In computer science, when dealing with difficult problems involving graphs and their associated metrics, one technique we usually resort to is to solve the problem on a simple subcase, in particular, on tree. In this lecture, we study the Padded Decomposition of graphs and how to build their Tree metrics with $O(\log n)$ distortion.

# 1 Tree Metrics

**Definition 1 Tree Metric** - A metric $(V, d)$ is a Tree Metric if there exists a tree $T = (V, E_T)$ with edge lengths such that $d(u, v) = d_T(u, v)$.

We can build tree metrics to approximate general graph metrics, with an $O(\log n)$ distortion, where $n$ is the number of vertices in the graph, thus the following theorem:

**Theorem 1** Given an undirected edge weighted graph $G = (V, E)$, there is a randomized poly-time algorithm that produces an random edge weighted tree $T = (V_T, E_T)$ such that

1. $V \subseteq V_T$;
2. $d_G(u, v) \leq d_T(u, v), \forall u, v$;
3. $E(d_T(u, v)) \leq O(\log |V|)d_G(u, v), \forall u, v$.

![Figure 1: Tree Embedding.](image-url)

The Problem that Theorem 1 indicates is as follows (refer to Figure 1 for illustration):

**Problem 1** Given a graph metric $G = (V, d)$, find a collection of trees $\mathcal{T} = \{T_1, T_2, \ldots, T_h\}$ and a distribution $\mu = \{P_1, P_2, \ldots, P_h\}$ on them such that

1. $\sum_{i=1}^{h} P_i = 1$;
2. $d_{T_i}(u, v) \geq d_G(u, v), \forall u, v, \forall i$;
3. $\sum_{i=1}^{h} P_i d_{T_i}(u, v) \leq O(\log n)d_G(u, v), \forall u, v$.

A tree embedding algorithm is discussed in the next section to show the correctness of Theorem 1.
2  Padded Decomposition and Tree Embeddings

The following algorithm is developed by Bartal in [1] to solve the tree embedding problem. Note that \( \Delta(G) \) denotes the diameter of graph \( G \).

**Tree Embedding Algorithm**\((G(V,d))\):

Start with \( G \).
Randomly decompose \( G \) into \( G_1, G_2, \ldots, G_h \) s.t.
- \( \Delta(G) \leq \frac{\Delta(G)}{2}, \forall 1 \leq i \leq h; \)
- \( Pr[(u,v) is cut] \leq 4\cdot \frac{\Delta(G)}{\Delta(G)}\).

Recursively construct “rooted” trees \( T_1, T_2, \ldots, T_h \) for \( G_1, G_2, \ldots, G_h \), respectively.
Create tree \( T \) for \( G \) by (see Figure 2)
- adding root \( r \);
- connecting \( T_1, T_2, \ldots, T_h \) to \( r \) with edge weighted \( \Delta(G) \).

![Figure 2: Create tree \( T \) from \( T_1, T_2, \ldots, T_h \).](image)

The above Tree Embedding Algorithm is a high-level algorithm, we first suppose there exists a graph decomposition procedure to randomly decompose \( G \) into \( G_1, G_2, \ldots, G_h \) as in the second step, we then have the following two Propositions that can be easily observed.

**Proposition 2** \( T \) dominates \( G \).

**Proposition 3** \( \Delta(T) \leq 4\Delta(G) \).

We subsequently have the following lemma:

**Lemma 4** \( E(d_T(u,v)) = O(\alpha \log \Delta(G))d_G(u,v) \).

**Proof of Lemma 4.** Let \( A_i \) denote the event that \((u,v) is cut in level i\), and let \( \bar{A}_i \) denote the event that \((u,v) is not cut in level i\).

\[
E[d_T(u,v)] \leq 4 \cdot \Delta(G) \cdot Pr[A_1] + 4 \cdot \frac{\Delta(G)}{2} \cdot Pr[A_2|\bar{A}_1] + \ldots + 4 \cdot \frac{\Delta(G)}{2i} \cdot Pr[A_i|\cap_{j=1}^{i-1} \bar{A}_j] + \ldots
\]

\[
= 4 \cdot \Delta(G) \cdot \frac{\alpha d_G(u,v)}{\Delta(G)} + 4 \cdot \frac{\Delta(G)}{2} \cdot \frac{\alpha d_G(u,v)}{\Delta(G)^{1/2}} + \ldots
\]

\[
\leq 4\alpha \log \Delta(G) \cdot d_G(u,v) = O(\alpha \log \Delta(G)) \cdot d_G(u,v)
\]
Note that now \( E[d_T(u, v)] \) is dependent on \( \log \Delta(G) \), and this can actually be replaced with \( \log n \) if the CKR GRAPH DECOMPOSITION procedure is incorporated into the TREE EMBEDDING ALGORITHM, so that correctness of Theorem \( \text{1} \) can be proved. The CKR GRAPH DECOMPOSITION procedure is introduced in next subsection.

### 2.1 CKR Graph Decomposition Procedure

<table>
<thead>
<tr>
<th>CKR GRAPH DECOMPOSITION ((G(V, d), \delta)):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pick ( \theta ) at random from ([\frac{\delta}{4}, \frac{\delta}{2}]).</td>
</tr>
<tr>
<td>Pick a random permutation ( \sigma ) on ( V ).</td>
</tr>
<tr>
<td>For ( i = 1 ) to ( n ) do</td>
</tr>
<tr>
<td>( G_i = B(V_{\sigma(i)}, \theta) \cup \bigcup_{j&lt;i} B(V_{\sigma(j)}, \theta) )</td>
</tr>
</tbody>
</table>

Note that \( B(u, \theta) \) denotes the ball that is centered at \( u \) and with radius as \( \theta \), and we say \( B(u, \theta) \) is cut if any vertex in \( B(u, \theta) \) is separated from \( u \).

**Theorem 5**  CKR decomposition decomposes \( G \) into \( G_1, G_2, \ldots, G_n \) such that:

1. \( \Delta(G_i) \leq \delta \);
2. \( \forall u, \Pr[B(u, \varrho) \text{ is cut}] \leq \frac{c \varrho}{\delta} \log n \), where \( c \) is a constant.

It is easy to see the following is an immediate Corollary of Theorem \( \text{5} \).

**Corollary 6**  \( \forall u, v, \Pr[(u, v) \text{ is cut}] \leq \frac{c d_G(u, v)}{\delta} \log n. \)

**Proof of Theorem \( \text{5} \)** We fix vertex \( u \), and assume w.l.o.g. \( \varrho < \frac{\delta}{4} (\frac{c \varrho}{\delta} > 1 \text{ if } \varrho \geq \frac{\delta}{4}) \). Let \( v_1, v_2, \ldots, v_n \) be an ordering of \( V \) s.t. \( d(v_1, u) \leq d(v_2, u) \leq \ldots \leq d(v_n, u) \). Let \( A_i \) be the event that \( v_i \) “first” cuts \( B(u, \varrho) \), i.e.,

1. \( B(v_i, \theta) \cap B(u, \varrho) \neq \emptyset \)
2. \( B(v_i, \theta) \cap B(u, \varrho) \neq B(u, \varrho) \)
3. \( \forall j < \arg \sigma(i), v_{\sigma(j)} \text{ does not cut or captures } B(u, \varrho) \).

As shown in Figure \( \text{3} \) obviously \( A_i \) is possible only when \( \theta = a \in [d(v_i, u) - \varrho, d(v_i, u) + \varrho] \), and note that \( \sigma \) is a random permutation, so

\[
\Pr[A_i | \theta = a] \leq \frac{1}{i}, \forall a \in [d(v_i, u) - \varrho, d(v_i, u) + \varrho]
\]

recall that \( \theta \in [\frac{\delta}{4}, \frac{\delta}{2}] \), thus

\[
\Pr[A_i] \leq \frac{1}{i} \cdot \frac{2\varrho}{\delta/4} \leq \frac{8}{i} \cdot \frac{\varrho}{\delta}
\]

Therefore, \( \Pr[B(u, \varrho) \text{ is cut}] \leq \sum_{i=1}^{n} \Pr[A_i] \leq \frac{8\varrho}{\delta} \log n. \)
Actually we are able to provide an even tighter bound for $\Pr[B(u, \varrho) \text{ is cut}]$. By observation, $v_i$ may cut $B(u, \varrho)$ only when $d(v_i, u) \in [\delta/4 - \varrho, \delta/2 + \varrho]$, therefore,

$$\Pr[B(u, \varrho) \text{ is cut}] \leq \sum_{i: d(v_i, u) \in [\delta/4 - \varrho, \delta/2 + \varrho]} \frac{1}{8} \cdot \frac{8\varrho}{\delta} \cdot \log \left| \frac{|B(u, \delta/2 + \varrho)|}{|B(u, \delta/4 - \varrho)|} \right|.$$

2.2 Incorporate CKR decomposition into Tree Embedding Algorithm

By applying CKR GRAPH DECOMPOSITION procedure to decompose $G$ into $G_1, G_2, \ldots, G_h$, at the second step in the TREE EMBEDDING ALGORITHM, we are able to improve the bound $E(d_T(u, v)) = O(\log \Delta(G))d_G(u, v)$ in Lemma 4 to $E(d_T(u, v)) = O(\log n)d_G(u, v)$:

$$E[d_T(u, v)] \leq 4 \cdot \Delta(G) \cdot \Pr[A_1] + 4 \cdot \frac{\Delta(G)}{2} \cdot \Pr[A_2 | A_1 \cap \cdots \cap A_j] + \ldots$$

$$\leq 4 \cdot \Delta(G) \cdot \frac{8d_G(u, v)}{\Delta(G)/2} \cdot \log \left| \frac{|B(u, \Delta(G)/2)|}{|B(u, \Delta(G)/16)|} \right|$$

$$+ 4 \cdot \frac{\Delta(G)}{2} \cdot \frac{8d_G(u, v)}{\Delta(G)/4} \cdot \log \left| \frac{|B(u, \Delta(G)/4)|}{|B(u, \Delta(G)/32)|} \right|$$

$$+ \ldots$$

$$+ 4 \cdot \frac{\Delta(G)}{2^{i+1}} \cdot \frac{8d_G(u, v)}{\Delta(G)/2^{i+1}} \cdot \log \left| \frac{|B(u, \Delta(G)/2^{i+1})|}{|B(u, \Delta(G)/2^{i+4})|} \right|$$

$$\leq 64 \cdot d_G(u, v) \cdot 3 \log n$$

$$= O(\log n) \cdot d_G(u, v)$$

Up to now, we have shown the correctness of the sc Tree Embedding Algorithm, thus Theorem 4 has been finally proved.

References