CS 598CSC: Approximation Algorithms | **Lecture date: April 1, 2009**
Instructor: Chandra Chekuri | **Scribe: Tianyi Wu**

1 The Sparsest Cut Problem

We define the SPARSESTCUT problem as follows. Given an undirected graph $G = (V, E)$ and a capacity function $c : E \rightarrow \mathbb{R}^+$ that assigns a capacity c_e to each edge $e \in E$. Also given a set of demand pairs $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$ and demand values $d_1, d_2, \ldots, d_k$ such that each demand pair (s_i, t_i) is associated with the demand value d_i for $1 \leq i \leq k$.

Given a set of edges $E' \subseteq E$ (a cut). Let $c(E') = \sum_{e \in E'} c(e)$. Let $dem(E') = \sum_{i:(s_i,t_i) \text{ is separated by } E'}$, where (s_i, t_i) is separated by E' if they are not connected in $G[E \backslash E']$. Let the sparsity of E' be $sparsity(E') = c(E')/dem(E')$. The goal of the sparsest cut problem is to find a cut E' of minimum sparsity (sparsest cut). This problem is NP-hard.

Claim 1 *There always exists a cut of minimum sparsity which induces exactly 2 connected components.*

Proof Sketch. For any cut E' which induces $m (\geq 3)$ connected components, each of the m components can be separated from the remaining components via a subset of E'. There must be a subset of E' with sparsity less than or equal to that of E'. □

Because of the above claim, the sparsest cut problem can be also formulated as to find $S \subset V$ so as to minimize $c(\delta(S))/dem(\delta(S))$.

A special case of the sparsest cut problem is the following: for each pair of vertices $(u, v) \in V \times V$ $(u \neq v)$, we have its corresponding demand value $d(uv) = 1$. This is called the *uniform* case. The *non-uniform* case refers to the general problem.

The sparsest cut problem can be formulated via integer programming (IP) as follows.

$$
\begin{aligned}
\min \quad & \frac{\sum_{e \in E} c_e x_e}{\sum_{i=1}^{k} d_i y_i} \\
\text{s.t.} \quad & \\
& \sum_{e \in p} x_e \geq y_i, \qquad p \in \mathcal{P}_{s_i t_i} \\
& y_i \in \{0, 1\}, \qquad 1 \leq i \leq k \\
& x_e \in \{0, 1\}, \qquad e \in E
\end{aligned}
$$

In the IP, y_i represents whether or not the pair (s_i, t_i) should be separated. The IP tries to directly minimize the sparsity function as a ratio of the total edge capacity to the total demand value separated. For its linear programming relaxation, since we cannot use ratios in LP, the denominator of the ratio is normalized to a constraint which says $\sum_{i=1}^{k} d_i y_i = 1$ (or $\sum_{i=1}^{k} d_i y_i \geq 1$ equivalently as will be shown in the LP shortly) and the numerator is minimized. Notice that the scaling does not affect the ratio. Therefore, we construct the LP as the following.

$$
\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e \\
\text{s.t.} \quad & \\
& \sum_{i=1}^{k} d_i y_i \geq 1 \\
& \sum_{e \in p} x_e \geq y_i, && p \in \mathcal{P}_{s_i t_i} \\
& y_i \geq 0, && 1 \leq i \leq k \\
& x_e \geq 0, && e \in E
\end{aligned}
$$

Remark 2 *The above LP is a valid relaxation to the IP.*

Now, let us construct the dual of the LP. For each path p between any demand pair (s_i, t_i) we have a dual variable $f(p)$, and the dual LP is formulated below.

$$
\begin{aligned}
\max \quad & \lambda \\
\text{s.t.} \quad & \\
& \sum_{p \in \mathcal{P}_{s_i t_i}} f(p) \geq \lambda d_i, && 1 \leq i \leq k \\
& \sum_{i=1}^{k} \sum_{\substack{p \ni e \\ p \in \mathcal{P}_{s_i t_i}}} f(p) \leq c_e, && \forall e \in E \\
& f(p) \geq 0 \\
& \lambda \geq 0
\end{aligned}
$$

This dual LP can be interpreted as the Maximum Concurrent Flow problem: Given $G = (V, E)$, edge capacities $c : E \to \mathbb{R}^+$, a set of demand pairs of vertices $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$, and their demand values $d_1, d_2, \ldots, d_k$, the goal is to maximize λ such that each pair (s_i, t_i) can concurrently send flow of λd_i. Assume the optimial value for the maximum concurrent flow problem is λ^*, and we have the following claims.

Claim 3 $\lambda^* \leq \min_{S \subset V} \dfrac{c(\delta(S))}{dem(\delta(S))}$.

Proof: This follows from the fact that, for any $S \subset V$, the demand crossing S (which is $dem(\delta(S)) * \lambda^*$) cannot exceed the capacity of the cut $\delta(S)$. □

Lemma 4 *λ^* is a lower bound on the minimum sparsity.*

Proof: This immediately follows from the above claim and the definition of sparsity. □

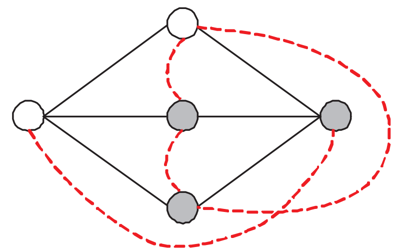

Figure 1: A case when λ^* is strictly less than minimum sparsity. The graph is in solid edges and demand pairs are in (red) dotted lines. All capacities/demands are 1.

Remark 5 *λ^* is equal to the minimum sparsity for $k \in \{1, 2\}$.*

When $k = 1$, the equality can be proved by the max-flow-min-cut theorem. The case for $k = 2$ is due to Hu's two-commodity flow theorem [1]. This means that when there is no more than 2 demand pairs of vertices, the minimum sparsity is exactly the optimal value of the maximum concurrent flow problem.

Remark 6 *λ^* may be strictly less than the minimum sparsity for $k \geq 3$.*

For $k \geq 3$, the equality between λ^* and the minimum sparsity does not always holds. To see why this is the case, we illustrate an example in Figure 1. Suppose the graph is in the solid edges (totally 6 edges) of capacity 1 and the dotted lines connect all 4 demand pairs with demand values being 1. Therefore, we have $\lambda^* \leq 3/4$ because the sum of the shortest paths between each pair is 8 but the total capacity of edges is only 6. On the other hand, the minimum sparsity of the graph is 1 (by letting S be the set of dark vertices). Thus, λ^* is strictly less than the minimum sparsity.

We already know that $\lambda^* \leq$ *minimum sparsity*. A natural question that arises is whether $\alpha * (\textit{minimum sparsity}) \leq \lambda^*$ for some $\alpha < 1$; this would allow us to get an $1/\alpha$ approximation for the sparsest cut problem since λ^* can be computed by solving the LP in polynomial time.

2 Flow-Cut Gap

Theorem 7 *For the uniform case, we have the following flow-cut gap: $\dfrac{\textit{minimum sparsity}}{c \log k} \leq \lambda^* \leq$ minimum sparsity, where c is a constant.*

Lemma 8 $\dfrac{\textit{minimum sparsity}}{c \log k} \leq \lambda^*$.

Proof Sketch. The idea is to first obtain the optimal solution for the LP, and then use it to construct a feasible solution for the multicut problem, which is in turn used to upper bound the sparsity.

First, recall that the variables for the LP are $\{y_i\}$ and $\{x_e\}$. Let us solve the LP and let $\{y_k^*\}$ and $\{x_e^*\}$ be the optimal values. Suppose that $y_i^* \in \{0, z\}$ for some $0 < z < 1$ and let $A = \{i | y_i^* = z\}$. Because of the LP constraint and that $d_i = 1$, we have $\sum_i y_i^* = 1$ and thus $z|A| = 1$.

Now, we round the optimal solution by letting $x'_e = x^*_e/z$. We have that $\{x'_e\}$ is a feasible solution for the LP of multicut problem (formulated on the pairs corresponding to A). This is because the constraints of the multicut LP can be expressed as $\sum_{e\in p} x'_e \geq 1$; since $\sum_{e\in p} x^*_e \geq y_i = z$, we can get $\sum_{e\in p} x^*_e/z \geq y_i/z = 1$, which means that the multicut's LP constraints $\sum_{e\in p} x'_e \geq 1$ can be satisfied.

From the previous lecture, we have learnt that when $\{x'_e\}$ is a feasible solution for the multicut problem, we can get a multicut E' in polynomial time such that it separates all pairs of vertices in A and its cost is no greater than $2H_k \sum_e c_e x'_e$.

Now we can upper bound the sparsity of E' as follows:

$$\begin{aligned} \text{sparsity of E'} &= \frac{c(E')}{dem(E')} \\ &\leq \frac{2H_k \sum_e c_e x'_e}{|A|} \\ &\leq \frac{2H_k \sum_e c_e x_e}{z|A|} \\ &= 2H_k \sum_e c_e x_e \\ &= 2H_k \lambda^*. \end{aligned}$$

Because *minimum sparsity* $\leq$ *sparsity of* E', we obtain *minimum sparsity* $\leq 2H_k\lambda^*$. Therefore, $\frac{\textit{minimum sparsity}}{c \log k} \leq \lambda^*$. □

Proof of the Theorem. Immediately from Lemmata 4 and 8. □

References

[1] T. C. Hu. Multi-Commodity Network Flows. Operations Research 11 (1963), 344-360.