Approximations for Steiner Tree Problem

Chihao Zhang

BASICS, Shanghia Jiao Tong University

January 7, 2013

Steiner Tree: Problem Statement

Input:	Given an undirected graph $G = (V, E)$, an edge cost $c_e \geq 0$ for each $e \in E$. V is partitioned into two sets, *terminals* and *Steiner vertices*.
Problem:	Find a minimum cost tree in G that contains all the terminals and any subset of the Steiner vertices.

MST Based Algorithm

- If G is a complete graph and the costs satisfy *the triangle inequality*, i.e.,

$$\mathrm{cost}(u, v) \leq \mathrm{cost}(u, w) + \mathrm{cost}(w, v),$$

then we call it metric Steiner tree problem.

Theorem

There is an 2-approximation algorithm for metric Steiner tree problem.

MST Based Algorithm (cont'd)

- The algorithm simply returns the minimum spanning tree on terminal vertices.
- To see it is a 2-approximation algorithm, oberseve that we can obtain a MST of terminals from optimal solution of Steiner tree problem by doubling its cost using triangular inequality.

Theorem

There is an approximation factor preserving reduction from the Steiner tree problem to the metric Steiner tree problem.

Steiner Forest Problem

Input:	Given an undirected graph $G = (V, E)$, nonnegative costs $c_e \geq 0$ for all edges $e \in E$ and k pairs of vertices $s_i, t_i \in V$..
Problem:	Find a minimum cost subset of edges $F \subseteq E$ such that every s_i-t_i pair is connected in the set of selected edges.

Integer ProgramLinear Program Relaxation

$$\begin{array}{rll} \text{minimize} & \sum_{e \in E} c_e x_e & \\ \text{subject to} & \sum_{e \in \delta(S)} x_e \geq 1, & \forall S \subseteq V : S \in \mathcal{S}_i \text{ for some } i, \\ & x_e \in \{0,1\} x_e \geq 0, & e \in E. \end{array}$$

where $\mathcal{S}_i := \{S \subseteq V \mid |S \cap \{s_i, t_i\}| = 1\}$ and
$\delta(S) := \{e = \{u, v\} \in E \mid u \in S, v \notin S\}$

The Dual Program

$$
\begin{array}{lll}
\text{maximize} & \sum_{S\subseteq V:\exists i, S\in\mathcal{S}_i} y_S & \\
\text{subject to} & \sum_{S:e\in\delta S} y_S \le c_e, & \forall e\in E, \\
& y_S \ge 0, & \exists i : S\in\mathcal{S}_i
\end{array}
$$

Standard Primal-Dual Schema

1. $y \leftarrow 0$
2. $F \leftarrow \varnothing$
3. **while** not all s_i-t_i pairs are connected in (V, F) **do**
    3.1 Let C be a connected component of (V, F) such that $|C \cap \{s_i, t_i\}| = 1$ for some i
    3.2 Increase y_C until there is an edges $e' \in \delta(C)$ such that $\sum_{S \in \mathcal{S}_i : e' \in \delta(S)} y_S = c_{e'}$
    3.3 $F \leftarrow F \cup \{e'\}$
4. **return** F

Standard Primal-Dual Schema (cont'd)

- Using the standard primal-dual analysis, we have

$$\sum_{e \in F} c_e = \sum_{e \in F} \sum_{S : e \in \delta(S)} y_S = \sum_S |\delta(S) \cap F| y_S .$$

- However, $|\delta(S) \cap F|$ can be as large as k!
- We will use an average case analysis instead of worst case analysis.

Primal-Dual Schema with Synchronization

1. $y \leftarrow 0$
2. $F \leftarrow \varnothing$
3. **while** not all s_i-t_i pairs are connected in (V, F) **do**
   - 3.1 Let $\mathcal{C}$ be the set of all connected components C of (V, F) such that $|C \cap \{s_i, t_i\}| = 1$ for some i
   - 3.2 Increase y_C for all C in $\mathcal{C}$ uniformly until for some $e_\ell \in \delta(C'), C' \in \mathcal{C}, c_{e_\ell} = \sum_{S: e_\ell \in \delta(S)} y_S$
   - 3.3 $F \leftarrow F \cup \{e_\ell\}$
4. $F' \leftarrow \{e \in F \mid F \setminus \{e\} \text{ is primal infeasible}\}$
5. **return** F'

Analysis

Using standard primal-dual analysis, we have

$$\sum_{e \in F'} c_e = \sum_{e \in F'} \sum_{S: e \in \delta(S)} y_S = \sum_S |F' \cap \delta(S)| y_S.$$

We will show that

$$\sum_S |F' \cap \delta(S)| y_S \leq 2 \sum_S y_S$$

This follows from the following lemma:

Lemma

For any $\mathcal{C}$ in any iteration of the algorithm,

$$\sum_{C \in \mathcal{C}} |\delta(C) \cap F'| \leq 2|\mathcal{C}|$$

Analysis (cont'd)

Proof.

The high-level idea is the following: If in each iteration, we contract every components into a single vertex, then F' is a forest in the contracted graph. Then we apply the fact that the average degree of vertices in a forest is at most 2.

Prize-collecting Steiner Tree Problem

Input:	An undirected graph $G = (V, E)$, an edge cost $c_e \geq 0$ for each $e \in E$, a selected root vertex $r \in V$, and a penalty $\pi_i \geq 0$ for each $i \in V$.
Problem:	Find 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

- A generalization of Steiner Tree Problem, where $\pi_i = \infty$ if i is terminal and $\pi_i = 0$ otherwise.

Integer ProgramLinear Program Relaxation

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

Digression: Ellipsoid Method

Definition (Separation Oracle)

A separation oracle takes as input a solution x and either verifies that x is a feasible solution or produces a constraint that is violated by x.

- Provided a polynomial-time separation oracle, the ellipsoid method can solve a linear program in polynomial time.

Polynomial-time Separation Oracle

The following algorithm is a polynomial-time separation oracle for our linear program relaxation of prize-collecting Steiner tree problem:

Given a solution (x, y),

1. Construct a network folow problem on G in which the capacity of each edge e is x_e.
2. For each $i \in V$
   2.1 If the maximum flow from from i to r is less than y_i
      2.1.1 Return the constraint (S, i) where S is the minimum cut from i to r
3. Return "(x, y) *is a feasible solution*".

Deterministic Rounding

1. Let $\alpha \in [0, 1)$ be a parameter to be fixed.
2. $U := \{i \in V \mid y_i \geq \alpha\}$.
3. Find a Steiner tree T on G with terminals U.
4. Return T.

Lemma

The tree T returned by the primal-dual algorithm for Steiner tree has cost

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

Proof.

Oberseve that if $\{x_e^*, y_i^*\}$ is a solution of LP for prize-collecting Steiner tree problem, then $\{\frac{x_e^*}{\alpha}\}$ is a solution of LP for Steiner tree problem.

Deterministic Rounding (cont'd)

Lemma

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

Theorem

The cost of the solution produced by the deterministic rounding algorithm is

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

Taking $\alpha = \frac{2}{3}$, the algorithm is a 3-approximation algorithm for the prize-collecting Steiner tree problem.

Randomized Rounding

- Let $0 < \gamma \leq 1$ be a constant to be fixed. Choose α uniformly from $[\gamma, 1]$.

Lemma

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

Lemma

$$E\left[\sum_{i \in V \setminus V(T)} \pi_i\right] \leq \frac{1}{1-\gamma} \sum_{i \in V} \pi_i (1 - y_i^*)$$

Randomized Rounding (cont'd)

Theorem

The expected cost of the solution produced by the randomized algorithms is

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

- Choosing $\gamma = e^{-1/2}$ gives a $(1-e^{-1/2})^{-1}$-approximation algorithm for the prize-collecting Steiner tree problem, where $(1-e^{-1/2})^{-1} \approx 2.54$.
- This algorithm can be easily derandomized.