IOE 691: Approximation Algorithms — Date: 03/13/2017

Lecture Notes: Prize Collecting Steiner Tree

Instructor: Viswanath Nagarajan — Scribe: Yutong Wang

1 Separation and Linear Optimization

Linear programs are of the form

$$\begin{aligned} \min \quad & c^t x \\ s.t. \quad & Ax \geq b \end{aligned}$$

where A is an m-by-n matrix, $n = \#$ of variables, $m = \#$ of constraints and $D =$ the maximum entry of the inputs to the problem, which we assume to be integer valued after scaling. Such a linear program can be solved exactly in $\text{poly}(n, m, \log_2 D)$.

Sometimes, m depends exponentially on n. In those cases, we may still be able to get solve the LP in polynomial time if the associated *separation* problem can be solved in polynomial time. The separation problem is the following: given linear constraints $Ax \geq b$ and $\bar{x} \in \mathbb{R}^n$, either

1. Certify that $A\bar{x} \geq b$, or
2. Find a violated constraint, i.e., $i = 1, \ldots, m$ such that $a_i^t \bar{x} < b_i$, where a_i is the i-th row of A.

It was proven in [1] that

Theorem 1.1 *The separation problem is polynomially equivalent to linear optimization.*

Note that for polynomially sized LP, the separation problem is trivial. However, in practice, a lot of interesting LP are large. Below, we will demonstrating solving a large LP by giving a polynomial time separation oracle (and invoking the above equivalence under the hood).

2 Steiner trees

Given a graph $G = (V, E)$, cost c_e for each $e \in E$, a root vertex $r \in V$ and a set of terminals $T \subseteq V$, the goal is to choose a set of edges which connects all terminals to the root with minimal cost. Consider the following LP formulation in which x_e is the decision variable for choosing the edge e in the solution:

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e x_e & & (P1) \\ s.t. \quad & \sum_{e \in \delta S} x_e \geq 1, & & \forall S \subseteq V : S \cap T \neq \emptyset,\ r \notin S, \\ & x \geq 0. \end{aligned}$$

The notation δS denotes the set of edges with exactly one end point in S. Observe that if $x_e \in \{0,1\}$ for all $e \in E$, then we get a valid solution. Indeed, if we fix any terminal $u \in T$, then the constraints imply that any r-u cut S satisfies $\sum_{e\in\delta S} x_e \geq 1$. Thus, u is connected to the root.

We will solve the separation problem for $(P1)$. Let $\bar{x} = \{\bar{x}_e \in \mathbb{R} : e \in E\}$ be given. For each terminal $u \in T$, let $\text{MinCut}(r,u,\bar{x})$ be the optimal value for the min r,u-cut with weights given by $\bar{x}$. Then

1. if $\text{MinCut}(r,u,\bar{x}) \geq 1$ for all $u \in T$, then $\bar{x}$ is feasible for $(P1)$.
2. otherwise, $\text{MinCut}(r,u,\bar{x}) < 1$ for some $u \in T$ and so we can output an r,u-cut S such that $\sum_{e\in\delta S} x_e < 1$.

This solves the separation problem for $(P1)$. Later in the course we will prove the following:

Theorem 2.1 *Given any feasible solution $\bar{x}$ for $(P1)$ (not necessarily optimal), one can find a tree H such that $\sum_{e\in H} c_e \leq 2c^t\bar{x}$. Consequently, we have a 2-approximation algorithm for the Steiner tree problem.*

We remark that $(P1)$ has an integrality gap of 2. To see this, consider when $G = (V,E)$ is the n-cycle graph with $T = V$ and $c_e = 1$ for all $e \in E$. In this case, the IP optimal value is $n-1$. On the other hand, setting $x_e = 1/2$ for each $e \in E$ yields a feasible solution to $(P1)$ whose objective value is $n/2$.

3 The Prize-Collecting Steiner Tree problem

Consider $G = (V,E)$, cost $c_e \geq 0$ for each $e \in E$, penalty π_v for each $v \in V$ and root $r \in V$. For a set of edges $H \subseteq E$, we write $c(H) = \sum_{e\in H} c_e$. The goal of the Prize-Collecting Steiner Tree problem (PCST) is to find a tree H rooted at r that minimizes $c(H) + \sum_{v\notin H} \pi_v$. Consider the following LP formulation of the problem with decision variables y_v = choose vertex v in the solution, i.e., $v \in H$, and x_e = choose edge e in the solution:

$$\begin{aligned} \min \quad & \sum_{e\in E} c_e x_e + \sum_{v\in V} \pi_v(1-y_v) \qquad\qquad (P2) \\ s.t. \quad & \sum_{e\in\delta S} x_e \geq y_v, \quad \forall v \in S : r \notin S \\ & x \geq 0,\ 1 \geq y \geq 0. \end{aligned}$$

We solve the separation problem for (P2). Let $(\bar{x},\bar{y})$ be a test point. Then

1. if $\text{MinCut}(r,v,\bar{x}) \geq \bar{y}_v$ for all $v \in V$, then $(\bar{x},\bar{y})$ is feasible for $(P2)$.
2. otherwise, $\text{MinCut}(r,u,\bar{x}) < \bar{y}_v$ for some $v \in V$ and so we can output an r,v-cut S such that $\sum_{e\in\delta S} x_e < \bar{y}_v$.

This solves the separation problem for $(P2)$ in polynomial time. Hence, by Theorem 1.1, the LP $(P2)$ can be solved in polynomial time.

3.1 First approach - a 3-approximate algorithm

Next, we will round the LP ($P2$) to get a 3-approximate algorithm. Let (x, y) be an optimal point for the LP ($P2$) with $C^* = \sum_{e \in E} c_e x_e$ and $\Pi^* = \sum_{v \in V} \pi_v (1 - y_v)$. Let α be a threshold to be chosen later. We observe that x/α is a feasible solution to and consider the following rounding scheme which we'll refer to as ALG_α

1. $T \leftarrow \{v \in V : y_v \geq \alpha\}$
2. Use Theorem 2.1 to find a Steiner tree H on the terminals T satisfying $c(H) \leq 2c^t x/\alpha$ by rounding the solutions x/α.

We observe that x/α is indeed feasible for ($P1$) because for each $v \in T$ and each r, v-cut S, we have $\sum_{e \in \delta S} x_e/\alpha \geq y_v/\alpha \geq 1$. Let $c(ALG_\alpha) = c(H)$ and $\pi(ALG_\alpha) = \sum_{v \notin H} \pi_v$. Thus, step 2 of ALG_α implies that $c(ALG_\alpha) \leq \frac{2}{\alpha} C^*$.

Lemma 3.1 $\pi(ALG_\alpha) \leq \sum_{v \notin T} \pi_v \leq (1 - \alpha)^{-1} \Pi^*$.

Proof: For each $v \notin T$, we have $y_v < \alpha$ and thus $1 - y_v > 1 - \alpha$. Hence,

$$\frac{\Pi^*}{1 - \alpha} \geq \sum_{v \notin T} \pi_v \frac{(1 - y_v)}{(1 - \alpha)} > \sum_{v \notin T} \pi_v \geq \sum_{v \notin H} \pi_v$$

By definition $\pi(ALG_\alpha) = \sum_{v \notin H} \pi_v$. Thus we're done. ■

Now, by setting $\alpha = 2/3$, we have

$$c(ALG_\alpha) + \pi(ALG_\alpha) \leq 3(C^* + \Pi^*).$$

Thus, we proved

Theorem 3.1 *There exists a 3-approximate algorithm for PCST.*

3.2 Refined approach - a 2.54-approximate algorithm

Below, we will show how we can get a better approximation ratio. Everything we did thus far was relative to various α values. If we check all the α values, i.e., by finding $\min_{\alpha \in [0,1]} ALG_\alpha$, we can get an 2.54-approximation algorithm. Abusing notation, let $ALG_\alpha := c(ALG_\alpha) + \pi(ALG_\alpha)$ denote the optimal objective value. For any distribution $\mathcal{D}$ on $[0, 1]$, we have

$$\min_{\alpha \in [0,1]} ALG_\alpha \leq \mathbb{E}_{\alpha \sim \mathcal{D}}[ALG_\alpha].$$

Thus, instead of analyzing $\min_{\alpha \in [0,1]} ALG_\alpha$, we will instead analyze $\mathbb{E}_{\alpha \sim \mathcal{D}}[ALG_\alpha]$. Let $a \in [0, 1]$ be a paramater to be chosen later. We take $\mathcal{D}$ to be the uniform distribution on $[a, 1]$. For simplicity, all expectations are taken with respect to $\alpha \sim \mathcal{D}$, i.e., $\mathbb{E} = \mathbb{E}_{\alpha \sim \mathcal{D}}$.

Lemma 3.2 $\mathbb{E}[\pi(ALG_\alpha)] \leq \mathbb{E}[\sum_{v \notin T} \pi_v] \leq (1 - a)^{-1} \Pi^*$

Proof: Let $v \in V$. By ALG_α's construction of T, we have $v \notin T \iff \alpha > y_v$. Thus,

$$\Pr(v \notin T) = \Pr(\alpha > y_v) \leq \frac{1 - y_v}{1 - a}.$$

Next, $\pi(ALG_\alpha) = \sum_{v \notin H} \pi_v \leq \sum_{v \notin T} \pi_v$, hence $\mathbb{E}[\pi(ALG_\alpha)] \leq \mathbb{E}[\sum_{v \notin T} \pi_v]$. To finish the proof, note that

$$\mathbb{E}[\sum_{v \notin T} \pi_v] = \mathbb{E}[\sum_{v \in V} \pi_v \mathbf{1}_{v \notin T}] = \sum_{v \in V} \pi_v \Pr(v \notin T) \leq \sum_{v \in V} \pi_v \frac{1 - y_v}{1 - a} = \frac{\Pi^*}{1 - a}$$

■

Lemma 3.3 $\mathbb{E}[c(ALG_\alpha)] \leq \left(\frac{2}{1-a} \ln \frac{1}{a}\right) C^*$

Proof: Earlier, we saw that $c(ALG_\alpha) \leq \frac{2}{\alpha} C^*$. Thus

$$\mathbb{E}[c(ALG_\alpha)] \leq 2C^* \int_a^1 \frac{1}{(1-a)\alpha} d\alpha = \frac{2}{1-a} C^* \ln \frac{1}{a}$$

■

Now, we simply choose a so that $\frac{2}{1-a} \ln \frac{1}{a} = \frac{1}{1-a}$ which gives us $a = e^{-1/2}$. This gives us an approximation ratio of

$$\frac{1}{1-a} = \frac{1}{1 - e^{-1/2}} \approx 2.54$$

References

[1] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.