SIAM J. COMPUT.
Vol. 25, No. 2, pp. 235–251, April 1996

001



# APPROXIMATE MAX-FLOW MIN-(MULTI)CUT THEOREMS AND THEIR APPLICATIONS*

NAVEEN GARG†, VIJAY V. VAZIRANI†, AND MIHALIS YANNAKAKIS‡

**Abstract.** Consider the multicommodity flow problem in which the object is to maximize the sum of commodities routed. We prove the following approximate max-flow min-multicut theorem:

$$\frac{\text{min multicut}}{O(\log k)} \leq \text{ max flow } \leq \text{ min multicut},$$

where $k$ is the number of commodities. Our proof is constructive; it enables us to find a multicut within $O(\log k)$ of the max flow (and hence also the optimal multicut). In addition, the proof technique provides a unified framework in which one can also analyse the case of flows with specified demands of Leighton and Rao and Klein et al. and thereby obtain an improved bound for the latter problem.

**Key words.** approximation algorithm, multicommodity flow, minimum multicut

**AMS subject classifications.** 68Q25, 90B10

## 1. Introduction.

Much of flow theory, and the theory of cuts in graphs, is built around a single theorem—the celebrated max-flow min-cut theorem of Fort and Fulkerson [FF] and Elias, Feinstein, and Shannon [EFS]. The power of this theorem lies in that it relates two fundamental graph-theoretic entities via the potent mechanism of a min-max relation.

The importance of this theorem has led researchers to seek its generalization to the case of *multicommodity flow.* In this setting, each commodity has its own source and sink, and the object is to maximize the sum of the flows subject to capacity and flow conservation requirements. The notion of a *multicut* generalizes that of a cut and is defined as a set of edges whose removal disconnects each source from its corresponding sink. Clearly, maximum multicommodity flow is bounded by minimum multicut; the question is whether equality holds. This can be established for some special cases, e.g., if there are only two commodities [Hu]; however, one can construct very simple examples to show that equality does not hold in general. Consider a tree of height one with three leaves. Each pair of leaf vertices form the source–sink pair for a commodity. All edges have unit capacities. The max flow in this graph is $\frac{3}{2}$, whereas the minimum multicut is 2.

Why does the theorem hold for a single commodity, and why does the generalization fail? For an explanation, consider the LP formulation of the maximum multicommodity flow problem. As shown in §5, the dual of this is the LP relaxation of the minimum multicut problem, i.e., the optimal *integral* solution to the dual is the minimum multicut. In general, the vertices of the dual polyhedron are not integral. However, for the case of a single commodity, they are integral (see [GV] for an exact characterization), and the max-flow min-cut theorem is simply a consequence of the LP-duality theorem. For the multicommodity case, the LP-duality theorem shows only that maximum flow is equal to the minimum fractional (i.e., relaxed) multicut.

In this situation, the best one can hope for is an approximate max-flow min-cut theorem. In ground-breaking work, Leighton and Rao [LR] gave the first such

* Received by the editors January 19, 1993; accepted for publication (in revised form) August 5, 1994.

† Department of Computer Science and Engineering, Indian Institute of Technology, Delhi, India.

‡ AT&T Bell Laboratories, Murray Hill, NJ 07974 (mihalis@research.att.com).

theorem. Let us consider a second formulation of the multicommodity flow problem that has also been widely studied in the past. In this formulation, a demand, $D_i$, is specified for each commodity, $i$. The object is to determine the maximum number, $f$, called *throughput*, such that $fD_i$ amount of each commodity $i$ can be routed simultaneously, subject to capacity and conservation constraints. (Equivalently, the object is to determine the minimum number, $u$, such that if the capacity of each edge is multiplied by $u$, then all the demands can be simultaneously satisfied. Clearly, at optimality $f = \frac{1}{u}$.) The analogue of a minimum cut in this case is a sparsest cut into two parts, one that minimizes the ratio of capacity of the cut to the demand across the cut. Let $\alpha$ be this minimum. Clearly, $f \leq \alpha$, and once again equality does not generally hold.

Leighton and Rao considered a special case of the above-stated formulation, called uniform multicommodity flow, in which there is a commodity corresponding to each pair of vertices and all the demands are unity. They proved the following approximate max-flow min-cut theorem:

$$\frac{\alpha}{O(\log n)} \leq f \leq \alpha,$$

where $n$ is the number of vertices in the graph. Subsequently, Klein et al. [KARR] managed to attack the arbitrary demands problem, and proved that

$$\frac{\alpha}{O(\log C \log D)} \leq f \leq \alpha,$$

where $C$ is the sum of capacities of all edges and $D$ is the sum of all demands. However, one restriction they impose is that all capacities and demands be integral. The lower bound was later improved to $\alpha/O(\log n \log D)$ by Tragoudas [Trag]. [LR], [KARR], and [Trag] also give polynomial-time algorithms for finding an approximation to the sparsest cut, the factors being $O(\log n)$ and $O(\log C \log D)$ (or $O(\log n \log D)$), respectively.

We address the first version of the multicommodity flow problem, henceforth referred to as the *maximum multicommodity flow* problem, and prove the following approximate max-flow min-multicut theorem:

$$\frac{M}{O(\log k)} \leq f \leq M,$$

where $f$ is the max flow, $M$ is the minimum multicut, and $k$ is the number of commodities. We also show that our theorem is tight up to a constant factor, and we give a polynomial-time algorithm for finding a multicut within $O(\log k)$ of the optimal fractional, and therefore also of the integral multicut.

Our general approach is similar to that of Leighton and Rao [LR] and Klein et al. [KARR]. We consider the LP-relaxation of the minimum multicut problem and use its optimal solution to define a graph with distance labels on the edges. Starting from a source or a sink, we grow a region in this graph until we find a cut of small enough capacity separating the root from its mate. The region is removed and the process is repeated. Our method differs in several respects from previous methods. It dispenses with the discretization of the edge distances and employs a technique that leads to quicker termination of the region growth process and thus produces a better bound on the capacity of the cut. These techniques are encapsulated in the two region-growing lemmas of §4, which use the idea of packing cuts to grow regions.

Our analysis is also useful in the demands version of the multicommodity flow problem. It establishes a unified framework in which simpler proofs of the theorems of Leighton and Rao and Klein et al. can be given by dispensing with tokenizing distances. We also avoid the dynamic resetting of the parameters for region growing and restarting the procedure. In both cases we use heavily ideas from the original papers. Using our lemmas, we improve the [KARR] and [Trag] results to

$$\frac{\alpha}{O(\log k \log D)} \leq f \leq \alpha.$$

We also dispense with the restriction that capacities be integral. Furthermore, Plotkin and Tardos [PT] give a method of scaling demands so that the $\log D$ factor in these results can be replaced by $\log k$, thus yielding an improved bound of $O(\log^2 k)$ on the gap between $f$ and $\alpha$.

The following problem is a generalization of the uniform multicommodity flow problem considered by Leighton and Rao and by Tragoudas. It was called the *product multicommodity flow problem* in [LR] and the *complete concurrent flow problem* in [Trag]. In this problem, each vertex has a nonnegative weight $w(v)$ (assumed to be positive integer in [LR] and [Trag] but this is not essential), and there is a commodity for each unordered pair $u$, $v$ of vertices, with a demand of $w(u)w(v)$. The object again is to maximize the throughput subject to capacity and conservation constraints. Let

$$\alpha = \min_{S \subseteq V} \frac{C_{\nabla(S)}}{w(S)w(\bar{S})},$$

where $w(S)$ in the sum of weights of vertices in $S$.

Let $W$ be the sum of weights of all vertices, and $C$ be the sum of all edge capacities. Then, Leighton and Rao prove $\alpha/O(\log \min(\frac{W}{w_{\min}}, \frac{C}{c_{\min}})) \leq f \leq \alpha$, where $w_{\min}$ is the weight of the lightest vertex, and $c_{\min}$ is the minimum over all vertices of the sum of capacities of edges incident at the vertex. Tragoudas improves the lower bound to $\alpha/O(\log n)$ [Trag]. Using our techniques, we improve this result to

$$\frac{\alpha}{O(\log k)} \leq f \leq \alpha,$$

where $k$ is the number of vertices having nonzero weight.

The multicut problem finds numerous applications, e.g., in circuit partitioning problems. It was first stated by Hu in 1963 [Hu]. For $k = 1$, the problem coincides with the ordinary min cut problem. For $k = 2$, it can also be solved in polynomial time by two applications of a max-flow algorithm [YKCP]. The problem was proven NP-hard and MAX SNP-hard for any $k \geq 3$ by Dalhaus et al. [DJPSY]. Because of the MAX SNP-hardness, there is no polynomial-time approximation scheme for multicut for $k \geq 3$ (assuming $P \neq NP$) [ALMSS]. Note that, in the demands case, the sparsest cut problem can be solved in polynomial time for fixed $k$ (or $k = O(\log n)$) because in this case we are concerned only with cuts into two parts, and thus we can try all possible partitions of the sources and sinks into two parts and compute the minimum cut for each partition.

Dahlhaus et al. [DJPSY] studied the *multiway cut* (or *multiterminal cut*) problem: given a set of "terminal" vertices $T$, find a minimum weight set of edges that disconnects every terminal from every other terminal. This is the special case of the multicut problem where there is one commodity for every pair of vertices from the subset $T$. [DJPSY] gave a factor-of-2 approximation algorithm for this case and showed that it can be used to approximate the general multicut problem within a factor of 2 with a running time that has $k$ in the exponent. Thus the running time is polynomial only for fixed $k$.

Klein et al. [KARR] used their approximation algorithm for the sparsest cut to give an $O(\log^3 n)$ approximation algorithm for multicut. They also gave some applications of the multicut problem obtaining approximation algorithms with ratio $O(\log^3 n)$ for the following problems: deleting the minimum number of clauses to make a 2CNF $\equiv$ formula satisfiable, deleting the minimum number of edges from a graph to make it bipartite, and a via minimization problem in VLSI. Our improvement for multicut gives us an $O(\log n)$ approximation algorithm for these problems.

**2. LP formulations for max multicommodity flow.** Given an undirected graph $G = (V, E)$, a capacity function $c : E \rightarrow \Re^+$, and $k$ pairs of vertices (not necessarily distinct) $\{s_i, t_i\}$ $1 \leq i \leq k$, we associate a commodity, $i$, with the pair $\{s_i, t_i\}$ and designate $s_i$ as the source and $t_i$ as the sink for commodity $i$. A *multicommodity flow* is a way of simultaneously routing commodities from their sources to sinks, subject to capacity and conservation constraints.

The assumption that each commodity has a single source and a single sink can be made without loss of generality. The more general case where a commodity $i$ may have a set $S_i$ of sources and a set $T_i$ of sinks can be easily reduced to this one by adding a new source $s_i$ with edges to the vertices in $S_i$ and a new sink $t_i$ with edges to $T_i$.

A multicommodity flow in which the sum of the flows over all the commodities is maximized will be called a *max (multicommodity) flow.* A *multicut* is defined as a set of edges whose removal disconnects each $\{s_i, t_i\}$ pair. The weight of the multicut is the sum of the capacities of the edges in it. The MULTICUT problem is to find a multicut of minimum weight.

We say that two vertices *share index* $i$ if they form the source–sink pair for commodity $i$.

Assume that there exist edges $(t_i, s_i)$ in $G, 1 \leq i \leq k$. These edges are special; the only flow allowable on edge $(t_i, s_i)$ is commodity $i$ flowing from $t_i$ to $s_i$. There are no capacity restrictions on these edges. This allows us to view max flow as a circulation in which the sum of the flows in the edges $(t_i, s_i), 1 \leq i \leq k$, is to be maximized. Let $f^l_{ij}$ denote the flow of commodity $l$ in edge $(i, j)$. The LP formulation of the problem is as follows:

$$
\begin{array}{llll}
\text{maximize} & \displaystyle\sum_{i=1}^{k} f^i_{t_i s_i} & & \\
\text{subject to} & \displaystyle\sum_{(j,i)\in E} f^l_{ji} - \sum_{(i,j)\in E} f^l_{ij} \leq 0 & \forall i \in V & \forall l \in [l, \ldots, k], \\
& \displaystyle\sum_{l=1}^{k} f^l_{ij} + \sum_{l=1}^{k} f^l_{ji} \leq c_{ij} & \forall (i,j) \in E - \bigcup_{i=1}^{k} \{(t_i, s_i)\}, & \\
& f^l_{ij} \geq 0 & \forall (i,j) \in E & \forall l \in [1, \ldots, k].
\end{array}
$$

The first set of inequalities says that the total flow of each commodity into vertex $i$ is at most the total flow out of it. Note that, if these inequalities hold for each vertex $i \in V$, then in fact they must all hold with equality, thereby implying flow conservation at each node. This is because a deficit in the flow balance at one node must imply a surplus at some other. The second set of inequalities are capacity constraints on the edges; the total flow over all commodities summed in both directions is at most the capacity of the edge.

The dual of this LP is

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{(i,j)\in E} d_{ij}c_{ij} & \\ \text{subject to} & d_{ij} \geq p_i^l - p_j^l \quad \forall (i,j) \in E - \bigcup_{i=1}^{k}\{(t_i, s_i)\} & \forall l \in [1,\ldots,k], \\ & p_{s_i}^i - p_{t_i}^i \geq 1 \quad \forall l \in [1..k], & \\ & p_i^l \geq 0 \quad \forall i \in V & \forall l \in [1,\ldots,k], \\ & d_{ij} \geq 0 \quad \forall (i,j) \in E - \bigcup_{i=1}^{k}\{(t_i, s_i)\}. & \end{array}$$

The variable $d_{ij}$ can be viewed as a distance label on the edge $(i,j)$ and $p_i^l$ as the potential corresponding to commodity $l$ on vertex $i$. Thus the dual problem is an assignment of potentials to vertices and distance labels to edges so that the potential difference (for each commodity) across each edge is no more than the distance label of that edge. Furthermore, the potential difference between the source and the sink for each commodity should be at least 1. These two conditions imply that the distance between each $s_i, t_i$ under the distance label assignment $d_{ij}$ is at least 1. The following LP (with, however, exponentially many constraints) expresses this much more simply.

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{e\in E} d_e c_e \\ \text{subject to} & \displaystyle\sum_{e\in E} d_e q_i^j(e) \geq 1 \quad \forall q_i^j \\ & d_e \geq 0 \quad \forall e \in E, \end{array}$$

where $q_i^j$ denotes the $j$th path in $G$ (under some arbitrary numbering) from $s_i$ to $t_i$ and $q_i^j(e)$ is the characteristic function of this path, i.e., $q_i^j(e) = 1$ if $e \in q_i^j, 0$ otherwise.

Clearly the distance labels of a feasible solution to the first LP give a feasible solution to the second LP with the same objective function. Conversely, given a feasible solution to the second LP compute potentials on the vertices for each commodity as follows:

$$p_i^l = \text{length of the shortest path from vertex } i \text{ to the sink for commodity } l,$$
$$\text{under distance labels } d_e.$$

It can be shown that these potentials, together with the distance labels $d_e$, are a feasible solution to the first LP with the same objective function. Hence the two formulations of the dual are equivalent.

The dual program can now be viewed as an assignment of nonnegative distance labels $d_e$ to edges $e \in E$, so as to minimize $\sum_{e\in E} d_e c_e$, subject to the constraint that each $\{s_i, t_i\}$ pair be at least a unit distance apart. An integral solution to the dual problem corresponds to a multicut; the edges with $d_e = 1$ form a multicut. Hence, the dual is the LP relaxation of the MULTICUT problem.

**3. Overview of the algorithm.** In this section we will give a high-level description of our algorithm, justifying the steps taken on intuitive grounds.

Our goal is to pick a set of edges of small capacity whose removal separates all $s_i, t_i$ pairs; the total capacity of edges picked should be within a small factor of the max flow (our factor is $O(\log k)$). Clearly, such edges will be bottlenecks for the max flow, so one possibility is to find a max flow using an LP subroutine and start with the set of saturated edges. A better possibility is to find an optimal solution to the dual LP and consider the set of edges having positive distance labels. Notice that, by complementary slackness, $d_e > 0 \Rightarrow \sum_{l=1}^{k} f_{ij}^l + \sum_{l=1}^{k} f_{ji}^l = c_e$, where $e = (i, j)$, i.e., $e$ must be saturated in every max flow. Moreover, the edges $D = \{e | d_e > 0\}$ constitute a multicut.

The entire set $D$ may have a very large capacity; we wish to pick a small capacity subset that is still a multicut. The optimal dual solution is the most cost effective way of picking a fractional multicut. This provides the clue that, for our purpose, edges with large distance labels should be more important than edges with small distance labels. Our algorithm indirectly gives preference to edges having large distance labels. We start by defining the length of edge $e$ in $G$ as $d_e$. We then find disjoint sets, called regions, such that for each set $S$, $C_{\nabla(S)} \leq \epsilon . wt(S)$, where $C_{\nabla(S)}$ is the capacity of the cut $(S, \overline{S})$, $\epsilon$ is an appropriately chosen parameter, and $wt(S)$ is roughly $\sum c_e d_e$, where the sum is over all edges having at least one endpoint in $S$. No region contains both source and sink of any pair, and for each commodity either the source or the sink is in some region. Under these conditions, the union of the cuts of the regions is a multicut and has capacity bounded by $2\epsilon F$, where $F$ is the value of the maximum flow.

The overall approach of finding the optimal fractional solution to the dual LP and then growing regions was introduced by Leighton and Rao [LR] for the uniform multicommodity flow problem. The procedure for growing regions is similar to a graph clustering technique first proposed by Awerbuch in [Aw] (for graphs without capacities or lengths on the edges), and is also similar to that used by Leighton and Rao [LR] and Klein et al. [KARR] (for graphs with capacities and lengths) in the context of multicommodity flows. Each region is formed by growing out radially, with respect to the edge lengths $d_e$, from one of the sources, as in the usual shortest path computation; the region is grown as long as it accumulates weight fast enough. The reason for adopting radial growth is that this maximizes the weight of the region for a given bound on the pairwise distance between vertices in the region. The region-growing process is formally described in the next section. The main differences from [LR] and [KARR] are in the initialization of the process (assignment of initial weights $wt$ for the roots of the regions), the elimination of the discretization of the lengths of the edges, and the use of auxiliary variables associated with the layers of the radial growth. The idea of packing cuts is used for growing the regions and for accounting. This yields simpler proofs, as well as a more precise bound on $\epsilon$.

**4. Two crucial lemmas.** In this section we shall prove two region-growing lemmas that will be central to our multicut algorithm. We shall prove these in sufficient generality so that they can be applied to the other versions of the multicommodity flow problem as well.

Given a graph $G = (V, E)$, a capacity function $c : E \to \Re^+$ and distance labels $d : E \to \Re^+$ on the edges, define $B = \sum_{e \in E} d_e c_e$. A subset of vertices $V' \subseteq V$ is provided to the region-growing algorithm as the set of candidate roots from which regions will be grown. In our case, $V'$ is the set of sources and sinks. We associate a variable $y_S$ with each subset $S \subset V$; initially $y_S = 0$ for all $S$. The cut associated with a set $S$, denoted by $\nabla(S)$, is the set of edges with exactly one end point in $S$. The capacity of the cut, $C_{\nabla(S)}$, is $\sum_{e \in \nabla(S)} c_e$.

**4.1. Growing a region.** A region is grown in a radial manner starting from a root vertex, $r$. The order in which vertices are included in the region is the same as the order in which Dijkstra's algorithm finds shortest paths to vertices from $r$. We begin by picking a vertex, $r \in V'$, and assign it a weight $wt(r) = B/q$, where $q = |V'|$.

At any point in the algorithm we identify a set, $A$, as the active set and raise its variable $y_A$. Initially, the active set is $\{r\}$. Define the weight enclosed by the set $A$ as

$$wt(A) = \sum_{S \subseteq A} y_S C_{\nabla(S)} + wt(r).$$

It is important that the $y_S$'s must form a packing, i.e., $\forall e \in E$: $\sum_{S: e \in \nabla(S)} y_S \leq d_e$. Thus, if while raising $y_A$ we find that $\sum_{e \in \nabla(S)} y_S = d_e$ for some edge $e = (u, v) \in \nabla(A)$, $u \in A$, we make the set $A \cup \{v\}$ active, i.e., $A \leftarrow A \cup \{v\}$, and start increasing the variable corresponding to it. We keep growing the active set in this manner, one vertex at a time, until

$$C_{\nabla(A)} \leq \epsilon \cdot wt(A) \tag{1}$$

is satisfied, where $\epsilon$ is a constant that will be set appropriately while applying the lemma. Let $\mathcal{R}$ denote the active set for which condition 1 is satisfied.

Define the *radius* of $A$, $rad(A) = \sum_{S \subseteq A} y_S$.

LEMMA 4.1. $rad(\mathcal{R}) < \ln(q+1)/\epsilon$.

*Proof.* The claim is trivial if $rad(\mathcal{R}) = 0$, so assume $rad(\mathcal{R}) > 0$. Let $S_1, S_2, \ldots, S_l$ denote the successive sets for which the variable $y_S > 0$. It is easy to check that these sets are nested, i.e., if $i < j$ then $S_i \subset S_j$. In what follows we denote the value of the variable $y_{S_i}$ by $y_i$ and $C_{\nabla(S_i)}$ by $C_i$.

Since, while raising the variable $y_{S_i}$ from 0 to $y_i$, condition 1 was not satisfied

$$C_i \geq \epsilon \cdot wt(S_i).$$

From our definition of the weight enclosed by a set, it follows that

$$\begin{aligned} wt(S_i) &= wt(S_{i-1}) + y_i C_i \\ &\geq wt(S_{i-1}) + \epsilon y_i wt(S_i), \end{aligned} \tag{2}$$

where for $i = 1$ we let $wt(S_0) = wt(r)$ in the above equation. Note that $wt(S_{i-1}) > 0$ for all $i$ and hence $0 < \epsilon y_i < 1$. Thus,

$$wt(S_i) \geq \frac{wt(S_{i-1})}{(1 - \epsilon y_i)}.$$

Hence,

$$wt(S_l) \geq \frac{wt(r)}{(1-\epsilon y_1)(1-\epsilon y_2)\cdots(1-\epsilon y_l)}$$
$$= \frac{B}{q(1-\epsilon y_1)(1-\epsilon y_2)\cdots(1-\epsilon y_l)}.$$

Since the $y_S$'s form a packing ($\forall e \in E : \sum_{S:e\in\nabla(S)} y_S \leq d_e$), it follows that

$$\sum_{i=1}^{l} y_i C_i = \sum_{i=1}^{l}\left(y_i \sum_{e\in\nabla(S_i)} c_e\right) = \sum_{e\in E}\left(c_e \sum_{S:e\in\nabla(S)} y_S\right) \leq \sum_{e\in E} c_e d_e = B,$$

and hence,

$$wt(S_l) = \sum_{i=1}^{l} y_i C_i + wt(r) \leq B + \frac{B}{q} = B\left(1+\frac{1}{q}\right).$$

Therefore,

$$\frac{B}{q(1-\epsilon y_1)(1-\epsilon y_2)\cdots(1-\epsilon y_l)} \leq wt(S_l) \leq B\left(1+\frac{1}{q}\right),$$

which implies that

$$\frac{1}{(1-\epsilon y_1)(1-\epsilon y_2)\cdots(1-\epsilon y_l)} \leq q+1.$$

Taking natural logs we get

$$\sum_{i=1}^{l} \ln(1-\epsilon y_i)^{-1} \leq \ln(q+1).$$

From (2) it follows that $0 < \epsilon y_i < 1$, $1 \leq i \leq l$. Since $\ln(1-x)^{-1} > x$ for $0 < x < 1$,

$$\epsilon \sum_{i=1}^{l} y_i < \ln(q+1).$$

Thus, $rad(\mathcal{R}) = \sum_{i=1}^{l} y_i < \ln(q+1)/\epsilon$. □

Let $dist_d(u,v)$ denote the shortest path distance between $u$ and $v$ under the distance label assignment $d$. Consider vertex $v \in \mathcal{R}$. Let $S_i$ be the first set containing $v$, i.e., $v \in S_i - S_{i-1}$. Then from the manner in which we grow the region it follows that

$$dist_d(r,v) = rad(S_{i-1}). \tag{3}$$

COROLLARY 4.2. *For all $u, v \in \mathcal{R}$, $dist_d(r,u) \leq rad(\mathcal{R})$ and $dist_d(u,v) \leq 2rad(\mathcal{R})$.*

**4.2. Growing disjoint regions.** Having grown a region rooted at an arbitrary vertex in $V'$, remove all vertices contained in the region and grow another region starting from a new root picked from $V'$. Continue in this manner until the residual graph contains no vertex of $V'$. Let $\mathcal{R}_1, \mathcal{R}_2 \ldots, \mathcal{R}_p$ denote the regions formed. It is easy to see that these are disjoint. Clearly, $p \leq |V'| = q$.

Let $M = \nabla(\mathcal{R}_1) \cup \nabla(\mathcal{R}_2) \cup \cdots \cup \nabla(\mathcal{R}_p)$.

LEMMA 4.3. $\sum_{e \in M} c_e \leq 2\epsilon B$.

*Proof.* Let $G_i = (V_i, E_i)$ be the graph obtained by deleting vertices contained in $\cup_{j=1}^{i-1} \mathcal{R}_j$ $(G_1 = G)$. Furthermore, let $C^i_{\nabla(S)}$ denote the capacity of the cut $\nabla(S)$ in $G_i$. Note that $\sum_{e \in M} c_e = \sum_{i=1}^{p} C^i_{\nabla(\mathcal{R}_i)}$.

Each region $\mathcal{R}_i$ satisfies condition 1. Therefore, $C^i_{\nabla(\mathcal{R}_i)} \leq \epsilon . wt(\mathcal{R}_i)$, $1 \leq i \leq p$. Hence,

$$\sum_{i=1}^{p} C^i_{\nabla(\mathcal{R}_i)} \leq \epsilon \sum_{i=1}^{p} wt(\mathcal{R}_i)$$

$$= \epsilon \left( \sum_{i=1}^{p} \sum_{S \subseteq \mathcal{R}_i} y_S C^i_{\nabla(S)} + \sum_{i=1}^{p} wt(r_i) \right),$$

where $r_i$ is the root of region $\mathcal{R}_i$. Since the $y_S$'s form a packing,

$$\sum_{i=1}^{p} \sum_{S \subseteq \mathcal{R}_i} y_S C^i_{\nabla(S)} \leq \sum_{e \in E} d_e c_e = B.$$

Also,

$$\sum_{i=1}^{p} wt(r_i) = \frac{B}{q} p \leq B.$$

Thus, $\sum_{i=1}^{p} C^i_{\nabla(\mathcal{R}_i)} \leq 2\epsilon B$, and hence $\sum_{e \in M} c_e \leq 2\epsilon B$. □

When growing region $\mathcal{R}_j$ in the graph $G_j$, we have that $\sum_{e \in E_j} d_e c_e \leq \sum_{e \in E} d_e c_e = B$. However, the proof of Lemma 4.1 goes through without any modifications. Regarding (3), note that $rad(S_{i-1})$ is now the shortest path distance between $r$ and $v$ in the graph $G_j$; the shortest distance between these vertices in $G$ might be even smaller. Hence Corollary 4.2 still holds.

COROLLARY 4.4. $\sum_{i=1}^{p} C_{\nabla(\mathcal{R}_i)} \leq 4\epsilon B$.

*Proof.* An edge in $M$ occurs in at most two cuts $\nabla(\mathcal{R}_i)$, $1 \leq i \leq p$. Thus,

$$\sum_{i=1}^{p} C_{\nabla(\mathcal{R}_i)} \leq 2 \sum_{e \in M} c_e \leq 4\epsilon B. \qquad \square$$

The time complexity of growing disjoint regions is $O(m+n \log n)$ as our algorithm is essentially the same as Dijkstra's algorithm for shortest paths.

**5. Approximate max-flow min-multicut theorem.** Clearly, the max flow, $F$, is less than the weight of the minimum multicut, $M$, i.e., $F \leq M$.

The main result of this section is an algorithm that finds a multicut of weight at most $F \cdot O(\log k)$. We state this as a theorem for later reference.

THEOREM 5.1 (approximating the minimum multicut). *Consider an instance of the* MULTICUT *problem specified by a graph* $G = (V, E)$, *a capacity function* $c : E \rightarrow \Re^+$, *and* $k$ *pairs of vertices. One can, in polynomial time, find a multicut separating the specified pairs of vertices having weight within a factor* $O(\log k)$ *of the maximum flow over these pairs.*

Since $M$ is the minimum multicut, $M \leq F \cdot O(\log k)$. Thus the ratio of the optimal integral solution to the optimal fractional solution of the dual program is at most $O(\log k)$.

COROLLARY 5.2 (approximate max-flow min-multicut theorem). $F \leq M \leq F \cdot O(\log k)$.

Furthermore, this bound on the ratio of the minimum multicut and maxflow is tight, as shown in Theorem 5.4.

For planar graphs, Tardos and Vazirani [TV] obtain a constant factor approximation for the minimum multicut. Garg, Vazirani, and Yannakakis [GVY] approximate the minimum multicut on trees to within twice the optimal. They also give a factor-$\frac{1}{2}$ approximation algorithm for maximum integral multicommodity flow on trees, and they show that even for planar graphs the integrality gap for flow is unbounded, thereby ruling out LP duality based methods for approximating maximum integral multicommodity flow. Both these results also establish approximate max-flow min-multicut theorems.

**5.1. Finding the multicut.** First, solve the dual LP to obtain a set of distance labels, $d_e$, $e \in E$. Next, grow regions as in §4. The constant $\epsilon$ and the set $V'$ are chosen to ensure that $\nabla(\mathcal{R}_1) \cup \nabla(\mathcal{R}_2) \cup \cdots \cup \nabla(\mathcal{R}_p)$ is a multicut.

The vertices of $V$ that are the source for some commodity form the set $V'$, i.e., $V' = \cup_{i=1}^{k} \{s_i\}$. Thus, $s_i \in \cup_{j=1}^{p} \mathcal{R}_j$, $1 \leq i \leq k$. Now if we can choose $\epsilon$ so that no two vertices in $\mathcal{R}_i$, $1 \leq i \leq p$, share an index, we shall be finished.

LEMMA 5.3. *If* $\epsilon = 2\ln(k+1)$, *then no two vertices in* $\mathcal{R}_i$ *share an index.*

*Proof.* Note that $q = |V'| \leq k$. Therefore, if $\epsilon = 2\ln(k+1)$, then, by Lemma 4.1, $rad(\mathcal{R}_i) < \frac{1}{2}$. Hence, by Corollary 4.2, the distance between any two vertices in $\mathcal{R}_i$ is less than 1. Since the assignment of distance labels to edges is such that the distance between any two vertices sharing an index is at least 1, no two vertices in $\mathcal{R}_i$ share an index. $\square$

Substituting this choice of $\epsilon$ into Lemma 4.3, we find that the multicut obtained has weight at most $4B\ln(k+1)$. Since $B = \sum_{e \in E} d_e c_e = F$, the max flow, the weight of the multicut is within a factor $4\ln(k+1)$ of the max flow.

**5.2. A tight example.**

THEOREM 5.4. *For all* $k, n, k \leq n$, *there exists an* $n$ *vertex graph,* $G$, *and* $\Omega(k^2)$ *pairs of vertices in* $G$ *such that the ratio between the minimum multicut and the max flow is* $\Omega(\log k)$.

*Proof.* As in [LR], we use an expander graph and similar arguments, but here we need to choose an appropriate set of source–sink pairs and consider cuts into many parts. Let $G = (V, E)$ be a $k$-vertex, bounded degree expander graph (each vertex has degree at most $d$, for an appropriate constant $d$). Every vertex has at most $\frac{k}{2}$ vertices within distance $\log_d\left(\frac{k}{2}\right)$. Thus, $G$ has $\Omega(k^2)$ pairs of vertices that are a distance $\log_d\left(\frac{k}{2}\right)$ or more apart. Let these be the pairs for the MULTICUT instance. All edges of the graph have unit capacity, and hence the total capacity of the edges is $O(k)$. Since each flow path is $\Omega(\log k)$ long, the maximum flow is $O(k/\log k)$.

The optimum multicut induces a partition of the vertex set of $G$. Since $G$ is an expander, any set $S$ in the partition has $\Omega(|S|)$ edges running across it, provided $|S| \leq \frac{k}{2}$. Any subgraph with more than $\frac{k}{2}$ vertices has pairs that are $\log_d\left(\frac{k}{2}\right)$ apart, and hence contains a pair of vertices that share an index. Thus, no set in the partition induced by the multicut has more than $\frac{k}{2}$ vertices, and so each set $S$ in the partition has $\Omega(|S|)$ edges running across it. Hence the number of edges in the multicut is $\Omega(k)$. This yields a ratio of $\Omega(\log k)$ between the weight of the minimum multicut and the maxflow.

Note that splitting an edge by adding vertices on the edge does not change the max flow or the minimum multicut. We modify $G$ into an $n$-vertex graph by adding an appropriate number of vertices on the edges of $G$. □

We remark that, in the case of the multiway cut problem [DJPSY] (i.e., the special case of the multicut problem where the given set of source–sink pairs for the commodities consists of all pairs of vertices from a given subset $S$ of terminals), the gap between min cut and max flow is much smaller; it is at most $2 - \frac{2}{k}$ [Cu]. Of course, if $S$ is the whole set of vertices (i.e., there is one commodity for every pair of vertices), the problem is trivial and there is no gap: max flow = min cut = total capacity of the graph.

## 6. Multicommodity flow with specified demands.

We next consider the case when along with the source and sink for commodity $i$, $1 \leq i \leq k$, we are also specified a demand, dem$(i)$, for the commodity. A multicommodity flow is *feasible* if it meets the demand for each commodity.

As in §4 we define the cut associated with $S$, denoted by $\nabla(S)$, as the set of edges with exactly one end point in $S$. The capacity of the cut, $C_{\nabla(S)}$, is $\sum_{e\in\nabla(S)} c_e$. The set $S$ separates commodity $i$ if and only if exactly one of $\{s_i, t_i\}$ is in $S$. The demand across the cut, $D_{\nabla(S)}$, is the sum of the demands of all commodities separated by set $S$. Clearly, the following condition is necessary for the existence of a feasible multicommodity flow.

*Cut condition.* For all $S \subseteq V$, $C_{\nabla(S)} \geq D_{\nabla(S)}$.

However, this condition is not sufficient. Extensive work has been done on characterizing graphs with distinguished sources and sinks for which the cut condition is both necessary and sufficient. Again, no complete characterization is known.

Klein et al. view this problem as follows: they wish to find the minimum factor, $u$, by which the capacity of the edges should be raised to ensure a feasible flow. Equivalently, they wish to find the maximum factor, $f$, such that it is possible to route simultaneously an amount $f \cdot$ dem$(i)$ of each commodity while satisfying the capacity constraints. At optimality $f = \frac{1}{u}$.

Let $q_i^j$ denote a path in $G$ from $s_i$ to $t_i$ and $q_i^j(e)$ be the characteristic function of this path, i.e., $q_i^j(e) = 1$ if $e \in q_i^j$, 0 otherwise. Then an LP formulation for this problem is

$$\begin{array}{lll} \text{maximize} & f & \\ \text{subject to} & \displaystyle\sum_{i,j} f_i^j q_i^j(e) \leq c_e & \forall e \in E, \\ & \displaystyle\sum_j f_i^j \geq f.\,dem(i), & 1 \leq i \leq k, \\ & f_i^j \geq 0 & \forall q_i^j, \end{array}$$

where $f_i^j$ is the flow along the path $q_i^j$. Thus, multicommodity flow is feasible if and only if $f$ is at least 1.

We can formulate the dual LP which now calls for an assignment of nonnegative distance labels, $d_e$, to edges $e \in E$ so as to minimize $\sum_{e \in E} d_e c_e$, subject to the constraint $\sum_{l=1}^{k} dist_d(s_l, t_l) dem(l) \geq 1$, where $dist_d(u, v)$ is the shortest path distance between $u$ and $v$ under this assignment of distance labels. At optimality, we have $\sum_{e \in E} d_e c_e = f$.

How does $f$ relate to the structure of the graph? The cut condition motivates the following definition. Define the *sparsest cut* as the cut that minimizes the ratio $C_{\nabla(S)}/D_{\nabla(S)}$. Let

$$\alpha = \min_{S \subseteq V} \frac{C_{\nabla(S)}}{D_{\nabla(S)}}.$$

Clearly, $f \leq \alpha$. Klein et al. show that the "throughput" $f$ is at least $\alpha/O(\log C \log D)$, where $C$ is the sum of all capacities and $D$ is the total demand. This was later improved to $\alpha/O(\log n \log D)$ by Tragoudas [Trag].

We improve this result by providing a tighter bound on $f$.

THEOREM 6.1.

$$\frac{\alpha}{O(\log k \log D)} \leq f \leq \alpha.$$

*Thus, for a multicommodity flow to be feasible it is necessary that the sparsest cut ratio, $\alpha$, be at least* 1 *and sufficient that it be* $O(\log k \log D)$.

Plotkin and Tardos [PT] give a method of scaling demands so that the log $D$ factor in Theorem 6.1 can be replaced by log $k$. Hence, they improve the bound in Theorem 6.1 to $O(\log^2 k)$.

For uniform multicommodity flow on bounded degree expanders, the ratio between $f$ and $\alpha$ is $O(\log n)$ ($n$ is the number of vertices) [LR]. We can, as in Theorem 5.4, modify this by adding new vertices on edges to get a graph on $n$ vertices and $k$ commodities ($k \leq \frac{n(n-1)}{2}$) for which $f = \alpha/O(\log k)$. It is an open problem to bridge the gap between the $O(\log^2 k)$ bound of [PT] and this $O(\log k)$ example.

For planar graphs, Klein, Plotkin, and Rao [KPR] show that multicommodity flow is feasible if the sparsest cut ratio, $\alpha$, is logarithmic.

**6.1. Proof of Theorem 6.1.** First solve the dual LP to obtain a set of distance labels, $d_e$, $e \in E$. By LP duality we have that the optimal "throughput" $f$ is equal to $B = \sum_{e \in E} d_e c_e$. We will find a cut whose ratio of capacity to demand is within a factor $\rho = 16 \ln(k+1) \log D$ of $f$ (and hence also of the optimal ratio $\alpha$).

As in Klein et al., the algorithm proceeds in phases. Each phase involves growing regions as in §4. If the source $s_j$ and sink $t_j$ of a commodity belong to the same region in a phase we will say that commodity $j$ is routed during that phase. In each phase we only consider the commodities that have not been routed so far, i.e., the set $V'$ of candidate roots consists of the vertices that are sources for some unrouted commodity. Thus, $q = |V'| \leq k$. Let the *residual demand* at phase $i$, $D_i$, be the total demand over commodities which have not been routed in phases 1 to $i-1$. We set the rate of expansion for phase $i$ to $\epsilon_i = \rho D_i/8$. The purpose is to route a significant fraction of the demand while keeping the cut small. (Contrast this with Lemma 5.3 where $\epsilon$ was chosen so that no commodity had its endpoints in the same region.)

We claim that at least one of the regions $\mathcal{R}_j$ in one of the phases has ratio of capacity to demand at most $\rho f$. Suppose that this is not the case, i.e., $C_{\nabla(\mathcal{R}_j)}/D_{\nabla(\mathcal{R}_j)} > \rho f$

for all regions. We will derive a contradiction to the constraint $\sum_{l=1}^{k} dist_d(s_l, t_l)dem(l) \geq 1$.

Consider phase $i$. The residual demand after phase $i$ is

$$D_{i+1} \leq \sum_{j=1}^{p} D_{\nabla(\mathcal{R}_j)} \leq \frac{1}{\rho f} \sum_{j=1}^{p} C_{\nabla(\mathcal{R}_j)}.$$

From Corollary 4.4 we get

$$D_{i+1} \leq \frac{4\epsilon_i B}{\rho f} = \frac{4\epsilon_i f}{\rho f} = \frac{D_i}{2}.$$

Since $q \leq k$ it follows from Lemma 4.1 and Corollary 4.2 that any two vertices in the same region are less than $2\ln(k+1)/\epsilon_i$ apart. Thus for any commodity $l$ that is routed in phase $i$ we have

$$dist_d(s_l, t_l) < \frac{2\ln(k+1)}{\epsilon_i} = \frac{16\ln(k+1)}{\rho D_i}.$$

Therefore, the sum of the quantities $dist_d(s_l, t_l)dem(l)$ over the commodities routed in phase $i$ is less than $16\ln(k+1)/\rho$. Since the residual demand is halved in each phase, all the commodities are routed after at most $\log D$ phases. Therefore, the sum over all commodities is

$$\sum_{l=1}^{k} dist_d(s_l, t_l)dem(l) < \frac{16\ln(k+1)\log D}{\rho} = 1,$$

a contradiction.

**7. Product multicommodity flow.** In this section, we will use our region-growing lemmas to establish an improved bound for the product multicommodity flow problem (defined in the introduction). If all the vertex weights are unity, this also gives a cleaner proof for the uniform multicommodity flow problem by dispensing with discretization (the rest of the ideas remain essentially the same). Another case of special interest is when a subset of the vertices (say $k$ in number) have unit weights and the rest have weights zero. We call this the *$k$-terminal uniform multicommodity flow problem.* It can be viewed as the analogue of the multiway cut problem in the demands case. For this case, the gap between the maximum throughput and the minimum ratio of a cut is $O(\log k)$.

As stated in the previous section, the example of Leighton and Rao can be easily adapted to show that the $O(\log k)$ gap is essentially tight for the $k$-terminal uniform multicommodity flow problem (up to a constant factor). It is interesting to note that, in both the minimum multicut problem and the sparsest cut problem, there is a $\log k$ discrepancy between the uniform case (where we have all pairs among a given set of terminals) and the general case (arbitrary set of pairs): in the multicut problem the integrality gap is $2 - \frac{1}{k}$ in the uniform case versus $O(\log k)$ in the nonuniform case. In the sparsest cut problem it is $O(\log k)$ versus $O(\log^2 k)$. The first three of these bounds are essentially tight. Only the last gap is not known to be tight; we do not have an example for the sparsest cut problem that takes advantage of nonuniformity.

In the product multicommodity flow problem each vertex $v$ has a (nonnegative) weight $w(v)$, and there is a commodity for each unordered pair $u$, $v$ of vertices with demand $w(u)w(v)$. Let

$$\alpha = \min_{S\subseteq V} \frac{C_{\nabla(S)}}{w(S)w(\overline{S})}.$$

The throughput $f$ is clearly bounded from above by $\alpha$.

The dual program again calls for an assignment of nonnegative distance labels, $d_e$, to the edges $e \in E$, so that $\sum_{e\in E} d_e c_e$ is minimized subject to the constraint that $\sum_{u,v\in V} w(u)w(v)\,dist_d(u,v) \geq 1$, where the sum extends over all unordered pairs of vertices. At optimality we have

$$\sum_{e\in E} d_e c_e = f.$$

THEOREM 7.1. *For the product multicommodity flow problem, the maximum throughput $f$ and the minimum ratio $\alpha$ of a cut satisfy the inequalities*

$$\frac{\alpha}{O(\log k)} \leq f \leq \alpha,$$

*where $k$ is the number of vertices having nonzero weight. We can find in polynomial time a cut whose ratio is within a factor $O(\log k)$ of the minimum ratio.*

A closely related quantity to the ratio of a cut $\nabla(S)$ is its *flux*:

$$\frac{C_{\nabla(S)}}{\min(w(S), w(\overline{S}))}.$$

Let

$$\beta = \min_{S\subseteq V} \frac{C_{\nabla(S)}}{\min(w(S), w(\overline{S}))}.$$

Recall that $W$ denotes the sum of the weights of all the vertices. Since $\frac{W}{2} \leq \max(w(S),\, w(\overline{S})) \leq W$, it follows that $\frac{\alpha W}{2} \leq \beta \leq \alpha W$. Therefore, Theorem 7.1 implies that the minimum flux $\beta$ lies between $\frac{fW}{2}$ and $fW \cdot O(\log k)$. Also, a cut approximates the minimum ratio $\alpha$ within a factor $O(\log k)$ if and only if it approximates the minimum flux $\beta$ within a factor $O(\log k)$.

**7.1. Proof of Theorem 7.1.** We follow the structure of the Leighton–Rao proof for the uniform multicommodity flow problem, except that we shall use the lemmas of §4. Also, it is not necessary to readjust adaptively the expansion rate $\epsilon$ of the regions and restart the procedure. First solve the dual LP to obtain a set of distance labels $d_e$. By LP duality, we have that $f$ is equal to $B = \sum_{e\in E} d_e c_e$. We shall find a cut whose flux is at most $\hat{\beta} = fW(4\ln(k+1)+1)$; thus its ratio of capacity to demand is within a factor $2(4\ln(k+1)+1)$ of the throughput $f$ and hence also of the optimal ratio $\alpha$.

We find a good cut in two stages. In the first stage we grow regions as in §4 with $V'$ as the set of vertices with nonzero weight; $|V'| = k$. We choose $\epsilon = \hat{\beta}W/4f$. If one of the regions has flux at most $\hat{\beta}$ we are finished. Suppose this is not the case. We show then that one of the regions contains vertices of weight at least $\frac{W}{2}$. If every region $\mathcal{R}_i$ contains vertices of weight less than $\frac{W}{2}$, then its flux is

$$\frac{C_{\nabla(\mathcal{R}_i)}}{\min(w(\mathcal{R}_i), w(\overline{\mathcal{R}_i}))} = \frac{C_{\nabla(\mathcal{R}_i)}}{w(\mathcal{R}_i)} > \hat{\beta}.$$

Therefore,

$$\sum_{i=1}^{p} C_{\nabla(\mathcal{R}_i)} > \hat{\beta} \sum_{i=1}^{p} w(\mathcal{R}_i) = \hat{\beta} W.$$

However, by Corollary 4.4 we have

$$\sum_{i=1}^{p} C_{\nabla(\mathcal{R}_i)} \leq 4\epsilon B = 4\epsilon f = \hat{\beta} W.$$

We conclude that one of the regions, say $\mathcal{R}*$, has weight at least $\frac{W}{2}$. Let $r$ be the root of region $\mathcal{R}*$.

In the second stage we reset all the variables $y_S$ to zero and grow a region from root $r$ as in §4. However, now we do not check for condition 1 and stop only when all vertices are included in the region. Let $\{r\} = S_1 \subset S_2 \subset \cdots \subset S_t$ denote the sets with $y_S > 0$. We will argue that one of these sets has flux at most $\hat{\beta}$. Assume that this is not the case; we shall derive a contradiction to the constraint $\sum_{u,v \in V} w(u)w(v)\,dist_d(u,v) \geq 1$.

First observe that $dist_d(u,v) \leq dist_d(r,u) + dist_d(r,v)$. Therefore,

$$\sum_{u,v \in V} w(u)w(v)\,dist_d(u,v) \leq W \sum_{v \in V} w(v)\,dist_d(r,v).$$

If $S_i$ is the smallest set containing vertex $v$ then $dist_d(r,v) = \sum_{S \subset S_i} y_S$. Let $S_l = \mathcal{R}$ be the first set in the chain that contains all the vertices of $\mathcal{R}*$. The set $\mathcal{R}$ may be a proper superset of $\mathcal{R}*$, because some nodes may have been deleted from the graph in the previous stage by the time we grew the region around $r$. In any case, however, $rad(\mathcal{R}) \leq rad(\mathcal{R}*)$. Since $|V'| = k$, by Lemma 4.1

$$rad(\mathcal{R}) < \frac{\ln(k+1)}{\epsilon} = \frac{4f\ln(k+1)}{W\hat{\beta}}.$$

The total weighted distance of all the vertices from $r$ is

$$\sum_{v \in V} w(v)\,dist_d(r,v) = \sum_{S} y_S(W - w(S)) = \sum_{S \subset \mathcal{R}} y_S(W - w(S)) + \sum_{S \supseteq \mathcal{R}} y_S(W - w(S)).$$

Now,

$$\sum_{S \subset \mathcal{R}} y_S(W - w(S)) \leq W \sum_{S \subset \mathcal{R}} y_S \leq W \cdot rad(\mathcal{R}) < \frac{4f\ln(k+1)}{\hat{\beta}}.$$

Since $w(\mathcal{R}) \geq \frac{W}{2}$, all supersets of $\mathcal{R}$ have weight at least $\frac{W}{2}$. Hence, the flux for these sets if $C_{\nabla(S)}/(W - w(S)) \geq \hat{\beta}$ and thus

$$\sum_{S \supseteq \mathcal{R}} y_S(W - w(S)) \leq \frac{1}{\hat{\beta}} \sum_{S \supseteq \mathcal{R}} y_S C_{\nabla(S)} \leq \frac{1}{\hat{\beta}} \sum_{e \in E} d_e c_e = \frac{f}{\hat{\beta}}.$$

Therefore, $\sum_{v \in V} w(v)\,dist_d(r,v) < \frac{f}{\hat{\beta}}(4\ln(k+1) + 1) < \frac{1}{W}$, and hence

$$\sum_{u,v \in V} w(u)w(v)\,dist_d(u,v) < 1,$$

a contradiction.

**8. Applications.** Several graph problems can be viewed as edge deletion problems. We wish to find a minimum weight set of edges whose removal yields a graph with a desired structure $\pi$ [Ya]. Klein et al. [KARR] propose a method for approximating such a problem when the property $\pi$ can be specified as a 2CNF $\equiv$ formula so that deleting edges from the graph corresponds to deleting clauses in the formula. In particular, they show how to model the minimum edge deletion graph bipartization problem, i.e., deleting a minimum weight set of edges so that the resulting graph is bipartite. The 2CNF $\equiv$ deletion problem is defined as follows.

> A 2 CNF $\equiv$ formula, $F$, is a weighted set of clauses of the form $P \equiv Q$ where $P$, $Q$ are literals. Find a minimum weight set of clauses the deletion of which makes the formula satisfiable.

Klein et al. showed that this problem can be reduced to the minimum multicut problem. Construct a graph $G(F)$ whose vertex set is the set of literals in $F$. For each clause of the kind $P \equiv Q$ include two edges $(P, Q)$ and $(\overline{P}, \overline{Q})$ of capacity equal to the weight of the clause $P \equiv Q$.

LEMMA 8.1. *A 2CNF $\equiv$ formula, $F$, is satisfiable if and only if no connected component of the graph $G(F)$ contains both a literal and its complement.*

*Proof.* An edge $(P, Q)$ in $G(F)$ implies that the literals $P$ and $Q$ take the same truth value. Thus, if a literal and its complement occur in the same connected component then the 2CNF $\equiv$ formula is not satisfiable.

Conversely, note that if two literals $P$, $Q$ are in the same connected component then their complementary literals $\overline{P}$, $\overline{Q}$ are also in the same connected component. Thus, the components can be paired, so that in each pair one component contains a set of literals and the other contains the complementary literals. We can now obtain a satisfying assignment by setting, for each pair of components, the literals of one component to true (and the other's to false). $\square$

Let $M$ be a minimum weight set of edges whose removal separates the pairs of complementary literals in $G(F)$ and let $W$ be the minimum weight set of clauses whose deletion makes $F$ satisfiable. Then we have the following lemma.

LEMMA 8.2. $wt(W) \leq wt(M) \leq 2wt(W)$.

*Proof.* The minimum multicut, $M$, in $G(F)$ corresponds to a set of clauses (of weight at most $wt(M)$) whose deletion makes the formula satisfiable. Hence, $wt(W) \leq wt(M)$.

Each clause of $F$ corresponds to two edges in $G(F)$. Thus the set $W$ corresponds to a multicut in $G(F)$ of weight at most $2wt(W)$. Therefore, $wt(M) \leq 2wt(W)$. $\square$

Finding the set of edges, $M$, is exactly the MULTICUT problem on the graph $G(F)$ with every pair of complementary literals forming a source–sink pair. Thus, the number of pairs, $k$, is equal to the number of variables in the formula, $n$, and hence by Theorem 5.1 we can approximate $M$ to within a factor $O(\log n)$. Using Lemma 8.2 we get the following theorem.

THEOREM 8.3. *Given a 2CNF $\equiv$ formula, one can in polynomial time find a set of clauses of weight at most a factor $O(\log n)$ of the minimum weight set of clauses whose deletion makes the formula satisfiable.*

COROLLARY 8.4. *The edge-deletion graph bipartization problem can be approximated within a factor $O(\log n)$ in polynomial time.*

We leave open the question of whether these problems can be approximated within some constant factor. We know they are both MAX SNP-hard [PY] and hence do not have a polynomial-time approximation scheme unless $P = NP$ [ALMSS].

**Acknowledgment.** We wish to thank Phil Klein for simplifying the formulation in §6, which made the presentation more uniform.



## REFERENCES

[ALMSS] S. Arora, C. Lund, R. Motwani, M. Sundan, and M. Szegedy, *Proof verification and hardness of approximation problems*, in Proc. 32nd IEEE Symposium on Foundations of Computer Science, IEEE Press, Piscataway, NJ, 1992, pp. 14–23.

[Aw] B. Awerbuch, *Complexity of network synchronization*, J. Assoc. Comput. Mach., 32 (1985), pp. 804–823.

[Cu] W. H. Cunningham, *The optimal multiterminal cut problem*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 5 (1991), pp. 105–120.

[DJPSY] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis, *The complexity of multiway cuts*, in Proc. 24th ACM Symposium on Theory of Computing, Association for Computing Machinery, New York, 1992, pp. 241–251.

[EFS] P. Elias, A. Feinstein, and C. F. Shannon, *A note on the maximum flow through a network*, IRE Trans. Inform. Theory IT, 2 (1956), pp. 117–119.

[FF] L. R. Fort, Jr. and D. R. Fulkerson, *Maximal flow through a network*, Canad. J. Math., 8 (1956).

[GV] N. Garg and V. V. Vazirani, *A characterization of the vertices and edges of the $s-t$ cut polyhedron, with algorithmic applications*, in Proc. Integer Programming and Combinatorial Optimization, 1993.

[GVY] N. Garg, V. V. Vazirani, and M. Yannakakis, *Primal-dual approximation algorithms for integral flow and multicut in trees, with applications to matching and set cover*, in Proc. 20th International Collection on Automata, Languages, and Programming, Springer-Verlag, Berlin, New York, Heidelberg, 1992, pp. 64–75.

[Hu] T. C. Hu, *Multicommodity network flows*, Oper. Res., 11 (1963), pp. 344–360.

[KARR] P. Klein, A. Agrawal, R. Ravi, and S. Rao, *Approximation through multicommodity flow*, in Proc. 31st Symposium on Foundations of Computer Science, IEEE Press, Piscataway, NJ, 1990, pp. 726–737.

[KPR] P. Klein, S. Plotkin, and S. Rao, *Excluded minors, network decomposition, and multi-commodity flow*, in Proc. 25th ACM Symposium on Theory of Computing, Association for Computing Machinery, New York, 1992, pp. 682–690.

[KST] P. Klein, C. Stein, and E. Tardos, *Leighton–Rao might be practical: faster approximation algorithms for concurrent flow with uniform capacities*, in Proc. 22nd ACM Symposium on Theory of Computing, Association for Computing Machinery, New York, 1990, pp. 310–321.

[LR] F. T. Leighton and S. Rao, *An approximate max-flow min-cut theorem for uniform multicommodity flow problems with application to approximation algorithms*, in Proc. 29th IEEE Symposium on Foundations of Computer Science, IEEE Press, Piscataway, NJ, 1988, pp. 422–431.

[PY] C. H. Papadimitriou and M. Yannakakis, *Optimization, approximation and complexity classes*, J. Comput. System Sci., 43 (1991), pp. 425–440.

[PT] S. Plotkin and E. Tardos, *Improved bounds on the max-flow min-cut ratio for multicommodity flows*, in Proc. 25th ACM Symposium on Theory of Computing, Association for Computing Machinery, New York, 1992, pp. 691–697.

[TV] E. Tardos and V. V. Vazirani, *Improved bounds for the max-flow min-multicut ratio for planar and $K_{r,r}$-free graphs*, Inform. Process. Lett., 47 (1993), pp. 77–80.

[Trag] S. Tragoudas, *Improved approximation for the minimum-cut ratio and the flux*, Math. Systems Theory, to appear.

[Ya] M. Yannakakis, *Edge-deletion problems*, SIAM J. Comput., 10 (1981), pp. 297–309.

[YKCP] M. Yannakakis, P. C. Kanellakis, S. C. Cosmadakis, and C. H. Papadimitriou, *Cutting and partitioning a graph after a fixed pattern*, in Proc. 10th International Collection on Automata, Languages, and Programming, Springer-Verlag, Berlin, New York, Heidelberg, 1983, pp. 712–722.