CS880: Approximations Algorithms

Scribe: Dave Andrzejewski **Lecturer:** Shuchi Chawla
Topic: Embedding metrics into trees **Date:** 3/30/07

This lecture introduces the idea of embedding a metric into a tree, and applies this technique to the development of an approximation algorithm for the Multicommodity "Buy at Bulk" Network Design problem.

22.1 Multicommodity Buy-At-Bulk Network Design

22.1.1 Problem formulation

GIVEN:

- a graph $G = (V, E)$
- edge lengths ℓ_e
- pairs of demand vertices (s_i, t_i)
- quantities q_i to be sent $s_i \to t_i$
- a concave cost function $f(c_e)$ for "buying" capacity c_e on edge e

DO: find

- a set of paths P_i from s_i to t_i such that $\sum_{p \in P_i} p \geq q_i$ for each i
- a set of edge capacity purchases c_e such that $\sum_{\{p|e \in p\}} p \leq c_e$

such that the total cost $\sum_e f(c_e)\ell_e$ is minimized.

The cost of purchasing capcity c_e on edge e is defined as $f(c_e)\ell_e$. This means that the cost of purchasing edge capacity is linearly related to the length of that edge, which will be important for our analysis. Also note that the cost function f is concave, and shared by all edges. This makes our formulation the "uniform" case. If each edge were allowed to have a different cost function f_e, it would be the non-uniform case, which is much harder, and was not known to have any sub-polynomial approximation until recently, when a poly-log approximation was discovered [1].

22.1.2 Algorithm design

Our first observation is that this problem would be greatly simplified for the special case where G is a tree, because each $s_i \to t_i$ path would be unique.

Furthermore, we recall that the total cost is linear in the edge lengths ℓ_e. This means that if we can find a low-distortion embedding from our graph G to some tree T, it will be relatively simple to analyze the impact of the distortion on our cost function.

22.1.3 Tree embeddings

To analyze potential embeddings, we must first ask which graph structures would be most difficult to embed into a tree. A natural first thought is to consider a complete graph. However, note that we could simply place any single node as the hub, and have all other nodes only be connected to the root as spokes. In the simplified case of a a graph with uniform edge costs, this clearly achieves an expansion factor of $\rho = 2$ (Figure 22.1.1).

The actual worst-case would be a graph which is simply a single large cycle of all n nodes (n-cycle). In this case, we can create a tree by simply removing any single edge. However, the distance between the 2 endpoints of that edge has now expanded by a factor of $\rho = \Omega(n)$ (Figure 22.1.1). It can be shown, in fact, that no embedding of the n-cycle into trees has distortion $o(n)$.

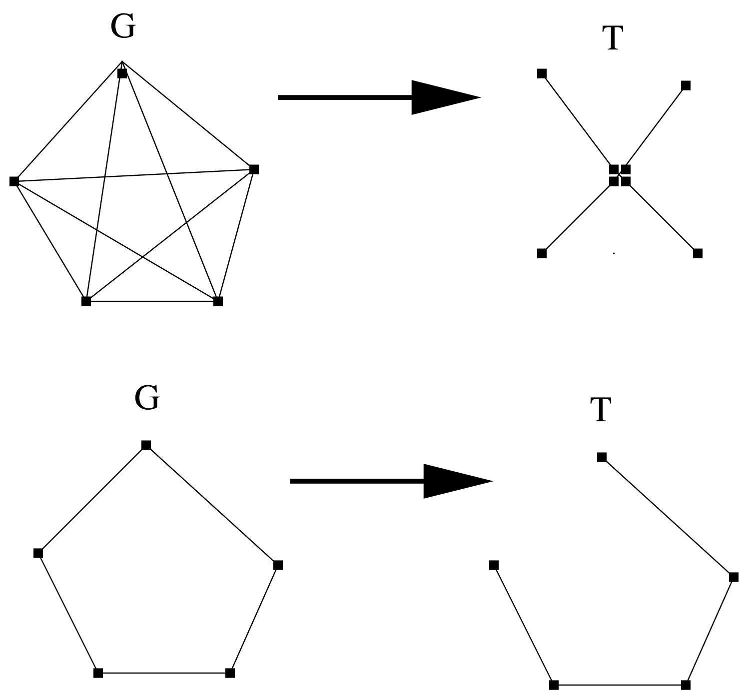

Figure 22.1.1: Embedding example graphs into trees.

22.1.4 Probabilistic tree embeddings

To avoid this worst-case scenario, we note that while embedding into a *single* tree may suffer large worst-case distortion, embedding into a *distribution* over trees can still achieve low expected distortion.

For our n-cycle graph example, define α as the uniform distribution over τ, the set of all trees T created by removing a single edge of the original graph. The expected distance between 2 vertices in an embedding drawn from α is then given by

$$d_\alpha(x,y) = E_{\alpha_T}[d_T(x,y)] = \sum_{T\in\tau} \alpha_T d_T(x,y)$$

where α_T is the probability of drawing tree T, and $d_T(x,y)$ is the distance between vertices x and y in T.

Theorem 22.1.1 *An n-cycle embeds into our distribution α with distortion $\rho = 2$.*

Proof: Each edge of the n-cycle is included in $n-1$ of the trees in τ. Therefore

$$E_\alpha[d_T(A,B)] = \frac{n-1}{n}(1) + \frac{1}{n}(n-1) \leq 2$$

■

We now formalize the our probabilistic embedding idea with the following definition.

Definition 22.1.2 *Given G, a β-*probabilistic embedding *into a distribution over trees is a distribution β over τ such that*

- $V[T_i] \supseteq V[G]\ \forall T_i \in \tau$
- $d_{T_i}(x,y) \geq d_G(x,y)\ \forall x,y\ \forall T_i \in \tau$
- $E_\alpha[d_T(x,y)] = \sum_T \alpha_T d_T(x,y) \leq \beta d_G(x,y)\ \forall x,y$

This definition sets up the main result from this lecture, which is that for all G there exists an $O(\log n)$-probabilistic embedding into trees.

22.1.5 Approximation of network design with a probabilistic embedding

Theorem 22.1.3 *Given a β-probabilistic embedding of G into trees, there exists a β-approximation for multicommodity uniform buy-at-bulk network design on G.*

Proof: Consider the following algorithm:

1. Given a β-probabilistic distribution α, pick $T \sim \alpha$

2. Solve uniform buy-at-bulk network design on T

3. Foreach $e = (u, v) \in T$, find shortest (u, v) path p in G and install capacity c_e^T on all edges in p

4. Map all p_i^T to their corresponding paths in G

This procedure clearly recovers a feasible solution on G, since all paths and capacities in the valid T solution are feasibly mapped to G.

What is the cost of our converted solution?

Claim 22.1.4 $E[cost_T] \leq \beta OPT$

Proof: Translate OPT_G to some solution in T by mapping each edge (u, v) in G to the unique path between u and v in T. Then

$$\begin{aligned} E[OPT_T] \leq E[cost_T] &\leq \sum_{(u,v)\in E_g} f(c_e^{OPT}) d_T(u, v) && (22.1.1) \\ &= \sum_{(u,v)\in E_g} f(c_e^{OPT}) \beta d_G(u, v) && (22.1.2) \\ &= \beta \sum_{(u,v)\in E_g} f(c_e^{OPT}) \ell_e && (22.1.3) \\ &= \beta OPT && (22.1.4) \end{aligned}$$

■

Claim 22.1.5 *Given a solution of cost X in T, our solution in G will have cost $\leq X$.*

Proof:

$$cost_G = \sum_{(u,v)\in E_T} f(c_e^T) d_G(u, v) \leq \sum_{(u,v)\in E_T} f(c_e^T) d_T(u, v) = X$$

■

These claims prove the theorem. ■

Note that this analysis relied on the fact that our objective function is linear in lengths $\ell_e = d_G(u, v)$.

These strategies were originally developed by Bartal, who derived $O(\log^2 n)$ and $O(\log n \log \log n)$ probabilistic embeddings of graphs into distributions over trees [4] [5]. Fakcharoenphol, Rao, and Talwar later improved these results to a $O(\log n)$ probabilistic embedding, which was also shown to be tight [3].

22.1.6 $O(\log n)$-probabilistic embedding

How can be get an $O(\log n)$-probabilistic embedding of general graphs into trees? The basic idea is to do a hierarchical clustering on all vertices in G (Figure 22.1.2).

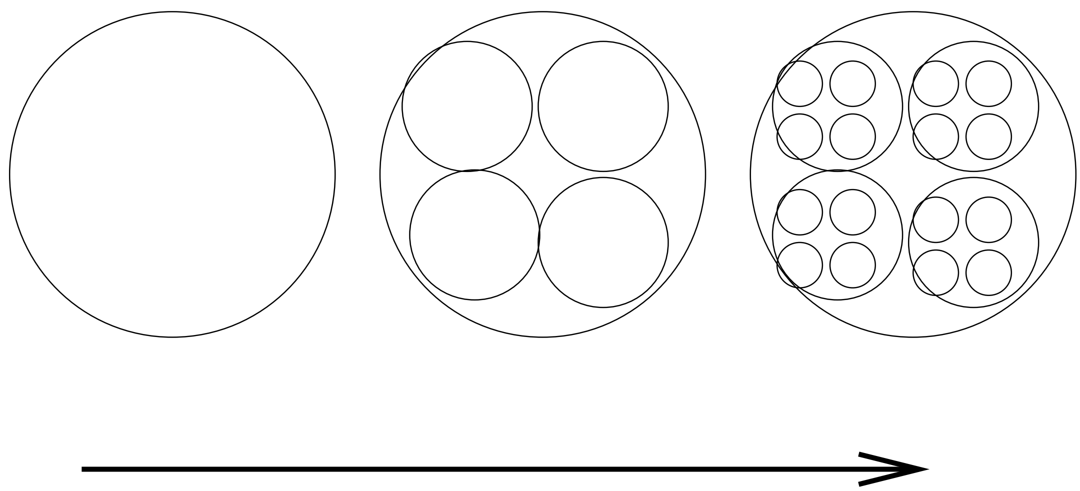

Figure 22.1.2: Hierarchical clustering by partitioning.

Under this scheme, all vertices of the original graph G are leaf nodes in our tree T. Each cluster in our hierarchical clustering then corresponds to a sub-tree in T. That is, all interior nodes of T are artifacts of our clustering scheme and were not originally present in G.

Starting from a graph with diameter Δ, we want our probabilistic embedding to have the property that $d_T(x, y) \geq d_G(x, y) \forall x, y$.

We can achieve this by building our clusters such that the diameter of the initial root cluster is Δ, the diameter of each child cluster is $\Delta/2$, and so on.

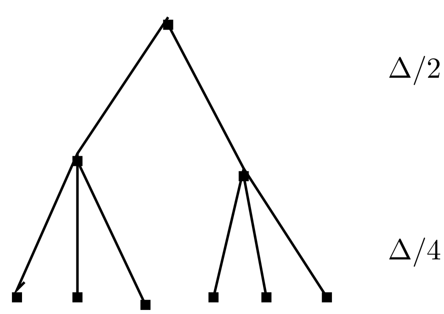

Figure 22.1.3: Tree representation of our hierarchical clustering.

Next, we need a partitioning scheme in order to build each level of our hierarchical clustering.

Definition 22.1.6 *A β* low-diameter low-distortion partitioning *with parameter δ is a partition of V into $\{V_1, V_2, ..., V_k\}$ such that*

1. *$diam(V_i) \leq \delta \ \forall i$*

2. $Pr[ecut] \leq \frac{d_e}{\delta}\beta \ \forall e \in E$

In this context, edge $e = (u, v)$ being "cut" by the partitioning means that $u \in V_i$ and $v \in V_j$ such that $i \neq j$.

Lemma 22.1.7 *Given a β low-diameter low-distortion partitioning scheme, there exists an $4\beta \log \Delta$-distortion probabilistic embedding of a graph into trees.*

Proof: Begin with a Δ-diameter graph. Partition with $\delta = \Delta/2$. Recursively embed $V_1, V_2, ..., V_k \hookrightarrow T_1, T_2, ..., T_k$.

Obviously, this approach satisfies properties 1 and 2 of Definition 22.1.6.

To see that it satisfies property 3, fix x, y and suppose that $e = (x, y)$ is an edge in E. Then suppose that they are separated at the top-level partitioning. Then $d_T(x, y) \leq 2\Delta$. If they are in the same subtree, then by induction

$$E_s[d_s(x, y)] \leq 4\beta \log \left(\frac{\Delta}{2}\right) d_G(x, y)$$

Since we must have 1 of these 2 cases (they are in the same subtree, or not), we can then calculate the expected distance between them in T. (Remember that for this first partition, $\delta = \Delta/2$.)

$$\begin{aligned}
E[d_T(x, y)] &= 2\Delta \left(\frac{d_e}{\delta}\beta\right) + 4\beta \log \left(\frac{\Delta}{2}\right) d_G(x, y) && (22.1.5)\\
&= 4\beta d_G(x, y) + 4\beta \log \left(\frac{\Delta}{2}\right) d_G(x, y) && (22.1.6)\\
&= 4\beta d_G(x, y)[1 + \log \left(\frac{\Delta}{2}\right)] && (22.1.7)\\
&= 4\beta d_G(x, y) \log \Delta && (22.1.8)\\
&= O(\beta \log \Delta) d_G(x, y) && (22.1.9)
\end{aligned}$$

Finally, it is easy to see that the worst distortion happens on edges present in the original graph G, so this bound will also hold for other pairs x, y.

■

The resulting tree is *k-hierarchically well-separated* (k-HST), with $k = 2$. This means that the edge weights between a parent and and all children are equal, and that the edge weights on any path from root to leaf decrease by a factor of at least k on each edge [4].

Our tree is also *ultrametric*, meaning that the root to leaf distances are the same for all leaves. Ultrametric trees have especially interesting applications in the construction of phylogenetic trees, which model the evolutionary relationships between the genomic sequences of different organisms [2].

The only missing component of our approach is now the β low-distortion low-diameter partition scheme, which will be introduced in the next lecture.

References

[1] C. Chekuri, M. T. Hajiaghayi, G. Kortsarz, M. R. Salavatipour. Approximation Algorithms for Non-Uniform Buy-at-Bulk Network Design. FOCS 2006, 677-686.

[2] Richard Durbin, Sean Eddy, Anders Krogh, and Graeme Mitchison. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, 1998.

[3] Jittat Fakcharoenphol, Satish Rao, Kunal Talwar. A Tight Bound on Approximating Arbitrary Metrics by Tree Metrics. STOC 2003, 448-455.

[4] Yair Bartal. Probabilistic Approximations of Metric Spaces and Its Algorithmic Applications. FOCS 1996, 184-193.

[5] Yair Bartal. On approximating arbitrary metrices by tree metrics. STOC 1998, 161-168.

[6] V. Vazirani. Approximation Algorithms. Springer, 2001.