Bounds on the Max-Flow Min-Cut Ratio for Directed Multicommodity Flows

Philip N. Klein Serge A. Plotkin Satish Rao and Eva Tardos

Department of Computer Science
Brown University
Providence, Rhode Island 02912

CS-93-30
September 1993

Bounds on the Max-Flow Min-Cut Ratio for Directed Multicommodity Flows

Philip N. Klein* Brown University

Serge A. Plotkin† Stanford University

Satish Rao NEC Research

Éva Tardos‡ Cornell University

Abstract

The most well-known theorem in combinatorial optimization is the classical max-flow min-cut theorem of Ford and Fulkerson. This theorem serves as the basis for deriving efficient algorithms for finding max-flows and min-cuts. Starting with the work of Leighton and Rao, significant effort was directed towards finding approximate analogs for the *undirected* multicommodity flow problem.

In this paper we consider an approximate max-flow min-cut theorem for *directed* graphs. We prove a polylogarithmic bound on the worst case ratio between the minimum multicut and the value of the maximum multicommodity flow in the special case when the demands are symmetric. The method presented in this paper can be used to give polynomial time polylogarithmic approximation algorithms for the corresponding minimum directed multicut problems.

The problem with symmetric demands extends the only previously known special case concerning directed graphs due to Leighton and Rao, who proved an $O(\log n)$ bound for the case when there is a unit demand between every pair of nodes. Computation of minimum cuts in directed multicommodity flow problems with symmetric demand is a basic step for approximation algorithms for a number of NP-complete problems. As an example, we show how to use our multicut approximation algorithm to approximately solve the minimum clause-deletion problem for 2-CNF formulae.

We also consider a generalization of the minimum-cut problem where instead of separating pairs of terminals, we have to separate sets of terminals. Our main result is a polynomial time polylogarithmic approximation algorithm for this generalized minimum cut problem.

---

*Research supported by NSF PYI award CCR-9157620, together with PYI matching funds from Thinking Machines Corporation and Xerox Corporation. Additional support provided by ARPA contract N00014-91-J-4052 ARPA Order No. 8225.

†Research supported by U.S. Army Research Office Grant DAAL-03-91-G-0102 and by a grant from Mitsubishi Electric Laboratories.

‡Research supported in part by a Packard Fellowship, an NSF PYI award, by the National Science Foundation, the Air Force Office of Scientific Research, and the Office of Naval Research, through NSF grant DMS-8920550, and by NEC.

1 Introduction

The multicommodity flow problem involves simultaneously shipping several different commodities from their respective sources to their sinks in a single network so that the total amount of flow going through each edge does not exceed its capacity. Each commodity has an associated demand, which is the amount of the commodity that we wish to ship.

Given a multicommodity flow problem, one often wants to know if there is a *feasible* flow, *i.e.* a flow which both satisfies the demands and obeys the capacity constraints. A cut whose capacity is below the sum of the demands that are separated by the cut proves that no feasible flow exists. The min-cut max-flow theorem for the single-commodity flow problem [1] states that the non-existence of such a "bad" cut proves that a feasible flow does exist. This theorem serves as the basis for deriving efficient algorithms for finding max-flows and min-cuts.

In contrast to single commodity flow, a multicommodity flow problem can be infeasible even if the cut condition is satisfied. A natural question to ask is how large a "safety margin" do we need, *i.e.* how much larger should capacity of a cut be than the sum of the demands separated by the cut in order to ensure existence of a feasible flow. Starting with the work of Leighton and Rao [10], significant effort was directed towards finding such approximate min-cut max-flow theorems for *undirected* multicommodity flow [8, 2, 13]. This sequence of results shows that for k-commodity problems, if the capacity of every cut is at least $O(\log^2 k)$ times the demand separated by the cut, than a feasible multicommodity flow exists.

In this paper we consider two extensions of these results. We consider the max-flow min-cut ratio for *directed* graphs, and a generalization on undirected graphs where instead of pairs of terminals we have *sets of terminals*.

On *directed* graphs we consider the special case of the multicommodity flow problem when the demands are symmetric, that is, each unit of flow sent from a source s to the corresponding sink t also has to return from t to s, though not necessarily along the same path. In the directed case simple cuts do not give a sufficiently strong bound on the maximum flow value, so we consider multicuts instead. A *multicut* is defined by a partition of the nodes into parts $U_1, \ldots, U_q$. An edge vw is in the multicut if v is in a lower-numbered part than w. A source-sink pair is *separated* by the multicut if the source and sink are in different parts. To prove infeasibility of a multicommodity flow problem with symmetric demands it is clearly sufficient to exhibit a multicut whose capacity is below the sum of the demands separated by it.

We show that if the capacity of every multicut is at least $O(\log^3 k)$ times the demand separated by the multicut, than a feasible multicommodity flow exists. The only previously known nontrivial result for directed graphs is due to Leighton and Rao, who proved the analogous theorem with $O(\log n)$ instead of the $O(\log^3 k)$ for the special case when there is a unit demand between every pair of nodes.

Our techniques give polynomial-time approximation algorithms for two variants of the minimum directed multicut problem. We give an $O(\log^2 k)$-approximation algorithm for finding a minimum-capacity multicut that separates all source-sink pairs. Equivalently, the algorithm finds a minimum-capacity set of edges whose deletion breaks the graph into strongly connected components such that, for each source-sink pair, the source and sink are in different components. The special case of this problem when every two nodes form a source-sink pair is the *feedback arc-set problem*, in which one seeks a minimum-capacity set of edges whose deletion makes the graph acyclic. Our results yield an $O(\log^2 n)$-approximation algorithm, which matches the best previously known bound for this special case [11, 14].

As a further application of our minimum capacity multicut algorithm we obtain an $O(\log^2 k)$-approximation algorithm for finding the minimum-weight set of clauses in a 2-CNF formula whose deletion makes the formula satisfiable, where k is the number of literals in the formula. The exact 2-CNF clause-deletion problem is equivalent to the MAX 2-SAT problem, in which one seeks the maximum subset of clauses comprising a satisfiable formula. However, in the context of approximation algorithms, these problems seem to differ in difficulty. It is easy to approximate MAX 2-SAT to within a factor of $1/2$ [4]. Somewhat more difficult algorithms achieve within a factor of $3/4$ [16, 3]. However, a large number of clauses may be discarded by these algorithms in obtaining a satisfiable formula, perhaps much larger than necessary.

We also give an $O(\log^3 k)$-approximation algorithm for the *minimum ratio multicut* problem, *i.e.*, the problem of finding a directed multicut minimizing the ratio of the capacity of the multicut to the sum of the demands that are separated by the multicut. The two minimum multicut problems correspond to two different maximization versions of the multicommodity flow problem. In the *concurrent multicommodity flow problem* the objective is to maximize a common percentage z such that the problem with demands zd_i is feasible. This problem is closely related to the minimum ratio multicut problem. The minimum required "safety margin" discussed above is exactly the maximum possible ratio of the minimum ratio multicut and the maximum concurrent flow.

The problem of finding the minimum capacity multicut separating all source sink pairs corresponds to the *maximum sum symmetric multicommodity flow* problem. Here, instead of satisfying given demands, the goal is to maximize the *sum of the flows* of all the commodities. Clearly, this sum is bounded from above by the capacity of a multicut that separates all of the source sink pairs. We prove that the ratio between the minimum capacity of the multicut separating all source sink pairs and the value of maximum sum symmetric multicommodity flow cannot exceed $O(\log^2 k)$.

We also consider a generalization of the minimum-cut problem where instead of separating pairs of terminals, we have to separate *sets of terminals*. As an example where this problem arises, consider the following problem on communication networks. For each edge of the (undirected) network we are given a nonnegative destruction cost. There are several groups of adversaries that need to communicate. All the members of each group must communicate with each other. Members of these groups are located at the network nodes, and communication is only possible along the edges. The minimum-cost of a multicut that separates all groups is the minimum total cost of communication links that one must be destroyed in order to disable communication in every group.

Analogously to the two versions of the min-cut and max-flow problems for directed multicommodity flows, we consider two versions of the problem. In the concurrent flow formulation we give an $O(\log^2(kt))$ bound on the min-cut and max-flow ratio, where k is the number of commodities and t is the maximum number of terminals per commodity, and we give an $O(\log^2(kt))$-approximation algorithm for finding a cut minimizing the ratio of the capacity of the cut and the sum of the demands separated by the cut. In the maximum sum version our result is as follows. The minimum multicut and maximum flow ratio is at most $O(\log^3(kt))$, and we give an $O(\log^3(kt))$-approximation algorithm for finding the minimum capacity of a multicut that separates all sets of terminals.

2 Preliminaries

Multicommodity flow definitions In a directed graph $G = (V, E)$, let $\mathcal{P}(s, t)$ denote the set of directed paths in G from node s to node t. An *st-flow* f is defined by nonnegative values $f(P)$ assigned to every $P \in \mathcal{P}(s, t)$. In a *symmetric st-flow* f each unit of flow sent from a source s to the corresponding sink t also has to be returned from t to s, though not necessarily along the same path. More precisely, nonnegative values $f(P)$ are assigned to every $P \in \mathcal{P}(s, t) \cup \mathcal{P}(t, s)$ such that $\sum_{P \in \mathcal{P}(s,t)} f(P) = \sum_{P \in \mathcal{P}(t,s)} f(P)$. The *value* of the symmetric flow thus defined is $\sum_{P \in \mathcal{P}(s,t)} f(P)$, the amount of flow sent from s to t.

A *multicommodity flow problem* (or *multiflow* problem for short) is defined by a directed graph G with nonnegative edge-capacities $u(vw)$, and k commodities. A *commodity* is specified by a source-sink pair (s_i, t_i), the terminals of commodity i. A *symmetric multiflow* f consists of symmetric $s_i t_i$-flows f_i for each commodity i. The amount of flow through an edge vw is defined to be the sum of the values of all paths in the flow that include the edge vw. A multiflow is *feasible* if, for each edge vw, the amount of flow through vw is at most its capacity $u(vw)$.

We consider two kinds of optimization problems concerning symmetric multiflows. The *maximum sum symmetric multiflow* problem is to find a feasible symmetric multiflow maximizing the sum of the values of the flows, $\sum_i \sum_{P \in \mathcal{P}(s_i,t_i)} f_i(P)$. In the *concurrent multiflow* problem we are given a nonnegative demand d_i for each commodity i. A multiflow f has *throughput* z if the value of commodity i's flow f_i is zd_i. The concurrent multiflow problem is to find a feasible multiflow with maximum throughput z^*. We will use D to denote the sum $\sum_i d_i$, and in bounds containing D we will assume that the demands are integral.

Multicuts Cuts provide tight combinatorial upper bounds for the maximum flow value in the case of a single commodity flows. In the directed multiflow problem simple cuts do not give strong enough bounds, so we will consider a slightly more general bound obtained using multicuts.

A *directed multicut* is an ordered partition of the nodes set V into disjoint sets $\mathcal{A} = (U_1, U_2, \ldots, U_p)$. The set of edges specified by $\mathcal{A}$ is

$$\{vw \ : v \in U_i, w \in U_j, i < j\}.$$

The capacity of $\mathcal{A}$, denoted $u(\mathcal{A})$, is defined to be the sum of the capacities of the edges in the multicut. A commodity is *separated* by $\mathcal{A}$ if no set U_j includes both terminals of the commodity. Let $d(\mathcal{A})$ denote the sum of the demands of the commodities separated by the multicut $\mathcal{A}$.

The maximum sum symmetric multiflow value is no more than the minimum capacity of a multicut separating all commodities. We show how to find a multicut whose capacity is in turn no more than $O(\log^2 k)$ times the maximum sum symmetric multiflow value.

The maximum throughput of a concurrent multiflow is no more than the minimum capacity-to-separated demand ratio $u(\mathcal{A})/d(\mathcal{A})$. We show how to find a multicut whose capacity-to-separated demand ratio is no more than $O(\log^3 k)$ times the maximum throughput.

The undirected analog of the max-flow min-cut theorem for the concurrent flow problem [10, 8, 2, 13] relates the maximum concurrent flow value to the capacity-to-separated demand ratio of a *cut*, as opposed to a multicut. Observe, however, that simple cuts do not provide strong enough bounds in the directed case. Consider the example of a simple directed cycle with $u(vw) = 1$ on every edge, and n terminal pairs defined to be nodes connected by edges. The minimum capacity-to-separated demand ratio of a simple cut is $1/2$, while the maximum concurrent flow value is $1/n$.

Dual linear programs The maximum sum symmetric multiflow and the concurrent multiflow problems can both be formulated as linear programs. Their dual linear programs provide tight bounds on the maximum value. The dual linear programs are similar. In each, there is a nonnegative length variable $\ell(vw)$ for each edge. Let $\mathrm{dist}_\ell(x, y)$ denote the distance from x to y in G with respect to the length function ℓ. The dual of the maximum sum symmetric multiflow problem is to minimize $W = \sum_{vw} u(e)\ell(e)$ subject to the conditions $\ell \geq 0$ and $\mathrm{dist}_\ell(s_i, t_i) + \mathrm{dist}_\ell(t_i, s_i) \geq 1$ for all commodities i. The dual of the concurrent flow problem is to minimize $W = \sum_{vw} u(e)\ell(e)$ subject to the conditions $\ell \geq 0$ and $\sum_i d_i[\mathrm{dist}_\ell(s_i, t_i) + \mathrm{dist}_\ell(t_i, s_i)] \geq 1$.

To understand the dual linear programs it is best to think of them a "volume conditions" for feasibility analogous to the "cut condition" that the sum of demands separated by a cut should not exceed the capacity of the cut. Imagine the graph as a network of pipes with edge vw having length $\ell(vw)$ and cross-section $u(vw)$. Then W is the total volume of the pipe system. Consider, for example, the dual of the maximum sum symmetric flow problem. The condition $\mathrm{dist}_\ell(s_i, t_i) + \mathrm{dist}_\ell(t_i, s_i) \geq 1$ in the dual problem guarantees that every unit of symmetric flow must use at least one unit of volume in the pipe system. Therefore the total volume of the pipe system is an upper bound on the maximum sum symmetric flow value.

Multicuts can be viewed as special solutions of the linear programming dual for both multiflow problems. In the case of the maximum sum symmetric multiflow problem multicuts correspond exactly to the integer solutions to the dual linear program.

3 Approximate Min-Multicut Max-Flow Theorems

The basic outline of our proof and algorithm is analogous to the outline of the algorithms for the undirected case [10, 8, 2]. Given a feasible solution to the linear programming dual of the multiflow problem we can find a multicut with value at most a logarithmic factor above the value of the dual solution. Since an optimal dual solution value is equal to the optimal value of the multiflow problem, this will prove both the approximate max-flow min-cut theorem, and that the multicut produced by the algorithm is close to optimal.

The decomposition theorem In essence, the decomposition theorem states that given a feasible solution to the linear programming dual of the maximum sum multiflow problem we can find a multicut whose capacity is not much more than the dual objective function value.

Let ℓ be a nonnegative length function on the edges, and let δ denote the minimum round trip distance, $\min_i[\mathrm{dist}_\ell(s_i,t_i)+\mathrm{dist}_\ell(t_i,s_i)]$, and let $W=\sum_{vw}\ell(vw)u(vw)$, denote the total volume of the corresponding pipe system.

Theorem 3.1 Consider a graph G with edge-capacities, edge-lengths, k pairs of terminals, and δ and W as defined above. In polynomial time one can find a multicut of capacity at most $\delta^{-1}O(\log^2 k)W$ that separates every terminal pair.

We are going to prove the theorem through a sequence of lemmas. The first lemma is a straightforward adaptation of the related undirected graph lemma in [2], and is stated without proof. For a node set U let $e(U)$ denote the set of edges with at least one endpoint in U, $\Gamma^{\text{out}}(U)\subseteq e(U)$ denote the set of edges leaving U, and $\Gamma^{\text{in}}(U)\subseteq e(U)$ denote the set of edges entering U. For a set of edges F let $u(F)=\sum_{vw\in F}u(vw)$, and $W(F)=\sum_{vw\in F}u(vw)\ell(vw)$.

Lemma 3.2 For any positive ϵ and p, and any node s there exists a set U containing s such that $u(\Gamma^{\text{out}}(U))\le\epsilon\left(W/p+W(e(U))\right)$, and every node in U is at a distance less than $\epsilon^{-1}\ln(p+1)$ from s.

Now set $\epsilon=4\delta^{-1}\ln(k+1)$. Let s be a source, and let R^{out} denote the "outregion" constructed using this ϵ, $p=k$, and node s. Similarly, (by considering graph with reverse edges) we construct an "inregion" R^{in}. By the choice of ϵ for any node x in the intersection $R^{\text{out}}\cap R^{\text{in}}$, there is an x-s-path and an x-s-path of length less than $\delta/4$. Therefore, the fact that the distance from s_i to t_i plus the distance from t_i to s_i is at least δ by the assumption on ℓ, implies the following lemma.

Lemma 3.3 There are no pairs of terminals $\{s_i,t_i\}$ in the intersection $R^{\text{out}}\cap R^{\text{in}}$.

Lemma 3.4 Consider a graph G with edge-capacities, edge-lengths, k pairs of terminals, and δ and W as defined above Theorem 3.1. In polynomial time one can find a multicut $\mathcal{A}=(U_1,\ldots,U_p)$ of capacity at most $\delta^{-1}O(\log k)W$ such that each part U_j contains at most $k/2$ pairs of terminals.

Proof: We give a recursive algorithm to construct the multicut. Select a source s, and construct the R^{out} and R^{in} using Lemma 3.2 as used in Lemma 3.3. It follows from Lemma 3.3 that one of the two regions contains at most $k/2$ terminal pairs. Let U be this region.

We delete region U from the graph along with all adjacent edges $e(U)$, and recursively construct a multicut $\mathcal{A}' = (U_1, \ldots, U_r)$ in the remaining graph. In applying Lemma 3.2 we use $p = k$ in all levels of the recursion. We add the part U to multicut $\mathcal{A}'$. If U was an outregion, the new multicut is $\mathcal{A} = (U, U_1, \ldots, U_r)$. If U was an inregion, the new multicut is $\mathcal{A} = (U_1, \ldots, U_r, U)$. In either case, the capacity we added in going from $\mathcal{A}'$ to $\mathcal{A}$ is at most $4\delta^{-1}\ln(k+1)(W/k + W(e(U))$.

Now consider the capacity of the resulting multicut $\mathcal{A} = (U_1, \ldots, U_p)$. Lemma 3.2 bounds the contribution of each region U_j to the capacity of the multicut. The bound consist of two parts, a term independent of the region, $4\delta^{-1}\ln(k+1)W/k$; and a term depending on the edges adjacent to the region. In each recursion we reduce the number of pairs by at least one, so the number p of parts is at most k. Furthermore, since the edges adjacent to set U have been deleted from the graph before the recursive call, the sets of edges whose weight is used to bound the contribution are disjoint. Hence the capacity of $\mathcal{A}$ is at most $8\delta^{-1}\ln(k+1)W$. ∎

Proof of Theorem 3.1: The Theorem follows from Lemma 3.4 by induction on k. First apply Lemma 3.4, then apply the inductive hypothesis to the problems defined by the graphs spanned by each of the regions U_j, the pairs of terminals included in U_j, and the length function ℓ. ∎

Maximum sum multiflow min-max theorem Let ℓ be an optimal dual solution of the maximum multicommodity flow problem. The dual objective value is $W = \sum_{vw} u(vw)\ell(vw)$. The dual constraints ensure that $\text{dist}_\ell(s_i, t_i) + \text{dist}_\ell(t_i, s_i) \geq 1$, for all commodities i. Hence by Theorem 3.1 there is a multicut of capacity at most $O(\log^2 k)W$, that is, at most $O(\log^2 k)$ times the dual objective value. Since the dual objective value equals the primal objective value, the maximum multiflow value, we obtain the following theorem.

Theorem 3.5 Given an instance of directed symmetric maximum sum multiflow problem with k commodities, one can find a multicut $\mathcal{A}$ such that

$$\phi^* \leq \min_{\mathcal{A}} u(\mathcal{A}) \leq O(\log^2 k)\phi^*$$

where ϕ^* is the value of the maximum sum multiflow.

Maximum concurrent multiflow min-max theorem For the concurrent multiflow problem we have the following min-max theorem.

Theorem 3.6 Given a directed concurrent flow problem with symmetric demands, one can find a multicut $\mathcal{A}$ such that

$$z^* \leq \frac{u(\mathcal{A})}{d(\mathcal{A})} \leq O(\log^3 k)z^*.$$

Similarly to the case of undirected graphs, the proofs of Theorems 3.1, and 3.5 can be modified to prove a weaker version of Theorem 3.6 with one of the $\log k$'s replaced by a $\log D$ in the upper bound. Kahale [5] has shown that in the undirected case the min-max theorem for concurrent multiflows can be derived essentially directly from the maximum sum multiflow theorem. His proof can easily be adapted to the directed case. The basis of his method is the following lemma.

Lemma 3.7 For given positive $\delta_1, \dots, \delta_k$, and integers $d_1, \dots, d_k$ there exists a set $Q \subseteq \{1, \dots, k\}$ such that

(1)
$$\frac{\sum_{i=1}^{k} d_i \delta_i}{\ln(1 + \sum_{i=1}^{k} d_i)} \leq (\sum_{i \in Q} d_i)(\min_{i \in Q} \delta_i).$$

To derive the weakened version of Theorem 3.6 using this lemma, let ℓ be an optimal dual solution, and set $\delta_i = \mathrm{dist}_\ell(s_i, t_i) + \mathrm{dist}_\ell(t_i, s_i)$ for every commodity i. Apply Lemma 3.7 to the these values of δ_i and the demands d_i to obtain a set of commodities Q. Next apply Theorem 3.1 to the length function ℓ, the commodities in Q, and $\delta = \min_{i \in Q} \delta_i$.

To get the stronger bound, independent of the size of the demands, we use a modification of the technique of Plotkin and Tardos [13]. In essence we prove that up to small constant factors the worst min-cut/max-flow ratios occur in problems with integer demands that are bounded by a small-degree polynomial in k. Although modifying the undirected techniques to apply to the directed case is nontrivial, we omit the details due to lack of space.

4 The 2-CNF Clause Deletion Problem

In this section we apply the approximation algorithm for the minimum capacity multicut problem presented in Section 3 to approximately solve the problem of finding the minimum-weight collection of clauses whose deletion make a given 2-SAT formula satisfiable.

The basis for the reduction is the following graph construction. Given a 2-SAT formula F, we can assume its clauses have the form $p \to q$, where p and q are literals (variables or negations of variables). We construct an auxiliary graph $G(F)$ with a node for each literal. For each clause $p \to q$ of weight w, there is an edge $p \to q$ and an edge $\bar{q} \to \bar{p}$, each of capacity w. It is well-known that the formula F is satisfiable if and only if no strongly connected component of F contains both a variable and its negation.

Given a 2-SAT formula F, we construct the auxiliary graph $G(F)$. It follows easily from the construction that the minimum capacity of a multicut in $G(F)$ separating every variable from its negation is at most twice the minimum weight of a set of clauses whose deletion makes F satisfiable. Conversely, given any multicut $\mathcal{A}$ that separates every variable from its negation, one can find a set of clauses having weight at most the capacity of $\mathcal{A}$ whose deletion makes the formula satisfiable.

We can use the algorithm of Theorem 3.5 to find a multicut separating each variable from its negation. The multicut found has capacity at most $O(\log^2 k)$ times the minimum total capacity of an edge set whose deletion separates each variable from its negation. We obtain the following theorem.

Theorem 4.1 There exists a polynomial-time algorithm to approximate the minimum-weight deletion problem for 2-CNF formulae. The solution output has weight $O(\log^2 k)$ times optimal, where k is the number of variables in the formula.

5 Steiner demands

Definitions In this section we consider multicommodity flow on undirected graphs where instead of pairs of terminal we are given *sets* of terminals. We address the problem of finding a minimum-capacity multicut that separates each such set of terminals. The main result in this section is an $O(\log^3(kn))$-approximation algorithm for the generalized problem.

Let S_i denote the sets of terminals for commodity i where $1 \leq i \leq k$. A multicut $\mathcal{A}$ is a partition of V into sets $U_0, \ldots, U_p$; we say that a multicut separates terminal set S_i if there is no set U_j containing all of S_i. The capacity of the multicut $u(\mathcal{A})$ is the sum of the capacities of all edges crossing from one set in the partition to another, i.e. $u(U_1, \ldots, U_p) = \frac{1}{2}\sum_i u(U_i)$. The goal is to find a minimum-capacity multicut that separates all k sets of terminals.

In the related *minimum ratio Steiner cut* problem there is a demand d_i associated with every set of terminals. The *demand* $d(U)$ across a cut U is the sum of the demands associated with the separated sets. The problem is to find a cut that minimizes the ratio of the capacity of the cut and the demand across the cut. In this minimum cut problem demands express the desirability of separating the corresponding sets of terminals.

The minimum-ratio Steiner cut problem is closely related to the following generalization of the multicommodity flow problems that we will refer to as the *concurrent Steiner flow* problem. In contrast to concurrent flow, where each commodity is associated with a single pair of terminals, here commodity i is associated with a set of terminals S_i, for $i = 1, \ldots, k$. Each commodity i also has an associated demand d_i. A *Steiner flow* f_i of commodity i in G connecting terminal set S_i is defined as a collection of Steiner trees spanning the terminal sets S_i; each tree has an associated real value. Let $\mathcal{T}_i$ denote a collection of Steiner trees that span terminals S_i, and let $f_i(\tau)$ be a nonnegative value for every tree $\tau \in \mathcal{T}_i$. The *value* of the Steiner flow thus defined is $\sum_{\tau \in \mathcal{T}_i} f_i(\tau)$. The amount of flow of commodity i through an edge vw is given by $f_i(vw) = \sum\{f_i(\tau) : \tau \in \mathcal{T}_i \text{ and } vw \in \tau\}$. A flow is *feasible* if it satisfies the capacity constraints, $\sum_i f_i(vw) \leq u(vw)$ for every edge $vw \in E$. The goal is to maximize a common percentage z such that there exists a feasible Steiner flow for which the flow of commodity i is zd_i.

One can view this concurrent Steiner flow problem as a fractional packing of Steiner trees into a capacitated graph, where for each i we pack at least zd_i trees associated with set of nodes S_i. Notice that the integer version of this tree packing problem, namely integer packing of

Steiner trees with given sets of terminals, arises in VLSI design, in particular in the problem of routing multiterminal nets.

The minimum-ratio Steiner cut provides a simple combinatorial bound on the maximum possible value of z. We prove that this bound is the best possible up to a factor of $O(\log^2(kn))$. We also give a polynomial time $O(\log^2(kn))$-approximation algorithm for finding the minimum-ratio Steiner cut. Using this algorithm as a subroutine we also give an $O(\log^3(kn))$-approximation algorithm for finding the minimum capacity Steiner multicut separating all sets of terminals.

Finding approximately minimum-ratio Steiner cuts The maximum concurrent Steiner flow problem can be formulated as a linear program, albeit one with an exponential number of variables. In order to find an optimal solution using the ellipsoid algorithm, or the packing algorithm of [12], we would have to solve the minimum-cost Steiner tree problem, which is NP-hard. Instead of working with this linear program, therefore, we work with a closely related one, in which the Steiner trees are *restricted*. Using restricted Steiner trees essentially corresponds to using the standard 2-approximation algorithm for the minimum-cost Steiner tree problem. In this extended abstract we will ignore this problem for simplicity of presentation.

The linear programming dual of the concurrent Steiner flow problem is as follows. There is a nonnegative cost variable $\ell(e)$ for every edge e. Let $\mathrm{weight}_\ell(S_i)$ denote the minimum cost of a Steiner tree with terminal set S_i. The goal is to minimize $\sum_e \ell(e)u(e)$ subject to the constraint $\sum_i d_i \,\mathrm{weight}_\ell(S_i) \geq 1$.

We will use W to denote $\sum_e \ell(e)u(e)$, and $W(F) = \sum_{e\in F} \ell(e)u(e)$ for a subset of the edges $F \subseteq E$. For a subset of nodes S we use $e(S)$ to denote the set of edges having at least one endpoint in S. The key of the proof is the following lemma due to Garg, Vazirani and Yannakakis [2].

Lemma 5.1 For any positive ϵ and p, and any node s, there exists a set U containing s such that $u(\Gamma(U)) \leq \epsilon\,(W/p + W(e(U)))$, and every node in U is at a distance less than $\epsilon^{-1}\ln(p+1)$ from s.

We need some additional notation in order to be able to state the main theorems of this section more precisely. For Steiner flow and cut problems with sets of terminals S_i for $i = 1, \ldots, k$, let t_i denote $|S_i|$, and let $t = \max_i t_i$. The number of terminals $|\cup_i S_i|$ is denoted by T.

Theorem 5.2 The minimum ratio of a Steiner cut is at most $O(\log T \log(kt))z^*$, where z^* is the maximum concurrent Steiner flow value. An $O(\log T \log(kt))$-approximation for the minimum cut value can be found in polynomial time.

Proof: In this extended abstract we prove a weaker version of this theorem with the $\log(kt)$ in the bounds replaced by a $\log D$ where $D = \sum_i (t_i - 1)d_i$, and we assume that the demands are integral. To improve the weaker bound involving $\log D$ to the bound claimed in the theorem we use the technique of Plotkin and Tardos [13].

We prove the weaker bound by induction on D. Let σ^* denote the minimum ratio of a cut. Let ℓ denote the optimum dual solution to the concurrent Steiner flow problem. Then z^* equals the dual optimum value $W = \sum_i d_i \operatorname{weight}_\ell(S_i)$. Since the dual solution satisfies the constraint $\sum_i d_i \operatorname{weight}_\ell(S_i) \geq 1$, it suffices to prove the following inequality:

$$\sum_i d_i \operatorname{weight}_\ell(S_i) \leq O(\log T \log D)\frac{W}{\sigma^*}. \tag{2}$$

We proceed along the lines of the proof of Klein, Rao, Agrawal and Ravi [8]. Apply Lemma 5.1 for a source s, $q = T$, and an appropriately chosen ϵ. Delete the resulting set U from the graph along with all the edges in $e(U)$, and apply the lemma to another terminal in the remaining graph. Repeat until no more terminals are left. Let $\mathcal{A} = (U_0, U_1, \ldots, U_p)$ denote the resulting multicut with U_0 denoting the set remaining when there are no more terminals left. By adding the bounds on capacities of the cuts we get that the total capacity of the multicut does not exceed $2\epsilon W$.

Consider a terminal set S_i. The multicut $\mathcal{A} = (U_0, \ldots, U_k)$ partitions the set S_i. For every j such that $S_i \cap U_j \neq \emptyset$, select one element in the intersection to represent the intersection, and let $\tilde{S}_i$ denote the set of these representatives. The fact that the diameter of each one of the sets U_j for $j \geq 1$ is bounded by $2\epsilon^{-1}\ln(T+1)$ implies that

$$\operatorname{weight}_\ell(S_i) \leq \operatorname{weight}_\ell(\tilde{S}_i) + 2|S_i - \tilde{S}_i|\epsilon^{-1}\ln(T+1). \tag{3}$$

We use the induction hypothesis on the problem with demands d_i and set of terminals $\tilde{S}_i$ for $i = 1, \ldots, k$. We want to set ϵ to guarantee that $\tilde{D} = \sum_i d_i(|\tilde{S}_i| - 1)$, the quantity corresponding to D in the new problem, is at most half of D. This is done by setting $\epsilon = \frac{\sigma^* D}{8W}$. By definition, for each j, we have $u(U_j)/d(U_j) \geq \sigma^*$. Hence we have

$$\sigma^* \leq \frac{\sum_j u(U_j)}{\sum_j d(U_j)} = \frac{2u(\mathcal{A})}{\sum_{i:|\tilde{S}_i|\neq 1} d_i|\tilde{S}_i|} \leq \frac{2u(\mathcal{A})}{\sum_i d_i(|\tilde{S}_i| - 1)} \leq \frac{4\epsilon W}{\tilde{D}}.$$

By the induction hypothesis

$$\sum_i d_i \operatorname{weight}_\ell(\tilde{S}_i) \leq O(\log T \log \tilde{D})\frac{W}{\sigma^*}.$$

Together with inequalities (3) and the fact that $\tilde{D} \leq D/2$, this implies the desired inequality (2).

Since we do not know the value of σ^*, in order to turn this proof into an algorithm we make do with an estimate of it. The estimate is initialized by considering single-node cuts and it is updated each time the algorithm discovers that the estimate is sufficiently far from the observed value. ∎

Using Theorem 5.2 repeatedly we obtain an approximation algorithm for finding minimum-capacity Steiner cuts that separate all of the demands.

Theorem 5.3 An $O(\log T \log k \log(kt))$-approximation to the minimum capacity of a multicut separating each set of terminals S_i for $i = 1, \ldots, k$, can be found in polynomial time.

Remark: The problem of finding a minimum capacity multicut separating all sets of terminals is naturally related to another Steiner flow problem, the *maximum sum Steiner flow* problem. It is not hard to show along the lines of the proof of Theorem 5.3 that the min-multicut/max-flow ratio for this problem is at most $O(\log T \log k \log(kt))$.

Acknowledgments

We are grateful to Jon Kleinberg for many helpful discussions. In particular, we would like to thank him for pointing out a crucial flaw in an earlier approach to making the min-multicut/max-flow bound for the directed symmetric concurrent multiflow problem independent of D, the sum of the demands.

References

[1] L. R. Ford, Jr. and D. R. Fulkerson. *Flows in Networks*. Princeton Univ. Press, Princeton, NJ, 1962.

[2] N. Garg, V. V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. To appear in *Proc. 25th ACM Symposium on the Theory of Computing*, May 1993.

[3] M. X. Goemans and D. P. Williamson. A new $\frac{3}{4}$-approximation algorithm for MAX SAT. *Proc., Third Conference on Integer Programming and Combinatorial Optimization*, pages 313-322, 1993.

[4] D. S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. System Sci.* vol. 9, pages 256-278.

[5] N. Kahale. On reducing the cut ratio to the multicut problem. Unpublished manuscript, 1993.

[6] P. Klein, S. Plotkin, and S. Rao. Planar graphs, multicommodity flow, and network decomposition. To appear in *Proc. 25th ACM Symposium on the Theory of Computing*, May 1991.

[7] P. Klein, S. Plotkin, C. Stein, and É. Tardos. Faster approximation algorithms for the unit capacity concurrent flow problem with applications to routing and finding sparse cuts. *SIAM J. Comput.*, to appear. Preliminary version appeared as "Leighton-Rao might be practical: faster approximation algorithms for concurrent flow with uniform capacities," *Proceedings, 22nd ACM Symposium on Theory of Computing*, pages 310-321, 1990.

[8] P. N. Klein, S. Rao, A. Agrawal, and R. Ravi. An approximate max-flow min-cut relation for multicommodity flow, with applications. *Combinatorica*, to appear. Preliminary version appeared as "Approximation through multicommodity flow," In *Proc. 31th IEEE Annual Symposium on Foundations of Computer Science*, pages 726–727, 1990.

[9] T. Leighton, F. Makedon, S. Plotkin, C. Stein, S. Tragoudas, and É. Tardos. Fast approximation algorithms for multicommodity flow problem. In *Proc. 23th ACM Symposium on the Theory of Computing*, pages 101–111, May 1991.

[10] T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *Proc. 29th IEEE Annual Symposium on Foundations of Computer Science*, pages 422–431, 1988.

[11] F. T. Leighton and S. Rao. An Approximate Max-Flow Min-Cut Theorem for Uniform Multicommodity Flow Problems with Applications to Approximation Algorithms. Unpublished Manuscript.

[12] S. Plotkin, D. B. Shmoys, and É. Tardos. *Fast approximation algorithms for fractional packing and covering problems*. In *Proc. 32th IEEE Annual Symposium on Foundations of Computer Science*, pages 495–505, 1991. Also available as Technical Report 999, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, 1992.

[13] S. Plotkin and E. Tardos. Improved bounds on the max-flow min-cut ratio for multicommodity flows. To appear in *Proc. 25th ACM Symposium on the Theory of Computing*, May 1993.

[14] S. Rao. Finding small cuts: Theory and Applications. Ph. D. Thesis, MIT, 1989.

[15] S. Tragoudas. *VLSI partitioning approximation algorithms based on multicommodity flow and other techniques*. PhD thesis, University of Texas at Dallas, 1991.

[16] M. Yannakakis. On the approximation of maximum satisfiability. *Proc., 3rd ACM-SIAM Symposium on Discrete Algorithms*, pages 1-9, 1992.