Advanced Algorithms	23 October 2018

Lecture 6: Approximation Algorithms VI

Lecturer: Mohsen Ghaffari *Scribe: Davin Choo*

1 Probabilistic tree embedding

Trees are a special kind of graphs without cycles and some **NP**-hard problems are known to admit exact polynomial time solutions on trees. Motivated by existence of efficient algorithms on trees, one hopes to design the following framework for a general graph $G = (V, E)$ with distance metric $d_G(u, v)$ between vertices $u, v \in V$:

1. Construct a tree T
2. Solve the problem on T efficiently
3. Map the solution back to G
4. Argue that transformed solution from T is a good approximation to G.

Ideally, we want to build a tree T such that (i) $d_G(u, v) \leq d_T(u, v)$ and (ii) $d_T(u, v) \leq c \cdot d_G(u, v)$, where c is the *stretch of tree embedding*. Unfortunately, such a construction is hopeless[1]. Instead, we consider a *probabilistic tree embedding* of G into a collection of trees $\mathcal{T} = \{T_1, \ldots, T_m\}$ such that

- (Over-estimates cost): $\forall T \in \mathcal{T}, d_G(u, v) \leq d_T(u, v)$
- (Expected over-estimation is not too much): $\forall T \in \mathcal{T}, \mathbb{E}_{T \in \mathcal{T}}[d_T(u, v) \leq c \cdot d_G(u, v)]$
- ($\mathcal{T}$ is a probability space): $\sum_{i=1}^{m} \Pr[T_i] = 1$

Bartal [Bar96] gave a construction[2] for probabilistic tree embedding with poly-logarithmic stretch factor c, and proved[3] that a stretch factor $c = \Omega(\log n)$ is required for general graphs. A construction that yields $c = \mathcal{O}(\log n)$, in expectation, was subsequently found by [FRT03].

2 A tight probabilistic tree embedding construction

In this section, we describe a probabilistic tree embedding construction due to [FRT03] with a stretch factor $c = \mathcal{O}(\log n)$. For a graph $G = (V, E)$, let $D = diam(G)$ and distance metric $d_G(u, v)$ denote the distance between two vertices $u, v \in V$. Denote $B(v, r) := \{u \in V : d_G(u, v) \leq r\}$ as the ball of distance r around vertex v, including v.

2.1 Idea: Ball carving

Before we present the actual construction, we argue that the following *ball carving* approach will yield a probabilistic tree embedding.

Definition 1 (Ball carving)**.** *Given a graph $G = (V, E)$, partition V into $V_1, \ldots, V_l$ such that*

(A) $\forall i \in \{1, \ldots, l\}, diam(V_i) \leq \frac{D}{2}$

(B) $\forall u, v \in V$, $\Pr[$*u and v not in same partition*$] \leq \alpha \cdot \frac{d_G(u,v)}{D}$, *for some* α

Using ball carving, Algorithm 1 recursively partitions the vertices of a given graph until there is only one vertex remaining. Figure 1 illustrates the process of building a tree T from a given graph G.

Lemma 2. *For any two vertices $u, v \in V$, if T separates them at level i, then* $\frac{2D}{2^i} \leq d_T(u, v) \leq \frac{4D}{2^i}$.

[1] For a cycle G with n vertices, the excluded edge in a constructed tree will cause the stretch factor $c \geq n - 1$.
[2] Theorem 8 in [Bar96]
[3] Theorem 9 in [Bar96]

Algorithm 1 ConstructT($G = (V, E)$ with diameter D)

if $|V| = 1$ **then**
  return V
else
  $V_1, \dots, V_l \leftarrow$ BallCarving(G)
  Create auxiliary vertex r ▷ r is root of current subtree
  for $i \in \{1, \dots, l\}$ **do**
    $G[V_i] \leftarrow$ Subgraph induced by vertices V_i ▷ By BallCarving(G), $diam(G[V_i]) \leq \frac{D}{2}$
    $r_i \leftarrow$ ConstructT($G[V_i]$) ▷ Either an auxiliary vertex or an actual vertex $v \in V(G)$
    Add edge (r, r_i) with weight D
  end for
  return Root of subtree r ▷ r is the root of the constructed T
end if

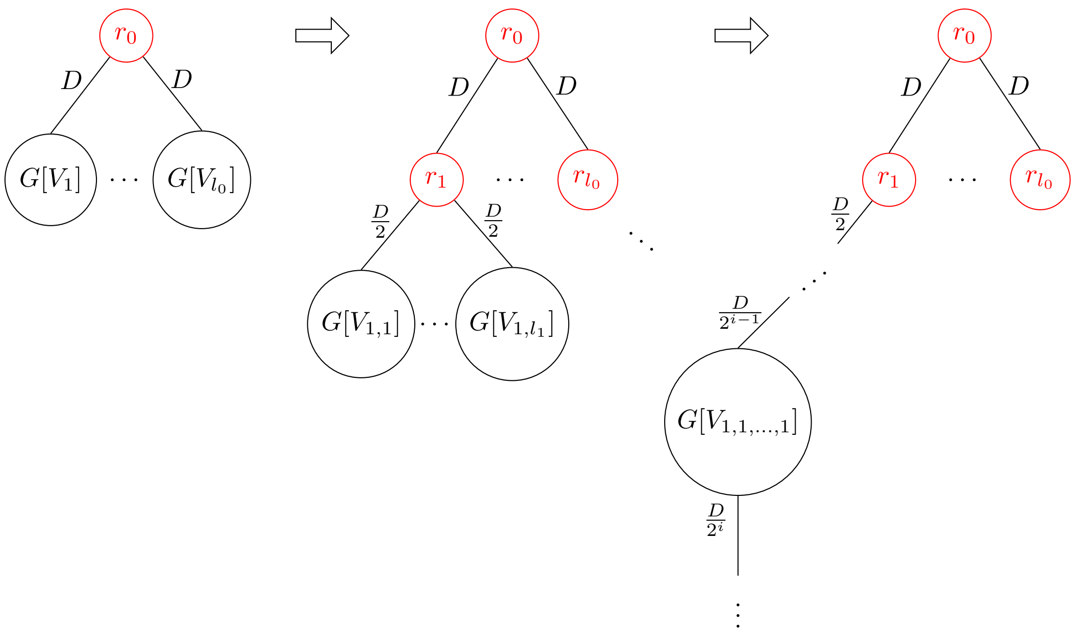
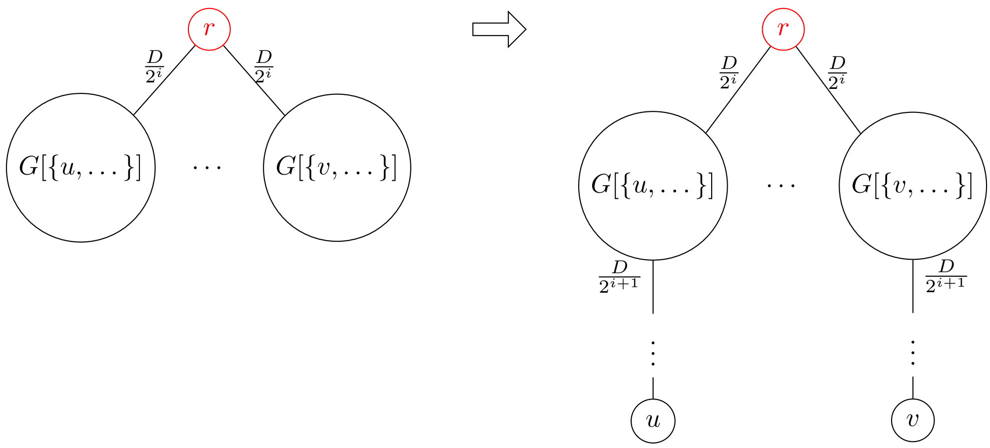


Figure 1: Recursive ball carving with $\lceil \log_2(D) \rceil$ levels. Red vertices are auxiliary nodes that are not in the original graph G. Denoting the root as the 0^{th} level, edges from level i to level $i + 1$ have weight $\frac{D}{2^i}$.

Proof. If T splits u and v at level i, then $d_T(u, v)$ has to include two edges of length $\frac{D}{2^i}$, hence $d_T(u, v) \geq \frac{2D}{2^i}$. To be precise, $d_T(u, v) = 2 \cdot (\frac{D}{2^i} + \frac{D}{2^{i+1}} + \cdots) \leq \frac{4D}{2^i}$. See picture.



□

Remark If $u, v \in V$ separate *before* level i, then $d_T(u,v)$ must still include the two edges of length $\frac{D}{2^i}$, hence $d_T(u,v) \geq \frac{2D}{2^i}$.

Claim 3. *Construct*T(G) *returns a tree* T *such that* $d_G(u,v) \leq d_T(u,v)$.

Proof. Consider $u, v \in V$. Say $\frac{D}{2^i} \leq d_G(u,v) \leq \frac{D}{2^{i-1}}$ for some $i \in \mathbb{N}$. By property (A) of ball carving, T will separate them at, or before, level i. By Lemma 2, $d_T(u,v) \geq \frac{2D}{2^i} = \frac{D}{2^{i-1}} \geq d_G(u,v)$. ☐

Claim 4. *Construct*T(G) *returns a tree* T *such that* $\mathbb{E}[d_T(u,v)] \leq 4\alpha \log(D) \cdot d_G(u,v)$.

Proof. Consider $u, v \in V$. Define $\mathcal{E}_i$ as the event that "vertices u and v get separated at the i^{th} level", for $i \in \mathbb{N}$. By recursive nature of ConstructT, a graph at the i^{th} level has diameter $\leq \frac{D}{2^i}$. So, property (B) of ball carving tells us that $\Pr[\mathcal{E}_i] \leq \alpha \cdot \frac{d_G(u,v)}{D/2^i}$. Then,

$$\begin{aligned}
\mathbb{E}[d_T(u,v)] &= \textstyle\sum_{i=0}^{\log(D)-1} \Pr[\mathcal{E}_i] \cdot [d_T(u,v), \text{ given } \mathcal{E}_i] && \text{Definition of expectation} \\
&\leq \textstyle\sum_{i=0}^{\log(D)-1} \Pr[\mathcal{E}_i] \cdot \frac{4D}{2^i} && \text{By Lemma 2} \\
&\leq \textstyle\sum_{i=0}^{\log(D)-1} (\alpha \cdot \frac{d_G(u,v)}{D/2^i}) \cdot \frac{4D}{2^i} && \text{Property (B) of ball carving} \\
&= 4\alpha \log(D) \cdot d_G(u,v) && \text{Simplifying}
\end{aligned}$$

☐

2.2 Ball carving construction

We now give a concrete construction of ball carving that satisfies properties (A) and (B) as defined.

Algorithm 2 BallCarving$(G = (V, E)$ with diameter $D)$

```
if |V| = 1 then
    return V
else                                                        ▷ Say there are n vertices, where n > 1
    θ ← Uniform random value from the range [D/8, D/4]
    Pick a random permutation π on V                        ▷ Denote π_i as the i^th vertex in π
    for i ∈ {1, ..., n} do
        V_i ← B(π_i, θ) \ ⋃_{j=1}^{i-1} B(π_j, θ)           ▷ This ensures that V_1, ..., V_n is a partition of V
    end for
    return Non-empty sets V_1, ..., V_l          ▷ V_i = ∅ when vertices in B(π_i, θ) exist in earlier balls
end if                                           ▷ i.e. V_i = ∅ ⟺ ∀v ∈ B(π_i, θ), [∃j < i, v ∈ B(π_j, θ)]
```

Notation For vertex $v \in V$, let us denote $\pi(v)$ as v's position in π. That is, $v = \pi_{\pi(v)}$.

Claim 5. BallCarving(G) *returns partition* $V_1, \ldots, V_l$ *such that* $\forall i \in \{1, \ldots, l\}, diam(V_i) \leq \frac{D}{2}$

Proof. Since $\theta \in [\frac{D}{8}, \frac{D}{4}]$, all constructed balls have diameter smaller than $\frac{D}{2}$. ☐

Definition 6 (Ball cut). *A ball* $B(u,r)$ *is cut if* BallCarving *puts the vertices in* $B(u,r)$ *in different partitions of* $V_1, \ldots, V_l$. *We say* V_i *cuts* $B(u,r)$ *if* $i = argmin_{i \in [l]}[V_i \cap B(u,r) \neq \emptyset$ *and* $B(u,r) \not\subseteq V_i]$.

Lemma 7. *For any vertex* $u \in V$ *and radius* $r \in \mathbb{R}^+$, $\Pr[B(u,r)$ *is cut in* BallCarving$] \leq \mathcal{O}(\log n) \cdot \frac{r}{D}$.

Proof. Let θ be the randomly chosen in BallCarving. Consider an ordering of vertices in increasing distance from u: $v_1, v_2, \ldots, v_n$, such that $d_G(u, v_1) \leq d_G(u, v_2) \leq \cdots \leq d_G(u, v_n)$. For $j < i$, since $d_G(u, v_j) \leq d_G(u, v_i)$, if $B(v_i, \theta) \cap B(u,r) \neq \emptyset$, then $B(v_j, \theta) \cap B(u,r) \neq \emptyset$. So,

$$\begin{aligned}
\Pr[B(u,r) \text{ is cut}] &= \Pr[\textstyle\bigcup_{i=1}^n \text{Event that } B(v_i,\theta) \text{ cuts } B(u,r)] \\
&\leq \textstyle\sum_{i=1}^n \Pr[B(v_i,\theta) \text{ cuts } B(u,r)] && \text{Union bound} \\
&= \textstyle\sum_{i=1}^n \Pr[\pi(v_i) < \min_{j \in [i-1]}\{\pi(v_j)\}] \Pr[V_i \text{ cuts } B(u,r)] && \text{Require } v_i \text{ to appear first} \\
&= \textstyle\sum_{i=1}^n (1/i) \cdot \Pr[V_i \text{ cuts } B(u,r)] && \text{By random permutation } \pi \\
&\leq \textstyle\sum_{i=1}^n (1/i) \cdot \frac{2r}{D/8} && diam(B(u,r)) \leq 2r,\ \theta \in [\tfrac{D}{8}, \tfrac{D}{4}] \\
&= 16\frac{r}{D} H_n && H_n = \textstyle\sum_{i=1}^n \frac{1}{i} \\
&\in \mathcal{O}(\log(n)) \cdot \frac{r}{D}
\end{aligned}$$

In the last inequality: For V_i to cut $B(u,r)$, we need $\theta \in (d_G(u,v_i) - r, d_G(u,v_i) + r)$, hence the numerator of $\leq 2r$; The denominator $\frac{D}{8}$ is because the range of values that θ is sampled from is $\frac{D}{4} - \frac{D}{8} = \frac{D}{8}$. ☐

Claim 8. BALLCARVING(G) *returns partition* $V_1, \dots, V_l$ *such that*

$$\forall u, v \in V, \Pr[u \textit{ and } v \textit{ not in same partition}] \le \alpha \cdot \frac{d_G(u,v)}{D}$$

Proof. Let $r = d_G(u,v)$, then v is on the boundary of $B(u,r)$.

$$\begin{aligned} \Pr[u \text{ and } v \text{ not in same partition}] &\le \Pr[B(u,r) \text{ is cut in BALLCARVING}] \\ &\le \mathcal{O}(\log n) \cdot \tfrac{r}{D} && \text{By Lemma 7} \\ &= \mathcal{O}(\log n) \cdot \tfrac{d_G(u,v)}{D} && \text{Since } r = d_G(u,v) \end{aligned}$$

Note: $\alpha = \mathcal{O}(\log n)$. □

If we apply Claim 8 with Claim 4, we get $\mathbb{E}[d_T(u,v)] \le \mathcal{O}(\log(n)\log(D)) \cdot d_G(u,v)$. To remove the $\log(D)$ factor, so that stretch factor $c = \mathcal{O}(\log n)$, a tighter analysis is needed by only considering vertices that may cut $B(u, d_G(u,v))$ instead of all n vertices. For details, see Theorem 16 in the appendix.

2.3 Contraction of T

Notice in Figure 1 that we introduce auxiliary vertices in our tree construction and wonder if we can build a T without additional vertices (i.e. $V(T) = V(G)$. In this section, we look at CONTRACT which performs *tree contractions* to remove the auxiliary vertices. It remains to show that the produced tree that still preserves desirable properties of a tree embedding.

Algorithm 3 CONTRACT(T)

```
while T has an edge (u, w) such that u ∈ V and w is an auxiliary node do
    Contract edge (u, w) by merging subtree rooted at u into w, and identifying the new node as u
end while
Multiply weight of every edge by 4
return Modified T'
```

Claim 9. CONTRACT *returns a tree* T *such that* $d_T(u,v) \le d_{T'}(u,v) \le 4 \cdot d_T(u,v)$.

Proof. Suppose auxiliary node w, at level i, is the closest common ancestor for two arbitrary vertices $u, v \in V$ in the original tree T. Then, $d_T(u,v) = d_T(u,w) + d_T(w,v) = 2 \cdot (\sum_{j=i}^{\log D} \frac{D}{2^j}) \le 4 \cdot \frac{D}{2^i}$. Since we do not contract actual vertices, at least one of the (u,w) or (v,w) edges of weight $\frac{D}{2^i}$ will remain. Multiplying the weights of all remaining edges by 4, we get $d_T(u,v) \le 4 \cdot \frac{D}{2^i} = d_{T'}(u,v)$.

Suppose we only multiply the weights of $d_T(u,v)$ by 4, then $d_{T'}(u,v) = 4d_T(u,v)$. Since we contract edges, $d'_T(u,v)$ can only decrease, so $d_{T'}(u,v) \le 4d_T(u,v)$. □

Remark Claim 9 tells us that one can construct a tree T' without auxiliary variables by incurring an additional constant factor overhead.

3 Application: Buy-at-bulk network design

Definition 10 (Buy-at-bulk network design problem). *Consider a graph* $G = (V, E)$ *with edge lengths* l_e *for* $e \in E$. *Let* $f : \mathbb{R}^+ \to \mathbb{R}^+$ *be a sub-additive cost function. That is,* $f(x+y) \le f(x) + f(y)$. *Given* k *commodity triplets* (s_i, t_i, d_i), *where* $s_i \in V$ *is the source,* $t_i \in V$ *is the target, and* $d_i \ge 0$ *is the demand for the* i^{th} *commodity, find a capacity assignment on edges* $c_e (\forall e \in E)$ *such that*

- $\sum_{e \in E} f(c_e) \cdot l_e$ *is minimized*
- $\forall e \in E, c_e \ge$ *Total flow passing through it*
- *Flow conservation is satisfied and every commodity's demand is met*

Algorithm 4 NETWORKDESIGN($G = (V, E)$)

```
c_e = 0, ∀e ∈ E                                        ▷ Initialize capacities
T ← CONSTRUCTT(G)                     ▷ Build probabilistic tree embedding T of G
T ← CONTRACT(T)                                  ▷ V(T) = V(G) after contraction
for i ∈ {1, . . . , k} do                                 ▷ Solve problem on T
    P^T_{s_i,t_i} ← Find shortest s_i − t_i path in T       ▷ It is unique in a tree
    for Edge (u, v) of P^T_{s_i,t_i} in T do
        P^G_{u,v} ← Find shortest u − v path in G
        c_e ← c_e + d_i, for each edge in e ∈ P^G_{u,v}
    end for
end for
return {e ∈ E : c_e}
```

Remark If f is linear (e.g. $f(x + y) = f(x) + f(y)$), one can obtain an optimum solution by finding the shortest path $s_i \to t_i$ for each commodity i, then summing up the required capacities for each edge.

Let us denote $I = (G, f, \{s_i, t_i, d_i\}_{i=1}^k)$ as the given instance. Let $OPT_G(I)$ be the optimal solution on G and $A_T(I)$ be the solution produced by NETWORKDESIGN. Denote the costs as $|OPT_G(I)|$ and $|A_T(I)|$ respectively. We now compare the solutions $OPT_G(I)$ and $A_T(I)$ by comparing edge costs $(u, v) \in E$ in G and tree embedding T.

Claim 11. *$|A_T(I)|$ using edges in $G \leq |A_T(I)|$ using edges in T.*

Proof. (Sketch) For any pair of vertices $u, v \in V$, $d_G(u, v) \leq d_T(u, v)$. □

Claim 12. *$|A_T(I)|$ using edges in $T \leq |OPT_G(I)|$ using edges in T.*

Proof. (Sketch) Since shortest path in a tree is unique, $A_T(I)$ is optimum for T. So, any other flow assignment has to incur higher edge capacities. □

Claim 13. $\mathbb{E}[|OPT_G(I)|$ *using edges in* $T] \leq \mathcal{O}(\log n) \cdot |OPT_G(I)|$

Proof. (Sketch) T stretches edges by at most a factor of $\mathcal{O}(\log n)$. □

By the three claims above, NETWORKDESIGN gives a $\mathcal{O}(\log n)$-approximation to the buy-at-bulk network design problem, in expectation. For details, refer to Section 8.6 in [WS11].

References

[Bar96] Yair Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*, pages 184–193. IEEE, 1996.

[FRT03] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 448–455. ACM, 2003.

[WS11] David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.

A Ball carving with $\mathcal{O}(\log n)$ stretch factor

If we apply Claim 8 with Claim 4, we get $\mathbb{E}[d_T(u, v)] \leq \mathcal{O}(\log(n) \log(D)) \cdot d_G(u, v)$. To remove the $\log(D)$ factor, so that stretch factor $c = \mathcal{O}(\log n)$, a tighter analysis is needed by only considering vertices that may cut $B(u, d_G(u, v))$ instead of all n vertices.

A.1 Tighter analysis of ball carving

Fix arbitrary vertices u and v. Let $r = d_G(u,v)$. Recall that θ is chosen uniformly at random from the range $[\frac{D}{8}, \frac{D}{4}]$. A ball $B(v_i, \theta)$ can cut $B(u,r)$ only when $d_G(u,v_i) - r \leq \theta \leq d_G(u,v_i) + r$. In other words, one only needs to consider vertices v_i such that $\frac{D}{8} - r \leq \theta - r \leq d_G(u,v_i) \leq \theta + r \leq \frac{D}{4} + r$.

Lemma 14. *For $i \in \mathbb{N}$, if $r > \frac{D}{16}$, then $\Pr[B(u,r) \text{ is cut}] \leq \frac{16r}{D}$*

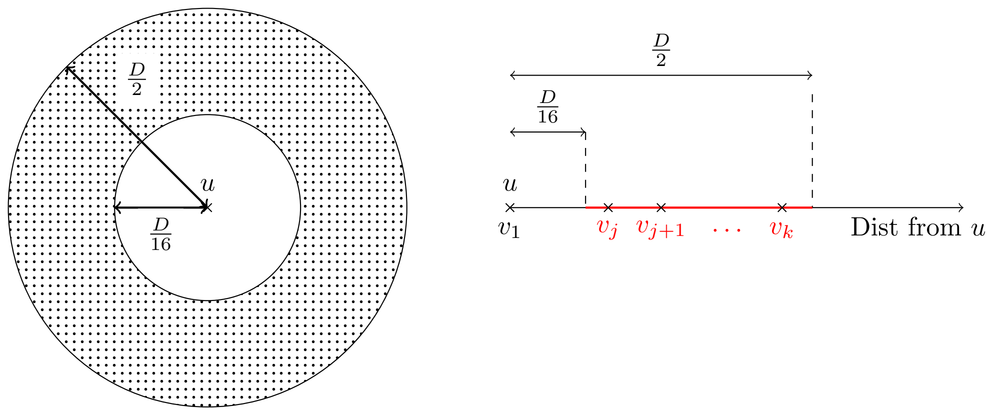
Proof. If $r > \frac{D}{16}$, then $\frac{16r}{D} > 1$. As $\Pr[B(u,r) \text{ is cut at level } i]$ is a probability ≤ 1, the claim holds. □

Remark Although lemma 14 is not a very useful inequality per se (since any probability ≤ 1), we use it to partition the value range of r so that we can say something stronger in the next lemma.

Lemma 15. *For $i \in \mathbb{N}$, if $r \leq \frac{D}{16}$, then*

$$\Pr[B(u,r) \text{ is cut}] \leq \frac{r}{D}\mathcal{O}(\log(\frac{|B(u,D/2)|}{|B(u,D/16)|}))$$

Proof. Since $B(v_i,\theta)$ cuts $B(u,r)$ only if $\frac{D}{8} - r \leq d_G(u,v_i) \leq \frac{D}{4} + r$, we have $d_G(u,v_i) \in [\frac{D}{16}, \frac{5D}{16}] \subseteq [\frac{D}{16}, \frac{D}{2}]$.



Suppose we arrange the vertices in ascending order of distance from u: $u = v_1, v_2, \ldots, v_n$. Denote:

- $j - 1 = |B(u, \frac{D}{16})|$ as the number of nodes that have distance $\leq \frac{D}{16}$ from u
- $k = |B(u, \frac{D}{2})|$ as the number of nodes that have distance $\leq \frac{D}{2}$ from u

We see that only vertices $v_j, v_{j+1}, \ldots, v_k$ have distances from u in the range $[\frac{D}{16}, \frac{D}{2}]$. Pictorially, only vertices in the shaded region could possibly cut $B(u,r)$. As before, let $\pi(v)$ be the ordering in which vertex v appears in random permutation π. Then,

$$\begin{array}{lll}
 & \Pr[B(u,r) \text{ is cut}] & \\
= & \Pr[\bigcup_{i=j}^{k} \text{Event that } B(v_i,\theta) \text{ cuts } B(u,r)] & \text{Only } v_j, v_{j+1}, \ldots, v_k \text{ can cut} \\
\leq & \sum_{i=j}^{k} \Pr[\pi(v_i) < min_{z<[i-1]}\{\pi(v_z)\}] \cdot \Pr[v_i \text{ cuts } B(u,r)] & \text{Union bound} \\
= & \sum_{i=j}^{k} \frac{1}{i} \cdot \Pr[B(v_i,\theta) \text{ cuts } B(u,r)] & \text{By random permutation } \pi \\
\leq & \sum_{i=j}^{k} \frac{1}{i} \cdot \frac{2r}{D/8} & diam(B(u,r)) \leq 2r, \theta \in [\frac{D}{8}, \frac{D}{4}] \\
= & \frac{r}{D}(H_k - H_j) & \text{where } H_k = \sum_{i=1}^{k} \frac{1}{i} \\
\in & \frac{r}{D}\mathcal{O}(\log(\frac{|B(u,D/2)|}{|B(u,D/16)|})) & \text{since } H_k \in \Theta(\log(k))
\end{array}$$

□

A.2 Plugging into ConstructT

Recall that ConstructT is a recursive algorithm which handles graphs of diameter $\leq \frac{D}{2^i}$ at each level. For a given pair of vertices u and v, there exists $i^* \in \mathbb{N}$ such that $\frac{D}{2^{i^*}} \leq r = d_G(u,v) \leq \frac{D}{2^{i^*-1}}$. In other words, $\frac{D}{2^{i^*-4}}\frac{1}{16} \leq r \leq \frac{D}{2^{i^*-5}}\frac{1}{16}$. So, lemma 15 applies for levels $i \in [0, i^* - 5]$ and lemma 14 applies for levels $i \in [i^* - 4, \log(D) - 1]$.

Theorem 16. $\mathbb{E}[d_T(u,v)] \in \mathcal{O}(\log n) \cdot d_G(u,v)$

Proof. As before, let $\mathcal{E}_i$ be the event that "vertices u and v get separated at the i^{th} level". For $\mathcal{E}_i$ to happen, the ball $B(u,r) = B(u, d_G(u,v))$ must be cut at level i, so $\Pr[\mathcal{E}_i] \leq \Pr[B(u,r)$ is cut at level $i]$.

$$
\begin{array}{lll}
 & \mathbb{E}[d_T(u,v)] & \\
= & \sum_{i=0}^{\log(D)-1} \Pr[\mathcal{E}_i] \cdot [d_T(u,v), \text{ given } \mathcal{E}_i] & \text{Definition of expectation} \\
\leq & \sum_{i=0}^{\log(D)-1} \Pr[\mathcal{E}_i] \cdot \frac{4D}{2^i} & \text{By Lemma 2} \\
= & \sum_{i=0}^{i^*-5} \Pr[\mathcal{E}_i] \cdot \frac{4D}{2^i} + \sum_{i=i^*-4}^{\log(D)-1} \Pr[\mathcal{E}_i] \cdot \frac{4D}{2^i} & \text{Split into cases: } \frac{D}{2^{i^*-4}} \frac{1}{16} \leq r \leq \frac{D}{2^{i^*-5}} \frac{1}{16} \\
\leq & \sum_{i=0}^{i^*-5} \frac{r}{D/2^i} \mathcal{O}(\log(\frac{|B(u,D/2^{i+1})|}{|B(u,D/2^{i+4})|})) \cdot \frac{4D}{2^i} + \sum_{i=i^*-4}^{\log(D)-1} \Pr[\mathcal{E}_i] \cdot \frac{4D}{2^i} & \text{By Lemma 15} \\
\leq & \sum_{i=0}^{i^*-5} \frac{r}{D/2^i} \mathcal{O}(\log(\frac{|B(u,D/2^{i+1})|}{|B(u,D/2^{i+4})|})) \cdot \frac{4D}{2^i} + \sum_{i=i^*-4}^{\log(D)-1} \frac{16r}{D/2^{i^*-4}} \cdot \frac{4D}{2^i} & \text{By Lemma 14 with respect to } D/2^{i^*-4} \\
= & 4r \sum_{i=0}^{i^*-5} \mathcal{O}(\log(\frac{|B(u,D/2^{i+1})|}{|B(u,D/2^{i+4})|})) + \sum_{i=i^*-4}^{\log(D)-1} 4 \cdot 2^{i^*-i} \cdot r & \text{Simplifying} \\
\leq & 4r \sum_{i=0}^{i^*-5} \mathcal{O}(\log(\frac{|B(u,D/2^{i+1})|}{|B(u,D/2^{i+4})|})) + 2^7 r & \text{Since } \sum_{i=i^*-4}^{\log(D)-1} 2^{i^*-i} \leq 2^5 \\
= & 4r\mathcal{O}(\log(n)) + 2^7 r & \log(\frac{x}{y}) = \log(x) - \log(y) \text{ and } |B(u,\infty)| \leq n \\
\in & \mathcal{O}(\log n) r &
\end{array}
$$

□