# Approximating Directed Multicuts

Joseph Cheriyan *
Department of Combinatorics and Optimization
University of Waterloo
Waterloo, Ontario
Canada N2L 3G1
jcheriyan@math.uwaterloo.ca

Howard Karloff
AT&T Labs—Research
180 Park Ave.
Florham Park, NJ 07932
howard@research.att.com

Yuval Rabani †
Computer Science Department, Technion—IIT
Haifa 32000
Israel
rabani@cs.technion.ac.il

**Abstract**

The seminal paper of Leighton and Rao (1988) and subsequent papers presented approximate min-max theorems relating multicommodity flow values and cut capacities in undirected networks, developed the divide-and-conquer method for designing approximation algorithms, and generated novel tools for utilizing linear programming relaxations. Yet, despite persistent research efforts, these achievements could not be extended to directed networks, excluding a few cases that are "symmetric" and therefore similar to undirected networks. This paper is an attempt to remedy the situation. We consider the problem of finding a minimum multicut in a directed multicommodity flow network, and give the first nontrivial upper bounds on the max flow-to-min multicut ratio. Our results are algorithmic, demonstrating nontrivial approximation guarantees.

## 1 Introduction

A *network* is a graph $G = (V, E)$, directed or undirected, with positive edge capacities $c : E \to \mathbb{R}^+$, together with a list of source-sink pairs of vertices $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$, sometimes called *commodities*. Usually, we use $k$ to denote the number of commodities. A multicut is a set of edges $M \subseteq E$ whose removal disconnects all commodities (that is, $G - M = (V, E - M)$ has no $s_i \to t_i$ path for $i = 1, \ldots, k$), and its capacity is the sum of the capacities of the edges in $M$. The problem of finding a multicut of minimum capacity may be formulated as a simple and elegant integer program, and dropping the integrality constraints gives a linear programming (LP) relaxation. The optimal value of this LP relaxation (which is

*Supported in part by NSERC research grant OGP0138432.

†Part of this work was done while visiting AT&T Labs—Research. Work at the Technion supported by Israel Science Foundation grant number 386/99, by BSF grants 96-00402 and 99-00217, by Ministry of Science contract number 9480198, by EU contract number 14084 (APPOL), by the CONSIST consortium (through the MAGNET program of the Ministry of Trade and Industry), and by the Fund for the Promotion of Research at the Technion.

a lower bound on the minimum capacity of a multicut) equals the maximum value of a multicommodity flow (see Section 2 for details). In the single-commodity ($k = 1$) case, the celebrated max flow-min cut theorem of Ford and Fulkerson [7] states that the minimum capacity of a multicut equals the maximum value of a flow. This is one of the key results in combinatorial optimization, and it has numerous important applications, both in theory and in practice. Unfortunately, this theorem does not generalize to multiple commodities, and moreover, the general problem of finding a minimum-capacity multicut is NP-hard (for $k \geq 3$ commodities for undirected networks, and for $k \geq 2$ commodities for directed networks). See [14] for more discussion on multicommodity flows.

Based on ground-breaking work by Leighton and Rao [15], and improving on earlier results due to Klein et al [11], Garg, Vazirani, and Yannakakis [8] proved an approximate minmax theorem for undirected networks: the minimum capacity of a multicut is at most $O(\log k)$ times the maximum value of a multicommodity flow; moreover, their proof is constructive and gives an $O(\log k)$-approximation algorithm (the algorithm runs in polynomial time and returns a multicut whose capacity is at most $O(\log k)$ times the maximum value of a multicommodity flow). Despite persistent research efforts, these results could not be extended to directed networks, excluding a few cases that are "symmetric" and therefore similar to undirected networks.

In this paper, we consider the problem of finding a minimum-capacity multicut in networks (without any symmetry assumptions), "network" without "undirected" meaning "directed network" from now on, and provide the first nontrivial upper bounds relating multicut capacities to multiflow values. For a network $G$, we denote by $C(G)$ the minimum capacity of a multicut, and by $F(G)$ the maximum value of a multicommodity flow. (For undirected networks $G'$, we denote the corresponding quantities by $C'(G')$ and $F'(G')$.) We prove four related theorems. Each of these theorems gives a bound on $C(G)$ in terms of $F(G)$ and other parameters of the network $G$; moreover, each proof gives an efficient algorithm for finding a multicut whose capacity is at most the bound on $C(G)$. The bounds given by the first three theorems are mutually incomparable in the sense that for each of the three bounds, there exist networks where that bound is better than the other bounds.

**Theorem 1** *There is a polynomial-time algorithm that takes a network $G$ satisfying $c(e) \geq 1$ for all arcs $e$ and finds a multicut $M$ satisfying $c(M) \leq 108\, F(G)^3$.*

We prove that without the "$c(e) \geq 1$ for all $e$" condition, no result of the form "$C(G) \leq g(F(G))$ for all $G$" is possible. (For undirected networks, Yannakakis [24] shows, via a variant of the region-growing procedure of [8], that $C(G) = O(F(G) \log F(G))$, if all capacities are at least 1.)

**Theorem 2** *There is a polynomial-time algorithm that takes a $k$-commodity network $G$ satisfying $c(e) \geq 1$ for all arcs $e$ and finds a multicut $M$ satisfying $c(M) \leq 39 \ln(k+1) F(G)^2$.*

Again, the "$c(e) \geq 1$ for all $e$" condition is necessary.

**Theorem 3** *There is a polynomial-time algorithm that takes an $n$-vertex, $k$-commodity network $G$ and finds a multicut $M$ satisfying*

$$c(M) \leq (45\sqrt{n \ln(k+1)})F(G) \leq (45\sqrt{2n \ln n})F(G).$$

We give a better approximation guarantee for some instances in planar digraphs.

**Theorem 4** *For every $\Delta$, there is a constant $\gamma$ such that there is a polynomial-time algorithm that takes an $n$-vertex, $k$-commodity ($k \geq 2$) network $G$ with uniform capacities, whose underlying undirected graph is planar, and in which the total degree of every vertex is at most $\Delta$, and finds a multicut $M$ satisfying $c(M) \leq (\gamma\sqrt{\lg k})n^{1/4}F(G)$.*

Tardos and Vazirani [23] use the methods of Klein, Plotkin, and Rao [12] to prove a constant ratio for undirected planar networks.

Theorem 1 is our basic result. The other three theorems are based on it, and derived using techniques such as region growing (Theorem 2), a trade-off via LP rounding (Theorem 3), and a trade-off via the planar separator theorem (Theorem 4).

In recent work, Saks, Samorodnitsky, and Zosin [20] construct a family of $k$-commodity networks, for all $k$ and $\epsilon > 0$, where the minimum multicut-to-maximum $k$-commodity flow ratio is no less than $k - \epsilon$, in contrast with the $O(\log k)$ upper bound in the undirected case. (An upper bound of $k$ is a trivial consequence of the Ford and Fulkerson theorem.) We note that in their graphs, $|V|$ is exponential in $k$, so an upper bound of $O(\log |V|)$, for example, is still possible. In fact, the networks in [20] have special structure. Each is obtained by adding $2k$ distinct new vertices $s_1, t_1, ..., s_k, t_k$ to an *undirected* graph $H$, together with arcs from the $s_i$'s to some vertices in $H$ and from some vertices in $H$ to the $t_i$'s, replacing each undirected edge by a pair of antiparallel arcs, and assigning positive capacities to the *vertices*. Each terminal gets infinite capacity. We show in Section 4 that any network $G$ of such special structure with $C(G) \geq (k/2)F(G)$ must, like the example of [20], have a number of vertices which is exponential in $k$. Indeed, the same result holds if capacities are instead assigned to arcs, provided that the arcs incident from the sources have infinite capacity, and so do the arcs incident to the sinks.

The best inapproximability result known for directed multicut is that the problem is MAX SNP-hard. This is also the strongest hardness result known for the undirected case [3].

The rest of this introduction gives our perspective on the previous work in this area, and is not essential for studying the new results in this paper.

In a seminal paper, Leighton and Rao [15] proved that for uniform multicommodity flow instances the sparsest cut-to-maximum concurrent flow ratio in undirected networks is at most logarithmic in the number of vertices.[1] They exhibited several applications of this result, mostly in the design and analysis of approximation algorithms for NP-hard optimization problems. Their paper inspired a significant research effort in the past decade. The results of this effort include the emergence of the divide-and-conquer method in approximation algorithms (see [22]), applications of their region-growing technique to other problems [2, 8, 11, 21], and the development of alternative proofs for their basic result and its generalizations [1, 5, 16]. In particular, Garg, Vazirani, and Yannakakis [8] gave an elegant analysis of the region-growing technique, and used it to derive asymptotically tight $O(\log k)$ bounds on the minimum multicut-to-maximum flow ratio in $k$-commodity undirected networks.

Most of the previous research on approximation algorithms for problems related to multicuts in directed networks exploits some sort of "symmetry" property that renders the problems similar to the undirected case; for example, the commodities occur in symmetric pairs $(s_i, t_i),\ (t_i, s_i)$ [4, 5, 6, 13, 15, 18, 21]. In particular, for such symmetric instances, Even, Naor, Schieber, and Sudan [6], improving upon a result of Klein, Plotkin, Rao, and Tardos [13], gave an $O((\log k) \log \log k)$ bound, and they gave efficient algorithms to find a "symmetric multicut" whose capacity is within the same factor of the optimum. (A symmetric multicut means a set of arcs whose removal disconnects either $s_i$ from $t_i$ or $t_i$ from $s_i$, for every symmetric

[1] The sparsity of a cut is the ratio between the cut capacity and the number of source-sink pairs that are disconnected. A concurrent flow delivers the same amount of flow of each commodity.

pair of commodities.) These papers use region-growing techniques, though the bounds that are proved are usually weaker than those that can be proved in the undirected case.

Unfortunately, the literature cited in the previous paragraph has almost no relevance for (asymmetric) directed multicuts because there is *no* relation between a (directed) multicut and a symmetric multicut. For example, consider a directed graph on two vertices $p, q$ with two arcs $(p, q)$ and $(q, p)$ having capacities $1$ and $1000$, respectively. There are two commodities $(s_1, t_1) = (p, q)$ and $(s_2, t_2) = (q, p)$. The unique multicut has capacity 1001, whereas there is a symmetric multicut of capacity one. Another way to see the contrast is to compare the integrality ratios of the linear programming relaxations: it is $O((\log k) \log \log k)$ for symmetric multicuts [6] but a construction due to Saks et al shows that it is $k$ for directed multicuts [20].

## 2 Preliminaries

A *network* $G$ is a directed graph $(V, E)$, without parallel arcs or self-loops, with an assignment of positive capacities to the arcs $c : E \to \mathbb{R}^+$, together with a positive integer $k$ and a set of $k$ distinct ordered pairs $(s_i, t_i)$ of vertices, $s_i \neq t_i$ for all $i$. Let $T = \{s_1, t_1, s_2, t_2, ..., s_k, t_k\}$ be the set of *terminals*. For any set of arcs $E'$, we use $c(E')$ to denote $\sum_{e \in E'} c(e)$. A *multicut* $M$ in $G$ is a subset $M \subseteq E$ such that the digraph $(V, E - M)$ has no $s_i \to t_i$ path, for each $i \in \{1, 2, ..., k\}$. (All paths are simple in this paper.) The *capacity* of a multicut $M$ is $c(M)$. DIRECTED MULTICUT is the problem of finding a minimum-capacity multicut in a specified network $G$. Let us denote the minimum capacity of a multicut in $G$ by $C = C(G)$. (When we work with undirected networks, the underlying graph $G'$ is undirected and the minimum capacity of a multicut is denoted $C' = C'(G')$.)

The problem of finding a minimum-capacity multicut in $G$ is precisely the following integer program: Find $x(e)$ for all $e \in E$, $x(e)$ integral, $x(e) \geq 0$, so as to minimize $\sum_{e \in E} c(e)x(e)$, such that for every $i = 1, 2, 3, ..., k$, and for every $s_i \to t_i$ path $P$ in $G$, $\sum_{e \in P} x(e) \geq 1$. An optimal solution will have $x(e) \leq 1$ for all $e \in E$.

Dropping the "$x(e)$ integral" condition gives a linear programming relaxation of DIRECTED MULTICUT: Find a nonnegative real length $x(e)$ for each arc $e$ such that for each $i = 1, \ldots, k$, the distance from $s_i$ to $t_i$, relative to these lengths, is at least 1, so as to minimize $\sum c(e)x(e)$. Its linear programming dual is easily seen to be equivalent to MULTICOMMODITY FLOW, which is this problem: Given a network $G$, find a sequence $(f_1, f_2, ..., f_k)$ such that $f_i$ is a single-source flow (of commodity $i$) in $G$ from source $s_i$ to sink $t_i$, such that $(f_1, f_2, ..., f_k)$ satisfies $\sum_{1 \leq i \leq k} f_i(e) \leq c(e)$ for all $e \in E$, and in which the sum over $i$ of the value of $f_i$ is maximized. Let $F = F(G)$ denote the optimal value of the multicommodity flow in $G$. It is easy to see that $C(G) \geq F(G)$ for all $G$. (In an undirected network $G'$, the optimal flow value is denoted $F' = F'(G')$.) Since MULTICOMMODITY FLOW can be written as a linear program of polynomial size, it can be solved in polynomial time.

We are interested in the relation between $C(G)$ and $F(G)$ in an arbitrary network $G$. Can $C(G)$ be bounded as a function of $F(G)$ for all $G$? More formally, is there a function $g : \mathbb{R} \to \mathbb{R}$ such that for all $G$, regardless of the number of vertices and commodities, $C(G) \leq g(F(G))$? We will (easily) see below that if the capacities can be arbitrarily small, then the answer is no. However, if we insist that $c(e) \geq 1$ for all $e \in E$, then it is a nontrivial fact that $F(G) \leq 1$ implies $C(G) \leq 1$.

Note that DIRECTED MULTICUT is *not* a generalization of UNDIRECTED MULTICUT obtained by replacing each undirected edge by a pair of antiparallel arcs and by replacing each commodity $\{s_i, t_i\}$ by a pair of "antiparallel" commodities $(s_i, t_i), (t_i, s_i)$. For example, consider a four-vertex undirected tree with root $r$ and leaves $l_1, l_2, l_3$. Let us define three commodities, one for each pair of leaves, and make all capacities one. Let $G'$ denote the network. Then, we have $C'(G') = 2 \ > \ F'(G') = 1.5$ (any two edges $\{r, l_i\}, \ \{r, l_j\}$

form a multicut, and an optimal flow assigns the value $1/2$ to each of the three undirected $s_i - t_i$ paths). However, if we now replace each edge by two antiparallel arcs (each of unit capacity) and define six commodities, one for each ordered pair of leaves, then the directed network $G$ has $C(G) = 3 = F(G)$ (the three arcs entering the root $r$ form a multicut, and an optimal flow assigns the value $1/2$ to each of the six directed $s_i - t_i$ paths).

# 3 Algorithms and bounds for multicut

## 3.1 Multicut is bounded by a function of flow

In this section we prove Theorem 1, that $C(G) \leq 108F(G)^3$, provided that $c(e) \geq 1$ for all arcs $e$. But first we prove that such a result is not possible without the "$c(e) \geq 1$ for all $e$" assumption. Garg, Vazirani, and Yannakakis show [8] that there exists $\gamma > 0$ such that for all sufficiently large $n$, there is an $n$-vertex, undirected, unit-capacities network $G'_n$ (on an expander), having $C'(G'_n)/F'(G'_n) \geq \gamma \lg n$. Create a (directed) network $G_n$ from $G'_n$ by replacing each edge by a pair of antiparallel, unit-capacity arcs. We have $F(G_n) \leq F'(2G'_n) = 2F'(G'_n)$ (because any flow in $G_n$ is feasible in $2G'_n$, which is $G'_n$ with its capacities doubled) and $C(G_n) \geq C'(G'_n)$ (because if $M$ is a minimum multicut in $G_n$, then $M' = \{\{u, v\} | (u, v) \in M$ or $(v, u) \in M\}$ is a multicut in $G'_n$ and $|M'| \leq |M|$). Hence $C(G_n)/F(G_n) \geq C'(G'_n)/(2F'(G'_n)) \geq (\gamma/2) \lg n$. Now suppose that $C(G) \leq g(F(G))$ for all directed networks $G$. Choose a large enough $n$ and set $\lambda = F(G_n)$. Let $H_n = G_n/\lambda$ (i.e., scale all capacities down by $\lambda$). We have (using $(\frac{\gamma}{2} \lg n)F(G_n) \leq C(G_n)$)

$$\frac{\gamma}{2} \lg n \quad = \quad \frac{1}{\lambda}(\frac{\gamma}{2} \lg n)F(G_n) \leq \frac{1}{\lambda}C(G_n) = C(H_n) \leq g(F(H_n)) = g(\frac{1}{\lambda}F(G_n)) = g(1),$$

which is a contradiction.

Now we prove Theorem 1, which is restated here for convenience.

**Theorem 1** *There is a polynomial-time algorithm that takes a network $G$ satisfying $c(e) \geq 1$ for all arcs $e$ and finds a multicut $M$ satisfying $c(M) \leq 108\, F(G)^3$.*

**Proof.** We give a polynomial-time algorithm to construct a multicut of capacity at most $108F^3$ in a network $G$ on digraph $(V, E)$ satisfying $c(e) \geq 1$ for all $e$, where $F = F(G)$. First, find a nonnegative, rational length function $x$ satisfying $\sum_e c(e)x(e) = F$ and $\sum_{e \in P} x(e) \geq 1$ for all $s_i \to t_i$ paths $P$, for all $i$. (Such an $x$ is given by an optimal solution to the linear programming relaxation of DIRECTED MULTICUT in Section 2; the optimal value $\sum_{e \in E} c(e)x(e)$ equals $F = F(G)$ by the duality theorem of linear programming.) Define $f = \sum_e x(e) \leq \sum_e c(e)x(e) = F$. For a technical reason, we need $x(e) \leq 1/6$ for all $e$. Replace any arc $e$ with $x(e) > 1/6$ by a path of $\lceil 6x(e) \rceil$ new arcs of length at most $1/6$ each, whose lengths add to $x(e)$, all of whose capacities are $c(e)$.

We need some more definitions. Let $E' \subseteq E$. Given any vertex $s$ and real $\rho$, let $B_{E'}(s, \rho) = \{u \in V|$ there is an $s \to u$ path in $(V, E')$ of length at most $\rho\}$. Define $\delta_{E'}(s, \rho) = \{(a, b) \in E' | a \in B_{E'}(s, \rho), b \notin B_{E'}(s, \rho)\}$. Informally speaking, $B_{E'}(s, \rho)$ denotes the ball with radius $\rho$ and centre $s$ in the digraph $(V, E')$, and $\delta_{E'}(s, \rho)$ denotes the set of arcs of $(V, E')$ that leave this ball.

For our purposes, *the prefix* of path $P =< u_0, u_1, u_2, ..., u_z >$ (whose length may exceed 1) is the path $P' =< u_0, u_1, u_2, ..., u_i >$ where $i$ is minimal such that the length of $P'$ (relative to $x$) is at least $1/6$, and *the suffix* of path $P =< u_0, u_1, u_2, ..., u_z >$ is the path $P' =< u_i, u_{i+1}, ..., u_z >$ where $i$ is maximal such that the length of $P'$ is at least $1/6$.

Here is the algorithm. The embedded comments are needed for the analysis.

/* Let $count(e) = 0$ for all $e \in E$. */
Let $E' = E$.
As long as there is a pair $(s_i, t_i)$ such that some $s_i \to t_i$ path exists in $G' = (V, E')$, repeat:

1. Choose any such $i$.

   /* Find a shortest $s_i \to t_i$ path $P_i$ in $G'$ with respect to $x$. */

2. Find a real number $\rho_i$ which minimizes $c(\delta_{E'}(s_i, \rho))$ among those $\rho$ in the interval $(1/3, 2/3)$.

   /* Let $B_i = B_{E'}(s_i, \rho_i)$. */

   /* Increment $count(e)$ for all arcs $e$ in the prefix of $P_i$. */

3. Remove from $E'$ all arcs in $\delta_{E'}(s_i, \rho_i)$.

Output $M = E - E'$.
End.

Obviously this process terminates and provides a multicut. We claim that the capacity of the multicut is at most $108f^2F \leq 108F^3$.

We need the following lemma, which is implicit in [8]. See also [22, p.204].

**Lemma 5** *Let $G = (V, E)$ be a digraph and let $s \in V$. Let $x : E \to \mathbb{R}^+$ be a length function, $c : E \to \mathbb{R}^+$ be a positive capacity function, and $E' \subseteq E$. Then there is a $\rho \in (1/3, 2/3)$ such that $c(\delta_{E'}(s, \rho)) \leq 3F'$, where $F' = \sum_{e \in E'} c(e)x(e) \leq F$.*

The lemma implies that in a given iteration we cut arcs of capacity at most $3F$.

Call the process of incrementing $count(e)$ *charging* $e$. In each iteration, we charge a set of arcs of total length at least $1/6$, all endpoints of which are in $B_{E'}(s_i, 1/3) \subseteq B_{E'}(s_i, \rho_i) = B_i$, because each arc has length at most $1/6$ and because $\rho_i > 1/3$. Since the total capacity added to $E - E'$ in an iteration is at most $3F$,

$$c(E - E') \leq 18F \sum_{e \in E} x(e)count(e)$$

is an invariant. We prove next that $count(e)$ never exceeds $6f$, and hence

$$c(E - E') \leq (18F)(6f) \sum_{e \in E} x(e) = 108f^2F.$$

Choose any arc $e = (u, v)$ in the original $G$ and relabel the commodities so that we charge $e$ in the iterations for commodities $1, 2, ..., b$, in that order (and no others); these need not be consecutive iterations, of course. We claim, for $i = 1, 2, ..., b$, that:
(1) None of the vertices on the suffix of $P_i$ are in $B_i$.
(2) All the vertices in the suffix of $P_i$ are in $B_1 \cap B_2 \cap B_3 \cap \cdots \cap B_{i-1}$.

Now (1) is trivial, because we chose a $\rho_i$ which is less than $2/3$, and each arc's length is at most $1/6$; hence the endpoints of the suffix are not in $B_i$.

Proving (2) is not much harder. Since the iteration for commodity $i$ charges $e$, $P_i$ must contain $e$. The head $v$ of $e = (u, v)$ must be in $P_1, P_2, \ldots, P_b$, and moreover, $v$ must be in $B_1, B_2, \ldots, B_b$ (in the iteration for commodity $\ell$, all endpoints of arcs we charge are in $B_\ell$). Consider now the subpath $Q$ of $P_i$ starting at $v$ and ending at the last vertex of $P_i$ (clearly, $Q$ contains the suffix of $P_i$). For each $\ell < i$, we claim that $B_\ell$

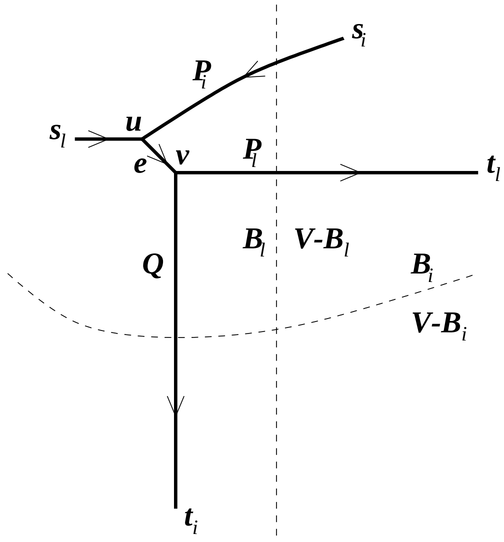

Figure 1: An illustration of the proof of Theorem 1. The dashed lines indicate the “balls” $B_\ell$ and $B_i$.

contains each vertex of $Q$. The reason is that we removed all arcs leaving $B_\ell$ (i.e., all arcs with tails in $B_\ell$ and heads in $V - B_\ell$) at the end of the iteration for commodity $\ell$. Hence, in the iteration for commodity $i$, any path in the current digraph that starts with a vertex in $B_\ell$ must have all its vertices in $B_\ell$ (the path cannot leave $B_\ell$). Since the start vertex $v$ of $Q$ is in $B_\ell$, every vertex of $Q$ is in $B_\ell$. This proves (2). See Figure 1.

We conclude that if $\ell < i$, then the suffix of $P_\ell$ is disjoint from the suffix of $P_i$, because each vertex of the suffix of $P_\ell$ is not in $B_\ell$ and each vertex of the suffix of $P_i$ is in $B_\ell$. Therefore, the sum of the lengths of arcs in $G$ is at least $(1/6)b$ (since there are $b$ disjoint suffixes, each of length at least $1/6$), and hence $(1/6)b \leq f$, or $b \leq 6f$. ■

## 3.2 The region-growing technique

Recall that the digraph is denoted $G = (V, E)$, each arc $e$ has a positive capacity $c(e)$, and there are $k$ commodities, each specified by a source-sink pair $(s_i, t_i)$. Let each arc $e$ have a nonnegative length $x(e)$. (The intention is that $x$ is a feasible solution to the linear programming relaxation of DIRECTED MULTICUT in Section 2.) Let $d_x(v, w)$ denote the shortest-path distance from vertex $v$ to vertex $w$ with respect to arc lengths $x$.

For a vertex set $S \subseteq V$, let $(S, V - S)$ denote the set of arcs leaving $S$, $\{(v, w) \mid v \in S, w \in V - S\}$, and for $E' \subseteq E$, let $c_{E'}(S, V - S)$ denote $c(E' \cap (S, V - S))$. Let $vol_E(S)$ denote the sum of $x(e)c(e)$ over all arcs $e \in E$ that have at least one end vertex (either tail or head) in $S$.

Recall that $F(G)$ denotes the optimal value of the linear program

$$\min\{\sum_e c(e)x(e) \ : \ d_x(s_i, t_i) \geq 1 \ (i = 1, \ldots, k); \ x \geq 0\}$$

and that $vol_E(V) = F(G)$ if the length function $x$ is optimal for the LP.

The next lemma extends Lemma 4.1 (on region growing) of Garg, Vazirani and Yannakakis [8] to directed networks, and has been previously applied by Klein et al. [13].

**Lemma 6 ([8, 13])** *Let $G$, $c$, $x$, and the $k$ commodities be as above. Let $r$ be any positive real and let $q$ be any vertex of $G$. Then there exists a real number $\rho$, $0 < \rho \leq \ln(k+1)/r$, such that*

$$c_E(B, V-B) \leq r \cdot (\mathit{vol}_E(B) + \mathit{vol}_E(V)/k),$$

*where $B$ denotes $B_G(q, \rho)$ (i.e., the set of vertices $v$ such that $G$ has a $q \to v$ path of length at most $\rho$). Moreover, there is an efficient algorithm to find $\rho$ and $B_G(q, \rho)$.*

### 3.3 An algorithm and proof for Theorem 2

Before describing the algorithm, we restate Theorem 2, for convenience.
**Theorem 2** *There is a polynomial-time algorithm that takes a $k$-commodity network $G$ satisfying $c(e) \geq 1$ for all arcs $e$ and and finds a multicut $M$ satisfying $c(M) \leq 39 \ln(k+1) F(G)^2$.*
**Proof of Theorem 2.** Here is the algorithm:

Let $E' = E$, let $M = \emptyset$, and for each $i \in \{1, \ldots, k\}$, let $B_i = \emptyset$.

**While** there is a commodity $i \in \{1, \ldots, k\}$ such that $G' = (V, E')$ has an $s_i \to t_i$ path **do**

Choose such an $i$.

Let $G_i = (V_i, E_i)$ be the subgraph of $G'$ obtained by keeping exactly those vertices and arcs that belong to some $s_i \to t_i$ path in $G'$.

Apply Lemma 6 (the GVY procedure) to $G_i$ with start vertex $q = s_i$ and $r = 3\ln(k+1)$, and let $B_i$ be the vertex set given by the lemma, i.e., $B_i = B_{G_i}(s_i, \rho)$, where $\rho \leq \frac{\ln(k+1)}{r} = \frac{1}{3}$.

Add to $M$ all the arcs of $E_i$ in the cut $(B_i, V_i - B_i)$ in $G_i$.
Replace $E'$ by $E' - M = E - M$.

**End While**

Output the multicut $M$ and stop.

For the analysis, it is convenient to have $x(e) \leq 1/6$ for all arcs $e$. As in the proof of Theorem 1, we replace each arc $e$ with $x(e) > 1/6$ by a path of $\lceil 6x(e) \rceil$ new arcs of length at most $1/6$ each, whose lengths add to $x(e)$, all of whose capacities are $c(e)$.

For each $i \in \{1, \ldots, k\}$ such that $B_i$ is nonempty and for each vertex $v$ in $B_i$, we assign a path of $G_i$, denoted $\sigma(i, v)$, and called the *suffix of $v$ with respect to commodity $i$*. To define $\sigma(i, v)$, take any $s_i \to t_i$ path $P$ of $G_i$ that contains $v$, and let $\sigma(i, v)$ be the suffix of $P$ of length at least $1/6$ and with the fewest vertices. Note that $P$ exists (by our choice of $G_i$ and the fact that $v \in B_i$) and has length at least 1 (since $d_x(s_i, t_i) \geq 1$). Clearly, $\sigma(i, v)$ has length less than $1/3$ (since $x(e) \leq 1/6$, $\forall e \in E$). Note that every vertex $w$ in $\sigma(i, v)$ has $d_x(w, t_i) < 1/3$, and every vertex $u$ in $B_i$ has $d_x(s_i, u) \leq 1/3$; hence, $\sigma(i, v)$ is disjoint from $B_i$.

We need a claim.
**Claim.** Every vertex $w$ of $G$ is in at most $6F(G)$ sets $B_i$, $i \in \{1, \ldots, k\}$.
**Proof of Claim.** Focus on any vertex $w$ and suppose that there are two commodities $i$ and $j$ such that $w$ is in $B_i$ and $B_j$. Assume without loss of generality that the algorithm processed $i$ before $j$.

Suppose that $\sigma(i, w)$ and $\sigma(j, w)$ have a vertex $y$ in common. Then $G_j$ contains a $w \to y$ path called, say, $P_j$. Focus on $G'$ at the start of the iteration for commodity $i$ and call this digraph $G^*$. Clearly, $G^*$ has

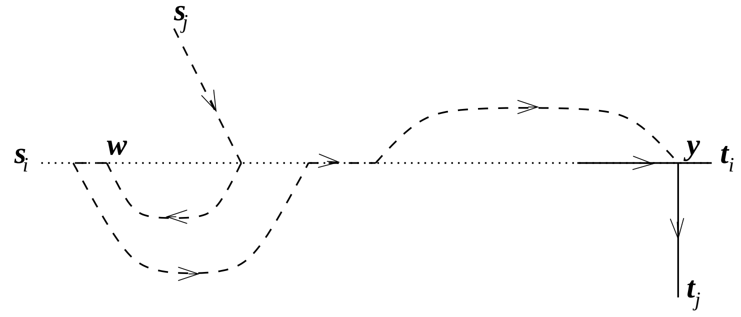


Figure 2: An illustration of the proof of Theorem 2. The solid lines indicate the suffixes $\sigma(i, w)$ (horizontal) and $\sigma(j, w)$ (vertical). $P_j$ is a subpath of the $s_j \to t_j$ path indicated by dashed and solid lines.

an $s_i \to w$ path that is contained in $B_i$ (since $w \in B_i$), $G^*$ contains the $w \to y$ path $P_j$ (since $i$ is processed before $j$), and $G^*$ has a $y \to t_i$ path that is a subpath of $\sigma(i, w)$. By concatenating these three paths, we see that $G^*$ has an $s_i \to t_i$ walk $W$ (allowing repeated vertices) that contains some arcs of $P_j$. Moreover, every arc of $W$ that is in the cut $(B_i, V_i - B_i)$ in $G^*$ is an arc of the middle path $P_j$, because the first of the three paths forming $W$ has all its vertices inside $B_i$ and the last of the three paths forming $W$ has all its vertices outside $B_i$. Shortcut the $s_i \to t_i$ walk $W$ to get an $s_i \to t_i$ path $P$ in $G^*$. Then every vertex of $P$ and arc of $P$ is present in $G_i$. Moreover, in $G_i$, every arc of $P$ in the cut $(B_i, V_i - B_i)$ is an arc of $P_j$, and there is at least one such arc. Hence, at least one arc of $P_j$ is removed from $E'$ by the iteration for commodity $i$. This is a contradiction, since $P_j$ is supposed to be a path of $G_j$. Hence, $\sigma(i, w)$ and $\sigma(j, w)$ must be vertex-disjoint.

This proves the claim, since the suffixes $\sigma(i, w)$, where $i$ is such that $w \in B_i$, are pairwise disjoint, and the number of suffixes is at most $\sum_e x(e)/(1/6) \le 6 \sum_e c(e)x(e) \le 6F(G)$, since each suffix has length at least $1/6$ and $c(e) \ge 1$ for all $e \in E$. See Figure 2. ■

Clearly, the theorem holds if $k = 0$ or if $F(G) = 0$, since $C(G) = 0$ in these cases (the algorithm returns $M = \emptyset$). If $F(G) \neq 0$, then note that $F(G) \ge 1$, by the assumption on the capacities. The rest of the proof follows from Lemma 6 and the claim, since

$$
\begin{aligned}
c(M) &= \sum_{i=1}^{k} c_{E_i}(B_i, V_i - B_i) \\
&\le \sum_{i=1}^{k} 3\ln(k+1)(\textit{vol}_{E_i}(V_i)/k + \textit{vol}_{E_i}(B_i)) \\
&\le 3\ln(k+1)(F(G) + \sum_{i=1}^{k} \textit{vol}_{E_i}(B_i)) \\
&\le 3\ln(k+1)(F(G) + 12F(G)^2) \\
&\le 39\ln(k+1)F(G)^2,
\end{aligned}
$$

where the inequality $\sum_{i=1}^{k} \textit{vol}_{E_i}(B_i) \le 12F(G)^2$ holds because

$$
\sum_{i=1}^{k} \textit{vol}_{E_i}(B_i) \quad = \quad \sum_{i=1}^{k} \sum \{c(u,v)x(u,v) \ : \ (u,v) \in E_i \text{ and } (u \in B_i \text{ or } v \in B_i)\}
$$

$$\leq \sum_{(u,v)\in E} c(u,v)x(u,v)(\kappa(u)+\kappa(v))$$
$$\leq \sum_{(u,v)\in E} c(u,v)x(u,v)(12F(G)) \;\leq\; 12F(G)^2,$$

where $\kappa(v)$ denotes the number of commodities $i \in \{1,\ldots,k\}$ such that vertex $v$ is in $B_i$, and we have $\kappa(v) \leq 6F(G)$ by the claim above. ■

**Remark.** The assumption "$c(e) \geq 1, \forall e \in E$" is used to get the bound "$\kappa(v) \leq 6F(G), \forall v \in V$," and also, it implies $F(G) \leq F(G)^2$. The next result, Theorem 3, uses a variant of this proof that avoids this assumption.

### 3.4 The proof of Theorem 3

We restate Theorem 3, for convenience.
**Theorem 3** *There is a polynomial-time algorithm that takes an $n$-vertex, $k$-commodity network $G$ and finds a multicut $M$ satisfying*
$$c(M) \leq (45\sqrt{n\ln(k+1)})F(G).$$

**Proof of Theorem 3.** The algorithm for Theorem 3 consists of two stages. Let $\alpha > 0$ be a parameter (later, we will fix $\alpha = 1/\sqrt{n\ln(k+1)}$).

In the first stage, we take $M_1$ to be the set of all arcs $e \in E$ such that $x(e) \geq \alpha$, and we take $E' = E - M_1$. $M_1$ is the subset of the multicut found by the first stage, and $E'$ is the arc set of the current digraph after the first stage. (Informally, we "cut" all the arcs in $M_1$ by "rounding up" the LP solution $x$, and these arcs are ignored by the second stage.)

The second stage applies the algorithm of Theorem 2 to $G' = (V, E')$. Let $M_2$ be the multicut found by the second stage. The final multicut obtained by the algorithm is $M = M_1 \cup M_2$.

Consider the capacity $c(M)$ of $M$. First, $c(M_1) = \sum_{e\in M_1} c(e) \leq \sum_{e\in M_1} c(e)x(e)/\alpha \leq \sum_{e\in E} c(e)x(e)/\alpha = F(G)/\alpha$, where the first inequality holds since the arcs $e$ in $M_1$ have been "rounded up" from $x(e) \geq \alpha$ to 1.

We estimate $c(M_2)$ by modifying the analysis in the proof of Theorem 2 to exploit the fact that $x(e) < \alpha$ for all arcs $e$ in the input. Choose any $i \in \{1,\ldots,k\}$ such that $B_i$ is nonempty, let $v$ be any vertex in $B_i$, and focus on the suffix $\sigma(i,v)$. Since $\sigma(i,v)$ has length at least $1/6$ and each arc $e$ (in Stage 2) has $x(e) < \alpha$, there must be at least $(1/6)/\alpha$ vertices in $\sigma(i,v)$. By the claim in the proof of Theorem 2, any two distinct suffixes $\sigma(i,v)$ and $\sigma(j,v)$, $i \neq j$, are vertex-disjoint. Consequently, for any vertex $v$, the number of distinct suffixes $\sigma(i,v)$, $i \in \{1,\ldots,k\}$, is at most $n/(1/(6\alpha)) = 6\alpha n$ (note that we did not use any assumption on the arc capacities). In other words, each vertex is in at most $6\alpha n$ distinct sets $B_i$, $i \in \{1,\ldots,k\}$. An argument similar to that of the proof of Theorem 2 (but without the assumption "$c(e) \geq 1, \forall e \in E$") implies $c(M_2)$ is at most $3\ln(k+1)(1+12\alpha n)F(G)$.

Then $c(M) = c(M_1) + c(M_2) \leq \dfrac{F(G)}{\alpha} + 3\ln(k+1)(1+12\alpha n)F(G)$. We balance the contribution of the two terms by choosing $\alpha = \dfrac{1}{\sqrt{n\ln(k+1)}}$ to get $c(M) \leq 3F(G)(\sqrt{n\ln(k+1)} + 14\sqrt{n\ln(k+1)}) = (45\sqrt{n\ln(k+1)})F(G)$. ■

**Remarks.** (1) Theorem 3 imposes no restrictions on the arc capacities. (2) Theorem 3 implies that the integrality ratio of the linear program is at most $45\sqrt{n\ln(k+1)}$, and hence any network with integrality ratio at least $k/2$ must have $n \geq k^2/(90^2\ln(k+1))$.

### 3.5 Bounded-degree planar digraphs

In this section we prove Theorem 4, which is restated here for convenience.

**Theorem 4** *For every $\Delta$, there is a constant $\gamma$ such that there is a polynomial-time algorithm that takes an $n$-vertex, $k$-commodity ($k \geq 2$) network $G$ with uniform capacities, whose underlying undirected graph is planar, and in which the total degree of every vertex is at most $\Delta$, and finds a multicut $M$ satisfying $c(M) \leq (\gamma\sqrt{\lg k})n^{1/4}F(G)$.*

The following planar separator lemma is implicit in Lipton and Tarjan [17]:

**Lemma 7 ([17])** *For every integer $\Delta > 0$ there exists a constant $\alpha = \alpha(\Delta) \geq 1$ such that for every (undirected) planar multigraph $G = (V, E)$ with maximum degree at most $\Delta$, there are disjoint subsets $L, R \subseteq V$ of size $\lfloor |V|/2 \rfloor$ each, and a set $Z$ of edges of size at most $\alpha\sqrt{|V|}$, such that every edge not in $Z$ either has both endpoints in $L$ or both in $R$. Furthermore, there is a polynomial-time algorithm that finds such a set of edges.*

**Proof of Theorem 4.** By Theorem 2, there is a universal constant $\beta$ such that the multicut size in a uniform-capacity network $G$ is at most $(\beta\lg k)F(G)^2$, if $k \geq 2$.

Fix $\Delta$. We prove the assertion in Theorem 4 with the constant $\gamma = \max\{\frac{\alpha}{1-2^{-1/4}}, \beta\}$. The proof can easily be converted into a polynomial-time algorithm.

Our proof proceeds by induction on $n$. The basis of the induction ($n = 1$) is trivial. Consider an $n$-vertex instance $G$, $n \geq 2$. If $F(G) \leq n^{1/4}/\sqrt{\lg k}$, then by Theorem 2, the minimum multicut is of size at most $(\beta\lg k)F(G)^2 \leq \beta\sqrt{\lg k}\cdot F(G)n^{1/4} \leq \gamma\sqrt{\lg k}\cdot F(G)n^{1/4}$. So, we may assume that $F(G) > n^{1/4}/\sqrt{\lg k}$.

By Lemma 7, we can find a set of at most $\alpha\sqrt{n}$ arcs whose removal partitions $G$ into two subgraphs of order $\lfloor n/2 \rfloor$ each (with perhaps one isolated vertex left over). Clearly, every commodity with terminals in two different components is cut by removing the at-most-$\alpha\sqrt{n}$ arcs. Let $f_1, f_2$ be the maximum flows for the remaining commodities in the two components. By the inductive assertion, for $i = 1, 2$ one can find a multicut of size at most $(\gamma\sqrt{\lg k})f_i\lfloor n/2 \rfloor^{1/4}$ in the $i$th component. (This holds even if either component has 0 or 1 commodity, though this case isn't covered by the inductive assertion.) The union of these multicuts and the separator is a multicut for the entire instance. Its size is at most

$$\begin{aligned}
\alpha\sqrt{n} + \gamma\sqrt{\lg k}\cdot\sum_{i=1}^{2} f_i\lfloor n/2 \rfloor^{1/4} &\leq \alpha\sqrt{n} + \gamma\sqrt{\lg k}\cdot(f_1+f_2)(n/2)^{1/4} \\
&\leq (\alpha + \gamma/2^{1/4})\sqrt{\lg k}\cdot F(G)n^{1/4} \\
&\leq \gamma\sqrt{\lg k}\cdot F(G)n^{1/4},
\end{aligned}$$

as $f_1 + f_2 \leq F(G)$, $F(G) > n^{1/4}/\sqrt{\lg k}$, and $\gamma \geq \alpha + \frac{\gamma}{2^{1/4}}$. ■

## 4 Some simple constructions must be large

In this section we prove that $k$-commodity arc-capacitated or vertex-capacitated networks with a particular structure and integrality ratio at least $k/2$ must have exponentially many vertices. The networks constructed by Saks et al [20] have the structure described in Theorem 10, and so this theorem explains why the number of vertices in these networks is exponential in $k$.

**Theorem 8** *Let $H' = (V', E')$ be an undirected graph in which each edge has some capacity $c(e) > 0$. Replace each edge $e$ by a pair of antiparallel arcs each of capacity $c(e)$ and call the resulting digraph $H = (V', E)$. Add $2k$ distinct new vertices $s_1, t_1, ..., s_k, t_k$, getting vertex set $V = V' \cup \{s_1, t_1, ..., s_k, t_k\}$. Choose subsets $S_i \subseteq V', T_i \subseteq V'$, $S_i \cap T_i = \emptyset$ for all $i = 1, 2, ..., k$ and add arcs $(s_i, v)$ for all $v \in S_i$ and $(u, t_i)$ for all $u \in T_i$, all of infinite (or very large) capacity. Where $G$ is the resulting network,*

$$C(G) \leq (4\gamma \lg n) F(G),$$

*where $\gamma$ is a universal constant and $n = |V'|$.*

Before giving the proof, we give an application of the theorem.

**Corollary 9** *Using the notation of Theorem 8, if $G$ has integrality ratio at least $k/2$, then $n \geq 2^{k/(8\gamma)}$.*

Informally, the theorem implies that if there exists an $n$-vertex network with integrality ratio at least $n^\epsilon$ (for a fixed $\epsilon > 0$), then it must exploit the asymmetry (or, directedness) more than the example of Saks et al.

**Proof of Theorem 8.** Starting with digraph $H$, define a network on $V'$ by constructing one commodity for each pair $(u, v)$ with $u \in S_i$ and $v \in T_i$ for some $i$, having source $u$ and sink $v$; call this network $H$ also. The key point is that $F(H) = F(G)$ and $C(H) = C(G)$.

We now construct an undirected version of $H$ and apply the result of [8] on integrality ratios of undirected networks. Build an undirected network called $H'$ by starting from undirected graph $H'$ and defining a commodity for every unordered pair $\{u, v\}$ such that $u \in S_i, v \in T_i$ for some $i$. We have $C(H) \leq 2C'(H')$ and $F(H) \geq F'(H')$. Apply [8] to infer that there is a universal constant $\gamma$ such that

$$C'(H') \leq \left(\gamma \lg \binom{n}{2}\right) F'(H') \leq (2\gamma \lg n) F'(H')$$

and then conclude that

$$C(G) \;=\; C(H) \;\leq\; 2C'(H') \;\leq\; (4\gamma \lg n) F'(H') \;\leq\; (4\gamma \lg n) F(H) \;=\; (4\gamma \lg n) F(G). \blacksquare$$

Now we state a vertex-capacitated version. A *vertex multicut* in a vertex-capacitated digraph $G$ is a subset of vertices containing at least one vertex on every $s_i \rightarrow t_i$ path, for all $i$. However, to discourage deletion of terminals, we insist that all terminals have infinite capacity. Similarly, a *vertex multicut* in a vertex-capacitated undirected graph $G'$ is a subset of vertices containing at least one vertex on every $s_i - t_i$ path, for all $i$. Again we insist that all terminals have infinite capacity. Let $NC(G), NC'(G')$ denote the minimum capacity of a vertex multicut in digraph $G$ or undirected graph $G'$, respectively.

There is an obvious LP relaxation with nonnegative variables $x(v)$ for all $v \in V$, constrained so that for all $i$, all $s_i \rightarrow t_i$ paths $P$ satisfy $\sum_{v \in P} x(v) \geq 1$, whose objective is the minimization of $\sum c(v)x(v)$; for undirected graphs, we have the same problem, except involving undirected $s_i - t_i$ paths. The corresponding duals are flow problems: Find a nonnegative value for each $s_i \rightarrow t_i$ (or $s_i - t_i$) path, for all $i$, such that the sum of the values on all paths containing vertex $v$ is at most $c(v)$, and maximize the sum of all variables. Let $NF(G), NF'(G')$ be the maximum flow value in digraph $G$ or undirected graph $G'$, respectively.

Garg, Vazirani, and Yannakakis [9] prove a vertex analogue to their arc result: There is a universal constant $\gamma$ such that $NC'(G') \leq (\gamma \lg k) NF'(G')$ for all $G'$ with $k \geq 2$ commodities.

**Theorem 10** *Let $H' = (V', E')$ be an undirected graph in which each vertex $v$ has some capacity $c(v) > 0$ (and edges are uncapacitated). Replace each edge $e$ by a pair of antiparallel arcs and call the resulting digraph $H = (V', E)$. Add $2k$ distinct new vertices $s_1, t_1, ..., s_k, t_k$ having infinite capacities, getting vertex set $V = V' \cup \{s_1, t_1, ..., s_k, t_k\}$. Choose subsets $S_i \subseteq V', T_i \subseteq V'$, $S_i \cap T_i = \emptyset$ for all $i = 1, 2, ..., k$ and add arcs $(s_i, v)$ for all $v \in S_i$ and $(u, t_i)$ for all $u \in T_i$. Where $G$ is the result,*

$$NC(G) \leq (4\gamma \lg n) NF(G),$$

*where $\gamma$ is a universal constant and $n = |V'|$.*

**Corollary 11** *Using the notation of Theorem 10, if $G$ has integrality ratio at least $k/2$, then $n \geq 2^{k/(8\gamma)}$.*

The proof of Theorem 10 is similar to that of Theorem 8, so we omit it, except to note that we make the terminals of $H$ and $H'$ *new* vertices (of infinite capacity) outside of $V'$, since effectively we cannot delete any terminal.

# 5 Further remarks

Anupam Gupta (personal communication, June 2001) has obtained the following improvements, based on a preliminary version of our results.

**Theorem 1′** *There is a constant $\gamma$ such that there is a polynomial-time algorithm that takes a network $G$ satisfying $c(e) \geq 1$ for all arcs $e$ and finds a multicut $M$ satisfying $c(M) \leq \gamma\, F(G)^2$.*

This implies the following improvement of Theorem 3, and also implies an improvement of Theorem 4 without the factor of $\sqrt{\log k}$.

**Theorem 3′** *There is a constant $\gamma'$ such that there is a polynomial-time algorithm that takes an $n$-vertex network $G$ and finds a multicut $M$ satisfying $c(M) \leq (\gamma' \sqrt{n}) F(G)$.*

**Acknowledgments**. We thank Joel Friedman and Mihalis Yannakakis for their help in the early phase of our work and we thank Aravind Srinivasan for his comments.

# References

[1] Y. Aumann and Y. Rabani. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM Journal on Computing* 27:291–301, 1998.

[2] B. Awerbuch and D. Peleg. Sparse partitions. In *Proc. Foundations of Computer Science '90*, pages 503–513.

[3] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing* 23:864–894, 1994. Preliminary version appeared in *Proc. Symposium on the Theory of Computing '92*.

[4] G. Even, J. Naor, S. Rao, and B. Schieber. Divide-and-conquer approximation algorithms via spreading metrics. *Journal of the ACM* 47(4):585–616, July 2000. Preliminary version appeared in *Proc. Foundations of Computer Science '95*.

[5] G. Even, J. Naor, S. Rao, and B. Schieber. Fast approximate graph partitioning algorithms. *SIAM Journal on Computing* 28(6):2187–2214, 1999. Preliminary version appeared in *SODA '97*.

[6] G. Even, J. Naor, B. Schieber, and M. Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica* 20:151–174, 1998. Preliminary version appeared in *IPCO '95*.

[7] L.R. Ford and D.R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics* 8:399–404, 1956.

[8] N. Garg, V.V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM Journal on Computing* 25:235–251, 1996. Preliminary version appeared in *Proc. Symposium on the Theory of Computing '93*.

[9] N. Garg, V. V. Vazirani, and M. Yannakakis. Multiway cuts in directed and node weighted graphs. In *Proc. ICALP '94*, pages 487–498.

[10] T.C. Hu. Multicommodity network flows. *Operations Research* 11:344-360, 1963.

[11] P. Klein, A. Agrawal, R. Ravi, and S. Rao. Approximation through multicommodity flow. *Combinatorica* 15:187–202, 1995. Preliminary version appeared in *Proc. Foundations of Computer Science '90*.

[12] P. Klein, S. Plotkin, and S. Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proc. Symposium on the Theory of Computing '93*, pages 682–690.

[13] P. Klein, S. Plotkin, S. Rao, and É. Tardos. Approximation algorithms for Steiner and directed multicuts. *Journal of Algorithms* 22:241–269, 1997.

[14] B. Korte, L. Lovász, H.J. Prömel, and A. Schrijver, editors. *Paths, Flows, and VLSI-Layout*. Springer-Verlag, 1990.

[15] F.T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. *Journal of the ACM* 46(6):787–832, November 1999. Preliminary version appeared in *Proc. Foundations of Computer Science '88*.

[16] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica* 15:215–245, 1995. Preliminary version in *Proc. Foundations of Computer Science '94*.

[17] R.J. Lipton and R.E. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics* 36(2):177–189, 1979.

[18] J. Naor and L. Zosin. A 2-approximation algorithm for the directed multiway cut problem. In *Proc. Foundations of Computer Science*, pages 548–553, 1997.

[19] S. Plotkin and É. Tardos. Improved bounds on the max-flow min-cut ratio for multicommodity flows. *Combinatorica* 15:425–434, 1995. Preliminary version appeared in *Proc. Symposium on the Theory of Computing '93*.

[20] M. Saks, A. Samorodnitsky, L. Zosin, A lower bound on the integrality gap for minimum multicut in directed networks. Manuscript, July 2000.

[21] P.D. Seymour. Packing directed circuits fractionally. *Combinatorica* 15:281–288, 1995.

[22] D.B. Shmoys. Cut problems and their application to divide-and-conquer. In: D.S. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company, 1997.

[23] É. Tardos and V. V. Vazirani. Improved bounds for the max-flow min-multicut ratio for planar and $K_{r,r}$-free graphs. *Information Processing Letters* 47:77–80, 1993.

[24] M. Yannakakis, Personal communication. February 2000.