Saarland
Informatics Campus

Script of Lecture 6, Approximation Algorithms
Summer term 2017

Tobias Mömke, Hang Zhou
`http://www-cc.cs.uni-saarland.de/course/61/`

Written by Hang Zhou

1 Programs

Suppose we want to solve the following linear program:

$$\begin{aligned} \text{maximize} \sum_{j=1}^{n} d_j x_j & \\ \text{subject to} \sum_{j=1}^{n} a_{ij} x_j \geq b_i, & \qquad i = 1, \ldots, m, \\ x_j \geq 0, & \qquad j = 1, \ldots, n. \end{aligned}$$

In general, the number of constraints m can be exponential in n. However, using an algorithm called the *ellipsoid method*, we can solve the LP in time polynomial in n as soon as there is a polynomial-time *separation oracle*.

Definition 1. *A* separation oracle *takes as input a supposedly feasible solution x to the linear program. It either verifies that x is indeed a feasible solution to the LP, or if it is not feasible, produces a constraint that is violated by x, i.e., with $\sum_{j=1}^{n} a_{ij} x_j < b_i$.*

The details of the ellipsoid method are beyond the scope of this course.

2 Prize-Collecting Steiner Tree

Input: Undirected graph $G = (V, E)$ with edge cost $c_e \geq 0$ for each $e \in E$, a root vertex $r \in V$, and a penalty $\pi_i \geq 0$ for each $i \in V$.

Output: a tree T that contains the root vertex r so as to minimize the cost of the edges in the tree plus the penalties of all vertices not in the tree, i.e., $\sum_{e \in T} c_e + \sum_{i \in V - V(T)} \pi_i$, where $V(T)$ is the set of vertices in the tree T.

Remark 1. *The standard Steiner tree problem is a special case of the above problem, where the penalty of a vertex is either ∞ or 0.*

2.1 Integer Program

For each vertex $i \in V$, create a variable y_i, such that $y_i = 1$ if i is in the solution tree and 0 otherwise. For each edge $e \in E$, create a variable x_e, such that $x_e = 1$ if e is in the solution tree and 0 otherwise. The goal is to minimize $\sum_{e \in E} c_e x_e + \sum_{i \in V} \pi_i (1 - y_i)$.

Now we need constraints to ensure that the edges in the solution lead to a tree that contains the root r and each vertex i with $y_i = 1$.

For a non-empty set of vertices $S \subset V$, let $\delta(S)$ be the set of edges in the *cut* defined by S, i.e., the set of all edges with exactly one endpoint in S. We introduce the constraints

$$\sum_{e \in \delta(S)} x_e \geq y_i$$

for each $S \subseteq V - r$ and each $i \in S$. We observe that each constraint must be satisfied in any feasible solution: if $y_i = 1$ but $\sum_{e \in \delta(S)} x_e = 0$, then r and i belong to different components in the solution, therefore the solution is not a tree that contains both r and i.

Now we show that these constraints together imply that the solution is a tree that contains the root r and each vertex i with $y_i = 1$. Consider any solution satisfying these constraints and consider the graph $G' = (V, E')$ where $E' = \{e \in E : x_e = 1\}$. Pick any $i \in V - r$ with $y_i = 1$. The constraints ensure that for any (r, i) cut S, there must be at least one edge of E' in $\delta(S)$, so the size of the minimum (r, i) cut in G' is at least 1. By the max-flow/min-cut theorem, the size of the maximum (r, i) flow is therefore at least 1. Thus there exists some r-to-i path in G'.

From the above, we know that r is connected to every vertex i with $y_i = 1$ if and only if all these constraints are satisfied.

We have the following integer program:

$$
\begin{aligned}
\text{minimize} \sum_{e \in E} c_e x_e + \sum_{i \in V} \pi_i (1 - y_i) & \\
\text{subject to} \sum_{e \in \delta(S)} x_e \geq y_i, & \qquad \forall S \subseteq V - r \text{ and } \forall i \in S, \\
y_r = 1, & \\
y_i \in \{0, 1\}, & \qquad \forall i \in V, \\
x_e \in \{0, 1\}, & \qquad \forall e \in E.
\end{aligned}
$$

2.2 Linear Program Relaxation

We relax the integer program to a linear program by replacing the constraints $y_i \in \{0, 1\}$ and $x_e \in \{0, 1\}$ by $y_i \geq 0$ and $x_e \geq 0$. We solve the LP using the ellipsoid method. In order to ensure polynomial running time, we only need to show that there is a polynomial time separation oracle for the constraints $\sum_{e \in \delta(S)} x_e \geq y_i$. The oracle is as follows: given a solution (x, y), we construct a network flow problem on the graph G in which the capacity of each edge e is set to x_e. For each vertex i, we check whether the maximum flow from the root r to i is at least y_i. If not, then the minimum cut S separating i from r corresponds to a violated constraint with $\sum_{e \in \delta(S)} x_e < y_i$. If the flow is at least y_i, then by the max-flow/min-cut theorem, for all cuts S separating i from r, we have $\sum_{e \in \delta(S)} x_e \geq y_i$. Hence, given a solution (x, y), we can find a violated constraint, if any exists, in polynomial time.

3 Deterministic Rounding for Prize-Collecting Steiner Tree

Definition 2. *Let $U \subseteq V$ be a set of vertices. A* Steiner tree on U *is a tree in G that contains all vertices of U.*

Given an optimal solution (x^*, y^*) to the linear program relaxation in Section 2, we have a simple deterministic rounding as follows:

a) Let $U \subseteq V$ be the set of vertices $i \in V$ such that $y_i^* \geq \alpha$; (α is a parameter to be defined later)

b) Use the primal-dual method to find a Steiner tree T on the set of terminals U, and return T.

Now we analyze the cost of our solution tree T. The cost contains two parts: the total edge costs $\sum_{e \in T} c_e$ and the total penalties $\sum_{i \in V - V(T)} \pi_i$. For the first part, we have:

Lemma 1.

$$\sum_{e \in T} c_e \le \frac{2}{\alpha} \sum_{e \in E} c_e x_e^*.$$

The proof is left as an exercise.

For the second part, we have:

Lemma 2.

$$\sum_{i \in V - V(T)} \pi_i \le \frac{1}{1-\alpha} \sum_{i \in V} \pi_i (1 - y_i^*).$$

The proof is simple: every vertex $i \in V - V(T)$ must have $y_i^* < \alpha$, therefore, $\frac{1-y_i^*}{1-\alpha} > 1$.

From the above two lemmas, we have:

Theorem 1.

$$\sum_{e \in T} c_e + \sum_{i \in V - V(T)} \pi_i \le \frac{2}{\alpha} \sum_{e \in E} c_e x_e^* + \frac{1}{1-\alpha} \sum_{i \in V} \pi_i (1 - y_i^*).$$

By setting $\alpha = \frac{2}{3}$ in the above theorem, we obtain:

Corollary 1. *We have a polynomial time 3-approximation algorithm for the prize-collecting Steiner tree problem.*

4 Randomized Rounding for Prize-Collecting Steiner Tree

The only difference from the deterministic algorithm is that the parameter α is not fixed, but taken uniformly at random from the range $[\beta, 1]$ for some $\beta > 0$ that we specify later.

Lemma 3.

$$E\left[\sum_{e \in T} c_e\right] \le \left(\frac{2}{1-\beta} \ln \frac{1}{\beta}\right) \sum_{e \in E} c_e x_e^*.$$

Proof.

$$\begin{aligned}
E\left[\sum_{e \in T} c_e\right] &\le E\left[\frac{2}{\alpha} \sum_{e \in E} c_e x_e^*\right] && \text{(by Lemma 1)} \\
&= E\left[\frac{2}{\alpha}\right] \sum_{e \in E} c_e x_e^* \\
&= \left(\frac{1}{1-\beta} \int_\beta^1 \frac{2}{x} dx\right) \sum_{e \in E} c_e x_e^* \\
&= \left[\frac{2}{1-\beta} \ln x\right]_\beta^1 \cdot \sum_{e \in E} c_e x_e^* \\
&= \left(\frac{2}{1-\beta} \ln \frac{1}{\beta}\right) \cdot \sum_{e \in E} c_e x_e^*
\end{aligned}$$

□

Lemma 4.

$$E\left[\sum_{i \in V - V(T)} \pi_i\right] \le \frac{1}{1-\beta} \sum_{i \in V} \pi_i (1 - y_i^*).$$

The proof follows from Lemma 2 and the fact that $\alpha \geq \beta$.

From the above two lemmas, we have:

Theorem 2.

$$E\left[\sum_{e \in T} c_e + \sum_{i \in V - V(T)} \pi_i\right] \leq \left(\frac{2}{1-\beta} \ln \frac{1}{\beta}\right) \sum_{e \in E} c_e x_e^* + \frac{1}{1-\beta} \sum_{i \in V} \pi_i (1 - y_i^*).$$

Corollary 2. *By setting $\beta = e^{-1/2}$, we obtain a polynomial time $\frac{1}{1-e^{-1/2}}$-approximation algorithm for the prize-collecting Steiner tree problem, where $\frac{1}{1-e^{-1/2}} \approx 2.54$.*

The derandomization of this algorithm is left as an exercise.