functions. It is not necessarily of polynomial time, however, even if c, f, g are integer-valued. We describe a machinery, the scaling technique, to make the above algorithm of polynomial time. The method is due to Edmonds and Karp (1972).

5.3. Scaling technique

Suppose that the cost function c is integer-valued and that a feasible circulation x and an integer-valued potential π satisfying the optimality criteria are available. Let c' be another cost function differing from c on one arc by 1. The basic observation is that the algorithm above finds in polynomial time a feasible circulation x' and an integer-valued potential π' satisfying the optimality criteria with respect to f, g, c' provided that we start with the available x and π. Indeed, if case 2 occurs, then $\varepsilon = 1$ and arc (t,s) violates no longer the optimality criteria. In other words, with one max-flow min-cut computation (whether case 1 or case 2 occurs) the required x' and π' can be obtained.

Consequently, if c'' is an integer-valued cost function differing from c on every arc by at most one, then, starting with x and π, at most $|A|$ max-flow min-cut computations yield an x'' and π'' satisfying the optimality criteria with respect to c''.

Another, trivial, observation is that if x and π satisfy the optimality criteria with respect to c, then so do x and 2π with respect to $2c$.

Assume now for convenience that c is non-negative and let c be given in binary base. Let the maximum of $c(a)$ have K digits. Then there are K 0–1 vectors $c_0, \ldots, c_{K-1}$ in $\mathbb{Z}_+^A$ such that $c = \sum (2^i c_i\colon i = 0, \ldots, K-1)$.

First solve the min-cost circulation problem for c_{K-1}. This needs at most $|A|$ MFMC computations. Let the solution be x_{K-1} and π_{K-1}. Starting with x_{K-1}, $2\pi_{K-1}$ solve the min-cost circulation problem for $2c_{K-1} + c_{K-2}$. This also needs at most $|A|$ MFMC computations. Continuing this way, after at most $K|A|$ applications of the MFMC algorithm, we obtain a feasible circulation x and a potential π satisfying the optimality criteria with respect to c.

Remark. Let us draw attention to a small technical difficulty which can, however, be easily overcome: it may happen that there is a min-cost circulation with respect to c but there is none with respect to an intermediate cost function.

5.4. Strongly polynomial algorithm

Comparing the complexity of the scaling technique and the maximum flow algorithm of Edmonds, Karp and Dinits there is a significant difference. Namely, the complexity of the latter algorithm does not depend on the magnitude of the numbers (if we assume that adding and comparing two numbers is one step) and in this sense this algorithm is "strongly" polynomial while the complexity of the scaling technique is proportional to the number of digits.

Tardos (1985) was the first who constructed a strongly polynomial algorithm for finding a min-cost circulation. Since her work many other strongly polynomial algorithms have been developed. The fastest one is due to Orlin (1988). Here we briefly outline the algorithm of Goldberg and Tarjan (1989) that seems to be conceptionally the most attractive.

Optimality criterion (2b) suggests the following procedure. Start with a feasible circulation x. If (2b) holds, x is optimal. Otherwise, choose a circuit C in D_x violating (2b) and cancel along C. *Canceling along* C means that we increase $x(u,v)$ by Δ if (u,v) is a forward arc of C and decrease $x(u,v)$ by Δ if (v,u) is a backward arc of C. Here Δ is the smaller value of $\min(g(u,v) - x(u,v)\colon uv$ a forward arc of $C)$ and $\min(x(u,v) - f(u,v)\colon (v,u)$ a backward arc of $C)$.

Clearly, the modified x' is a feasible circulation and its cost is smaller than that of x. The algorithm consists of repeating this canceling procedure as long as (2b) is violated. This procedure is not necessarily of polynomial time. However, Goldberg and Tarjan proved that the following selection rule makes the algorithm strongly polynomial: each time choose a circuit C in G_x to be one of minimum mean cost. In section 3 we indicated how to compute such a circuit.

A beautiful feature of the algorithm of Goldberg and Tarjan is that it can be considered as a straight generalization of the Edmonds–Karp–Dinits algorithm for computing a maximum flow. Indeed, in the proof of Theorem 4.3 it was shown how a max-flow problem can be formulated as a minimum cost circulation problem. The Goldberg–Tarjan algorithm, when applied to this special min-cost circulation problem, yields precisely the Edmonds–Karp–Dinits algorithm.

5.5. Minimum cost flows

Let us be given again a digraph $D = (V, A)$ with a source s and a sink t. A non-negative capacity function g and a non-negative cost function c are given on A. We assume that both g and c are integer-valued. We have seen how to compute the maximum value M of an st-flow. This time we are interested in finding a minimum cost st-flow of value m for all possible integers m, $0 \le m \le M$. The cost of a flow z is defined by $cz = \sum (c(e)z(e): e \in A)$. We say that an st-flow z is a *min-cost flow* if z has the minimum cost among the feasible st-flows of value $\mathrm{val}(z)$.

We have seen the equivalence between the feasible circulation problem and the maximum flow problem. Using the same elementary construction the min-cost st-flow problem could be solved in strongly polynomial time with the help of a strongly-polynomial min-cost circulation algorithm.

Here we briefly survey a direct algorithm due to Ford and Fulkerson (1962). This algorithm is strongly polynomial only if the capacities are small integers. The reason why we include this algorithm is that it has a nice combinatorial application.

A flow z is of minimum cost if and only if there is a function $\pi: V \to Z_+$, called a *potential*, ($\pi(s) = 0 \le \pi(v) \le \pi(t)$ for $v \in V$) for which the following two optimality criteria hold:

$$\pi(v) - \pi(u) < c(u,v) \Rightarrow z(u,v) = 0\,, \tag{5.3i}$$

$$\pi(v) - \pi(u) > c(u,v) \Rightarrow z(u,v) = g(u,v)\,. \tag{5.3ii}$$

We use the notation $\bar{c}(u,v) = c(u,v) - \pi(v) + \pi(u)$ for $(u,v) \in A$. The method (Ford and Fulkerson 1962) can be considered as a refinement of the max-flow min-cut algorithm of Ford and Fulkerson. It constructs a min-cost flow for all possible (integer) flow values m.

The algorithm starts with the identically zero flow and the identically zero potential. Then the flow value is increased one by one and the potential is appropriately increased so that the optimality criteria are throughout maintained. The algorithm terminates when a maximum flow (and a minimum cut) is found.

Iterative step. At the general step we are given a flow z and a potential π satisfying (i) and (ii). Construct an auxiliary digraph $D' = (V, A')$ as follows. D' has two types of arcs: forward and backward. An arc (u,v) is a forward arc if $uv \in A$, $\bar{c}(u,v) = 0$ and $z(u,v) < g(u,v)$. An arc (u,v) is a backward arc if $(v,u) \in A$, $\bar{c}(v,u) = 0$ and $z(v,u) > 0$. Let S be the set of nodes reachable from s in D'. There are two cases.

Case 1: $t \notin S$. Define $\varepsilon_1 = \min(\bar{c}(u,v): (u,v) \in \delta^+(S), z(u,v) < g(u,v))$ and $\varepsilon_2 = \min(-\bar{c}(u,v): uv \in \delta^+(V-S), z(u,v) > 0)$ where the minimum is defined to be ∞ if it is taken over the empty set. Let $\varepsilon = \min(\varepsilon_1, \varepsilon_2)$. The optimality criteria and the construction of S imply that ε is positive.

If $\varepsilon = \infty$, the algorithm terminates since we have $\delta_g^+(S) = \mathrm{val}(z)$ and thus the current flow z is maximum and $\delta^+(S)$ is a minimum cut.

If $\varepsilon < \infty$, revise π by increasing $\pi(v)$ for every $v \in V - S$ by ε.

Claim. *The revised potential and the unchanged flow satisfy the optimality criteria.*

Repeat the procedure. Observe that in the new auxiliary digraph the set of

reachable nodes from s is strictly larger than S. Therefore, after at most $|V| - 1$ occurrences of case 1 either $\varepsilon = \infty$ or case 2 occurs.

Case 2: $t \in S$. Let P be a path in D' from s to t. Modify z as follows. Let $z'(u,v) = z(u,v) + 1$ if (u,v) is a forward arc of P and let $z'(u,v) = z(u,v) - 1$ if (v,u) is a backward arc of P.

Claim. *The revised flow and the unchanged potential satisfy the optimality criteria.*

What can we say about the complexity of the algorithm? We need roughly M flow augmentations. So the algorithm is polynomial if all the M minimum cost flows (of value $1, 2, \ldots, M$) are required. The algorithm is not necessarily polynomial if one wants to compute only a min-cost flow of value M since the complexity is proportional to M.

If the maximum capacity is not too big (namely, its value can be bounded by a polynomial of $|V|$, then the algorithm is (strongly) polynomial, irrespective of the cost function.

5.6. An application to partially ordered sets

Let $P = \{p_1, p_2, \ldots, p_n\}$ be a partially ordered set. Dilworth's (1950) famous theorem asserts that the maximum cardinality a of an antichain is equal to the minimum number of covering chains. Another result of this type asserts that the maximum cardinality c of a chain is equal to the minimum number of covering antichains. In this section we discuss common generalizations of these results.

For a family $\mathscr{B} = \{B_1, B_2, \ldots, B_k\}$ denote $\bigcup \mathscr{B} = \bigcup (B_i: i = 1, \ldots, k)$. By a *chain family* $\mathscr{C}_\gamma = \{C_1, C_2, \ldots, C_\gamma\}$ we mean a set of γ disjoint non-empty chains. Let $\boldsymbol{C}_\gamma$ denote the set of chain families of γ chains and $\boldsymbol{C}$ the set of all chain families. Let $c_\gamma = \min(|\bigcup C_\gamma|: \mathscr{C} \in \boldsymbol{C}_\gamma)$.

By an *antichain family* $\mathscr{A}_\alpha = \{A_1, A_2, \ldots, A_\alpha\}$ we mean a set of α disjoint non-empty antichains. Let $\boldsymbol{A}_\alpha$ denote the set of antichain families of α antichains and $\boldsymbol{A}$ the set of all antichains. Let $a_\alpha = \min(|\bigcup \mathscr{A}_\alpha|: \mathscr{A}_\alpha \in \boldsymbol{A}_\alpha)$.

By Dilworth's theorem $c_a = n$, by its polar $a_c = n$. What can be said about c_γ $(1 \le \gamma \le a)$ and about a_α $(1 \le \alpha \le c)$?

Theorem 5.4a (Greene and Kleitman 1976). $a_\alpha = \min(q\alpha + |P - \bigcup \mathscr{C}_q|: \mathscr{C}_q \in \boldsymbol{C})$.

Theorem 5.5a (Greene 1976). $c_\gamma = \min(q\gamma + |P - \bigcup \mathscr{A}_q|: \mathscr{A}_q \in \boldsymbol{A})$.

Since a chain and an antichain can share at most one element, a_α and c_γ do not exceed the minimum in question.

Definition. An antichain family $\mathscr{A}_\alpha = \{A_1, A_2, \ldots, A_\alpha)$ and a chain family $\mathscr{C}_\gamma = \{C_1, C_2, \ldots, C_\gamma\}$ are said to be *orthogonal* if

$$P = (\bigcup \mathscr{A}_\alpha) \cup (\bigcup \mathscr{C}_\gamma) \tag{a}$$

and

$$A_i \cap C_j \neq \emptyset \quad \text{for } 1 \leq i \leq \alpha,\ 1 \leq j \leq \gamma. \tag{b}$$

The non-trivial parts of Theorems 5.4a and 5.5a can be reformulated as follows.

Theorem 5.4b. *For every α, $1 \leq \alpha \leq c$, there are $\mathcal{A}_\alpha \in \mathbf{A}_\alpha$ and $\mathcal{C}_\gamma \in \mathbf{C}$, for some γ, which are orthogonal.*
Theorem 5.5b. *For every γ, $1 \leq \gamma \leq a$, there are $\mathcal{C}_\gamma \in \mathbf{C}$ and $\mathcal{A}_\alpha \in \mathbf{A}$, for some α, which are orthogonal.*

A common generalization is due to Frank (1980).

Theorem 5.6. *There exists a sequence $\mathcal{C}_a | \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{i_1} | \mathcal{C}_{a-1}, \mathcal{C}_{a-2}, \ldots, \mathcal{C}_{a-j_1} | \mathcal{A}_{i_1+1}, \ldots$ which arises as a combination of two sequences $\mathcal{C}_a, \mathcal{C}_{a-1}, \ldots, \mathcal{C}_1$ and $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_c$, where $\mathcal{C}_j \in \mathbf{C}$ and $\mathcal{A}_i \in \mathbf{A}$, with the property that any member of the sequence (whether $\mathcal{C}_j$ or $\mathcal{A}_i$) is orthogonal to the last member of other type preceding it. (That is, $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{i_1}$ are orthogonal to $\mathcal{C}_a$ and $\mathcal{C}_{a-1}, \mathcal{C}_{a-2}, \ldots, \mathcal{C}_{a-j_1}$ are orthogonal to $\mathcal{A}_{i_1}$, and so on.)*

Proof. Associate a digraph $D = (V, A)$ with P where $V := \{s, t, x_1, x_2, \ldots, x_n, y_1, y_2, \ldots y_n\}$, and $A := \{(s, x_i): i = 1, 2, \ldots, n\} \cup \{(y_i, t): i = 1, 2, \ldots, n\} \cup \{(x_i, y_j): \text{if } p_i \geq p_j\}$. Define all arc capacities $g(a)$ to be 1, while the costs are $c(e) = 1$ if $e = (x_i, y_i)$ and 0 otherwise.

Apply the min-cost flow algorithm to this network and let z and π be a flow and a potential at an intermediate stage of the algorithm. By analyzing the effect of a flow augmentation and a potential change and using the optimality criteria the following lemma can easily be proved.

Lemma 5.7. (a) *Either $\pi(y_i) = \pi(x_i)$ or $\pi(y_i) = \pi(x_i) + 1$.*
(b) *If $p_i > p_j$ and $z(x_i, y_j) = 1$, then $\pi(x_i) = \pi(y_j)$.*

The arcs (x_i, y_j) $(i < j)$ for which $z(x_i, y_j) = 1$ correspond to a chain family $\mathcal{C}_\gamma$ where $\gamma = n - \text{val}(z)$. For $\alpha = \pi(t)$ define a family $\mathcal{A}_\alpha = \{A_1, A_2, \ldots, A_\alpha\}$ where $A_i = \{p_j: \pi(x_j) + 1 = \pi(y_j) = i\}$.

Lemma 5.8. *$\mathcal{A}_\alpha$ is an antichain family and is orthogonal to $\mathcal{C}_\gamma$.*

The proof easily follows from the optimality criteria and from Lemma 5.7.

Now the Ford–Fulkerson algorithm and Lemma 5.8 immediately imply Theorem 5.6. □

6. Trees and arborescences

6.1. Minimum cost trees and arborescences

Given a connected graph $G = (V, E)$ and a cost function $c: E \to \mathbb{R}$, find a minimum cost spanning tree. This problem is one of the earliest combinatorial optimization problem that has been solved. For an excellent historical survey, see Graham and Hell (1985). For algorithmic details, see Tarjan (1983).

The following property of trees is crucial.

Lemma 6.1. *Let T_1 and T_2 be the edge sets of two trees on the same node set. Then for any edge $e \in T_1$ there is an edge $f \in T_2$ such that both $T_1 - e + f$ and $T_2 - f + e$ are trees.*

Proof. If $e \in T_2$, then $f := e$ will do. Suppose that $e = st \notin T_2$. $T_1 - e$ has two components C_1 and C_2. T_2 contains a path P connecting s and t. Let f be an edge of P that connects C_1 and C_2. This f satisfies the requirement of the lemma. □

Let T be a spanning tree of a graph $G = (V, E)$. A *fundamental cut* belonging to an element e of F is a cut of G determined by the two components of $T - e$. A *fundamental circuit* belonging to an edge $f = uv \in E - T$ is a circuit consisting of f and the unique path in T connecting u and v. Clearly, an edge $e \in T$ is in the fundamental cut of an edge $f \in E - T$ if and only if e is in the fundamental circuit of f.

Lemma 6.1 immediately implies the following.

Theorem 6.2. *For a spanning tree T of G the following are equivalent:*
(a) *T is of minimum cost.*
(b) *$c(e) \leq c(f)$ for any edge $e \in T$ and edge f of the fundamental cut of e.*
(c) *$c(e) \geq c(f)$ for any edge $e \notin T$ and edge e of the fundamental circuit of f.*

One of the simplest (and earliest) algorithms in combinatorial optimization is the greedy algorithm to construct a minimum-cost spanning tree of a connected graph $G = (V, E)$.

Greedy algorithm (Boruvka 1926, Kruskal 1956). The procedure consists of building a spanning forest by adding edges one by one. It starts with a forest of node-set V that has no edges and stops when the current forest is a spanning tree. The general step consists of adding an edge of minimum cost that connects two distinct components of the current forest.

There is another version of the greedy algorithm.

Dijkstra–Prim algorithm (Dijkstra 1959, Prim 1957). Choose an arbitrary node

x_0. Starting at x_0 build a tree edge by edge. At a general step choose a least cost edge to be added that has exactly one end in the current tree.

The following algorithm does not build a tree or forest directly but gets rid of edges of big cost and the remaining graph is the desired tree.

Reverse greedy algorithm. The procedure consists of discarding edges one by one so that the remaining graph is connected. At a general step choose an edge of maximum cost that is not a cut-edge of the current graph and delete it. The algorithm stops when the remaining graph is a spanning tree.

All of these algorithms can be formulated in a general framework.

General algorithm. The algorithm consists of applications of the following two operations in arbitrary order. The first operation builds a spanning forest F by adding edges one by one while the second operations deletes edges one by one. More precisely, let F denote the forest already constructed (at the beginning F is the forest of no edges.)

Step 1. If F is a spanning tree, halt. Otherwise, choose an arbitrary cut B disjoint from F and let $e \in B$ be a cheapest edge of B. Add e to F.

Step 2. Choose an arbitrary circuit C (if there is none, step 2 does not apply any longer) and let $e \in C$ be the most expensive edge of $C - F$. Delete e from G.

Theorem 6.3. *The final tree of the algorithm is of minimum cost.*

Proof. Any stage of the algorithm can be specified by a pair (F, D) of disjoint subsets of E where F denotes the forest constructed so far and D denotes the set of edges deleted so far.

We prove by induction that at each stage (F, D) of the algorithm there is a minimum cost tree T of G for which $F \subseteq T \subseteq E - D$. Any min-cost tree will do when $F = D = \emptyset$. Suppose we have already proved the statement for (F, D), that is, there is min-cost tree T with $F \subseteq T \subseteq E - D$.

Assume first that step 1 is applied and let $e \in B$ be the newly added edge and $F' := F + e$. If $e \in T$, we are done. Otherwise, let C_e be the fundamental circuit of e with respect to T. Edge e is in cut B and in circuit C_e therefore there must be another edge f in $B \cap C_e$. Since $e, f \in B$, by the rule in step 1 we have $c(e) \leqslant c(f)$. Since e, $f \in C_e$ and T is of minimum cost we have $c(f) \geqslant c(e)$. Hence $c(e) = c(f)$ and $T' := T - f + e$ is another min-cost tree for which $F' \subseteq T' \subseteq E - D$.

Second, assume that step 2 is applied and let $e \in C$ be the newly deleted edge. If T does not contain e, we are done. Otherwise let B_e be the fundamental cut of e belonging to T. There is an edge $f \neq e$ with $f \in C \cap B_e$. Since e, $f \in C$, by the rule in step 2 we have $c(e) \geqslant c(f)$. Since e, $f \in B_e$ and T is of minimum cost we have $c(e) \leqslant c(f)$. Hence $c(e) = c(f)$ and $T' := T - f + e$ is another min-cost tree for which $F \subseteq T' \subseteq E - (D + e)$. □

Remark. The above algorithm can be extended to matroids. That is, there is a greedy algorithm for finding a minimum cost basis of a matroid. (See chapter 11.)

Let us turn to a directed counter-part of the minimum cost spanning tree problem. Let us be given a digraph $D = (V, A)$ and a cost function $c: A \to \mathbb{R}_+$. Assume that every node can be reached from a specified node s, that is, D includes a spanning s-arborescence. Our next problem is to find a minimum cost s-arborescence.

This problem has been solved by Fulkerson. Note that the min-cost tree problem can be reduced to a min-cost arborescence problem: replace each edge of G by a pair of oppositely directed arcs.

Call a set-function $z: 2^{V-s} \to \mathbb{R}_+$ *c-feasible* if

$$c(a) \geqslant \sum (z(X): a \text{ enters } X) \quad \text{for every } a \in A\,. \tag{6.1}$$

Theorem 6.4 (Fulkerson 1974). *The minimum cost of a spanning s-arborescence is* $\max(\sum (z(x): X \subseteq V - s)$: z *is c-feasible*). *Furthermore, if c is integer-valued, the optimal z can be chosen integer-valued.*

Proof. Let F be a spanning s-arborescence and z a c-feasible vector. We have

$$\begin{aligned} c(F) &= \sum (c(a): a \in F) \\ &\geqslant \sum \Big(\sum (z(x): a \text{ enters } X): a \in F\Big) \\ &\geqslant \sum (z(X): X \subseteq V - s) \end{aligned} \tag{6.2}$$

from which $\max \leqslant \min$ follows. In (6.2) we have equality if the following *optimality criteria* hold.

$$c(a) = \sum (z(X): a \text{ enters } X) \text{ for every } a \in F\,, \tag{6.3a}$$

$$z(X) > 0 \text{ implies } \delta_F^-(X) = 1\,. \tag{6.3b}$$

The algorithm below finds a spanning s-arborescence F and a feasible z for which (6.3a–b) holds. It consists of two parts. The first part constructs z while the second constructs F. In the course of the first part we revise the cost function. The current cost function is denoted by c'. We call an arc a a 0-arc if $c'(a) = 0$.

Part 1. Iterate the following step. Choose a minimal set $X \subseteq V - s$ with no entering 0-arc. Define $z(X) := \min(c'(a)$: a enters X) and revise c' as follows. $c'(a) := c'(a) - z(X)$ if a enters X. The new c' is non-negative and its value is zero on at least one more arc.

Part 1 terminates if every set $X \subseteq V - s$ has an entering 0-arc, equivalently, there is a spanning s-arborescence consisting of 0-arcs.

Part 2. Starting at s and using only 0-arcs build up a spanning s-arborescence F. If, during the building process, there is more than one 0-arc leaving the sub-arborescence already constructed, choose that one which became a 0-arc earliest during the first part.

Obviously the constructed z is c-feasible and (6.3a) holds.

Lemma. *z and F satisfy* (6.3b).

Proof. Let X be a set with $z(X)>0$. If, indirectly, $\delta_F^-(X)>1$, there is a moment during part 2 when a sub-arborescence F' is at hand for which $\delta_{F'}^-(X)=1$ and the arc e currently added to F' enters X. Consider the moment of part 1 when $z(X)$ became positive. Then no 0-arc entered X and every proper non-empty subset of X had an entering 0-arc. In particular, there is a 0-arc f entering $X-V(F')$ which does not enter X. Therefore, when $z(X)$ became positive, f was a 0-arc while e was not and we are in a contradiction with the rule of part 2: arc f should have been chosen instead of e. □

A related problem is as follows. Given a digraph $D=(V,A)$ and a cost function d on A, find a maximum cost branching. We call a pair (p,y) a *covering*, where $p:V\to R_+$ is a non-negative function on V and $y:2^V\to\mathbb{R}_+$ a non-negative function on the subsets of V if $d(u,v)\leq p(v)+\sum(y(B): u,v\in B\subseteq V)$ for every $(u,v)\in A$. The *value* of a covering is $\sum(p(v): v\in V)+\sum(y(B)(|B|-1): B\subseteq V)$.

Theorem 6.5 (Edmonds 1967, Chu and Liu 1965). *The maximum cost of a branching of D is the minimum value of a covering. If d is integer-valued, the optimal covering can be chosen integer-valued.*

Proof. Obviously max ≤ min. To see the other direction, extend D by a new node s and new arcs (s,v) $(v\in V)$. Define a cost function c on the arcs of the extended digraph D', as follows. Let the cost of the new arcs of D' be $M:=\max(d(a): a\in A)$ and $c(a)=M-d(a)$ $(a\in A)$. By Theorem 6.4 there is a c-feasible vector $z:2^V\to\mathbb{R}_+$ and a spanning s-arborescence F of D' satisfying 6.3. Define $p(v):=M-\sum(z(B): v\in B)$ and for $|B|>1$ define $y(B):=z(B)$ $(B\subseteq V)$. It is straightforward that (p,y) is a covering and its value is equal to the d-cost of the branching $F\cap A$ of D. □

6.2. Packing and covering

A basic result on this field is due to Edmonds (1973).

Theorem 6.6. *Given a digraph $D=(V,A)$ with a specified node s, there are k pairwise arc-disjoint spanning arborescences rooted at s if and only if $\delta^-(X)\geq k$ for every $X\subseteq V-s$.*

Proof (Lovász 1976a, *sketch*). Starting at s we build up an arborescence F such that

$$\delta_{A-F}^-(X)\geq k-1 \quad \text{for every } X\subseteq V-s. \tag{$*$}$$

By induction on k, this will prove the sufficiency of the condition. Suppose that F is sub-arborescence satisfying $(*)$ which is not spanning (that is, $V(F)\neq V$). Call a set $X\subseteq V-s$ *critical* (with respect to F) if it satisfies $(*)$ with equality. By submodularity, if X and Y are critical and $X\cap Y\neq\emptyset$, then $X\cap Y$ and $X\cup Y$ are critical. By the hypothesis there is no critical subset of $V-V(F)$. Let M be a minimal critical subset for which $M-V(F)\neq\emptyset$. Then there is an arc $(u,v)\in A$ such that $u\in M\cap V(F)$ and $v\in M-V(F)$. This arc does not enter any critical set therefore the sub-arborescence $F'=F+(u,v)$ continues to satisfy $(*)$. (If no such an M exists, any arc (u,v) leaving $V(F)$ will do.) □

Note that Theorem 6.6 of Edmonds immediately implies the directed arc-disjoint version of Menger's theorem. Indeed, adjoin k parallel arcs from t to v for every $v\in V-\{s,t\}$ and apply Edmonds' theorem.

The following conjecture is a kind of node-disjoint counterpart of Theorem 6.6.

Conjecture 6.7. (a) Suppose that, given a digraph $D=(V,A)$ and a specified node $s\in V$, there are k openly disjoint paths from s to any other node of D. Then there are k arc-disjoint s-arborescences of D such that for any node $v\in V-s$ the k paths from s to v uniquely determined by the arborescences are openly disjoint.

(b) Suppose that, given a graph $G=(V,E)$ and a specified node $s\in V$, there are k openly disjoint paths from s to any other node of G. Then there are k spanning trees of G such that for any node $v\in V-s$ the k paths from s to v uniquely determined by the k trees are openly disjoint.

(c) The same as (b) except replace "openly disjoint" by "edge-disjoint".

Conjecture 6.7 (a) easily implies Conjecture 6.7 (b). Whitty (1986) proved Conjecture 6.7 (a) for $k=2$ while Conjecture 6.7 (b) has been proved for $k=3$ by Cheriyan and Maheshwari (1988) and by Zehavi and Itai (1989). Conjecture 6.7 (c) is proved for $k=2$ (using the ear-decomposition of 2-edge-connected graphs).

Recently, A. Huck proved (a) for acyclic digraphs and disproved it for general digraphs when $k\geq 3$.

One may be interested in finding k arc-disjoint spanning arborescences which need not be rooted at the same node.

Theorem 6.8. *There are k arc-disjoint spanning arborescences if and only if*

$$\sum_{i=1}^{t}\delta^-(X_i)\geq k(t-1)$$

for every family of disjoint non-empty sets $X_1, X_2, \ldots, X_t$.

More generally, the arborescence packing problem can be solved when lower and upper bounds are imposed at every node for the number of arborescences rooted at that node.

Another extension of Edmonds' theorem is due to Schrijver (1982). Let

$D=(V,A)$ be a digraph and let V_1, V_2 be a bipartition of V. Call a subset B of arcs a *bi-branching* if $\delta_B^-(X) \geq 1$ for every $X \subseteq V_1$ and $X \supseteq V_2$.

Theorem 6.9. (Schrijver 1982). *There are k arc-disjoint bi-branchings if and only if $\delta^-(X) \geq k$ for every $X \subseteq V_1$ and for every $X \supseteq V_2$.*

When $|V_2| = 1$ we are back at Theorem 6.6. When both V_1 and V_2 are independent König's edge coloring theorem is obtained as a special case. (See chapter 3.)

Schrijver used this result to prove the following conjecture of Woodall for acyclic digraphs in the special case where each sink can be reached from each source.

Conjecture 6.10. In an acyclic digraph if every directed cut contains at least k arcs, there are k disjoint sets of arcs each of which covers all directed cuts.

A counterpart of Theorem 6.6 is due to Vidyasankar (1978).

Theorem 6.11. *Let s be a specified node of a digraph $D=(V,A)$ with no entering arc. The arcs of D can be covered by k spanning s-arborescences if and only if* (i) *$\delta^-(v) \leq k$ for $v \in V-s$ and* (ii) *$k - \delta^-(X) \leq \sum (k - \delta^-(v): v \in T(X))$ for every $X \subseteq V - s$ where $T(X) := \{v \in X$: there is an arc $(u,v) \in A$ with $u \in V - X\}$.*

Proof. By elementary construction. For every $v \in V-s$ adjoin to D a copy of v, denoted by v', and k parallel arcs from v to v' and $k - \delta^-(v)$ parallel arcs from v' to v. Furthermore, for $(u,v) \in A$ adjoin k parallel arcs from u to v'. Apply Theorem 6.6 to the extended digraph D' and observe that k arc-disjoint spanning arborescences in D' correspond to k covering s-arborescences in D. Moreover, if $\delta'^-(X') < k$ for some $X' \subseteq V' - s$, then $X = \{v \in X': v' \notin X'\}$ violates (ii). □

Another interesting consequence of Edmonds' theorem is the following.

Theorem 6.12. *The arc-set of a digraph $D=(V,A)$ can be covered by k branchings if and only if* (i) *$\delta^-(v) \leq k$ for every $v \in V$ and* (ii) *$|A(X)| \leq k(|X|-1)$ for every $X \subseteq V$ (where $A(X)$ denotes the set of arcs induced by X).*

Proof. By elementary construction. Adjoin a new node s to V and for $v \in V$ adjoin $k - \delta^-(v)$ parallel arcs from s to v. In the new digraph D' we have

$$\begin{aligned}\delta'^-(X) &= \delta^-(X) + \textstyle\sum (k - \delta^-(v):\ v \in X)\\ &= \delta^-(X) - \delta^-(X) - |A(X)| + k|X| \geq k\end{aligned}$$

for every $X' \subseteq V$. By Theorem 6.6 there are k arc-disjoint spanning s-arborescences in D'. These determine k covering branchings of D. □

For undirected graphs we have the following (historically earlier) theorem by Nash-Williams (1964).

Theorem 6.13. *The edge set of an undirected graph $G=(V,E)$ can be covered by k forests if and only if $|E(X)| \leq k(|X|-1)$ for every $X \subseteq V$.*

Proof. Theorem 6.12 and the following easy lemma imply the result. □

Lemma. *The edges of a graph $G=(V,E)$ have an orientation for which $\delta^-(v) \leq k$ for every $v \in V$ if and only if*

$$|E(X)| \leq k|X| \quad \textit{for every } X \subseteq V . \tag{$*$}$$

Proof. (*Sufficiency*) In an orientation of G call a node v *bad* if $\delta^-(v) > k$. Choose an orientation where the "badness" $\sum (\delta^-(v) - k$: v bad) is minimal. If there is no bad node, we are done. Otherwise, let t be a bad node and let X be the set of nodes from which t is reachable in the current orientation. Then X contains a node s with $\delta^-(s) < k$ since otherwise $|E(X)| = \sum (\delta^-(v): v \in X) > k|X|$, contradicting ($*$). Reorienting the arcs of a dipath from s to t results in an orientation with smaller badness. □

(Note that this lemma easily follows from König's theorem, as well). For connected graphs the problems of covering the edge set by k forests or by k spanning trees are clearly equivalent. The packing problem of spanning trees was solved by Tutte (1961a).

Theorem 6.14. *A connected graph $G=(V,E)$ contains k pairwise edge-disjoint spanning trees if and only if $e_t \geq k(t-1)$ holds for every partition $\{V_1, V_2, \ldots, V_t\}$ of V ($V_i \neq \emptyset$) where e_t denotes the number of edges connecting different V_i.*

Remark. Edmonds extended Tutte's theorem to matroids by providing a good characterization of the existence of k disjoint bases of a matroid. See chapter 11.

7. Higher connectivity

7.1 Connectivity between two nodes

We start this section with a result which is undoubtedly the central theorem of this whole chapter, the Menger (1927) theorem. In what follows s and t are two specified nodes of the graph or digraph in question. A set of paths is called *openly disjoint* if the paths are pairwise disjoint except, possibly, for their end nodes.

Theorem 7.1. (a) *In a digraph (graph) the maximum number of arc-disjoint (edge-disjoint) st-paths is equal to the minimum number of arcs (edges) covering all* *st-paths. (Moreover, the minimum is attained on a set of type $\Delta^+(S)$ where $S \subseteq V$ is an $s\bar{t}$-set.)*

(b) *In a digraph (graph) if there is no arc (edge) from s to t, the maximum number of openly disjoint st-paths is equal to the minimum number of nodes distinct from s and t covering all st-paths.*

Actually this is four theorems according to whether we consider directed or undirected and edge-(arc-)disjoint or openly disjoint *st*-paths. Menger originally proved the undirected, openly disjoint version.

Proof. We have already seen two proofs for the arc-disjoint case (as a consequence of the max-flow min-cut theorem and a consequence of Edmonds' disjoint arborescences theorem). From this the other three cases follow by elementary construction. Namely, in case (a) replace each edge by a pair of oppositely directed arcs and observe that if there is a set of k arc-disjoint paths in the resulting digraph, then there is one that does not use both arcs assigned to an original edge. The same construction yields the undirected openly-disjoint version from the directed one.

To see the directed openly-disjoint version construct a new digraph D' from D, as follows. Replace each node v ($v \neq s, t$) of D by a pair of new nodes v' and v''. Let (v', v'') be an arc of D' and for an arc (u, v) of D let (u'', v') be an arc of D'. Arc-disjoint *st*-paths in D' correspond to openly disjoint paths in D. Moreover, if there are k arcs in D' covering all *st*-paths, then these arcs can be assumed to be of type (v', v'') and this set of arcs corresponds to a set of k nodes of D covering all *st*-paths. □

There exist other versions of Menger's theorem. For example, given a graph and two disjoint subsets S, T of its node set, there are k disjoint paths between S and T if and only if there are no $k-1$ nodes covering all such paths. By elementary construction this result easily follows from the original Menger theorem. Yet another version, sometimes called the *fan lemma*, is as follows. Let s be a node of a graph and T a subset of nodes not containing s. There are k paths connecting s and some elements of T so that they are disjoint except at s if and only if there are no $k-1$ nodes in $V - s$ covering all such paths.

Hoffman found the following unifying approach to the different versions of Menger's theorem. Let S be a finite set and let $\mathcal{P}$ be a set of ordered subsets of S. We call the members of $\mathcal{P}$ paths. Suppose that for any two paths $P = \{p_1, p_2, \ldots, p_k\}$ and $T = \{t_1, t_2, \ldots, t_l\}$ sharing an element $p_i = t_j$ the sequence $\{p_1, \ldots, p_i, t_{j+1}, \ldots, t_l\}$ includes a path.

Theorem 7.2 (Hoffman 1974). *The maximum number of disjoint paths is equal to the minimum number of elements covering all the paths.*

As a consequence, a Menger-type theorem can be formulated for disjoint shortest paths.

Corollary 7.3. *The maximum number of openly disjoint shortest paths from s to t is equal to the minimum number of nodes covering all shortest paths from s to t.*

This type of min–max results fails to be true for paths of bounded length in general, however we have the following.

Theorem 7.4 (Lovász et al. 1978). *Let $G = (V, E)$ be an undirected graph with two specified non-adjacent nodes s and t. The maximum number of openly disjoint st-paths of length at most k ($k \geq 2$) is at least $2/k$ times the minimum number of nodes (distinct from s, t) covering all st-paths of length at most k.*

A possible generalization of Menger's theorem (undirected, openly disjoint) is the following. In an undirected graph given a subset of nodes T, what is the maximum number of disjoint paths connecting nodes of T. This problem was answered by Gallai (see Corollary 8.25). Mader found a min–max formula for the maximum number of openly disjoint paths with end nodes in T. (See Theorem 8.24.)

7.2. Global connectivity

Let k be a positive integer. A graph $G = (V, E)$ is called *k-connected* (sometimes k-node-connected) if $|V| > k$ and for any subset $X \subseteq V$ with less than k elements $G(V - X)$ is connected. G is called *k-edge-connected* if deleting any subset of edges of less than k elements leaves a connected graph. This is equivalent to requiring $d^+(X) \geq k$ for any $\emptyset \neq X \subset V$. A digraph $D = (V, A)$ is called *k-arc-connected* (often the term strongly k-arc-connected is used) if deleting any subset of arcs of less than k elements leaves a strongly connected digraph. This is equivalent to saying that $\delta^+(X) \geq k$ for any $\emptyset \neq X \subset V$. We will say that a digraph D with a specified node s is k-arc-connected from s if $\delta^-(X) \geq k$ for any subset $X \subseteq V - s$. By Menger's theorem this is equivalent to the property that there are k arc-disjoint paths from s to any other node. This is, in turn, equivalent to that there are k spanning arborescences rooted at s (Theorem 6.6). Finally, let us call a digraph strongly k-connected if deleting any subset of nodes of less than k elements leaves a strongly connected digraph.

These definitions reflect one of our intuitive expectations for a graph to be "pretty much connected": it is not possible to destroy the connectivity by taking away a small part of the graph. Another natural definition for high connectivity is that there are many disjoint paths between any pair of nodes. The following result, due to Whitney, says that these two approaches coincide. (One may have other intuitions for high connectivity: for example, if the graph contains k edge-disjoint spanning trees. Let us call such a graph *k-tree-connected*. For a characterization of k-tree-connected graphs, see Theorems 6.13 and 7.11).

Theorem 7.5 (Whitney 1932). *A graph on more than k nodes is k-connected if and only if there are k openly disjoint paths between any two nodes. A graph (digraph)* *is k-edge-(arc-)connected if and only if there are k-edge-(arc-)disjoint paths from any node to any other.*

Proof. The second part immediately follows from Theorem 7.1. To see the non-trivial direction of the first part let s and t be two nodes. If they are not adjacent, we are done by Menger. Otherwise let e be an edge connecting s and t. If there are no $k-1$ openly disjoint st-paths in $G-e$, there is (by Menger) a subset of nodes of at most $k-2$ elements not containing s and t for which s and t belong to different components of $G-e-X$. Since G has more than k nodes either $X+s$ or $X+t$ is a disconnecting set of $k-1$ elements. □

Robbins' (1939) theorem (Corollary 2.13) described a relation between 2-edge-connected graphs and strongly connected digraphs. The following generalization is due to Nash-Williams (1960).

Theorem 7.6. *An undirected graph is 2k-edge-connected if and only if it has an orientation which is k-arc-connected.*

Actually Nash-Williams proved a much stronger result:

Theorem 7.7. *An undirected graph has an orientation such that for every ordered pair (x, y) of nodes there are $\lfloor \lambda(x, y)/2 \rfloor$ arc-disjoint xy-paths where $\lambda(x, y)$ denotes the maximum number of edge-disjoint xy-paths.*

One may be interested in the existence of a (strongly) k-connected orientation. The ("sufficiency" part of the) following conjecture is open even for $k=2$.

Conjecture 7.8. A graph $G=(V, E)$ has a k-connected orientation if and only if deleting any subset X of j nodes $(0 \leq j \leq k-1)$ results in a $2(k-j)$-edge-connected graph.

In section 2 we saw an ear-decomposition theorem for strongly-connected digraphs. This can be interpreted so that every strongly connected digraph can be obtained from a node by consecutively adding and subdividing arcs. Subdividing an arc (u, v) with a new node z means that we replace arc (u, v) by arcs (u, z) and (z, v) where z is a new node. The following generalization is due to Mader.

Theorem 7.9 (Mader 1982). *A digraph D is k-arc-connected if and only if D can be obtained from a node by adding arcs (connecting old nodes) and applying operation O_k:*

Operation O_k: Pick up k arbitrary arcs, subdivide them by nodes $z_1, \ldots, z_k$ and identify $z_1, \ldots, z_k$ to a new node z.

Corollary 7.10. *A digraph D (with $\delta^-(s)=0$) is k-arc-connected from a node s if and only if D can be built up from s by repeated applications of the following operation: For some j, $0 \leq j \leq k$, first apply O_j and add then $k-j$ new arcs entering z.*

Using this characterization one can easily derive Edmonds' Theorem 6.6 on arc-disjoint arborescences (see Mader 1983). Also we have the following.

Corollary 7.11. *A graph $G=(V, E)$ is k-tree-connected if and only if G can be obtained from a node by sequentially adding edges (connecting old nodes) and applying the following operation: choose j $(0 \leq j \leq k)$ distinct edges and $k-j$ (not necessarily distinct) nodes, subdivide the j edges by j nodes, identify these j nodes to a new node z and connect z and the $k-j$ old nodes by $k-j$ edges.*

Theorems 7.6 and 7.9 imply a result of Lovász.

Theorem 7.12 (Lovász 1979, Problem 6.53). *An undirected graph G is 2k-edge-connected if and only if G can be obtained from a node by adding edges and applying Operation Q_k: pick up k arbitrary edges, subdivide them by nodes $z_1, \ldots, z_k$ and identify $z_1, \ldots, z_k$ into a new node z.*

Notice on the other hand that Theorem 7.12 implies Theorem 7.6. What about $(2k+1)$-edge-connected graphs? We need two other operations.

Operation Q_k^+: Proceed as in Q_k, then choose a node x of the graph and add a new edge joining x and the new z.

Operation Q_k^2: Proceed as in Q_k thereby constructing G', choose k distinct edges $e'_1, \ldots, e'_k$ of G' not all incident to z, subdivide each e'_i by a node z'_i, identify the z'_i's into a new node z' and add a new edge joining z and z'.

Theorem 7.13 (Mader 1978a). *A graph G is $(2k+1)$-edge-connected if and only if G can be obtained from a node by successive addition of edges and repeated applications of Q_k^+ and Q_k^2.*

Let us be given a k-edge-connected undirected graph $G=(V, E)$. It is not difficult to prove that for k odd the k-element cuts are pairwise non-crossing. (Two cuts $\Delta(X), \Delta(Y)$ are called *crossing* if none of $X-Y$, $Y-X$, $X \cap Y$, $V-(X \cup Y)$ is empty). Dinits et al. (1976) showed that the structure of minimum cuts can also be described when k is even. Let $k=2l$.

Let us call a 2-edge-connected (loop-free) graph $T=(U, F)$ a *circuit-tree* if each block of T is a (possibly 2-element) circuit. Intuitively, T consists of edge-disjoint circuits which are joined to each other in a tree-like manner.

Any minimum cut of T consists of two edges belonging to the same circuit of T. If we replace each edge of T by l parallel edges, we obtain a $2l$-edge-connected graph T'. Clearly, the minimum cuts of T' correspond to the minimum cuts of T.

The content of the next theorem is that the structure of minimum cuts of every $2l$-edge-connected graph can be described with the help of a circuit-tree.

Theorem 7.14 (Dinits et al. 1976). *Let $G=(V,E)$ be a $2l$-edge-connected graph. There exists a circuit-tree $T=(U,F)$ and a mapping $\varphi: V \to U$ so that for every minimum cut of T determined by a partition $[X, U-X]$ of U the cut of G determined by the partition $[\varphi^{-1}(X), \varphi^{-1}(U-X)]$ is a minimum (i.e., of $2l$ elements) cut of G and every minimum cut of G arises this way.*

We close this subsection by mentioning two results on constructing k-edge connected graphs and k-arc-connected digraphs.

Theorem 7.15. (a) (Watanabe and Nakamura 1987) *An undirected graph $G=(V,E)$ can be made k-edge-connected ($k \geq 2$) by adding at most γ new edges if and only if*

$$\sum (k - d(X_i): i = 1, \ldots, t) \leq 2\gamma$$

holds for every family $\{X_i\}$ of disjoint non-empty subsets of V.

(b) (Frank 1992a) *A directed graph $D=(V,A)$ can be made k-edge-connected ($k \geq 1$) by adding at most γ new edges if and only if*

$$\sum (k - \delta^-(X_i): i = 1, \ldots, t) \leq \gamma \quad and \quad \sum (k - \delta^+(X_i): i = 1, \ldots, t) \leq \gamma$$

holds for every family $\{X_i\}$ of disjoint non-empty subsets of V.

In Frank (1992a) the first part has been generalized to the case when the prescribed edge-connectivity between each pair of nodes is arbitrary (and not necessarily the same number k). Also augmentations with minimum node-costs are tractable. For a survey see Frank (1994).

7.3. 3-connected graphs

In the preceding subsection we saw how to construct all the k-edge-connected graphs. As far as node-connectivity is concerned there is an ear-decomposition result for 2-connected graphs (Proposition 2.6). Tutte (1966) developed a theory for decomposing a 2-connected graph into 3-connected "components". (The reader is refered to the original work since even formulating the result needs too much space.)

Unfortunately there are no known analogous constructions for k-connected graphs, in general. For 3-connectivity, however, the situation is much better. In order to be able to work with 3-connected graphs we must have "reductions" that preserve 3-connectivity. Two simple reductions are deleting and contracting an edge e. We use the notation $G-e$ and G/e for the graphs arising from G by deleting and contracting e, respectively. For 2-connectivity we saw (Proposition 2.7) that any edge of a 2-connected graph can be either deleted or contracted without destroying 2-connectivity. For 3-connectivity one has the following.

Theorem 7.16 (Tutte 1966, Theorem 12.65). *If e is any edge of a 3-connected graph on at least four nodes, then either G/e is 3-connected or $G-e$ is a subdivision of a 3-connected graph.*

Proof (Thomassen 1984). Suppose G/e is not 3-connected for an edge $e=xy$. Then there is a node z such that $G' = G - \{x,y,z\}$ is not connected. What we have to show is that there are three openly disjoint paths between x' and y' not using e for any two nodes x', y' distinct from x and y. In G there are three openly disjoint paths connecting x' and y'. If one of these contains e, then x' and y' belong to the same component of G' or one of x', y' equals z. Then there is a component C of G' not containing x' and y' and there is a path P in $C + \{x,y\} - e$ connecting x and y. But now replacing e by P we obtain three openly disjoint paths between x' and y' not using e. □

We say that an edge e of a 3-connected graph is *contractible* if G/e is 3-connected. The next result, due to Tutte (1961b) shows that there always exists a contractible edge.

Theorem 7.17. *A 3-connected graph $G=(V,E)$ with at least five nodes has a contractible edge.*

Proof (*sketch*, Thomassen 1980b). If an edge xy is not good, there is a node z such that $\{x,y,z\}$ is a disconnecting set of G. Choose xy in such a way that the largest component C of $G - \{x,y,z\}$ is as big as possible. Let C' be another component of $G - \{x,y,z\}$ and $u \in V(C')$ such that $uz \in E$. The contraction of uz leaves a 3-connected graph. □

One way to generate new 3-connected graphs is applying the following splitting operation (that may be considered as a converse to contracting an edge). Note that the graph may have parallel edges.

Operation S: Choose a node v of degree at least four. Partition the edges incident to v into two parts E_1 and E_2 so that $|\{u: uv \in E_i\}| \geq 2$ for $i=1,2$. Replace v by two nodes v_1 and v_2, replace each edge $vu \in E_i$ by an edge v_iu ($i=1,2$) and join v_1 and v_2 by an edge.

Theorem 7.17 provides a kind of converse.

Corollary 7.18. *A (not necessarily simple) graph G is 3-connected if and only if G can be obtained from K_4 by repeatedly adding edges (connecting old nodes) and applying operation S.*

A slight drawback of this theorem is that, though parallel edges do not play any role in 3-connectivity, it may happen that even if the 3-connected graph G to be constructed is simple the graphs occurring in the intermediate steps are not. This is the case, for example, if G is a wheel (bigger than K_4). (A *wheel* is a circuit plus an extra node connected to all nodes of the circuit.) In a sense wheels are the only essential examples of this type since Tutte (1961b) proved the following.

Theorem 7.19. *A simple graph is 3-connected if and only if G can be obtained from a wheel by repeatedly adding edges connecting non-adjacent old nodes and applying operation S.*

This result is a reformulation of the following one.

Theorem 7.20 (Tutte 1961b). *A 3-connected graph G is either*
(a) *a wheel or*
(b) *contains an edge e for which $G - e$ is 3-connected or*
(c) *contains a contractible edge which is not in a triangle.*

Proof (Thomassen 1984). Suppose that neither (b) nor (c) occurs. By Theorem 7.17 there is a triangle T. Let $V(T) = \{x, y, z\}$. We claim that at least two nodes of T have degree three. Suppose, indirectly, x has neighbours x_1, x_2 not in T and y has neighbours y_1, y_2 not in T. Since $G - xy$ is not 3-connected, there is a node z' such that $G' = G - \{z, z'\} - xy$ is not connected. Let G_x and G_y denote the components of G' containing x and y, respectively. Since $G - y$ is 2-connected, there are two disjoint paths P_1, P_2 from $\{y_1, y_2\}$ to $\{z', z\}$. One of x_1 and x_2, say x_1, is distinct from z'. In $G - x$ there are two openly disjoint paths Q_1, Q_2 from x_1 to $\{z', z\}$. Since P_1, P_2 are in $G_x + \{z', z\}$ and Q_1, Q_2 are in $G_y + \{z', z\}$, $P'_1 = (P_1 + Q_1 + Q_2)$ is a path from y to z. Now P'_1, P_2 and $P_3 = (yx, xz)$ are three openly disjoint paths between y and z, therefore edge yz satisfies (b), a contradiction.

So T has at least two nodes of degree 3, say x and y. If x' ($\neq y, z$) is a neighbour of x, then it is easy to see that G/xx' is 3-connected. Hence xx' must be in a triangle. The third node of this triangle must be z (unless $G = K_4$) and we conclude that x' has degree 3. We then consider the neighbour x'' of x' distinct from x and z and continuing this way we see that G is a wheel with center z. □

Obviously Tutte's theorem, in turn, implies Theorem 7.17. Theorem 7.17 is a highly powerful device in proving results apparently not related to connectivity. For example, with the help of it, Thomassen found an easy proof of Kuratowski's theorem on planarity of graphs as well as Tutte's theorem stating that every 3-connected planar graph has a convex representation in the plane. Note that Tutte originally used a different approach. He relied on the following result.

Theorem 7.21 (Tutte 1963). *Every edge uv of a* 3-connected *graph is contained in two peripherical circuits C_1 and C_2 (that is a chordless circuit the deletion of which results in a connected graph) for which $V(C_1) \cap V(C_2) = \{u, v\}$ A* 3-connected *graph is planar if and only if every edge is contained in exactly two peripherical circuits.*

Note that the first part of the theorem is straightforward for planar graphs as the two circuits determined by the faces incident to uv satisfy the requirements. Here are three variations of Theorem 7.17.

Theorem 7.22 (Thomassen and Toft 1981). *Every simple 3-connected graph with no triangles contains a circuit C such that every edge of C is contractible.*

Theorem 7.23 (Halin 1969a). *If v is a node of degree 3 in a 3-connected graph, then there is an edge uv for which G/uv is 3-connected.*

Theorem 7.24 (Ando et al. 1987). *Every 3-connected graph $G = (V, E)$ with $|V| > 4$ has at least $|V|/2$ contractible edges.*

There is a counterpart of Theorem 7.17.

Theorem 7.25 (Barnette and Grünbaum 1969, Titov 1975). *Every 3-connected graph with at least five nodes has an edge e such that $G - e$ is a subdivision of a 3-connected graph.*

Proof. If there is no such an edge, then by Theorem 7.16 every edge is contractible without destroying 3-connectivity. Let G' be a graph obtained by contracting an edge st. By induction there is an edge e of G' such that $G' - e$ is a subdivision of a 3-connected graph. If $G - e$ is not a subdivision of a 3-connected graph, then there are nodes x, y such that $G - \{x, y\} - e$ has a component consisting of two nodes and e is incident to one of them, denoted by z. Since G is 3-connected there is an edge f from z to $\{x, y\}$. But $\{z, x, y\}$ is a separating set of G so the contraction of f destroys 3-connectivity. □

Corollary 7.26. *A graph $G = (V, E)$ is 3-connected if and only if G can be obtained from K_4 by sequentially adding edges and applying the following operations:*
(a) *Pick up two non-parallel edges, subdivide them by nodes u, v and join u and v by an edge.*
(b) *Pick up an edge xy and a node v ($\neq x, y$), subdivide xy by a node u and join u and v by an edge.*

Barnette and Grünbaum (1969) used Theorem 7.25 to provide a short proof of a theorem of Steinitz stating that the 1-skeletons of the 3-dimensional polytopes are precisely the 3-connected planar graphs.

Finally, here is a theorem consisting deletion of nodes rather than edges.

Theorem 7.27 (Chartrand et al. 1972). *Every 3-connected graph of minimum degree at least 4 has a node v such that $G - v$ is 3-connected.*

7.4. Preserving connectivity

In the preceding subsection on 3-connected graphs we have encountered theorems saying that a 3-connected graph remains 3-connected under certain operations (Theorems 7.16, 7.17, 7.27). In this part we survey further operations that preserve connectivity properties of graphs. As far as node-connectivity is concerned, only a few general reduction results are known. Here is one.

Theorem 7.28 (Thomassen 1981). *A k-connected graph with no triangle contains an edge whose contraction results in a k-connected graph.*

The situation is much better for edge connectivity. Let G be a graph (or digraph) and let $e = uz$ and $f = zv$ be two edges (or arcs) of G incident to a common node z. We say that a pair $\{e, f\}$ is split off (at z) if we replace e and f by a new edge (arc) uv. The resulting graph (digraph) is denoted by G^{ef}.

The following fundamental result is due to Mader (1978a).

Theorem 7.29. *Let $G = (V, E)$ be a (not necessarily simple but loopless) graph and z a node of degree at least 4 so that there is no cut-edge incident to z. There exist edges e, f incident to z such that $\lambda(x, y; G) = \lambda(x, y; G^{ef})$ for every pair of distinct nodes x, y different from z. (Here $\lambda(x, y; G)$ denotes the maximum number of edge-disjoint paths in G connecting x and y).*

A relatively simple proof can be found in Frank (1992b). Mader used it to derive Nash-Williams' orientation result (Theorem 7.7). Theorem 7.13 was also obtained from this result. Theorem 7.6 follows already from a weaker form of Theorem 7.29 due to Lovász (1979, Problem 6.53): If z is a node of even degree and $\lambda(x, y{:}\ G) \geqslant k$ whenever $x, y \in V - z$, then for any edges e incident to z there is an edge f incident to z such that $\lambda(x, y{:}\ G^{ef}) \geqslant k$ for every x, y $(\neq z)$. While the directed analogue of Theorem 7.29 is not true in general the counterpart of Lovász' version holds, as follows.

Theorem 7.30 (Mader 1983). *Let $D = (V, A)$ be a digraph and z a node for which $\delta^-(z) = \delta^+(z)$. Suppose that $\lambda(x, y{:}\ D) \geqslant k$ for every x, $y \in V - z$. For any arc entering z there is an arc f leaving z such that $\lambda(x, y{:}\ D^{ef}) \geqslant k$ for every x, $y \in V - z$.*

This result is one ingredient to Theorem 7.9. The other one will be mentioned later (Theorem 7.41).

If we restrict ourselves to Eulerian digraphs (that is, $\delta^-(z) = \delta^+(z)$ for every node z) then the counterpart of Theorem 7.28 does hold (although it is much easier).

Theorem 7.31 (Frank 1989, Jackson 1988). *Let $D = (V, A)$ be an Eulerian digraph and z a node. For any arc e entering z there is an arc f leaving z such that $\lambda(x, y{:}\ D) = \lambda(x, y{:}\ D^{ef})$ for every x, $y \in V - z$.*

Sometimes we can maintain connectivity properties under "bigger" reductions.

Theorem 7.32 (Thomassen and Toft 1981). *Every 3-connected graph with minimum degree at least 4 contains a circuit whose contraction results in a 3-connected graph.*

Theorem 7.33 (Jackson 1980). *A simple 2-connected graph of minimum degree at least 4 contains a circuit C such that the removal of its edges leaves the graph 2-connected and, in addition* (Thomassen and Toft 1981), *$G - V(C)$ is connected. The first part holds for not necessarily simple planar graphs* (Fleischner and Jackson 1985).

Theorem 7.34 (Thomassen and Toft 1981). *Every simple 2-connected graph G of minimum degree at least 3 contains an induced circuit C (that is a circuit without chords) such that $G - V(C)$ is connected.*

The same conclusion was proved by Tutte for 3-connected graphs (see Theorem 7.21).

Theorem 7.35 (Thomassen and Toft 1981). *If G is a 2-connected graph with minimum degree at least 5, then G contains an induced circuit such that $G - V(C)$ is 2-connected. If G is 3-connected with minimum degree at least 4, then G has a circuit such that $G - V(C)$ is a block.*

For higher node connectivity we have the following results.

Theorem 7.36 (Thomassen 1981). *Every $(k + 3)$-connected graph G contains an induced circuit C such that $G - V(C)$ is k-connected.*

Theorem 7.37 (Egawa 1987). *Every $(k + 2)$-connected triangle-free graph G contains an induced cycle C such that $G - V(C)$ is k-connected.*

Theorem 7.38 (Mader 1974a). *Every k-connected graph G with minimum degree at least $k + 2$ contains a circuit C such that $G - E(C)$ is k-connected.*

For edge-connectivity Mader proved the following.

Theorem 7.39 (Mader 1985a). *For every pair of nodes s, t of a connected graph there is a path P connecting s and t such that deleting the edges of P the local connectivity $\lambda(x, y)$, for any pair x, y of nodes, can decrease by at most two.*

A directed counterpart of this result is also due to Mader.

Theorem 7.40 (Mader 1981). *For every pair of nodes s, t of a k-arc-connected digraph there is a path P from s to t such that deleting the arcs of P leaves the digraph $(k - 1)$-arc-connected.*

7.5. *Minimal and critical graphs*

In graph theory it is a typical way to prove things by starting with a graph critical (or minimal) with respect to a certain property. For example, if this property is "having no perfect matching" we arrive at the concept of factor-critical graphs (see chapter 3). Therefore, it is a general program to investigate "critical" graphs. Typically, we use the adjective "minimal" (resp. critical) if deleting any edge (resp. node) destroys the property considered.

Call a graph (digraph) G minimally k-edge-connected (k-arc-connected) if G is k-edge-connected (k-arc-connected) but $G-e$ is not for each edge (arc) e. Similarly, a graph G is minimally k-connected if G is k-connected but $G-e$ is not for each edge e. Strongly minimally k-connected (or, briefly, minimally k-connected digraphs) are defined analogously.

A k-connected (k-edge-connected) graph is called critically k-connected (critically k-edge-connected) if deleting any node destroys k-connectivity (k-edge-connectivity). The corresponding notions for digraphs are defined analogously.

Actually there are eight classes to be investigated corresponding to the possible choices: directed or undirected graph, edge- (arc-)connectivity or node-connectivity, critical or minimal. There are interesting results concerning each but one of these classes (critically k-arc-connected digraphs have not yet been investigated). Here we list only the most important theorems. The starting point is a theorem due to Halin.

Theorem 7.41. (a) (Halin 1969a) *Every minimally k-connected graph has at least one node of degree k.*

(b) *Every minimally k-edge-connected graph G (with at least two nodes) contains a node of degree k.*

Mader extended these results.

Theorem 7.42 (Mader 1972). *Every minimally k-connected graph contains at least $k+1$ nodes of degree k. Furthermore, every circuit of G contains a node of degree k* (Mader 1971b). *Every minimally k-edge-connected simple graph contains at least $k+1$ nodes of degree k.*

Another interesting generalization of Halin's result is also due to Mader (1973).

Theorem 7.43. *In a simple graph if every degree is at least $k+1$, there are two adjacent nodes s and t which are connected by $k+1$ openly disjoint paths.*

Let us see critical graphs.

Theorem 7.44 (Mader 1986). *Every critically k-edge-connected simple graph G contains a node of degree k. Furthermore, G contains a node x such that $G-x$ is $(k-1)$-edge-connected.*

To formulate results on critically k-connected graphs we need the following concepts. In a k-connected graph G a set X is a separating set if $G-X$ is not connected. A subset C of nodes called an *end* of G if C is one of the components in $G-X$ for a k-element separating set X. An *atom* is an end with smallest cardinality. The name "atom" is justified by the following.

Theorem 7.45 (Mader 1971a). *If C is an atom and K is an end, then either $K \cap C \neq \emptyset$ or $C \subseteq K$.*

It is not true, in general, that a critically k-connected graph G has a node of degree k (or, equivalently, the atoms of G are of cardinality one). But one has the following.

Theorem 7.46 (Mader 1985c). *Every critically k-connected (not complete) graph G contains two disjoint ends with cardinality at most $k/2$. Furthermore, G contains four disjoint ends.*

Corollary 7.47 (Chartrand et al. 1972). *Every critically k-connected simple (not complete) graph G contains a node of degree at most $\lfloor 3k/2 \rfloor$ and this bound is best possible. Actually, G contains at least two such nodes* (Hamidoune 1980 and Veldman 1983).

What about directed graphs?

Theorem 7.48 (Mader 1985b). *Every minimally k-connected digraph contains at least k nodes of in-degree k and at least k nodes of out-degree k.*

Conjecture 7.49 (Mader 1979). Every minimally k-connected digraph contains a node of in-degree and out-degree k.

The k-arc-connected version of this statement is true and plays a central role in constructing all k-arc-connected digraphs (Theorem 7.9).

Theorem 7.50 (Mader 1974b). *Every minimally k-arc-connected digraph contains at least two nodes having both in-degree and out-degree k.*

For critical strongly connected digraphs it is true again that there is a node of in-degree 1. More specifically, the following holds.

Theorem 7.51 (Mader 1989). *Every critical strongly connected digraph with at least four nodes contains four distinct nodes x_1, x_2, y_1, y_2 for which $\delta^-(x_i) = \delta^+(y_i) = 1$ $(i = 1, 2)$.*

Theorem 7.52 (Mader 1991). *Every critically k-connected digraph G contains a node s for which $\delta^-(s) \leq 2k-1$ or $\delta^+(s) \leq 2k-1$. If G is antisymmetric (that is, if*

(x, y) is an arc, then (y, x) is not), then G contains a node s for which $\delta^-(s) \leq \lfloor 3k - \frac{1}{2} \rfloor$ or $\delta^+(s) \leq \lfloor 3k - \frac{1}{2} \rfloor$. These bounds are best possible.

Actually this theorem is a consequence of a result of Mader that can be considered as a directed counterpart of Theorem 7.46. It is not true that the minimum in-degree in a critically k-connected digraph ($k \geq 2$) is at most $2k - 1$.

7.6. *Connected subgraphs*

By Whitney's Theorem 7.5 we know that a k-connected graph contains k openly disjoint paths connecting two specified nodes. It is a natural feeling that a highly connected graphs must contain some other type of subgraphs. In this subsection we briefly summarize such results. The starting point is a theorem by Dirac.

Theorem 7.53 (Dirac 1960). *In a k-connected graph G every subset of k nodes is included in a circuit. If, in addition, G is non-bipartite, every subset of $k - 1$ nodes is included in an odd circuit* (Bondy and Lovász 1981).

Theorem 7.54 (Mesner and Watkins 1967). *In a k-connected graph ($k \geq 3$) a subset H of $k + 1$ nodes is included in a circuit if and only if there is no set $X \subseteq V - H$ with $|X| = k$ such that each node in H belongs to a different component of $G - X$.*

Suppose we want much more: find $k(k - 1)/2$ openly disjoint paths between k specified nodes $s_1, \ldots, s_k$ (one path for one pair). Such a configuration can be considered as a subdivision of K_k with principal nodes $s_1, \ldots, s_k$. With sufficiently high connectivity this property can also be guaranteed.

Theorem 7.55 (Jung 1970, Larman and Mani 1970). *If G is $2^{3k(k-1)/2}$ connected, then for distinct nodes $s_1, \ldots, s_k$ there is a subdivision of K_k in G having $s_1, \ldots, s_k$ as principal nodes.*

Theorem 7.56 (Häggkvist and Thomassen 1982). *In a k-connected graph every subset of $k - 1$ independent edges is included in a circuit.*

Conjecture 7.57 (Lovász). In a k-connected graph every subset of k independent edges is included in a circuit unless k is odd and the k edges disconnect the graph.

For $k = 3$ this was shown by Lovász. The following result, due to Lovász (1977) and Györi (1978), is about partitions of graphs or digraphs into connected parts of given size.

Theorem 7.58. *In a digraph (graph) let $S = \{s_1, s_2, \ldots, s_k\}$ be a set of k nodes and $n_1, n_2, \ldots, n_k$ positive integers such that $\sum n_i = |V|$. Suppose that for any $v \in V - S$ there are k paths from S to v pairwise disjoint except at v. There is a partition $\{V_1, V_2, \ldots, V_k\}$ of V into k parts such that $V_i \cap S = \{s_i\}$, $|V_i| = n_i$ and the digraph (graph) induced by V_i contains an arborescence rooted at s_i (is connected).*

7.7. *Extremal results*

In this last subsection we briefly mention some extremal-type results concerning connectivity. More detailed accounts are found in Mader (1979) and in Bollobás, (1978). Let us start with a result on digraphs.

Theorem 7.59 (Dalmazzo 1977). *A minimal k-edge-connected digraph $D = (V, A)$ on n nodes has at most $2k(n - 1)$ arcs.*

Proof. Let s be an arbitrary node of D. By Edmonds' Theorem 6.6 there are k edge-disjoint spanning arborescences of root s. Let G_1 denote the union of these arborescences. Similarly, there are k edge-disjoint spanning co-arborescences of root s (a co-arborescence of root s is a directed tree such that re-orienting all of its edges results in an arborescence of root s). Let G_2 denote the union of these co-arborescences. Clearly, both G_1 and G_2 have $k(n - 1)$ arcs and their union is k-edge-connected from which the result follows. □

Note that the bound in the theorem is sharp as is shown by a digraph obtained from any tree by replacing each edge uv by $2k$ parallel arcs among which k are in one direction and the other k are in the other direction.

Theorem 7.59 immediately implies that a minimal k-edge-connected graph on n nodes has at most $k(n - 1)$ edges. Indeed, if we replace each edge by two oppositely directed arcs, we obtain a minimal k-edge-connected digraph and then Theorem 7.59 applies.

Mader proved that for simple graphs a better bound exists.

Theorem 7.60 (Mader 1974b). *A minimal k-edge-connected simple graph on n nodes has at most $kn - k(k + 1)/2$ edges.*

Since a minimal k-edge-connected graph must not have a $(k + 1)$-edge-connected subgraph, Theorem 7.60 is an immediate consequence of the following.

Theorem 7.61 (Mader 1974b). *Every simple graph on n nodes with more than $kn - k(k + 1)/2$ edges has a $(k + 1)$-edge-connected subgraph.*

Again, the bound is sharp as is shown by a graph constructed from a complete bipartite graph $K_{k,n-k}$ by adding all the possible edges in the k-element part.

8. Multicommodity flows and disjoint paths

8.1. *Problem formulation*

In this section we address the following problem, called *the disjoint paths problem*. Given a graph or a digraph and k pairs of nodes (s_1, t_1),

$(s_2, t_2), \ldots, (s_k, t_k)$, find k pairwise openly disjoint paths connecting the corresponding pairs (s_i, t_i). If we are interested in finding edge-disjoint paths we speak about the *edge-disjoint paths problem*. In the book "Paths, Flows, and VLSI-Layout" (B. Korte et al., eds., Springer 1990), several survey papers are included related to the material of this section (Frank 1990, Schrijver 1990, Robertson and Seymour 1990].

A capacitated version of the edge-disjoint paths problem is the following. For every edge of the graph a non-negative capacity is specified and, similarly, for every pair to be connected a non-negative demand is given. The *integer multicommodity* flow problem is that of finding as many paths between the corresponding terminals as their demands are so that every edge occurs in at most as many paths as its capacity. If we allow fractional paths as well, we speak about the *multicommodity flow problem* or, in short, *multiflow problem*. That is, a multicommodity flow is defined by paths $P_1, P_2, \ldots, P_k$ and non-negative numbers $\lambda_1, \ldots \lambda_k$ such that each path P_i is a path from s_i to t_i and for each edge e the sum of coefficients assigned to paths using e is at most the capacity of e.

Actually this kind of problem can be considered as a *feasibility* problem. The *maximization problem* is that when no demands are specified and one is interested in finding a maximum number of paths connecting the corresponding terminal pairs.

Sometimes it is convenient to mark the terminal pairs to be connected by an edge. The graph $H = (U, F)$ formed by the marking edges is called a *demand graph* while the original graph $G = (V, E)$ is the *supply graph*. Let us call a circuit of $G + H$ a *good* circuit if it contains precisely one demand edge. In this terminology the edge-disjoint paths problem is equivalent to seeking for $|F|$ edge-disjoint good circuits.

The multiflow problem can be formulated as a linear program. One way to do so is as follows. Let A be a 0–1 matrix the rows of which correspond to the edges of G the columns correspond to the good circuits. An entry (i, j) is 1 if the edge corresponding to i is in the circuit corresponding to j and 0 otherwise. Similarly let B be a 0–(-1) matrix the rows of which correspond to the edges of H, the columns correspond to the good circuits. An entry (i, j) is -1 if the edge corresponding to i is in the circuit corresponding to j and 0 otherwise. (The structure of B is simple: every column has exactly one non-zero entry.) The multiflow problem is equivalent to the following linear inequality system. $Ax \leqslant \mathbf{1}$, $Bx = -\mathbf{1}$, $x \geqslant 0$, where $\mathbf{1}$ and $-\mathbf{1}$ are appropriately sized vectors of 1s and -1s, respectively.

By Farkas's lemma this system has no solution if and only if there is a vector w in $\mathbb{R}_+^E$ and a vector z in $\mathbb{R}^F$ such that $\sum (w(e): e \in E) - \sum (z(f): f \in F) < 0$ and such that $\sum (w(e): e \in C - f) - z(f) \geqslant 0$ holds for every demand edge f and every circuit C for which $C \cap F = \{f\}$. Obviously, if there is such a w and z, then z can be chosen so as to satisfy $z(f) = \mathrm{dist}_w(u, v)$ where $f = uv$ and $\mathrm{dist}_w(u, v)$ is the minimum w-weight of a path in G connecting the end nodes of demand edge f. We obtain the following.

Theorem 8.1 (Iri 1970, Onaga and Kakusho 1971). *The multiflow problem has a*

solution if and only if the

$$\textit{Distance criterion}: \sum (\mathrm{dist}_w(u, v): uv \in F) \leqslant \sum (w(e): e \in E) \tag{8.1}$$

holds for every $w \in \mathbb{R}_+^E$.

As general linear programs can be solved in polynomial time, so is the multiflow problem. Since the constraint matrix above has entries 0, ± 1 there is a strongly polynomial algorithm as well (Tardos 1986). This is why we concentrate only on integer multicommodity flows or disjoint paths.

First we survey results concerning undirected graphs.

Theorem 8.2 (Karp 1975). *The undirected (edge-) disjoint path problem (when k is a part of the input) is* NP-*complete*.

Even et al. (1976) proved that the problem is NP-complete even in the special case when the demand graph consists of two sets of parallel edges. In other words, the integer 2-commodity flow problem is NP-complete. On the other hand we have the following very difficult result.

Theorem 8.3 (Robertson and Seymour 1986b). *For fixed k the undirected (edge-disjoint) disjoint paths problem can be solved in polynomial time.*

8.2. Characterizations for edge-disjoint paths

First, let us concentrate on edge-disjoint paths. A natural necessary condition is the cut-criterion:

Cut-criterion: $d_G(X) \geqslant d_H(X)$ for every $X \subseteq V$.

Note that the cut-criterion is a special case of the distance-criterion. We call the difference $d_G(X) - d_H(X)$ the *surplus* of cut $\Delta(X)$ and denote it by $s(X)$. A cut $\Delta(X)$ is called *tight* if $s(X) = 0$.

The cut criterion is not sufficient, in general, as the simple example in fig. 8.1 shows. It is sufficient, however, if the demand graph is a star (that is, the demand edges share a common endpoint). (This immediately follows from the undirected edge version of Menger's theorem.)

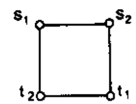

Figure 8.1

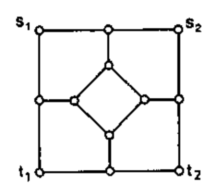

Figure 8.2.

The next two simplest demand graphs are $2K_2$ (a graph on four nodes with two disjoint edges) and C_3 (a triangle). The following characterization for $2K_2$ is due to Seymour (1980c) and Thomassen (1980a).

Theorem 8.4. *Let G be a graph such that no cut edge separates both of the two terminal pairs (s_1, t_1) and (s_2, t_2). There are no two edge-disjoint paths between the corresponding terminals if and only if some edges of G can be contracted so that the resulting graph G' is planar, the four terminals have degree two while the other nodes are of degree 3 and the terminals are positioned on the outer face in this order: s_1, s_2, t_1, t_2.*

Figure 8.2 shows a typical example where the two edge-disjoint paths do not exist.

If we want k_i paths between s_i and t_i $(i = 1, 2)$ the problem becomes NP-complete. The situation is much better for the other special H mentioned above.

Theorem 8.5 (Seymour 1980c). *If the demand graph H consists of three sets of parallel edges between three nodes v_1, v_2 and v_3, the edge-disjoint paths problem has a solution if and only if the cut criterion holds and*

$$q(V_1 \cup V_2 \cup V_3) \leq s(V_1) + s(V_2) + s(V_3)$$

for every choice of disjoint sets V_i with $v_i \in V_i$ $(i = 1, 2, 3)$ where $s(X)$ denotes the surplus and $q(X)$ denotes the number of components C in $G - X$ for which $d_G(X) + d_H(X)$ is odd.

This result is a rather easy consequence of a theorem of Mader (Theorem 8.23 below) on edge-disjoint T-paths (when $|T| = 3$).

Let us call a set X, given G and H, an *odd set* and the cut $\Delta(X)$ an *odd cut* (with respect to $G + H$) if $d_G(X) + d_H(X)$ is odd (or equivalently, the surplus $s(X)$ is odd). A basic feature of odd cuts is that in any solution to the edge-disjoint paths problem an odd number of edges of an odd cut, in particular, at least one edge, will not be used.

What if there are no odd cuts at all, that is, $G + H$ is Eulerian? The cut criterion is still not sufficient as is shown in fig. 8.3. Even worse, Middendorf and Pfeiffer (1990) proved that the edge-disjoint paths problem is NP-complete even if $G + H$ is Eulerian.

However, in the special cases listed below the cut criterion proves to be

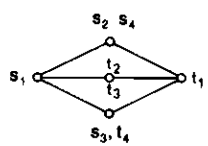

Figure 8.3.

sufficient. Given a demand graph $H = (V, A)$, H' will denote the graph arisen from H by replacing each set of parallel edges by one edge.

Theorem 8.6. *Suppose that $G + H$ is Eulerian. In the following cases the cut criterion is necessary and sufficient for the solvability of the edge-disjoint paths problem.*

(a) *H' is $2K_1$* (Rothschild and Whinston 1966b).

(b) *H' consists of two stars* (Papernov 1976, Seymour 1980a, Lomonosov 1985).

(c) *H' is K_4* (Papernov 1976, Seymour 1980a, Lomonosov 1985).

(d) *H' is C_5* (Lomonosov 1985).

(e) *G is planar and each terminal is on one face* (Okamura and Seymour 1981).

(f) *G is planar and there are two faces F_1, F_2 such that each demand edge connects two nodes of either F_1 or F_2* (Okamura 1983).

(g) *$G + H$ is planar* (Seymour 1981).

(h) *G is planar and there are two specified inner faces C_1 and C_2 of G. The demand edges $s_1t_1, \ldots, s_kt_k$ are positioned in such a way that each s_i is on C_1, each t_i is on C_2 and their cyclic order is the same* (Schrijver 1989).

Note that part (g) of this theorem immediately follows (by planar dualization) from a theorem by Seymour asserting that a ± 1 weighted bipartite graph (planar or not) has no circuit of negative total length if and only if the edge set can be partitioned into cuts such that each cut contains at most one negative edge. This is an equivalent formulation of Seymour's theorem on the maximum number of disjoint T-cuts.

As we mentioned earlier the cut criterion is not sufficient, in general, even if $G + H$ is Eulerian. Sometimes the stronger distance criterion (8.1) helps.

Theorem 8.7 (Karzanov 1987). *Suppose that $G + H$ is Eulerian and the demand edges form a graph arising from K_5 by adding parallel edges. Then the distance criterion is necessary and sufficient for the solvability of the edge-disjoint paths problem. (In other words, if there is a fractional solution, there is an integral one.)*

Theorem 8.8 (Karzanov 1994). *Suppose that G is planar, $G + H$ is Eulerian and each demand edge connects two nodes of one of three specified faces of G. Then the distance criterion is necessary and sufficient for the solvability of the edge-disjoint paths problem.*

Theorem 8.5 provided an example where parity played a basic role in a good characterization. Here are two more cases.

Theorem 8.9 (Frank 1990). *Suppose that $G + H$ is planar and the demand edges are on two faces of G. The edge-disjoint paths problem has a solution if and only if the cut criterion holds and $d_{G+H}(X \cap Y)$ is even for every pair of tight sets X, Y.*

This theorem is a generalization of an earlier theorem of Seymour (1981) where H consisted of two sets of parallel edges. Sebő (1993) proved that if $G+H$ is planar and the number of demand edges nodes is bounded by a constant, then there is a polynomial time algorithm to solve the integer multiflow problem. Another result, due to Schrijver (1990) asserts, that if $G+H$ is planar and the number of faces covering all the terminal nodes is bounded by a constant, then the edge-disjoint paths problem is polynomially solvable. On the other hand, the problem is NP-complete if there is no such a bound (Middendorf and Pfeiffer 1990).

Theorem 8.10 (Frank 1985). *Suppose that G is planar, the terminals are on the outer face and the degree of every node not on the outer face is even. The edge disjoint paths problem has a solution if and only if $\sum s(C_i) \geq q/2$ for every family $(C_1, C_2, \ldots, C_k)$ of cuts ($k \leq |V|$) where q denotes the number of odd components in $G - C_1 - C_2 - \cdots - C_k$ and $s(C)$ is the surplus of C.*

To close this subsection we mention a theorem by van Hoesel and Schrijver (1986) where topology plays a role.

Theorem 8.11. *Let G be a planar graph embedded in $\mathbb{R}^2$. Let O denote the interior of the unbounded face and I the interior of a specified bounded face. Let $C_1, C_2, \ldots, C_k$ be curves in $\mathbb{R}^2 - (I \cup O)$ each of which connects a node on $I \cup O$ with a node on $I \cup O$ so that for each node v of G the degree of v has the same parity as the number of curves ending at v. Then there exist pairwise edge-disjoint paths $P_1, P_2, \ldots, P_k$ in G so that P_i is homotopic to C_i in $\mathbb{R}^2 - (O \cup I)$ ($i = 1, 2, \ldots, k$) if and only if for each dual path Q from $I \cup O$ to $I \cup O$ the number of edges in Q is not smaller than the number of times Q necessarily intersects the curves C_i.*

Note that this theorem generalizes part (e) of Theorem 8.6. It is an open problem to find a common generalization of Theorems 8.6(f) and 8.11. This last theorem is a prototype of theorems belonging to the area one may call homotopic paths packing. An excellent survey of this topic occurs in Schrijver (1990).

8.3. Sufficient conditions for edge-disjoint paths

We call a graph *k-linked on the edges* if for any choice of k pairs of terminals there are k edge-disjoint paths connecting the corresponding terminal pairs.

Theorem 8.4 implies that a 3-edge-connected graph is 2-linked on the edges. Actually such a graph is 3-linked as the following even stronger result shows.

Theorem 8.12 (Okamura 1984). *In a graph three terminal pairs (s_i, t_i) ($i = 1, 2, 3$) are specified such that for each i there are three edge-disjoint paths connecting s_i and t_i. Then there are s_it_i-paths ($i = 1, 2, 3$) pairwise edge-disjoint.*

Okamura's theorem is an answer to the following conjecture of Thomassen, when $k = 3$.

Conjecture 8.13 A k-edge-connected graph is k-linked on the edges if k is odd and $(k-1)$-linked on the edges if k is even.

Note that by Tutte's Theorem 6.14 a $(2k)$-edge-connected graph always has k edge-disjoint spanning trees and therefore it is k-linked on the edges.

The following theorem gets very close to the conjecture.

Theorem 8.14 (Huck 1991). *A k-edge-connected graph is $(k-1)$-linked on the edges if k is odd and $(k-2)$-linked on the edges if k is even.*

We note that Thomassen's conjecture is open for $k = 5$.

In certain cases the cut condition is not strong enough to ensure the existence of the required paths but the demands can almost be met.

Theorem 8.15 (Korach and Penn 1992). *Suppose that $G + H$ is planar, for each terminal pair (s_i, t_i) an integer demand d_i is given and the cut condition holds with respect to d. Then there are $d_i - 1$ paths connecting s_i and t_i for each terminal pair such that all these paths are pairwise disjoint.*

Theorem 8.16 (Itai and Zehavi 1984). *Assume that in a graph G (s_i, t_i) are terminal pairs ($i = 1, 2$) such that there are k edge-disjoint paths connecting s_i and t_i ($i = 1, 2$). Then for each m, $0 \leq m < k$ there are k edge-disjoint paths P, $S_1, S_2, \ldots, S_m$, $Q_1, Q_2, \ldots, Q_{k-m-1}$ such that each S_i connects s_1 and t_1, each Q_j connects s_2 and t_2 and P connects either s_1 and t_1 or s_2 and t_2.*

8.4. Node-disjoint paths

We call a graph *k-linked* if for any choice of k pairs of terminals there are k openly disjoint paths connecting the corresponding terminal pairs. A counter-part of the cut-condition is:

Node-cut condition: No subset S of nodes can separate more than $|S|$ terminal pairs.

This condition is sufficient if the terminal pairs share a common node (a version of the node-Menger theorem) but not in general.

Theorem 8.17 (Thomassen 1980a, Seymour 1980c). *Let G be a graph such that no cut node separates s_1 from t_1 and s_2 from t_2. There are no disjoint paths between s_1 and t_1 and between s_2 and t_2 if and only if G arises from a planar graph G', where the four terminals are on the outer face in this order s_1, s_2, t_1, t_2, by placing an arbitrary graph into some faces of G' bounded by two or three edges.*

The problem was solved algorithmically by Shiloach (1980).

Corollary 8.18 (Jung 1970). *A 4-connected non-planar graph is 2-linked. A 6-connected graph is 2-linked.*

Here the second statement follows from the first one since a planar graph always has a node of degree at most 5. Note that there is a 5-connected planar graph that is not 2-linked. For higher connectivity we have the following.

Theorem 8.19 (Jung 1970, Larman and Mani 1970). *A 2^{3k} connected graph is k-linked.*

It is not known if 2^{3k} can be replaced by a linear bound. The natural $2k+2$ is not enough as can be seen from a K_{3k-1} with edges $x_1y_1, \ldots, x_ky_k$ removed (an example due to Strange and Toft 1983).

The following pretty result is not difficult to prove.

Theorem 8.20 (Robertson and Seymour 1986a). *Suppose that G is planar and the terminals are on the outer face. The disjoint paths problem has a solution if and only if the node-cut condition holds and there are no two "crossing" terminal pairs (that is, any two pairs (s_1, t_1) and (s_2, t_2) are in this order on the outer face; s_1, t_1, s_2, t_2).*

Robertson and Seymour also found a characterization for the disjoint paths problem when G is planar and the terminals are positioned on two specified faces.

8.5. Maximization

So far we have studied multicommodity flow problems of feasibility type. One can also be interested in the maximization form: Given a graph $G=(V,E)$ with non-negative integer capacity function c on the edges and a set of terminal pair $(s_1,t_1), \ldots, (s_k,t_k)$, find flows between s_i and t_i $(i=1,2,\ldots,k)$ that maximize the sum M of flow values under the condition that for each edge e the sum of edge-values of flows on this edge is at most $c(e)$. Let us denote by M_I the maximum sum of flow values if we restrict ourselves to integer flows.

We will use the notation $d_c(v)$ for the sum of capacities of edges incident to v. Let $V_1, V_2, \ldots, V_t$ be a family $\mathcal{P}$ of disjoint subsets of V such that each demand edge connects different V_i. By a *multicut* defined by $\mathcal{P}$ we mean the set of edges uv of G such that $u \in V_i$, $v \notin V_i$ for some i. The capacity of a multicut is defined to be $\sum d(V_i)/2$. Let m denote the minimum capacity of a multicut. Let m_1 denote the minimum capacity of a cut separating each terminal pair (if there is any). Obviously, $m_1 \geq m \geq M \geq M_I$. If $k=1$, then $m_1 = M_I$ by Menger's theorem.

Theorem 8.21 (Hu 1963). *If $k=2$, $m_1 = M$. If $k=2$ and $d_c(v)$ is even for each non-terminal node, then $m_1 = M_I$* (Rothschild and Whinston 1966a).

Theorem 8.22 (Lovász 1976b, Cherkasskij 1977b). If the demand edges form a complete graph induced by T $(T \subseteq V)$, then $m=M$. In addition, if $d_c(v)$ is even for $v \in V-T$, then $m = M_I$.

Generalizing this result to non-Eulerian graphs, Mader (1978b) found the following characterization for M_I.

Theorem 8.23. *Let $G=(V,E)$ be a graph and T a specified subset of nodes. The maximum number of edge-disjoint paths connecting distinct elements of T is $\min[\sum d(V_i) - q_0(\cup V_i)]/2$ where the minimum is taken over all collections of disjoint subsets $V_1, V_2, \ldots, V_{|T|}$ for which $|V_i \cap T| = 1$. (Here $d(X)$ denotes the edges leaving X and $q_0(X)$ denotes the number of components C of $G-X$ for which $d(C)$ is odd.)*

To formulate a node-disjoint version of Theorem 8.23 suppose that T is independent. For a subset X of V and a subgraph G' of G let $b(X; G') := |\{x \in X$: there is an $xy \in E(G')$ with $y \notin X\}|$. $E(X)$ denotes the set of edges induced by X.

Theorem 8.24 (Mader 1978c). *The maximum number of openly node-disjoint paths connecting distinct members of T is equal to $\min(|V_0| + \sum \lfloor b(V_i; G-V_0)/2 \rfloor)$ where the minimum is taken over all collections of disjoint subsets $V_0, V_1, \ldots, V_k$ of $V-T$ $(k \geq 0)$ (where only V_0 can be empty) such that $G - V_0 - \bigcup(E(V_i): i = 1, \ldots, k)$ contains no path connecting distinct nodes of T.*

This result can be regarded as a common generalization of Menger's theorem and the Berge–Tutte theorem. An immediate corollary of Theorem 8.24 is a result of Gallai (1961).

Corollary 8.25. *The maximum number of disjoint paths having end nodes in T is $\min(|K| + \sum \lfloor |C \cap T|/2 \rfloor : K \subseteq V)$ where the sum is taken over the components C of $G-K$.*

Let us turn back to edge-disjoint paths. A common generalization of Theorems 8.21 and 8.22 is as follows.

Theorem 8.26 (Karzanov and Lomonosov 1978). *Let $H=(T,F)$ denote the demand graph. If the maximal independent sets of H can be partitioned into two classes such that both classes consist of disjoint sets (which is equivalent to saying that the complement of H is the line graph of a bipartite graph), then $m=M$. In addition, if $d_c(v)$ is even for $v \in V-T$, then $m = M_I$* (Karzanov 1985).

A proof relying on the polymatroid intersection theorem of Edmonds can be found in Frank et al. (1992). Let us continue our survey with two results where no parity restrictions are imposed.

Theorem 8.27 (Lomonosov 1983). *Suppose that $k=2$ and $G+H$ is planar. Then*

M_I is either $m-1$ or m. $M_I = m-1$ if and only if there are three cuts of value m (each separating both (s_i, t_i) $(i=1,2)$) the union of which includes a cut B of odd capacity so that B does not separate (s_i, t_i) $(i=1,2)$.

Note that in Theorem 8.27 $m = M$ immediately follows from Theorem 8.6(g).

Theorem 8.28 (Kleitman et al. 1970). *If every node of $v \in V-(s_1, \ldots, s_k)-(t_1, \ldots, t_k)$ is adjacent to a member of at least $k-1$ terminal pairs and the terminals form an independent set of G, then $m = M_I$.*

8.6. Directed graphs

All the results we have considered so far in this section concerned undirected graphs. Let $D=(V, A)$ be a digraph and let (s_i, t_i) $(i=1,2,\ldots,k)$ be (ordered) pairs of terminals. The problem is to find (arc-) disjoint paths from s_i to t_i. Let $H=(U, F)$ denote the demand graph, where $F=\{(t_i, s_i): i=1,2,\ldots,k\}$. We call a directed circuit of $D+H$ *good* if it contains precisely one demand arc. Then the arc-disjoint paths problem is equivalent to finding k arc-disjoint good circuits of $D+H$.

Unfortunately, much less is known about directed graphs. One negative result is as follows.

Theorem 8.29 (Fortune et al. 1980). *The (arc-) disjoint paths problem is* NP-*complete for $k=2$.*

Notice that the corresponding undirected problem is tractable (see Theorem 8.3). In what follows we briefly list some special cases when good characterizations and/or polynomial time algorithms are available.

The following criterion is clearly necessary in the arc-disjoint case:

Directed cut criterion: $\delta^-_D(X) \geq \delta^+_H(X)$ for every $X \subseteq V$. If $s_1 = \cdots = s_k$ and $t_1 = \cdots = t_k$ then the directed cut criterion is sufficient as well (directed arc-version of Menger's theorem). A counter-part of Theorem 8.8(a) is also true.

Theorem 8.30. *If H consists of two sets of parallel arcs and $D+H$ is Eulerian (that is, the in-degree of any node is equal to the out-degree), then the directed cut criterion is necessary and sufficient for the solvability of the arc-disjoint paths problem.*

Proof. Assume that H consists of α_i arcs from t_i to s_i $(i=1,2)$. By Menger's theorem, it follows from the hypothesis of the theorem that there are α_1 arc-disjoint paths in D from s_1 to t_1. If we leave out these paths and the α_1 demand edges from $D+H$ we obtain an Eulerian digraph. This partitions into arc-disjoint circuits, and hence it contains α_2 edge-disjoint paths from s_2 to t_2. □

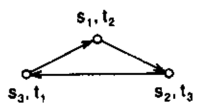

Figure 8.4.

In the example in fig. 8.4 the directed cut criterion is satisfied but there is no solution to the directed edge-disjoint paths problem.

The reason is that the following necessary condition, called *covering criterion*, is violated: the directed circuits of $D+H$ cannot be covered by less than k arcs.

Theorem 8.31. *When $D+H$ is planar and D is acyclic the directed edge-disjoint paths problem has a solution if and only if the covering criterion holds.*

Proof. By planar dualization we obtain from the Lucchesi–Younger Theorem 2.15 that in a planar digraph the maximum number of directed circuits is equal to the minimum number of arcs covering all the directed circuits. Since D is acyclic the set $A(H)$ of demand arcs covers all directed circuits of $D+H$. By the covering criterion this is a minimum covering of directed circuits and hence there are $k=|A(H)|$ arc-disjoint circuits in $D+H$ which must be good circuits. □

In the following theorem D may be non-planar but the number k of demand arcs is considered as a constant.

Theorem 8.32 (Fortune et al. 1980). *In acyclic digraphs the (arc-) disjoint paths problem can be solved in polynomial time if k is fixed.*

In the special case $k=2$, Thomassen (1985) found a complete description of acyclic digraphs having no solution to the disjoint paths problem. The core of his result is as follows.

Theorem 8.33. *Let us be given an acyclic digraph $D=(V, A)$ and terminal pairs (s_1, t_1), (t_2, t_2) such that $|V| \geq 5$, $\delta^-(v)$, $\delta^+(v) \geq 2$ for each non-terminal node v and $\delta^-(s_1) = \delta^-(s_2) = \delta^+(t_1) = \delta^+(t_2) = 0$. If there are no disjoint paths from s_1 to t_1 and from s_2 to t_2, then D is planar and has a plane representation in such a way that s_1, t_2, t_1, s_2 are on the outer face occurring in that cyclic order.*

Ibaraki and Poljak (1991) solved the arc-disjoint paths problem when $k=3$ and $D+H$ is Eulerian. Let D be an Eulerian digraph with three distinct specified nodes a, b, c, called *terminals*. The *three-terminal* problem consists of finding (altogether three) arc-disjoint paths from a to b, from b to c and from c to a. Clearly, this is a special case of the three arc-disjoint paths problem but Ibaraki and Poljak observed that, conversely, the three arc-disjoint paths problem can also be easily reduced to the three-terminal problem.

Suppose that D is a planar digraph with no cut-nodes that has a plane representation such that each face is bounded by a directed circuit, the terminals have degree 2, and they lie in one face where their order with respect to the orientation of the face is a, c, b. It is easy to see that in such case the three-terminal problem has no solution. Therefore we call such a representation *bad*.

Theorem 8.34 (Ibaraki and Poljak 1991). *Given an Eulerian digraph D with terminals a, b, c, the three-terminal problem has a solution if and only if D cannot be contracted to a planar digraph that has a bad plane representation.*

As far as the maximization problem is concerned for digraphs we mention the following (rather easy) counter-part of Theorem 8.22.

Theorem 8.35 (Frank 1989). *In an Eulerian digraph $D = (V, A)$ the maximum number of arc-disjoint paths connecting distinct nodes of a specified subset T of V is equal to the minimum of $\sum \delta^-(V_i)$ over all families of disjoint subsets $V_1, V_2, \ldots, V_{|T|}$ of V for which $|V_i \cap T| = 1$ $(i = 1, 2, \ldots, |T|)$.*

We close this section by mentioning an interesting sufficient condition by Shiloach (1979). Let us call a digraph $D = (V, A)$ *k-linked on the arcs* if for any choice of k pairs $\{s_1, t_1\}, \ldots, (s_k, t_k\}$ of (not necessarily distinct) terminals there are arc-disjoint paths P_i from s_i to t_i $(i = 1, \ldots, k)$. Obviously such a digraph is strongly k-arc connected (that is every non-empty proper subset of nodes has k entering arcs.)

Theorem 8.36. *A strongly k-arc connected digraph is k-linked on the arcs.*

Proof. Add a new node r to D and new arcs (r, s_i) $(i = 1, 2, \ldots, k)$ and apply Edmonds' disjoint arborescence Theorem 6.9. □

The theorem also easily follows from Theorem 7.30 of Mader.

References

Ahuja, R.K., T.L. Magnanti and J.B. Orlin
[1993] *Network Flows: Theory, Algorithms, and Applications* (Prentice-Hall, Englewood Cliffs, NJ).

Alon, N.
[1990] Generating pseudo-random permutations and maximum flow algorithms, *Inform. Process. Lett.* **35**, 201–204.

Ando, K., H. Enomoto and A. Saito
[1987] Contractible edges in 3-connected graphs, *J. Combin. Theory B* **42**, 87–93.

Barnette, D.W., and B. Grünbaum
[1969] On Steinitz's theorem concerning convex 3-polytopes and on some properties of planar graphs, in: *The Many Facets of Graph Theory, Lecture Notes in Mathematics*, Vol. 110, eds. G. Chartrand and S.F. Kapoor (Springer, Berlin) pp. 27–40.

Bellman, R.E.
[1958] On a routing problem, *Quart. Appl. Math.* **16**, 87–90.

Boesch, F., and R. Tindell
[1980] Robbins's theorem for mixed multigraphs, *Amer. Math. Monthly* **87**, 716–719.

Bollobás, B.
[1978] *Extremal Graph Theory* (Academic Press, New York).

Bondy, J.A., and L. Lovász
[1981] Cycles through specified vertices of a graph, *Combinatorica* **1**, 117–140.

Boruvka, O.
[1926] O jistem problemu minimalnim, *Prace Mor. Prirodoved. Spol. v Brne (Acta Societ. Scient. Natur. Moravicae)* **3**, 37–58.

Chartrand, G., A. Kaugars and D.R. Lick
[1972] Critically n-connected graphs, *Proc. Amer. Math. Soc.* **32**, 63–68.

Cheriyan, J., and T. Hagerup
[1990] A randomized maximum flow algorithm, in: *Proc. 30th IEEE Conf. on the Foundations of Computer Science*, pp. 118–123.

Cheriyan, J., and S.N. Maheshwari
[1988] Finding nonseparating induced cycles and independent spanning trees in 3-connected graphs, *J. Algorithms* **9**, 507–537.
[1989] Analysis of preflow-push algorithms for maximum network flow, *SIAM J. Comput.* **18**, 1057–1086.

Cherkasskij, B.V.
[1977a] Algorithm of construction of maximal flow in networks with complexity $O(n^2p^{1/2})$ operations, Akad. Nauk SSSR, CEMI (*Mathematical Methods for the Solution of Economical Problems*) **7**, 117–126 (in Russian).
[1977b] A solution of a problem of multicommodity flows in a network, *Ekonom. i Mat. Metody* **13**(1), 143–151 (in Russian).

Chu, Y.-J., and T.-H. Liu
[1965] On the shortest arborescence of a directed graph, *Sci. Sinica* **4**, 1396–1400.

Dalmazzo, M.
[1977] Nombre d'arcs dans les graphes k-arc-fortement connexes minimaux, *C.R. Acad. Sci. Paris A* **2853**, 341–344.

Dijkstra, E.W.
[1959] A note on two problems in connexion with graphs, *Numer. Math.* **1**, 269–271.

Dilworth, R.P.
[1950] A decomposition theorem for partially ordered sets, *Ann. of Math.* **51**, 161–166.

Dinits, E.A.
[1970] Algorithm for solution of a problem of maximum flow in a network with power estimation, *Dokl. Akad. Nauk SSSR* **194**, 754–757 (in Russian) [Soviet Math. Dokl. **111**, 277–1289].

Dinits, E.A., A.V. Karzanov and M.L. Lomonosov
[1976] On the structure of a family of minimal weighted cuts in a graph, in: *Studies in Discrete Mathematics* (in Russian), ed. A.A. Fridman (Nauka, Moscow) pp. 290–306.

Dirac, G.A.
[1960] In abstrakten Graphen vorhandene vollständige 4-Graphen und ihre Unterteilungen, *Math. Nachr.* **22**, 61–85.

Edmonds, J.
[1967] Optimum branchings, *J. Res. Nat. Bur. Standards B* **71**, 233–240.
[1973] Edge-disjoint branchings, in: *Combinatorial Algorithms*, ed. R. Rustin (Academic Press, New York) pp. 91–96.

Edmonds, J., and R.M. Karp
[1972] Theoretical improvements in algorithmic efficiency for network flow problems, *J. Assoc. Comput. Mach.* **19**, 248–264.

Egawa, Y.
[1987] Cycles in k-connected graphs whose deletion results in a $(k-2)$-connected graph, *J. Combin. Theory B* **42**, 371–377.

Elias, P., A. Feinstein and C.E. Shannon
[1956] A note on the maximum flow through a network, *IRE Trans. Information Theory* **IT-2**, 117–119.
Even, S., A. Rai and A. Shamir
[1976] On the complexity of timetable and multicommodity flow problems, *SIAM J. Comput.* 5(4), 691–703.
Fleischner, H., and B. Jackson
[1985] Removable cycles in planar graphs, *J. London Math. Soc.* **31**(2), 193–199.
Ford, L.R., and D.R. Fulkerson
[1956] Maximum flow through a network, *Canad. J. Math.* **8**, 399–404.
[1962] *Flows in Networks* (Princeton University Press, Princeton, NJ).
Ford Jr, L.R.
[1956] *Network Flow Theory*, Paper P-923 (RAND Corporation, Santa Monica, CA).
Fortune, S., J. Hopcroft and J. Wyllie
[1980] The directed subgraph homeomorphism problem, *Theoret. Comput. Sci.* **10**, 111–121.
Frank, A.
[1980] On chain and antichain families of partially ordered sets, *J. Combin. Theory B* **29**, 176–184.
[1981] How to make a digraph strongly connected, *Combinatorica* **1**, 145–153.
[1985] Edge-disjoint paths in planar graphs, *J. Combin. Theory B* **39**(2), 164–178.
[1989] On connectivity properties of Eulerian digraphs, *Ann. Discrete Math.* **41**, 179–194.
[1990] Packing paths, circuits, and cuts – a survey, in: *Paths, Flows, and VLSI-Layout*, eds. B. Korte, L. Lovász, H.J. Prömel and A. Schrijver (Springer, Berlin) pp. 47–100.
[1992a] Augmenting graphs to meet edge-connectivity requirements, *SIAM J. Discrete Math.* **5**(1), 25–53.
[1992b] On a theorem of Mader, *Ann. Discrete Math.* **101**, 49–57.
[1994] Connectivity augmentation problems in network design, in: *Mathematical Programming: State of the Art 1994*, eds. J.R. Birge and K.G. Murty (University of Michigan) pp. 34–63.
Frank, A., A.V. Karzanov and A. Sebő
[1992] On multiflow problems, in: *Proc. Meeting on Integer Programming and Combinatorial Optimization, Pittsburgh, PA*, eds. E. Balas, G. Cornuéjols and R. Kannan, pp. 85–101.
Fulkerson, D.R.
[1974] Packing rooted directed cuts in a weighted directed graph, *Math. Programming* **6**, 1–13.
Galil, Z.
[1980] An $O(|V|^{5/3}|E|^{2/3})$ algorithm for the maximal flow problem, *Acta Inform.* **14**, 221–242.
Gallai, T.
[1961] Maximum-Minimum Sätze und verallgemeinerte Faktoren von Graphen, *Acta Math. Acad. Sci. Hungar.* **12**, 131–163.
Goldberg, A.V., and R.E. Tarjan
[1986] A new approach to the maximum flow problem, in: *Proc. 18th Annu. ACM Symp. on Theory of Computing*, pp. 136–146. Full paper: 1988, *J. ACM* **35**, 921–940.
[1989] Finding minimum-cost circulations by canceling negative cycles, *J. Assoc. Comput. Mach.* **36**(4), 873–886.
Goldberg, A.V., R.E. Tarjan and É. Tardos
[1990] Network flow algorithms, in: *Paths, Flows, and VLSI-Layout*, eds. B. Korte, L. Lovász, H.J. Prömel and A. Schrijver (Springer, Berlin) pp. 101–164.
Gomory, R., and T.C. Hu
[1961] Multiterminal network flows, *J. SIAM* **9**, 551–570.
Graham, R.L., and P. Hell
[1985] On the history of the minimum spanning tree problem, *Ann. Hist. Comput.* 7(1), 43–57.
Greene, C.
[1976] Some partitions associated with a partially ordered set, *J. Combin. Theory A* **20**, 69–79.
Greene, C., and D.J. Kleitman
[1976] The structure of Sperner k-families, *J. Combin. Theory A* **20**, 41–68.
Győry, E.
[1978] On division of graphs to connected subgraphs, in: *Combinatorics 1*, eds. A. Hajnal and V.T. Sós, *Colloq. Math. Soc. János Bolyai* **18**, 485–494.

Häggkvist, R., and C. Thomassen
[1982] Circuits through specified edges, *Discrete Math.* **41**, 29–34.
Halin, R.
[1969a] A theorem on n-connected graphs, *J. Combin. Theory*, 150–154.
[1969b] Untersuchungen über minimale n-fach zusammenhängende Graphen, *Math. Ann.* **182**, 175–188.
Hamidoune, Y.O.
[1980] On critically, h-connected simple graphs, *Discrete Math.* **32**, 257–262.
Hoffman, A.
[1960] Some recent applications of the theory of linear inequalities to extremal combinatorial analysis, in: *Combinatorial Analysis*, eds. R. Bellman and M. Hall (AMS, Providence, RI) pp. 113–128.
Hoffman, A.J.
[1974] A generalization of max flow–min cut, *Math. Programming* **6**, 352–359.
Hu, T.C.
[1963] Multicommodity network flows, *Oper. Res.* **11**, 344–360.
Huck, A.
[1991] A sufficient condition for a graph to be weakly weakly k-linked, *Graphs Combin.* **7**, 323–351.
Ibaraki, T., and S. Poljak
[1991] Weak three-linking in Eulerian digraphs, *SIAM J. Discrete Math.* **4**(1), 84–89.
Iri, M.
[1970] On an extension of the max-flow min-cut theorem for multicommodity flows, *J. Oper. Res. Soc. Japan* **13**, 129–135.
Itai, A., and A. Zehavi
[1984] Bounds on path connectivity, *Discrete Math.* 25–34.
Jackson, B.
[1980] Removable cycles in 2-connected graphs of minimum degree at least four, *J. London Math. Soc.* **21**(2), 385–392.
[1988] Some remarks on arc-connectivity, vertex splitting and orientation in digraphs, *J. Graph Theory* **12**(3), 429–436.
Jung, H.A.
[1970] Eine Verallgemeinerung des n-fachen Zusammenhangs für Graphen, *Math. Ann.* **187**, 95–103.
Karp, R.M.
[1975] On the computational complexity of combinatorial problems, *Networks* **5**, 45–68.
[1978] A characterization of the minimum cycle mean in a digraph, *Discrete Math.* **23**, 309–311.
Karzanov, A.V.
[1974] The problem of finding the maximal flow in a network by the method of preflows, *Dokl. Akad. Nauk SSSR* **215**, 49–52 (in Russian) [*Soviet Math. Dokl.* **215**, 434–437].
[1985] On multicommodity flow problems with integer-valued optimal solutions, *Dokl. Akad. Nauk SSSR* **280**(4) [*Soviet Math. Dokl.* **32**, 151–154].
[1987] Half-integral five-terminus flows, *Discrete Appl. Math.* **18**, 263–278.
[1994] Paths and metrics in a planar graph with three or more holes I and II, *J. Combin. Theory B* **60**(1), 1–35.
Karzanov, A.V., and M.V. Lomonosov
[1978] Multiflows in undirected graphs, in: *Mathematical Programming, The Problems of Social and Economic Systems, Operations Research Model*, Vol. 1 (in Russian) (The Institute for System Studies, Moscow).
Kleitman, D.J., A. Martin-Löf, B. Rothschild and A. Whinston
[1970] A matching theorem for graphs, *J. Combin. Theory* **8**, 104–114.
König, D.
[1915] Line systems and determinants, *Math. Termész. Ért.* **33**, 221–229 (in Hungarian).
Korach, E., and M. Penn
[1992] Tight integral duality gap in the Chinese Postman Problem, *Math. Programming* **55**, 183–191.

Kruskal, J.B.
[1956] On the shortest spanning tree of a graph and the travelling salesman problem, *Proc. Amer. Math. Soc.* **7**, 48–50.

Larman, D.G., and P. Mani
[1970] On the existence of certain configurations within graphs and the 1-skeletons of polytopes, *Proc. London Math. Soc.* **20**, 144–160.

Lawler, E.L.
[1976] *Combinatorial Optimization: Networks and Matroids* (Holt, Rinehart & Winston, New York).

Lomonosov, M.V.
[1983] On the planar integer two-flow problem, *Combinatorica* **3**(2), 207–219.
[1985] Combinatorial approaches to multiflow problems, *Discrete Appl. Math.* **11**(1), 1–93.

Lovász, L.
[1976a] On two minimax theorems in graph theory, *J. Combin. Theory B* **21**, 96–103.
[1976b] On some connectivity properties of Eulerian graphs, *Acta Math. Acad. Sci. Hungar.* **28**, 129–138.
[1977] Homology theory for spanning trees of a graph, *Acta Math. Acad. Sci. Hungar.* **30**, 241–251.
[1979] *Combinatorial Problems and Exercises* (North-Holland, Amsterdam).

Lovász, L., and M.D. Plummer
[1986] *Matching Theory, Ann. Discrete Math.* **29**.

Lovász, L., V. Neumann-Lara and M.D. Plummer
[1978] Mengerian theorems for paths of bounded length, *Period. Math. Hungar.* **9**, 269–276.

Lucchesi, C.L., and D.H. Younger
[1978] A minimax relation for directed graphs, *J. London Math. Soc.* **17**(2), 369–374.

Mader, W.
[1971a] Eine Eigenschaft der Atome endlicher Graphen, *Arch. Math.* **22**, 257–262.
[1971b] Minimale n-fach kantenzusammenhangende Graphen, *Math. Ann.* **191**, 21–28.
[1972] Ecken vom Grad n in minimalen n-fach zusammenhangende Graphen, *Arch. Math.* **23**, 219–224.
[1973] Grad und lokaler Zusammenhang in endlichen Graphen, *Math. Ann.* **205**.
[1974a] Kreuzungsfreie (a, b)-Wege in endlichen Graphen, *Abh. Mat. Sem. Univ. Hamburg* **42**, 187–204.
[1974b] Ecken vom Innen- und Aussengrad k in minimal n-fach kantenzusammenhangenden Digraphen, *Arch. Math.* **25**, 107–112.
[1978a] A reduction method for edge-connectivity in graphs, *Ann. Discrete Math.* **3**, 145–164.
[1978b] Über die Maximalzahl kantendisjunkter A-Wege, *Arch. Math. (Basel)* **30**, 325–336.
[1978c] Über die Maximalzahl kreuzungsfreier H-Wege, *Arch. Math. (Basel)* **31**, 387–402.
[1979] Connectivity and edge-connectivity in finite graphs, in: *Surveys on Combinatorics, London Math. Soc. Lecture Notes Series*, Vol. 38, ed. B. Bollobás (London Mathematical Society, London) pp. 293–309.
[1981] On a property of n-edge connected digraphs, *Combinatorica* **1**(4), 385–386.
[1982] Konstruktion aller n-fach kantenzusammenhängenden Digraphen, *European J. Combin.* **3**, 63–67.
[1983] On n-edge-connected digraphs, *Ann. Discrete Math.* **17**, 439–441.
[1985a] Paths in graphs reducing the edge-connectivity only by two, *Graphs and Combinatorics* **1**, 81–89.
[1985b] Minimal n-fach zusammenhangende Digraphen, *J. Combin. Theory B* **38**(2), 102–117.
[1985c] Disjunkte Fragmente in kritisch n-fach zusammenhängende Graphen, *European J. Combin.* **6**, 353–359.
[1986] Kritisch n-fach kantenzusammenhängende Graphen, *J. Combin. Theory B* **40**(2), 152–158.
[1989] On critically connected digraphs, *J. Graph Theory* **13**(4), 513–522.
[1991] Ecken von kleinem Grad in kritisch n-fach zusammenhängende Digraphen, *J. Combin. Theory B* **53**(2), 260–272.

Malhotra, V.M., M.P. Kumar and S.N. Maheswari
[1978] An $O(|V|^3)$ algorithm for finding maximum flows in a network, *Inform. Process. Lett.* **7**, 277–278.

Menger, K.
[1927] Zur allgemeinen Kurventheorie, *Fund. Math.* **10**, 96–115.

Mesner, D.M., and M.E. Watkins
[1967] Cycles and connectivity in graphs, *Canad. J. Math.* **19**, 1319–1328.

Middendorf, M., and F. Pfeiffer
[1990] On the complexity of disjoint paths problem, in: *Polyhedral Combinatorics, ACM Series in Discrete Mathematics and Theoretical Computer Science*, Vol. 1, eds. W. Cook and P.D. Seymour (ACM, New York) pp. 171–178.

Nagamochi, H., and T. Ibaraki
[1992] A linear time algorithm for finding a sparse k-connnected spanning subgraph of a k-connected graph, *Algorithmica* **7**, 583–596.

Nash-Williams, C.St.J.A.
[1960] On orientations, connectivity and odd vertex pairings in finite graphs, *Canad. J. Math.* **12**, 555–567.
[1961] Edge-disjoint spanning trees of finite graphs, *J. London Math. Soc.* **36**, 445–450.
[1964] Decomposition of finite graphs into forests, *J. London Math. Soc.* **39**, 12.

Okamura, H.
[1983] Multicommodity flows in graphs, *Discrete Appl. Math.* **6**, 55–62.
[1984] Multicommodity flows in graphs II, *Japan J. Math.* **10**, 99–116.

Okamura, H., and P.D. Seymour
[1981] Multicommodity flows in planar graphs, *J. Combin. Theory B* **31**, 75–81.

Onaga, K., and O. Kakusho
[1971] On feasibility conditions of multicommodity flows in networks, *IEEE Trans. Circuit Theory* **CT-18**, 425–429.

Orlin, J.P.
[1988] A faster strongly polynomial minimum cost flow algorithm, in: *Proc. 20th ACM Symp. on Theory of Computing* (ACM, New York) pp. 377–387. Full paper: *J. Oper. Res.*, to appear.

Padberg, M.W., and M.R. Rao
[1982] Odd minimum cut-sets and b-matchings, *Math. Oper. Res.* **7**, 67–80.

Papernov, B.A.
[1976] Feasibility of multicommodity flows, in: *Studies in Discrete Optimization* (Issledovaniya po Discretnoy Optimizacii), ed. A.A. Friedman (Isdat. Nauka, Moscow) pp. 230–261 (in Russian).

Phillips, D.T., and A. Garcia-Diaz
[1981] *Fundamentals of Network Analysis* (Prentice-Hall, Englewood Cliffs, NJ).

Prim, R.C.
[1957] Shortest connection networks and some generalization, *Bell Syst. Techn. J.* **36**, 1389–1401.

Robbins, H.E.
[1939] A theorem on graphs with an application to a problem of traffic control, *Amer. Math. Monthly* **46**, 281–283.

Robertson, N., and P.D. Seymour
[1986a] Graph minors VI: Disjoint paths across a disc, *J. Combin. Theory B* **41**, 115–138.
[1986b] Graph minors XIII: The disjoint paths problem, *J. Combin. Theory B*, to appear.
[1990] An outline of a disjoint paths algorithm, in: *Paths, Flows, and VLSI-Layout*, eds. B. Korte, L. Lovász, H.J. Prömel and A. Schrijver (Springer, Berlin) pp. 267–291.

Rothschild, B., and A. Whinston
[1966a] On two-commodity network flows, *Oper. Res.* **14**, 377–387.
[1966b] Feasibility of two-commodity network flows, *Oper. Res.* **14**, 1121–1129.

Schrijver, A.
[1982] Min–max relations for directed graphs, *Ann. Discrete Math.* **16**, 261–280.
[1989] The Klein bottle and multicommodity flows, *Combinatorica* **9**(4), 375–384.
[1990] Homotopic routing methods, in: *Paths, Flows, and VLSI-Layout*, eds. B. Korte, L. Lovász, H.J. Prömel and A. Schrijver (Springer, Berlin) pp. 329–371.

Sebő, A.
[1990] Undirected distances and the postman-structure of graphs, *J. Combin. Theory B* **49**(1), 10–39.
[1993] Integer plane multicommodity flows with a fixed number of demands, *J. Combin. Theory B* **59**(2), 163–171.

Seymour, P.D.
[1980a] Four-terminus flows, *Networks* **10**, 79–86.
[1980b] Decomposition of regular matroids, *J. Combin. Theory B* **28**, 305–359.
[1980c] Disjoint paths in graphs, *Discrete Math.* **29**, 293–309.
[1981] On odd cuts and plane multicommodity flows, *Proc. London Math. Soc.* **42**, 178–192.
Shiloach, Y.
[1978] *An $O(nI(\log I)^2)$ maximum flow algorithm*, Tech. Report STAN-CS-78-802 (Stanford University Computer Science Department, Stanford, CA).
[1979] Edge-disjoint branchings in directed multigraphs, *Inform. Proc. Lett.* **8**, 24–27.
[1980] A polynomial solution to the undirected two paths problem, *J. Assoc. Comput. Mach.* **27**, 445–456.
Shiloach, Y., and U. Vishkin
[1982] An $O(n^2 \log n)$ parallel max-flow algorithm, *J. Algorithms* **3**, 128–146.
Sleator, D.D.K.
[1980] *An $O(nm\log n)$ algorithm for maximum network flow*, Tech. Report STAN-CS-80-831 (Stanford University Computer Science Department, Stanford, CA).
Strange, K.E., and B. Toft
[1983] An introduction to the subgraph homeomorphism problem, in: *Proc. Third Czech Symp. on Graph Theory* (B.G. Teubner, Stuttgart) pp. 296–301.
Tardos, É.
[1985] A strongly polynomial minimum cost circulation algorithm, *Combinatorica* **5**, 247–255.
[1986] Strongly polynomial algorithm to solve combinatorial linear programs, *Oper. Res.* **34**(2), 250–256.
Tarjan, R.E.
[1972] Depth-first search and linear graph algorithms, *SIAM J. Comput.* **1**, 146–160.
[1983] Data structures and network algorithms, *CBMS-NFS Regional Conf. Ser. in Appl. Math.*, Vol. 44 (Society for Industrial and Applied Mathematics, Philadelphia, PA).
Tarjan, R.E., and M. Yannakakis
[1984] Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs, *SIAM J. Computing*, **13**(3).
Thomassen, C.
[1980a] 2-linked graphs, *European J. Combin.* **1**, 371–378.
[1980b] Planarity and duality of finite and infinite graphs, *J. Combin. Theory B* **29**, 244–271.
[1981] Non-separating cycles in k-connected graphs, *J. Graph Theory* **5**, 351–354.
[1984] Plane representations of graphs, in: *Progress in Graph Theory*, eds. A. Bondy and R. Murty (Academic Press) pp. 43–69.
[1985] The 2-linkage problem for acyclic digraphs, *Discrete Math.* **55**, 73–87.
Thomassen, C., and B. Toft
[1981] Non-separating induced cycles in graphs, *J. Combin. Theory B* **31**(2), 199–224.
Titov, V.K.
[1975] *A constructive description of some classes of graphs*, Doctoral Dissertation (Moscow).
Tutte, W.T.
[1961a] On the problem of decomposing a graph into n-connected factors, *J. London Math. Soc.* **36**, 221–230.
[1961b] A theory of 3-connected graphs, *Indag. Math.* **23**, 441–455.
[1963] How to draw a graph, *Proc. London Math. Soc.* **13**, 743–767.
[1966] *Connectivity in Graphs* (University of Toronto Press, Toronto).
[1984] *Graph Theory* (Addison-Wesley, Reading, MA).
Van Hoesel, C., and A. Schrijver
[1986] Edge-disjoint homotopic paths in a planar graph with one hole, *J. Combin. Theory B* **48**, 77–91.
Veldman, H.J.
[1983] *Three topics in graph theory*, Dissertation (Technical University Twente, Enschede).
Vidyasankar, K.
[1978] Covering the edge-set of a directed graph with trees, *Discrete Math.* **24**, 79–85.
Watanabe, T., and A. Nakamura
[1987] Edge-connectivity augmentation problems, *J. Computer & System Sci.* **35**(1), 96–144.

Whitney, H.
[1932] Non-separable and planar graphs, *Trans. Amer. Math. Soc.* **34**, 339–362.
Whitty, R.H.
[1986] Vertex-disjoint paths and edge-disjoint branchings in directed graphs, *J. Graph Theory*, **11**(3), 349–358.
Zehavi, A., and A. Itai
[1989] Three tree-paths, *J. Graph Theory*, **13**(2), 175–188.