Approximating Sparsest Cuts

Naveen Garg[*]

Abstract

In this paper we prove that the max-flow min-cut ratio for multicommodity flow is no more than $2\mathcal{H}(n)$ and show a randomized polynomial time algorithm for finding a cut of sparsity no more than $4\mathcal{H}(n)$ times the optimum; where $\mathcal{H}(n) = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n}$ is the n^{th} harmonic sum. This represents a significant improvement over the $40 \log n$ approximation guarantee in [5]. More importantly over approach, which uses the notion of cut packings, seems to be fairly powerful and general and holds promise for approximating other **NP**-hard problems involving cuts.

1 Introduction

Given a multicommodity flow problem, with a demand associated with each commodity, which is the amount of the commodity that we wish to ship, one often needs to know if there is a *feasible flow*, i.e. a flow that satisfies the demands and obeys the capacity constraints. For a feasible flow to exist it is necessary that the capacity of any cut exceed the sum of the demands of the commodities whose source and sink are separated by the cut. The max-flow min-cut theorem for single commodity flow implies that this *cut condition* is also sufficient. In contrast, a multicommodity flow problem can be infeasible even if the cut condition is satisfied.

The optimization version of the multicommodity flow problem, called the *concurrent flow* problem, first formulated by Shahrokhi and Matula [6], is to maximize the *throughput*, which is defined as the value, f, such that there exists a multicommodity flow shipping a fraction f of the demand of each commodity.

An upper bound on the throughput can be obtained by considering cuts in the network. For any cut, the throughput times the sum of the demands of the commodities whose source and sink lie on different sides of the cut, cannot exceed the capacity of the cut. Thus the throughput is at most the minimum, taken over all cuts, of the ratio of the capacity of the cut to the demand across the cut; the cut achieving this minimum ratio, α, is called the *sparsest cut*.

The maximum throughput is also referred to as the *maximum concurrent flow* or the *maximum flow* and the sparsest cut is commonly called the *minimum cut*. The max-flow min-cut theorem for single commodity flow states that the maximum flow and minimum cut as defined above are equal. However, equality of maximum flow and minimum cut does not hold for multicommodity flow instances in general.

[*]Max-Planck-Institut für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany. Work supported by the EU ESPRIT LTR Project N. 20244 (ALCOM-IT).

In a recent result Linial, London and Rabinovich [5] use ideas from the low-distortion embeddings of graphs in normed spaces [1] to show that

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

Alternatively, this means that for the case of uniform demands, it is sufficient that the capacity of every cut exceed the demand across the cut by an $O(\log k)$ factor. Further, this approximate max-flow min-cut theorem is tight; Leighton and Rao show an example where the ratio between the minimum cut and the maximum flow is $O(\log k)$.

The proof of this theorem also yields a randomized polynomial time algorithm for approximating the sparsest cut within a factor of $O(\log k)$. Computation of such cuts is a basic step for a variety of approximation algorithms for **NP**-hard optimization problems.

In this paper we provide a $2\mathcal{H}(n)$ bound on the max-flow min-cut ratio and present a randomized polynomial time algorithm for approximating the sparsest cut with an approximation guarantee of $4\mathcal{H}(n)$ Besides improving significantly on the constants involved, our work provides a potentially powerful approach to approximating other **NP**-hard problems involving cuts in graphs. We elaborate on this below.

Given a fractional solution (obtained by solving either a linear-programming or semi-definite programming relaxation of the integer program) to an **NP**-hard problem involving cuts in graphs, can it be expressed as a convex combination of the integer solutions? The integer solutions in our case are cuts. Hence we would like to assign values to cuts in such a manner that the sum of the values assigned to cuts that include a certain edge e, is 'close' to the value of this edge in the fractional solution; we refer to the value of edge e in the fractional solution as the *length* of e. In particular, we require that the sum of the values assigned to cuts including e be no more than its length and it is for this reason that we refer to this assignment of values to cuts as a *cut packing*. Furthermore, if the cut packing is such that the sum of the values of the cuts including an edge is at least β times the length of the edge then one of the cuts has sparsity no more than $1/\beta$ times the value of the fractional solution.

Another instance where such an approach has been used for approximating an **NP**-hard problem is that of the max-cut. In [2] the fractional solution obtained by solving the semi-definite program is packed with cuts that are defined by hyperplanes. Since the problem involved is a maximization problem we only require that each edge is packed to a sufficiently large extent. In fact by considering only those cuts that are defined by hyperplanes and assigning them suitable values Goemans and Williamson argue that each edge is packed to an extent of at least 0.878 thereby obtaining an algorithm with this approximation guarantee.

In the case of max cut, geometry plays a crucial role in that it determines what cuts to include in the packing and at what value. Our result demonstrates that geometry is of no significance for the sparsest cut problem as [5] seems to suggest. Thus although the approach we use for packing cuts is as suggested implicitly in the paper by Bourgain [1] and the problem addressed there was of embedding metric spaces into finite-dimensional normed spaces, the cut packing approach can be viewed in purely graph-theoretic and combinatorial terms. The geometric view advanced in [5] fails for the case of multicommodity flows in directed graphs since we cannot hope to embed a directed graph (with vastly different lengths on two anti-parallel edges) with any bounded distortion. We believe that this break from geometry shall lead to improve bounds on the max-flow min-cut ratio for symmetric multicommodity flow in directed graphs [4].

The paper is organized as follows. Section 2 introduces the notations used. In Section 3 we show how a cut-packing implies a bound on the max-flow min-cut ratio. We also give a procedure for obtaining a cut-packing. Section 4 proves the main theorem of the paper while Section 5 gives a randomized polynomial time algorithm which approximates the sparsest cut within a factor $4\mathcal{H}(n)$.

2 Notation and Preliminaries

An instance of the multicommodity flow problem consists of an undirected graph $G = (V, E)$, a capacity function, $\mathbf{c} : E \to \mathbf{R}^+$, on the edges and k commodities, numbered 1 through k, where for commodity i, besides the source and sink for that commodity we are also specified a non-negative demand. The *demand graph* H corresponding to a multicommodity flow instance is the graph obtained by putting an edge for each source-sink pair.

Any multicommodity flow instance is equivalent to another instance where the graph G is the complete graph and every pair of vertices is the source-sink pair of a commodity; we assign zero capacities to the edges and zero demands for the commodities that are not part of the original instance. For this paper we assume that the multicommodity flow instance is specified as a complete graph $G = (V, E)$ with non-negative capacities $\mathbf{c} : E \to \mathbf{R}^+$ and demands $\mathbf{d} : E \to \mathbf{R}^+$ where $\mathbf{d}(e)$ represents the demand of the commodity whose source-sink pair are the end-points of the edge e.

For a subset of edges, $E' \subseteq E$, let $\mathbf{c}(E')$ denote the sum of the capacities of the edges in E'. Similarly, $\mathbf{d}(E') = \sum_{e \in E'} \mathbf{d}(e)$. We define the cut associated with a set S, denoted by $\bigtriangledown(S)$, as the set of edges with exactly one end point in S. Thus $\mathbf{c}(\bigtriangledown(S)), \mathbf{d}(\bigtriangledown(S))$ denote the capacity and the demand across the cut $\bigtriangledown(S)$. Using this notation the Cut condition can be formulated as

Cut condition: $\forall S \subseteq V$, $\mathbf{c}(\bigtriangledown(S)) \geq \mathbf{d}(\bigtriangledown(S))$

The sparsest cut is the cut that minimizes the ratio $\mathbf{c}(\bigtriangledown(S))/\mathbf{d}(\bigtriangledown(S))$ and the sparsest cut ratio, α, is given by

$$\alpha = \min_{S \subseteq V} \frac{\mathbf{c}(\bigtriangledown(S))}{\mathbf{d}(\bigtriangledown(S))}$$

Let $l : E \to [0, 1]$ be an assignment of lengths to the edges that satisfies the triangle inequality *ie.*, for all vertices u, v, w, $l(u, v) + l(v, w) \geq l(u, w)$, and which minimizes the ratio

$$\frac{\sum_{e \in E} l(e)\mathbf{c}(e)}{\sum_{e \in E} l(e)\mathbf{d}(e)}$$

Note that a 0/1 assignment of lengths which satisfies triangle inequality and minimizes the above ratio yields a sparsest cut; the edges of length 1 are the edges in the cut. Thus l can be viewed as a fractional solution to the sparsest cut problem . In fact it follows from the duality theory of linear programming that the above ratio is equal to the maximum throughput of the multicommodity flow instance, *ie.*

$$f = \frac{\sum_{e \in E} l(e)\mathbf{c}(e)}{\sum_{e \in E} l(e)\mathbf{d}(e)}$$

For a non-empty subset $S \subset V$, let $l(u,S)$ denote the minimum length of an edge whose one endpoint is u and the other is in S *ie.*, $l(u,S) = \min_{w \in S} l(u,w)$. If $u \in S$ then $l(u,S) = 0$.

3 Cut Packings

Let $y : 2^V \to \mathbf{R}^+$ be an assignment of non-negative values to subsets of vertices. This assignment of values to subsets can also be viewed as an assignment of values to cuts with $y(S) + y(\overline{S})$ being the value assigned to the cut $\bigtriangledown(S)$. Such an assignment is a *cut packing* if for all edges $e \in E$, the sum of the values assigned to subsets which contain exactly one end point of e is at most $l(e)$ *ie.*, $\sum_{S: e \in \bigtriangledown(S)} y(S) \leq l(e)$. Furthermore if it is the case that for all edges e, $\sum_{S: e \in \bigtriangledown(S)} y(S) \geq \beta \cdot l(e)$ then

$$\begin{aligned} f &= \frac{\sum_{e \in E} l(e)\mathbf{c}(e)}{\sum_{e \in E} l(e)\mathbf{d}(e)} \\ &\geq \beta \frac{\sum_{e \in E} \mathbf{c}(e) \left(\sum_{S: e \in \bigtriangledown(S)} y(S)\right)}{\sum_{e \in E} \mathbf{d}(e) \left(\sum_{S: e \in \bigtriangledown(S)} y(S)\right)} \\ &= \beta \frac{\sum_{S \subseteq V} y(S)\mathbf{c}(\bigtriangledown(S))}{\sum_{S \subseteq V} y(S)\mathbf{d}(\bigtriangledown(S))} \\ &\geq \beta \min_{S \subseteq V} \frac{\mathbf{c}(\bigtriangledown(S))}{\mathbf{d}(\bigtriangledown(S))} \end{aligned}$$

and hence the sparsest cut is of value no more than f/β. Thus to prove an $1/\beta$ bound on the max-flow min-cut ratio for multicommodity flow it suffices to show a way of assigning values to subsets of vertices $y : 2^V \to \mathbf{R}^+$ such that for every edge e, $\beta \cdot l(e) \leq \sum_{S: e \in \bigtriangledown(S)} y(S) \leq l(e)$.

Bourgain's definition of an embedding [1] suggests a natural way of packing cuts. We begin by picking a set of vertices S (we call this set the *seed*) and increase $y(S)$ till for some edge $e \in \bigtriangledown(S)$, $l(e) = y(S)$. Since we cannot increase $y(S)$ any further, we include the other endpoint of e, say v, into the set S and start raising $y(S \cup \{v\})$. The set whose y value we raise is referred to as the *active set*. Let A be the active set at some point (A is initially the same as S). We raise $y(A)$ till some edge $e = (u,v)$, $u \in A$, becomes tight *ie.*, $\sum_{S: e \in \bigtriangledown(S)} y(S) = l(e)$. We then update the active set to $A \cup \{v\}$ and continue in this manner till all vertices are included in the active set. Note that if $l(u,S) < l(v,S)$ then vertex u is included before vertex v. We scale the values assigned to the sets in this process by $1/(|S|\binom{n}{|S|}\mathcal{H}(n))$ and repeat this procedure for all 2^n initial choices of the seed set. Finally, the value assigned to a set is the sum of the scaled values assigned to the set in the 2^n runs. This then yields the cut packing.

Note that for every edge $e = (u,v)$

$$\sum_{S: e \in \bigtriangledown(S)} y(S) = \frac{1}{\mathcal{H}(n)} \sum_{S \subseteq V} \frac{|l(u,S) - l(v,S)|}{|S|\binom{n}{|S|}}$$

and we denote this quantity by $\hat{l}(e)$.

Theorem 3.1

$$\forall e \in E: \quad \frac{l(e)}{2\mathcal{H}(n)} \leq \hat{l}(e) \leq l(e)$$

This, by our argument from before, implies that the sparsest cut in the graph is of sparsity no more than $2\mathcal{H}(n) \cdot f$. Hence the max-flow min-cut ratio for multicommodity flow is bounded by $2\mathcal{H}(n)$.

A weaker version of Theorem 3.1 first appeared in [1] where it was shown that $l(e)/O(\log n) \leq \hat{l}(e) \leq l(e)$ with the constant in the Big-Oh being roughly 40. Our proof of this theorem yields a better constant *ie.* two and this translates to a significantly better approximation guarantee for the sparsest cut problem.

4 Proof of Theorem 3.1

For this section we fix an edge $e = (u, v)$ and provide upper and lower bounds on $\hat{l}(e)$ in terms of $l(e)$. To simplify presentation we assume that $l(e) = 1$; the same arguments apply for any particular value of $l(e)$.

Lemma 4.1 (Bourgain) $\hat{l}(e) \leq 1$

One important technical contribution of this paper is to prove that $\hat{l}(e)$ is minimum when every vertex (besides u, v) has an edge of length $1/2$ to both u and v and for this choice of lengths $\hat{l}(e) = (1/2 + 1/n)/\mathcal{H}(n)$. Formally,

Lemma 4.2 *$\hat{l}(e)$ is minimum when*

$$\forall w \in V - \{u, v\} : l(u, w) = l(v, w) = 1/2$$

The minimum value of $\hat{l}(e)$ is $(1/2 + 1/n)/\mathcal{H}(n)$.

From amongst all possible length functions which satisfy the property that for all vertices w different from u and v, $l(u, w) + l(v, w) \geq 1$ let $l^* : E \rightarrow \mathbf{R}^+$ be the one for which $\hat{l}(e)$ is minimum. We call the quantity $|l^*(u, S) - l^*(v, S)|$ the *contribution of set S* (to $\hat{l}(e)$).

Claim 4.1 $\forall w \in V : l^*(u, w) + l^*(v, w) = 1$.

Proof: Since l^* is a metric, $\forall w \in V : l^*(u, w) + l^*(v, w) \geq 1$. Let w be such that $l^*(u, w) + l^*(v, w) > 1$. Decrease the larger of $l^*(u, w), l^*(v, w)$ (both, if they are equal) by an amount ϵ. We now argue that there is no set whose contribution increases as a result of this change. Let S be a set whose contribution changes and let $l^*(u, S) \geq l^*(v, S)$. Clearly $w \in S$. The contribution of S increases only when $l^*(u, S)$ remains unchanged and $l^*(v, S)$ decreases. $l^*(v, S)$ decreases when $l^*(v, S) = l^*(v, w) \geq l^*(u, w)$. $l^*(u, S)$ remains unchanged when $l^*(u, w) < l^*(v, w)$ or $l^*(u, S) < l^*(u, w)$. However both these cases imply that $l^*(u, S) < l^*(v, S)$ contradicting our assumption. Furthermore, for a suitable ϵ, the contribution of at least one set, either $\{u, w\}$ or $\{v, w\}$, decreases. Hence if l^* minimizes $\hat{l}(e)$ it must be the case that $\forall w \in V : l^*(u, w) + l^*(v, w) = 1$. ■

Claim 4.2 *There exists a l^* such that $\forall w \in V : l^*(u,w), l^*(v,w) \in \{0, 1/2, 1\}$.*

Proof: Given a length function l^* we show how to obtain from it another length function which also minimizes $\hat{l}(e)$ and under which the length of any edge incident at u or v is either $0, 1/2$ or 1.

Modify l^* by decreasing (resp. increasing) all lengths in the range $(1/2, 1)$ (resp. $(0, 1/2)$) by an amount ϵ. Let S be a set whose contribution changes and let $l^*(u,S) \geq l^*(v,S)$. Then S satisfies one of the following

- $l^*(u,S) > 1/2$. This implies that $l^*(v,S) < 1/2$ (consequence of Claim 4.1) and hence the contribution of S decreases by 2ϵ.
- $l^*(u,S) = 1/2$ and $l^*(v,S) < 1/2$. Contribution of S decreases by ϵ.
- $0 < l^*(u,S) < 1/2$ and $l^*(v,S) = 0$. Contribution of S increases by ϵ.

Let δ be the change in $\hat{l}(e)$ as a result of this modification in lengths ($\delta > 0$ implies an increase and $\delta < 0$ a decrease in $\hat{l}(e)$). Now consider the complementary modification which increases (resp. decreases) all lengths in the range $(1/2, 1)$ (resp. $(0, 1/2)$) by the same amount ϵ. It is easy to see that the only sets whose contribution changes are exactly those as before. Sets whose contribution increased (resp. decreased) earlier have their contributions decreasing (resp. increasing) now by the same amount. Since $\hat{l}(e)$ is a weighted sum of the contributions of the sets and δ was the change in $\hat{l}(e)$ earlier, the change in $\hat{l}(e)$ now is $-\delta$. If $\delta \neq 0$ then one of these two ways of modifying l^* reduces $\hat{l}(e)$ contradicting the fact that l^* minimizes $\hat{l}(e)$. On the other hand, if $\delta = 0$, we could use any one of these two techniques for modifying lengths and finally obtain a length function which takes values from $\{0, 1/2, 1\}$ for the length between any vertex and u, v. ■

Henceforth we assume that l^* is as claimed above. Let S_u be the set of vertices which have an edge of 0 to u (and hence an edge of length 1 to v), i.e. $S_u = \{w \in V | l^*(u,w) = 0\}$. Similarly, $S_v = \{w \in V | l^*(v,w) = 0\}$. Further, let $|S_u| = a$ and $|S_v| = b$. The only sets that have a non-zero contribution are those that intersect exactly one of S_u, S_v. Of these, the sets that are contained in either S_u or S_v contribute 1 while the rest contribute 1/2 to $\hat{l}(e)$. There are $\binom{n-b}{r} - \binom{n-a-b}{r}$ r-element sets that intersect S_u but not S_v and of these $\binom{a}{r}$ sets are contained in S_u. Hence,

$$\hat{l}(e) = \frac{1}{\mathcal{H}(n)} \left(\frac{1}{2} \sum_{r=1}^{n-b} \frac{\binom{n-b}{r} - \binom{n-a-b}{r}}{r\binom{n}{r}} + \frac{1}{2} \sum_{r=1}^{a} \frac{\binom{a}{r}}{r\binom{n}{r}} + \frac{1}{2} \sum_{r=1}^{n-a} \frac{\binom{n-a}{r} - \binom{n-a-b}{r}}{r\binom{n}{r}} + \frac{1}{2} \sum_{r=1}^{b} \frac{\binom{b}{r}}{r\binom{n}{r}} \right)$$

Claim 4.3

$$\sum_{r=1}^{n-p} \frac{\binom{n-p}{r} - \binom{n-q}{r}}{r\binom{n}{r}} = \mathcal{H}(q) - \mathcal{H}(p)$$

where $0 \leq p \leq q \leq n$.

Proof: Telescoping the difference in the numerator we obtain

$$\binom{n-p}{r} - \binom{n-q}{r} = \sum_{i=p+1}^{q} \binom{n-i+1}{r} - \binom{n-i}{r} = \sum_{i=p+1}^{q} \binom{n-i}{r-1}$$

Therefore,

$$\begin{aligned}
\sum_{r=1}^{n-p} \frac{\binom{n-p}{r} - \binom{n-q}{r}}{r\binom{n}{r}} &= \sum_{r=1}^{n-p} \sum_{i=p+1}^{q} \frac{\binom{n-i}{r-1}}{r\binom{n}{r}} \\
&= \sum_{i=p+1}^{q} \sum_{r=1}^{n-p} \frac{\binom{n-i}{r-1}}{r\binom{n}{r}} \\
&= \sum_{i=p+1}^{q} \left(\frac{1}{n} \sum_{r=1}^{n-p} \frac{\binom{n-i}{r-1}}{\binom{n-1}{r-1}} \right)
\end{aligned}$$

The following identity is proved in [3, pages 173,174].

$$\sum_{k=0}^{l} \frac{\binom{l}{k}}{\binom{m}{k}} = \frac{m+1}{m+1-l}, \quad 0 \le l \le m$$

Hence

$$\sum_{r=1}^{n-p} \frac{\binom{n-i}{r-1}}{\binom{n-1}{r-1}} = \frac{(n-1)+1}{(n-1)+1-(n-i)} = \frac{n}{i}$$

since $i \ge p+1$ and $p \ge 0$. Substituting the above into equation 1 yields

$$\sum_{r=1}^{n-p} \frac{\binom{n-p}{r} - \binom{n-q}{r}}{r\binom{n}{r}} = \sum_{i=p+1}^{q} \frac{1}{i} = \mathcal{H}(q) - \mathcal{H}(p)$$

■

Substituting the identity in Claim 4.3 into the expression for $\hat{l}(e)$ gives

$$\hat{l}(e) = \frac{1}{\mathcal{H}(n)} \left(\frac{1}{2}(\mathcal{H}(a+b) - \mathcal{H}(b)) + \frac{1}{2} \sum_{r=1}^{a} \frac{\binom{a}{r}}{r\binom{n}{r}} + \frac{1}{2}(\mathcal{H}(a+b) - \mathcal{H}(a)) + \frac{1}{2} \sum_{r=1}^{b} \frac{\binom{b}{r}}{r\binom{n}{r}} \right)$$

Since $a, b \ge 1$, the expression on the right is minimum when $a = b = 1$ at which point its value is $(1/2 + 1/n)/\mathcal{H}(n)$. This proves Lemma 4.2.

5 A $4\mathcal{H}(n)$-approximation algorithm for sparsest cut

In this section we present a randomized polynomial time algorithm that finds a cut of sparsity at most $4\mathcal{H}(n) \cdot f$ with probability at least $1 - e^{-1}$. Our arguments in the previous sections were existential *ie.* we only showed the existence of a cut of sparsity at most $2\mathcal{H}(n)$. Besides the cut packing we used required an assignment of values to exponentially many sets and hence it is not clear how this could lead to a polynomial time algorithm.

Note however that there is a simple procedure for picking a cut with probability proportional to its value in the packing. Hence by picking cuts in this manner we should be able to obtain a cut with sparsity less than the 'average' sparsity. The actual procedure we use for picking cuts is slightly different and is detailed below.

Recall that in the procedure for finding a cut packing we picked a seed set and then using a Dijkstra like procedure for shortest paths, assigned values to some cuts. We now pick, randomly, a set S with probability $1/(\mathcal{H}(n)|S|\binom{n}{|S|})$ and perform this procedure with S as the seed. From amongst all cuts encountered in this procedure (the cuts corresponding to the active sets) pick the one with the minimum sparsity.

Lemma 5.1 *The probability, p, that the cut picked in this manner has sparsity more than $4\mathcal{H}(n) \cdot f$ is at most $1 - 1/4\mathcal{H}(n)$.*

Proof: For contradiction assume that $p > 1 - 1/4\mathcal{H}(n)$. Let $\mathcal{S}$ be the collection of seed sets for which the sparsest cut found is of sparsity more than $4\mathcal{H}(n) \cdot f$. Then

$$p = \sum_{S \in \mathcal{S}} \frac{1}{\mathcal{H}(n)|S|\binom{n}{|S|}}$$

Recall that $y(S)$ is the value assigned to set S in the cut packing obtained by considering all possible seed sets. We split $y(S)$ into $y_1(S)$ and $y_2(S)$ where $y_1(S)$ is the value assigned to S when considering those seed sets which are in $\mathcal{S}$ and $y_2(S)$ is the value assigned to S when considering those seed sets that are not in the collection $\mathcal{S}$. Thus $y(S) = y_1(S) + y_2(S)$.

Claim 5.1

$$\sum_{S \subseteq V} y_2(S)\mathbf{d}(\bigtriangledown(S)) < \frac{1}{4\mathcal{H}(n)} \sum_{e \in E} l(e)\mathbf{d}(e)$$

Proof: Note that for any edge $e \in E$

$$\sum_{S: e \in \bigtriangledown(S)} y_2(S) \leq l(e)(1-p)$$

Thus

$$\sum_{S \subseteq V} y_2(S)\mathbf{d}(\bigtriangledown(S)) = \sum_{e \in E} \mathbf{d}(e) \sum_{S: e \in \bigtriangledown(S)} y_2(S)$$

which implies the claim ■

Since

$$\sum_{S \subseteq V} y(S)\mathbf{d}(\bigtriangledown(S)) \geq \frac{1}{2\mathcal{H}(n)} \sum_{e \in E} l(e)\mathbf{d}(e)$$

we have

$$\sum_{S \subseteq V} y_1(S)\mathbf{d}(\bigtriangledown(S)) > \frac{1}{4\mathcal{H}(n)} \sum_{e \in E} l(e)\mathbf{d}(e)$$

Further, any set S with $y_1(S) > 0$ has sparsity at least $4\mathcal{H}(n) \cdot f$ and therefore

$$\sum_{e \in E} l(e)\mathbf{c}(e) \quad \geq \quad \sum_{S \subseteq V} y(S)\mathbf{c}(\bigtriangledown(S))$$

$$
\begin{aligned}
&\geq \sum_{S \subseteq V} y_1(S)\mathbf{c}(\bigtriangledown(S)) \\
&\geq 4\mathcal{H}(n) \cdot f \sum_{S \subseteq V} y_1(S)\mathbf{d}(\bigtriangledown(S)) \\
&> f \sum_{e \in E} l(e)\mathbf{d}(e)
\end{aligned}
$$

which contradicts the equality

$$
f = \frac{\sum_{e \in E} l(e)\mathbf{c}(e)}{\sum_{e \in E} l(e)\mathbf{d}(e)}
$$

Hence the probability that we pick a cut of sparsity more than $4\mathcal{H}(n) \cdot f$ is at most $1 - 1/4\mathcal{H}(n)$. ■

Thus if we repeat this experiment $4\mathcal{H}(n)$ times then the probability that every trial gives us a cut of sparsity more than $4\mathcal{H}(n) \cdot f$ is at most $(1 - 1/4\mathcal{H}(n))^{4\mathcal{H}(n)}$. Thus the probability that one of the trials gives us a cut of sparsity at most $4\mathcal{H}(n) \cdot f$ is at least $1 - e^{-1}$.

Theorem 5.2 *There is a randomized polynomial time algorithm that with probability at least* $1-e^{-1}$ *finds a cut of sparsity at most* $4\mathcal{H}(n)$ *times the optimum.*

6 Sparsest Node Cuts

Acknowledgment I wish to thank Volker Priebe for his help in proving Claim 4.3 and for comments on a first draft of the paper.

References

[1] J. Bourgain. On lipschitz embedding of finite metric spaces in hilbert space. *Israel Journal of Mathematics*, 52(1):46–52, 1985.

[2] M.X. Goemans and D.P. Williamson. .878-approximation algorithms for max cut and max 2sat. In *Proceedings, ACM Symposium on Theory of Computing*, pages 422–431, 1994.

[3] R.L. Graham, D.E. Knuth, and O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison Wesley, New York, 1994.

[4] P.N. Klein, S.A. Plotkin, S. Rao, and E. Tardos. Bounds on the max-flow min-cut ratio for directed multicommodity flows. Unpublished Manuscript, 1993.

[5] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–246, 1995.

[6] F. Shahrokhi and D. Matula. The maximum concurrent flow problem. *Journal of the ACM*, 37(2):318–334, 1990.