| COM S 783 / OR&IE 634 Approximation Algorithms | Fall Semester, 2003 |
|---|---|

Lecture 10: Oct 1, 2003

*Lecturer: Yuval Rabani* *Scribe: Ara Hayrapetyan*

# 1 Sparsest Cut

The problem that we will address today and during the next lecture will be the *sparsest cut problem* - yet another question related to multicommodity flow. We are given an undirected graph $G = (V, E)$ with edge capacities $c : E \to Q^+$ and $k$ pairs of vertices (called *commodities*) $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$, where each pair $i$ would like to ship demand $d_i$ between its source $s_i$ and its target $t_i$. If we attempt to route the demands from their sources to their sinks we may have to violate some of the capacities. The *Concurrent Flow* problem wants to find the minimum $\gamma$ such that if we scale all the edge capacities by $\gamma$ it becomes feasible to route $d_i$ units of flow between $s_i$ and $t_i$. This problem can be solved precisely using Linear Programming.

Given a cut $(S, \bar{S})$ let us denote its cost by

$$c(S, \bar{S}) = \sum_{i \in S, j \in \bar{S}} c_{i,j}, \tag{1}$$

Let us also denote by $s(S, \bar{S})$ the set of indices $i$ such that the pair $(s_i, t_i)$ is separated by the cut. Finally, denote the total demand that has to cross this cut by

$$d(S, \bar{S}) = \sum_{i : i \in s(S, \bar{S})} d_i. \tag{2}$$

The *cut ratio* of the cut $(S, \bar{S})$ is the defined as $\frac{c(S,\bar{S})}{d(S,\bar{S})}$, that is the ratio between the total capacity crossing the cut and the demand that has to be shipped accross the cut. Clearly the reciprocal the cut ratio is a lower bound on the $\gamma$ from the Concurrent Flow problem.

The goal of the *sparsest cut* problem is to minimize this cut ratio. For simplicity we will consider the *uniform sparsest cut* problem, where the commodities are all pairs of vertices and all the demands equal to one [3]. In this case the cut ratio of $(S, \bar{S})$ is simply

$$\frac{c(S, \bar{S})}{|S||\bar{S}|}, \tag{3}$$

The *flux* or the *edge expansion* of the graph, is defined as

$$\frac{c(S, \bar{S})}{min\{|S|, |\bar{S}|\}}. \tag{4}$$

Thus a good approximation algorithm for sparsest cut would yield a good approximation algorithm for the flux and vice versa (this follows from observing that $max\{|S|, |\bar{S}|\}$ is between $[\frac{n}{2}, n]$).

## 2 Linear Programming Relaxation

Let us consider a cut $(S, \bar{S})$. Define a distance measure $\delta_{i,j}$ to be 0 if $i$ and $j$ are on the same side of the cut, and some real number $l$ (to be set later) if they are on different sides. Notice that now the objective function can be written in terms of the measure $\delta$. More specifically, $\sum \delta_{i,j} c_{i,j} = l\ c(S, \bar{S})$, and $\sum \delta_{i,j} = l|S||\bar{S}|$. The objective function is the ratio between these two equations, and since we want to make it linear, we set $l$ such that $\sum \delta_{i,j} = 1$.

So our linear programming relaxation will consider the set of all semimetrics satisfying the condition $\sum \delta_{i,j} = 1$ and we will optimize the objective function $\sum \delta_{i,j} c_{i,j}$ over this relaxation.

More rigorously, the following will be our linear program (LP):

$$min \sum_{i,j} \delta_{i,j} c_{i,j} \tag{5}$$

$$\sum_{i,j} \delta_{i,j} = 1, \tag{6}$$

$$\delta \ is\ a\ semimetric, \tag{7}$$

where the last condition is our relaxation. Before proceeding to the rounding let us make two observations.

- The dual of this program would try to ship one unit of flow between the demands so as to maximize the reciprocal of the concurrent flow ratio $\gamma$.
- The approximate cut that we find will be an upper bound on the flow that could be shipped, therefore yielding an approximate max-flow/min-cut theorem (see [1] for details).

## 3 Rounding

We will round this linear program in two steps. First we will embed $\delta$ into an $L_1$ metric of low distortion using *Bourgain's theorem* [2]. Then we will find a good cut using the properties of $L_1$.

The second step is easier, so let us consider it first. Let $\ell_1^d$ be the $d$-dimensional real space with $L_1$ as the distance metric. Suppose that $\delta$ is such an $L_1$ metric. Let us consider hyperplanes perpendicular to the axis of our space $R^d$. Let $(j, \xi)$-hyperplane be the hyperplane that cuts the $j$-th axis at coordinate $\xi$ and is perpendicular to the $j$-th direction. And let $(j, \xi)$-cut be the cut defined by the $(j, \xi)$- hyperplane. Let us use $f(j, \xi)$ to denote the capacity of the $(j, \xi)$-cut, and $g(j, \xi)$ to denote the product of the sizes of the sides of the $(j, \xi)$-cut. Our goal is to find a good point $(j, \xi)$, that is a $(j, \xi)$-cut with low $\frac{f(j,\xi)}{g(j,\xi)}$ ratio.

Notice that it is sufficient to prove the existence of such a good point, since if $d$ is polynomial, there are only polynomially many different $(j, \xi)$-cuts (since in a given direction each such cut changed only when a node moves from one side of the cut to the other) and we can search through all of them.

The existence proof will be done by averaging. Let us first consider the average of the $f(j, \xi)$'s: $\sum_j \int f(j, \xi) d\xi$. Notice that the $j$-th coordinate of the edge $(x, y)$ will have its contribution equal to the length of the projection of the edge $(x, y)$ onto the $j$-th axis. Therefore,

$$\sum_{j=1}^{d} \int f(j,\xi) d\xi = \sum_{j} \sum_{x,y} |x_j - y_j| c_{x,y} = \sum_{x,y} c_{x,y} ||x-y||_1 \leq \sum_{x,y} c_{x,y} \delta_{x,y} = LP, \tag{8}$$

where the last inequality follows from the metric embedding. Similarly for the $g$ function we observe that the product of the sizes of the two sides is just the number of edges crossing the cut, therefore

$$\sum_{j=1}^{d} \int g(j,\xi) d\xi = \sum_{x,y} ||x-y||_1 \geq \frac{\sum_{x,y} \delta_{x,y}}{O(\log n)} \geq \frac{1}{O(\log n)}, \tag{9}$$

where again the last inequality is implied from the metric embedding. Therefore there exists a pair $(j,\xi)$ such that

$$\frac{f(j,\xi)}{g(j,\xi)} \leq O(\log n) LP. \tag{10}$$

Therefore if we succeed in embedding the $\delta$ in the LP objective function into $L_1$ with polynomially many dimensions such that the embedding has low distortion, we would get a good approximation algorithm for the sparsest cut problem.

Next lecture we will prove Bourgain's theorem, which claims that any semimetric of $n$ nodes can be embedded into $L_1$ with distortion $O(\log n)$ and $O(\log^2 n)$ many dimensions (the latter fact was proved in [4]). This will therefore give us a $O(\log n)$ approximation algorithm for the sparsest cut.

# References


[1] Yonatan Aumann and Yuval Rabani. An $o(\log k)$ approximate min-cut max-flow theorem and approximation algorithms. *SIAM Journal of Computing*, 27:291–301.

[2] Jean Bourgain. On Lipschitz embeddings of finite metric spaces in Hilbert space. *Israel Journal of Mathematics*, 52:46–52, 1985.

[3] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46:787–832, 1999.

[4] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15:215–245, 1995.