15.1 Introduction

Recall the definition of a metric space:

Definition 15.1.1 *A pair (V, d) is a metric space if for all $u, v, w \in V$:*

1. $d(u, v) = 0 \iff u = v$
2. $d(u, v) = d(v, u)$
3. $d(u, v) \leq d(u, w) + d(w, v)$

Note that it is common to simply refer to the metric as d instead of the pair (V, d). We're going to be concerned with a special type of metric known as a *tree metric*.

Definition 15.1.2 *A* tree metric *(V', T) for a set of nodes V is a tree T on vertices V', where $V \subseteq V'$ are the leaves of T. Every edge of T has a nonnegative length.*

The distance in T between any two vertices $u, v \in V'$ is denoted $d_T(u, v)$, where the distance in T is the length of the unique $u - v$ path in T.

Definition 15.1.3 *Let (V, d) be a metric and (V', T) a tree metric for V. Then (V, d) embeds into T with distortion α if $d(u, v) \leq d_T(u, v) \leq \alpha \cdot d(u, v)$ for all $u, v \in V$.*

Intuitively, if we can embed (V, d) into some tree (V', T) with small distortion, then T is "like" the original metric space so we might hope that we can just solve any problem that we care about it on T instead of on the original metric. Unfortunately, this is not always possible: even simple metric spaces like the cycle C_n might require large distortion to embed into any tree. This is trivial to see if we required T to be a subtree of the input graph, but since we're not requiring that, this is a bit harder to prove. It is possible to show that C_n requires distortion at least $\frac{n-1}{8}$ to embed into any tree.

What can we do? Let's take inspiration from the cycle: there's no tree which allows small distortion, but if we fix some pair $u, v \in V$, then a *random* subtree of C_n is pretty good in expectation! For example, if u and v are adjacent in C_n, then with probability $1/n$ they get distance n, while with probability $\frac{n-1}{n}$ they're still at distance 1. So the expected distance is at most 2. So for any pair of nodes the expected distortion is small, even though once we instantiate some particular tree, there *will be* some pair which is badly distorted. As it turns out, though this kind of expected distortion is enough for many applications.

The best and provably optimal result for doing this is due to Fakcharoenphol, Rao, and Talwar, who proved the following theorem.

Theorem 15.1.4 ([FRT04]) *Let (V, d) be a metric. Then there is a randomized, polytime algorithm that produces a tree metric (V', T) for V such that*

1. $d(u,v) \le d_T(u,v)$ *for all* $u,v \in V$*, and*

2. $\mathbf{E}[d_T(u,v)] \le O(\log n) \cdot d(u,v)$ *for all* $u,v \in V$*.*

In other words, this theorem gives an embedding into a *distribution* of *dominating* trees (a distribution of trees each of which does not contract any pair). This algorithm is tight: there are metrics for which any embedding into a distribution of dominating trees requires distortion $\Omega(\log n)$. FRT is the best possible result of this form, but it built off of ideas pioneered by Bartal, who introduced the definition of probabilistic tree embeddings and gave one with $O(\log^2 n)$ distortion [Bar96], and then improved this to $O(\log n \log\log n)$ distortion [Bar98].

We're going to spend the next couple of classes proving this theorem and analyzing tree embeddings, but before we do that, let's show why they're useful. It's not hard to see that almost any problem which involves distances can be turned into a problem on trees by using this theorem and losing an extra $O(\log n)$ in the approximation ratio, but let's see this for a particular problem: Group Steiner Tree.

15.2 Group Steiner Tree on General Metrics

Recall the GST problem:

- **Input**:, A graph $G = (V,E)$, edge costs $c : E \to \mathbb{R}_{\ge 0}$, a root vertex $r \in V$, and groups $g_1, \dots, g_k \subseteq V$.
- **Feasible solution**: A tree T such that for all $i \in [k]$, there is some $v \in g_i$ such that T has a path between r and v.
- **Objective**: $\min \sum_{e \in T} c(e)$

We now know that Garg, Konjevod, and Ravi (GKR) gave an $O(\log n \log k)$-approximation when the input graph is a tree, and that the problem is $\Omega(\log^{2-\epsilon} n)$-hard to approximate even on trees. How can we design an approximation algorithm for general metrics? Use FRT to change the input into a tree!

Slightly more formally, consider the following algorithm:

1. Extend c to a metric space (V,d) where $d(u,v)$ is the minimum cost of any $u-v$ path.
2. Use FRT (Theorem 15.1.4) to embed (V,d) into a tree (V',T) with distortion $O(\log n)$. Note that since V are the leaves of T, all of the terminals (vertices in the groups) are now leaves.
3. Make a new group which is just $\{r\}$, and then use the GKR algorithm to get a subtree T' of T which is an $O(\log |V'| \log k)$-approximation to the optimal solution on T.
4. Shortcut T' to get a cycle C only on terminals.

5. Use C (with one arbitrary edge removed) as our solution in the metric space (V, d). To get a solution on G, replacing any edge of C which doesn't exist in G with a path of the same length.

This algorithm clearly gives a feasible solution: GKR returns a tree which connects at least one terminal from each group (including r) to the root of T, so C has r and at least one terminal from each group. Thus C gives a feasible solution. The algorithm also clearly takes only polynomial time. So we just need to analyze the approximation ratio.

Theorem 15.2.1 *This algorithm is a $O(\log^2 n \log k)$-approximation, i.e.,*

$$\mathbf{E}[c(C)] \leq O(\log^2 n \log k) \cdot OPT.$$

Proof: Let's set up some notation.

- Let S be the terminals connected by OPT (so $S \cap g_i \neq \emptyset$ for all $i \in [k]$)
- Let C_S be the cycle on S obtained by shortcutting OPT (so $c(C_S) \leq 2 \cdot OPT$).
- T will be the (random) tree built by FRT, with costs c_T or d_T.
- Let $OPT(T)$ denote the optimal solution in T.
- Let T_S be the subtree of T induced by S (i.e., the subtree of T which consists of all the paths from nodes in S up to the LCA of S.

Now we can actually prove the theorem.

$$\begin{aligned}
\mathbf{E}[c(C)] &\leq \mathbf{E}[c_T(C)] && \text{distances in } T \text{ are nondecreasing} \\
&\leq \mathbf{E}\big[2 \cdot c_T(T')\big] && \text{shortcutting costs at most a factor of 2} \\
&= 2 \cdot \mathbf{E}\big[c_T(T')\big] && \text{linearity of expectation} \\
&\leq 2\mathbf{E}[O(\log n \log k) \cdot c_T(OPT(T))] && \text{GKR} \\
&= O(\log n \log k) \cdot \mathbf{E}[c_T(OPT(T))] && \text{linearity of expectations} \\
&\leq O(\log n \log k) \cdot \mathbf{E}[c_T(T_S)] && \text{by definition of } OPT(T) \\
&\leq O(\log n \log k) \cdot \mathbf{E}[c_T(C_S)] && C_S \text{ a cycle on leaves of } T_S \\
&= O(\log n \log k) \cdot \mathbf{E}\left[\sum_{(u,v) \in C_S} d_T(u,v)\right] && \text{by definition} \\
&= O(\log n \log k) \cdot \sum_{(u,v) \in C_S} \mathbf{E}[d_T(u,v)] && \text{linearity of expectations} \\
&\leq O(\log n \log k) \cdot \sum_{(u,v) \in C_S} (O(\log n) \cdot d(u,v)) && \text{FRT} \\
&= O(\log^2 n \log k) \cdot \sum_{(u,v) \in C_S} d(u,v) && \text{linearity of expectations}
\end{aligned}$$

$$\le O(\log^2 n \log k) \cdot 2 \cdot OPT \qquad \text{shortcutting}$$
$$= O(\log^2 n \cdot \log k) \cdot OPT \qquad \text{asymptotic notation}$$

■

So combining FRT with GKR gives an $O(\log^2 n \log k)$-approximation to GST in general! This is still the state of the art. The question of whether this extra $\log n$ loss can be avoided is still an extremely important open question in approximation algorithms.

15.3 Metric Embeddings in General

We're not going to talk too much about general metric embeddings, but our approach for GST can be generalized to many other problems and other metrics. Let's see this a bit abstractly.

Definition 15.3.1 *(V, d) embeds into (V, d') with distortion α if $d(u, v) \le d'(u, v) \le \alpha d(u, v)$ for all $u, v \in V$.*

There are equivalent definitions based on contraction rather than expansion or on both, which are slightly more natural in some contexts, but this definition is more intuitive based on what we've been doing.

Now suppose that we have a β-approximation for some problem in d', but not in d. Then consider the algorithm which first embeds d into d' with distortion α, and then uses the β-approximation for d'. If the problem that we care about has costs which are just sums of distances (like many of the problems we've been thinking about), then we get that

$$\begin{aligned} c(ALG) &= \sum_{\{u,v\} \in ALG} d(u, v) \le \sum_{\{u,v\} \in ALG} d'(u, v) \le \beta \sum_{\{u,v\} \in OPT(d')} d'(u, v) \le \beta \sum_{\{u,v\} \in OPT} d'(u, v) \\ &\le \beta\alpha \sum_{\{u,v\} \in OPT} d(u, v) = \beta\alpha \cdot c(OPT) \end{aligned}$$

Handling probabilistic embeddings, like we did with FRT for GST, just involves putting expectations in the right places, but it all works out the same. So as long as our problem is "about" distances, we can use metric embeddings to transform the input metric into a "simpler" metric (like a tree) by paying the distortion in the approximation ratio.

15.4 The FRT Algorithm

We are now going to prove Theorem 15.1.4, the FRT theorem. We first give the algorithm in this section, and then analyze it in the next section.

15.4.1 Hierarchical Cut Decomposition

The first key idea, which is what FRT will actually construct, is (the tree corresponding to) a *hierarchical cut decomposition*. This is a special type of tree metric for V. For any $u \in V$ and $r \in \mathbb{R}_{\ge 0}$, let $B(u, r) = \{v \in V : d(u, v) \le r\}$ be the ball around u of radius r. Without loss of generality, we may assume (by scaling) that $\min_{u,v \in V : u \ne v} d(u, v) = 1$. For any set $S \subseteq V$, let

$\text{diam}(S) = \max_{u,v \in S} d(u,v)$ denote its diameter. Let $\Delta = 2^{\lceil \log \text{diam}(V) \rceil}$ be the smallest power of 2 such that $\Delta \geq \text{diam}(V)$.

Hierarchical Cut Decomposition: A tree metric (V', T) for (V, d) so that

1. Every vertex $\ell \in T$ is associated with the subset $S_\ell \subseteq V$ where $v \in S_\ell$ if and only if v is a descendent of ℓ in T. Note that this implies:
   - The root r of T has $S_r = V$.
   - If u has children $w_1, \ldots w_k$, then $\{S_{w_i}\}_{i \in [k]}$ partition S_u (i.e., $S_u = \cup_{i=1}^k S_{w_i}$ and $S_{w_i} \cap S_{w_j} = \emptyset$ for all $i \neq j$).
2. If u is at level i of T, then $\text{diam}(S_u) < 2^i$ (leaves at level 0, root at level $\log \Delta$).
3. The length of an edge between a level i node and a level $i+1$ node is 2^{i+1}.

This can be summed up with the following picture, which is directly from the textbook [WS11]:

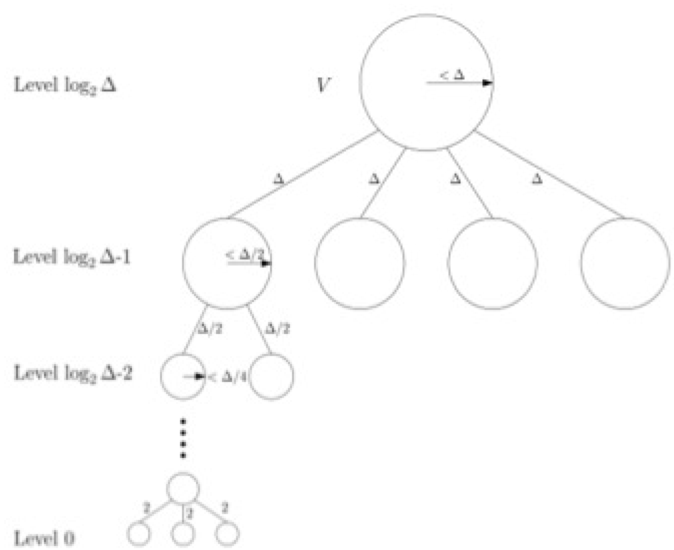

Figure 15.4.1: Hierarchical Decomposition

The FRT algorithm will construct a hierarchical cut decomposition, but before we give the algorithm, let's start by showing a simple lemma which holds for any hierarchical cut decomposition. Consider a hierarchical cut decomposition (V', T) of some metric (V, d).

Lemma 15.4.1 *If the least common ancestor of two leaf nodes u and v in T is at level i, then $d_T(u,v) \leq 2^{i+2}$. Furthermore, $d_T(u,v) \geq d(u,v)$ for all $u, v \in V$*

Proof: Let u and v be leaf nodes in T, and let w be u and v's least common ancestor (so w is at level i). Then by construction we know that $d_T(u,w) = \sum_{j=1}^{i} 2^j$, so $2^i \leq d_T(u,w) < 2^{i+1}$. Similarly,

$2^i \leq d_T(v, w) < 2^{i+1}$. Since $d_T(u, v) = d_T(u, w) + d_T(w, v)$, we get that $2^{i+1} \leq d_T(u, v) < 2^{i+2}$. That proves the first part of the lemma. And because u, v are both contained in S_w, we know that $d(u, v) \leq \mathrm{diam}(S_w) \leq 2^i$, which implies the second part. ■

15.4.2 Constructing the FRT tree

We can now finally give the FRT algorithm for constructing a tree embedding. FRT constructs a hierarchical decomposition in a certain way, but since it does construct a hierarchical decomposition, we know that no pair is contracted, and the distance between two nodes in the tree depends only on the level of their LCA. This is going to make reasoning about distances in the tree much easier.

(As a side note, you might have noticed that these trees are not just trees, they're special trees where the distance between two nodes grows exponentially with the level of their LCA. So we're actually doing more than just giving a tree embedding: we're giving a tree embedding into a special class of trees known as Hierarchically Well-Separated Trees (HSTs). Occasionally it is useful to utilize this property algorithmically: for GST we didn't care whether we were in a HST or a general tree, but for other problems it is sometimes easier to handle HSTs than general trees, and thanks to FRT we only need to handle HSTs).

Algorithm 1 FRT embedding

```
Let π be a permutation of V, chosen uniformly at random
Let r_0 be a value in [1/2, 1), chosen uniformly at random
Let r_i = r_0 · 2^i for all i such that 1 ≤ i ≤ log Δ
Let T be a tree with only a root node (at level log Δ) which represents V
for i ← log Δ − 1 to 0 do
  Let 𝒞 be the set of nodes at level i + 1
  for C ∈ 𝒞 do
    S ← C
    for j ← 1 to n do
      P ← B(π(j), r_{i−1}) ∩ S
      if P ≠ ∅ then
        S ← S \ P
        Add P to T as a child of C at level i
      end if
    end for
  end for
end for

return T
```

Note that there are two sources of randomness in this algorithm: the choice of π, and the choice of r_0.

15.5 Analysis of FRT

Since FRT gives a hierarchical cut decomposition we know that no distance is smaller in T than it is in the original metric. So we just need to prove that the expected expansion is at most $O(\log n)$, i.e., we want to prove the following theorem.

Theorem 15.5.1 $\mathbf{E}[d_T(u,v)] \leq O(\log n) d(u,v)$ *for all* $u, v \in V$

For the rest of this section, let's fix u and v. Let's introduce a couple definitions. Recall that $B(w,r)$ denotes the ball with center w and radius r.

Definition 15.5.2 w settles u, v *at level* i *if* w *is the first vertex in* π *s.t.* $B(w, r_{i-1}) \cap \{u, v\} \neq \emptyset$.

Definition 15.5.3 w cuts u, v *at level* i *if* $|B(w, r_{i-1}) \cap \{u, v\}| = 1$.

From these definitions we can make the following obvious observation. Recall that $LCA(u,v)$ is the least common ancestor of u and v, and we know from our previous analysis of hierarchical cut decompositions that the distance between u and v is essentially determined by their LCA.

Observation 15.5.4 $LCA(u,v)$ *is at level* $i+1$ *if* i *is the largest value such that the vertex* w *which settles* u, v *at level* i *also cuts* u, v *at level* i.

To analyze the expected distortion we'll need to analyze a few random variables:

$$S_{iw} = \begin{cases} 1, & \text{if } w \text{ settles } u, v \text{ at level } i, \\ 0, & \text{otherwise.} \end{cases}$$

$$X_{iw} = \begin{cases} 1, & \text{if } w \text{ cuts } u, v \text{ at level } i, \\ 0, & \text{otherwise.} \end{cases}$$

We can now start analyzing the expected distortion, although we'll have to stop a few places along the way to prove useful lemmas. Using our random variables, there is a vertex which both settles and cuts u, v at level i if $\sum_{w \in V} S_{iw} X_{iw} = 1$. Let i^* be the level of $LCA(u,v)$. Then using our observation, $i^* - 1$ is the largest i such that $\sum_{w \in V} S_{iw} X_{iw} = 1$. Moreover, we know from Lemma 15.4.1 that $d_T(u,v) \leq 2^{i^*+2}$. Putting this together and changing the order of summation, we get that

$$d_T(u,v) \leq 2^{i^*+2} = \max_{i : \sum_{w \in V} S_{iw} X_{iw} = 1} 2^{i+3} \leq \sum_{i=0}^{\log \Delta} 2^{i+3} \sum_{w \in V} S_{iw} X_{iw} = \sum_{w \in V} \sum_{i=0}^{\log \Delta} 2^{i+3} S_{iw} X_{iw}.$$

Now if we take the expectation, by using linearity of expectations and the definition of conditional probabilities, we get that

$$\begin{aligned} \mathbf{E}[d_T(u,v)] &\leq \sum_{w \in V} \sum_{i=0}^{\log \Delta} 2^{i+3} \mathbf{E}[S_{iw} X_{iw}] \\ &= \sum_{w \in V} \sum_{i=0}^{\log \Delta} 2^{i+3} \mathbf{Pr}[S_{iw} = 1 \wedge X_{iw} = 1] \end{aligned}$$

$$= \sum_{w \in V} \sum_{i=0}^{\log \Delta} 2^{i+3} \mathbf{Pr}[S_{iw} = 1 | X_{iw} = 1] \mathbf{Pr}[X_{iw} = 1]$$

So we've (very formally) broken this up into analyzing two events: that w cuts u, v at level i (which has nothing to do with π), and that w settles u, v at level i conditioned on it cutting u, v at level i. We're going to prove a few lemmas which let us analyze these events, but consider the following intuition. w cutting u, v is independent of π: it only has to do with r_0. On the other hand, if we assume that w does cut u, v, then whether it also settles depends on π (and on r_0). So the hope is that this will be easier to analyze since we've removed the dependence on π from one of them.

The first lemma gives us a bound on the conditional event.

Lemma 15.5.5 *For every vertex w there is some $b_w \in \mathbb{R}_{\geq 0}$ such that:*

1. $\mathbf{Pr}[S_{iw} = 1 | X_{iw} = 1] \leq b_w$ *for all i, and*
2. $\sum_{w \in V} b_w \leq O(\log n)$.

The second lemma gives us a bound on the cutting probability.

Lemma 15.5.6 $\sum_{i=0}^{\log \Delta} 2^{i+3} \mathbf{Pr}[X_{iw} = 1] \leq 32 d(u, v)$ *for all $w \in V$.*

Let's now finish the proof of the main theorem, assuming these two lemmas. Continuing from our previous inequalities:

$$\begin{aligned}
\mathbf{E}[d_T(u, v)] &\leq \sum_{w \in V} \sum_{i=0}^{\log \Delta} 2^{i+3} \mathbf{Pr}[S_{iw} = 1 | X_{iw} = 1] \mathbf{Pr}[X_{iw} = 1] \\
&\leq \sum_{w \in V} b_w \sum_{i=0}^{\log \Delta} 2^{i+3} \mathbf{Pr}[X_{iw} = 1] \\
&\leq \sum_{w \in V} b_w 32 d(u, v) = 32 d(u, v) \sum_{w \in V} b_w \\
&\leq O(\log n) d(u, v)
\end{aligned}$$

So now we just need to prove these two lemmas!

Proof of Lemma 15.5.5: We're trying to analyze $\mathbf{Pr}[S_{iw} = 1 | X_{iw} = 1]$ for every $w \in V$. To do this, let's order V by distance to $\{u, v\}$, so

$$d(w_i, \{u, v\}) \leq d(w_{i+1}, \{u, v\})$$

for all i.

Now let's fix some w_j, and suppose that w_j cuts $\{u, v\}$ at level i, i.e., $|B(w_j, r_{i-1}) \cap \{u, v\}| = 1$. Then by the definition of our ordering, every w_k with $k < j$ must have $|B(w_k, r_{i-1}) \cap \{u, v\}| > 0$. Thus if *any* of these nodes come before w_j in π, we know that w_j will not settle u, v at level i, since at least one of u, v will have already been clustered by the time w_j gets to form clusters. Since π is a random permutation, the probability that w_j comes before the x_k for all $k < j$ is exactly $1/j$. Thus $\mathbf{Pr}\big[S_{iw_j} = 1 | X_{iw_j} = 1\big] \le 1/j$. So by setting $b_{w_j} = 1/j$, we have proved the first part of the lemma.

The proof of the second part of the lemma is now straightforward:

$$\sum_{w \in V} b_w = \sum_{j=1}^{n} b_{w_j} = \sum_{j=1}^{n} \frac{1}{j} = H_n = O(\log n),$$

as claimed. ■

Proof of Lemma 15.5.6: Now we're trying to prove that $\sum_{i=0}^{\log \Delta} 2^{i+3} \mathbf{Pr}[X_{iw} = 1] \le 16 d(u, v)$ for all $w \in V$. Without loss of generality, let's assume that $d(w, u) \le d(w, v)$. In order for w to cut u, v at level i (i.e., for $X_{iw} = 1$), it needs to be the case that $r_{i-1} \in [d(w, u), d(w, v))$. Moreover, r_{i-1} is distributed uniformly in $[2^{i-2}, 2^{i-1})$. Thus

$$\mathbf{Pr}[X_{iw} = 1] = \frac{|[2^{i-2}, 2^{i-1}) \cap [d(w, u), d(w, v))|}{|[2^{i-2}, 2^{i-1})|} = \frac{|[2^{i-2}, 2^{i-1}) \cap [d(w, u), d(w, v))|}{2^{i-2}}.$$

So we have that

$$\begin{aligned} 2^{i+3} \mathbf{Pr}[X_{iw} = 1] &= \frac{2^{i+3}}{2^{i-2}} |[2^{i-2}, 2^{i-1}) \cap [d(w, u), d(w, v))| \\ &= 32 |[2^{i-2}, 2^{i-1}) \cap [d(w, u), d(w, v))|. \end{aligned}$$

Thus

$$\begin{aligned} \sum_{i=0}^{\log \Delta} 2^{i+3} \mathbf{Pr}[X_{iw} = 1] &\le \sum_{i=0}^{\log \Delta} 32 |[2^{i-2}, 2^{i-1}) \cap [d(w, u), d(w, v)]| \\ &= 32 |[d(w, u), d(w, v)]| = 32(d(w, v) - d(w, u)) \le 32 d(u, v), \end{aligned}$$

where the final inequality is from the triangle inequality. ■

15.6 Steiner Point Removal

Recall that our definition of a tree embedding for (V, d) involved us creating a tree where V was the leaves. A natural question is whether this is actually necessary: can we probabilistically embed into trees on V itself (so without any "extra" nodes)? Or even more basically, forgetting the probabilistic embedding:

Question 15.6.1 *If (V', T') is a tree metric for V, is there a (weighted) tree $T = (V, E)$ such that $d_{T'}(u, v) \le d_T(u, v) \le \alpha d_{T'}(u, v)$ for all $u, v \in V$, where $\alpha = O(1)$?*

This question asks whether we can turn any tree metric which uses steiner nodes ("extra" nodes) into a tree without any steiner nodes. This question was resolved in a seminal paper by Anupam Gupta [Gup01], who showed that this was possible with $\alpha = 8$. Today we're going to prove an easier result which only holds for the kinds of tree embeddings that we construct, i.e., for hierarchical cut decompositions.

Theorem 15.6.2 *If (V', T') is a tree embedding for V which is a hierarchical cut decomposition, then can find some other T with vertex set V s.t. $d_{T'}(u,v) \le d_T(u,v) \le 4d_{T'}(u,v)$ for all $u, v \in V$.*

Proof: Use the following algorithm to construct T.

1. While there exists a node $x \in V$, s.t. $p(x) \notin V$, contract $(x, p(x))$. This gives a tree with vertex set V.

2. Multiply all edge weights by 4.

Here contracting edge $(x, p(x))$ means we just merge the subtree at x into $p(x)$ and identify the newly merged node as x. Contracting makes distance go down, and hence $d_T(u,v) \le 4d_{T'}(u,v)$.

Suppose the least common ancestor of u, v in T' is w at level i. Then $d_{T'}(u,v) \le 2^{i+2}$. After contractions, their distance in T is at least 2^i (consider w and its child). So $d_T(u,v) \ge 2^{i+2}$ as we multiply each edge weights by 4. So $d_{T'}(u,v) \le d_T(u,v)$. ■

References

[Bar96] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *Proceedings of 37th Conference on Foundations of Computer Science*, FOCS '96, pages 184–193, 1996.

[Bar98] Yair Bartal. On approximating arbitrary metrices by tree metrics. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pages 161–168, 1998.

[FRT04] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *Journal of Computer and System Sciences*, 69(3):485 – 497, 2004. Special Issue on STOC 2003.

[Gup01] Anupam Gupta. Steiner points in tree metrics don't (really) help. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '01, pages 220–227, 2001.

[WS11] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, New York, NY, USA, 1st edition, 2011.