U.C. Berkeley — CS294: PCP and Hardness of Approximation — Professor Luca Trevisan
Handout N4 — January 30, 2006 — Scribe: Luca Trevisan

Notes for Lecture 4

These notes are based on my survey paper [3]. L.T.

Inapproximability of Metric Steiner Tree and Independent Set

1 Steiner Tree

In the Steiner tree problem we are given a metrix space (X, d), where X is a finite set of points and $d : X \times X \to \mathbb{R}$ is a metric, and a subset $C \subseteq X$ of points.

We want to find a tree $T = (V, E)$ such that $C \subseteq V \subseteq X$ of minimum cost, where the cost of a tree is $\sum_{(u,v)\in E} d(u, v)$.

This problem is different from the minimum spanning tree problem because the tree is not required to contain the points in $X - C$, although it may contain some of them if this is convenient.

We describe a reduction from the Vertex Cover problem in bounded degree graphs to the Steiner Tree problem. The reduction is due to Bern and Plassmann [1].

We start from a graph $G = (V, E)$, and we assume that G is connected.[1] We define a metric space (X, d) where X has $|V| + |E|$ points, that is, a point $[v]$ for each vertex v of G and a point $[u, v]$ for each edge (u, v) of G.

The distances in G' are defined as follows:

- For every edge $(u, v) \in E$, the points $[u]$ and $[u, v]$ are at distance one, and so are the points $[v]$ and $[u, v]$.
- Any two points $[u]$, $[v]$ are at distance one.
- All other pairs of vertices are at distance two.

We let C be the set of points $\{[u, v] : (u, v) \in E\}$. This completes the description of the instance of the Steiner Tree problem. Notice that, since all the distances are either one or two, they satisfy the triangle inequality, and so the reduction always produces an instance of Metric Steiner Tree.

Claim 1 *If there is a vertex cover in G with k vertices, then there is a Steiner tree in (X, d) of cost $m + k - 1$.*

[1] We proved inapproximability of Vertex Cover without guaranteeing a connected graph. Clearly, if we have an approximation algorithm that works only in connected graphs, we can make it work on general graphs with same factor. It follows that any inapproximability for general graphs implies inapproximability of connected graphs with the same factor.

PROOF: Let S be the vertex cover. Create a graph G' whose vertices are the points $\{[v] : v \in S\}$ and the points $\{[u,v] : (u,v) \in E\}$, and whose edges connect pairs of points that are distance 1 in (X,d). We have described a connected graph G', because every vertex in $\{[u] : u \in S\}$ is connected to every other vertex in the same set, and every vertex $[u,v]$ is connected to a vertex in $\{[u] : u \in S\}$. Let us take any spanning tree of this subgraph. It has $m+k-1$ edges of weight one, and so it is of cost $m+k-1$, and it is a feasible solution to the Steiner Tree problem. $\square$

Claim 2 *If there is a Steiner tree in (X,d) of cost $\leq m+k-1$, then there is a vertex cover in G with $\leq k$ vertices.*

PROOF: Let T be a feasible Steiner tree. We first modify the tree so that it has no edge of cost 2. We repeatedly apply the following steps.

- If there is an edge of cost 2 between a point $[w]$ and a point $[u,v]$, we remove it and add the two edges $([w],[u])$ and $([u],[u,v])$ of cost 1.
- If there is an edge of cost 2 between a point $[u,v]$ and a point $[v,w]$, we remove it and add the two edges $([u,v],[v])$ and $([v],[v,w])$ of cost 1.
- Finally, and this case is more interesting, if there is an edge of cost 2 between the points $[u,v]$ and $[w,z]$, we remove the edge, and then we look at the two connected components into which T has been broken. Some points $[u,v]$ are in one component, and some such points are in the other. This corresponds to a partition of the edges of G into two subsets. Since G is connected, we see that there must be two edges on different sides of the partition that share an endpoint. Let these edges be (u,v) and (v,w) then we can reconnect T by adding the edges $([u,v],[v])$ and $([v],[v,w])$.

We repeat the above steps until no edges of cost two remain. This process will not increase the cost, and will return a connected graph. We can obtain a tree by removing edges, and improving the cost, if necessary.

The final tree has only edges of weight one, and it has a cost $\leq m+k-1$, so it follows that it spans $\leq m+k$ vertices. The m points $\{[u,v] : (u,v) \in E\}$ must be in the tree, so the tree has $\leq k$ points $[v]$. Let S be the set of such vertices. We claim that this is a vertex cover for G. Indeed, for every edge (u,v) in G, the point $[u,v]$ is connected to the rest of the tree using only edges of weight one, which means that either $[u]$ or $[v]$ is in the tree, and that either u or v is in S. $\square$

If we combine the reduction with the results of past lectures, we prove the following theorem.

Theorem 3 *There is a constant ϵ_3 such that if there is a polynomial time $(1+\epsilon_3)$-approximate algorithm for Metric Steiner Tree then $\mathbf{P} = \mathbf{NP}$.*

2 More About Independent Set

In this section we describe a direct reduction from PCP to the Independent Set problem. This reduction is due to Feige et al. [2].

Let L be **NP**-complete, and V be a verifier showing that $L \in \mathbf{PCP}_{c,s}[q(n), r(n)]$. For an input x, let us consider all possible computations of $V^w(x)$ over all possible proofs w; a complete description of a computation of V is given by a specification of the randomness used by V, the list of queries made by V into the proof, and the list of answers. Indeed, for a fixed input x, each query is determined by x, the randomness, and the previous answers, so that it is enough to specify the randomness and the answers in order to completely specify a computation. We call such a description a *configuration*. Note that the total number of configuration is at most $2^{r(n)} \cdot 2^{q(n)}$, where n is the length of x.

Consider now the graph G_x that has a vertex for each *accepting* configuration of V on input x, and has an edge between two configurations c, c' if c and c' are *inconsistent*, that is, if c and c' specify a query to the same location and two different answers to that query. We make the following claims.

Claim 4 *If $x \in L$, then G_x has an independent set of size $\geq c \cdot 2^{r(n)}$.*

Proof: If $x \in L$, then there is a proof w such that $V^w(x)$ accepts with probability at least c, that is, there is a proof w such that there are at least $c \cdot 2^{r(n)}$ random inputs that make $V^w(x)$ accept. This implies that there are at least $c \cdot 2^{r(n)}$ mutually consistent configurations in the graph G_x, and they form an independent set. □

Claim 5 *If $x \notin L$, then every independent set of G_x has size $\leq s \cdot 2^{r(n)}$.*

Proof: We prove the contrapositive: we assume that there is an independent set in G_x of size $> s \cdot 2^{r(n)}$, and we show that this implies $x \in L$. Define a witness w as follows: for every configuration in the independent set, fix the bits in w queried in the configuration according to the answers in the configurations. Set the bits of w not queried in any configuration in the independent set arbitrarily, for example set them all to zero. The $> s \cdot 2^{r(n)}$ configurations in the independent set correspond to as many different random strings. When $V^w(x)$ picks any such random string, it accepts, and so $V^w(x)$ accepts with probability bigger than s, implying $x \in L$. □

It follows that

Theorem 6 *If there is a ρ-approximate algorithm for the independent set problem, then every problem in $\mathbf{PCP}_{c,s}[r(n), q(n)]$ can be solved in time poly$(n, 2^{r(n)+q(n)})$, provided $c/s < \rho$.*

From the PCP Theorem we immediately get that there cannot be a ρ-approximate algorithm for the independent set problem with $\rho < 2$ unless $\mathbf{P} = \mathbf{NP}$, but we can do better, as we shall prove in the next lecture.

References

[1] M. Bern and P. Plassmann. The Steiner tree problem with edge lengths 1 and 2. *Information Processing Letters*, 32:171–176, 1989. 1

[2] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268–292, 1996. Preliminary version in *Proc. of FOCS91*. 3

[3] Luca Trevisan. Inapproximability of combinatorial optimization problems. Technical Report TR04-065, Electronic Colloquium on Computational Complexity, 2004. 1