

# Embedding Bounded Bandwidth Graphs into $\ell_p$

Douglas E. Carroll\*      Adam Meyerson†

February 16, 2007

## Abstract

We prove that graphs can be embedded randomly into  $\ell_p$  (for any  $p \geq 1$ ) with expected distortion polynomial in  $k$ , where  $k$  is the *bandwidth* of the graph. This implies the existence of an infinite family of graphs which embed well into  $\ell_p$ , which is surprising since *trees* do not form such a class, having a lower bound of  $\Omega(\log \log n)$  for  $\ell_2$  embedding. Our results extend to graphs of bounded *tree-bandwidth* when the target metric is  $\ell_1$ , or when we allow an additional term of size  $O(\log \log n)$  in our distortion.

## 1 Introduction

Metric embedding is a powerful technique with many applications in approximation algorithms. The idea is to find a one-to-one function to transform a *source* metric into a *target* metric. The *distortion* of the embedding measures the expansion and contraction of the distance function under this transformation. For embedding to be useful, the target metric should have properties which enable us to produce better algorithmic results; typical target metrics include *trees*, *graphs of low tree width*, and the *infinite normed spaces*. The class of possible source metrics should be as broad as possible, and the distortion should be low.

In this paper, we will focus our attention on embedding finite graph metrics into the normed spaces  $\ell_p$ . The main result in this area is the work of Bourgain [4], showing that *any* metric with  $n$  nodes can be embedded into high-dimensional  $\ell_p$  with distortion bounded by  $O(\log n)$ . Subsequent work has proven this to be tight (the particular metrics which are hard to embed are based upon expander graphs) [14]. For this reason, attention has turned to producing infinite graph families where better results are possible. The major result of this form is by Rao [20], showing that any graph family which excludes minor  $M$  can be embedded into  $\ell_p$  with distortion  $O(|M|^3 \sqrt{\log n})$ . The dependence on  $|M|$  has been improved [13], but there are lower bounds indicating that we cannot embed even *tree-width two* graphs into  $\ell_2$  with distortion  $\Omega(\sqrt{\log n})$  [17]. In fact, even *trees* have matching bounds of  $\Theta(\sqrt{\log \log n})$  for embedding into  $\ell_2$  [5, 15]. Since we'd like to find infinite families which embed with constant distortion, research in this direction has changed focus to the special case of the  $\ell_1$  metric, for which a number of results are now known [8, 11, 18].

Our work revives the interest in embedding infinite graph families into  $\ell_p$  for  $p > 1$ . We show that graphs of *bandwidth*  $k$  can be embedded into  $\ell_p$  with distortion polynomial in  $k$ . Of course, for this to be possible it is critical to observe that trees can have unbounded bandwidth. This implies that graph bandwidth may have interesting connections to  $\ell_2$ -embedding. We make use of the technique of *iterative embedding* previously introduced in [7], partitioning a low bandwidth graph into many small pieces and embedding

---

\*Department of Computer Science, University of California, Los Angeles. Email: dcarroll@cs.ucla.edu.

†Department of Computer Science, University of California, Los Angeles. Email: awm@cs.ucla.edu.

each one separately. We thereby produce an  $\ell_1$  embedding with no contraction and bounded expansion; by a carefully selected local embedding technique and improved methods for combining the pieces, we will additionally improve the distortion to polynomial in the bandwidth (previous results were exponential in the bandwidth). Since the  $\ell_p$  distance between two coordinates cannot exceed the  $\ell_1$  distance, this implies an  $\ell_p$  embedding which also has bounded expansion. We additionally observe that for any pair of nodes  $x, y$ , our technique guarantees that a small number of coordinates (at most  $k$ ) support the entire distance between  $x$  and  $y$ . This allows us to bound the *contraction* of the  $\ell_p$  embedding, and to prove our main result.

We can extend our result to *tree-bandwidth* as defined by [7]. Since all trees have tree-bandwidth one, we will not be able to embed into  $\ell_p$  with distortion dependent only on the tree bandwidth. Instead, we show that we can embed into  $\ell_p$  with distortion  $O(k^4 + k\alpha_p(T))$  where  $\alpha_p(T) \leq O(\sqrt{\log \log n})$  is the minimum distortion to embed the tree decomposition  $T$  into  $\ell_p$ . This gives a polynomial-in- $k$  distortion result for  $\ell_1$  embedding, and also improves over previous results in  $\ell_p$  by replacing the dependence on  $\sqrt{\log n}$  with a dependence on  $\sqrt{\log \log n}$ .

## 1.1 Related Work

A great deal of recent work has concentrated on achieving tight distortion bounds for  $\ell_1$  embedding of restricted classes of metrics. For general metrics with  $n$  points, the result of Bourgain[4] showed that embedding into  $\ell_1$  with  $O(\log n)$  distortion is possible. A matching lower bound (using expander graphs) was introduced by LLR [14]. It has been conjectured by Gupta *et al.* [11], and Indyk [12] that the shortest-path metrics of planar graphs can be embedded into  $\ell_1$  with constant distortion. Gupta *et al.* [11] also conjecture that excluded-minor graph families can be embedded into  $\ell_1$  with distortion that depends only on the excluded minors. In particular, this would mean that for any  $k$  the family of treewidth- $k$  graphs could be embedded with distortion  $f(k)$  independent of the number of nodes in the graph<sup>1</sup>. Such results would be the best possible for very general and natural classes of graphs.

Since Okamura and Seymour [18] showed that outerplanar graphs can be embedded isometrically into  $\ell_1$ , there has been significant progress towards resolving several special cases of the aforementioned conjecture. Gupta *et al.* [11] showed that treewidth-2 graphs can be embedded into  $\ell_1$  with constant distortion. Chekuri *et al.* [8] then followed this by proving that  $k$ -outerplanar graphs can be embedded into  $\ell_1$  with constant distortion. Note that all these graph classes not only have low treewidth, but are planar. We give the first constant distortion embedding for a non-planar subclass of the bounded treewidth graphs.

Rao [20] proved that any minor excluded family can be embedded into  $\ell_1$  with distortion  $O(\sqrt{\log n})$ . This is the strongest general result for minor-excluded families. Rabinovich [19] introduced the idea of *average distortion* and showed that any minor excluded family can be embedded into  $\ell_1$  with constant *average distortion*.

Graphs of low treewidth have been the subject of a great deal of study. For a survey of definitions and results on graphs of bounded treewidth, see Bodlaender [2]. More restrictive graph parameters include domino treewidth [3] and bandwidth [9], [10].

---

<sup>1</sup> There is a lower bound of  $\Omega(\log k)$  arising from expander graphs.

## 2 Definitions: Bandwidth and Tree-Bandwidth

Given two metric spaces  $(G, \nu)$  and  $(H, \mu)$  and an embedding  $\Phi : G \rightarrow H$ , we say that the *distortion* of the embedding is  $\|\Phi\| \cdot \|\Phi^{-1}\|$  where

$$\begin{aligned}\|\Phi\| &= \max_{x,y \in G} \frac{\mu(\Phi(x), \Phi(y))}{\nu(x, y)}, \\ \|\Phi^{-1}\| &= \max_{x,y \in G} \frac{\nu(x, y)}{\mu(\Phi(x), \Phi(y))}\end{aligned}$$

Parameter  $\|\Phi\|$  will be called the *expansion* of the embedding and parameter  $\|\Phi^{-1}\|$  is called the *contraction*.

We now define bandwidth and tree-bandwidth.

**Definition 2.1.** Given graph  $G = (V, E)$  and linear ordering  $f : V \rightarrow \{1, 2, \dots, |V|\}$  the *bandwidth* of  $f$  is  $\max\{|f(v) - f(w)| \mid (v, w) \in E\}$ . The *bandwidth* of  $G$  is the minimum bandwidth over all linear orderings  $f$ .

**Definition 2.2.** Given a graph  $G = (V, E)$ , we say that it has *tree-bandwidth*  $k$  if there is a rooted tree  $T = (I, F)$  and a collection of sets  $\{S_i \subset V \mid i \in I\}$  such that:

1.  $\forall i, |S_i| \leq k$
2.  $V = \bigcup S_i$
3. the  $S_i$  are disjoint
4.  $\forall (u, v) \in E$ ,  $u$  and  $v$  lie in the same set  $S_i$  or  $u \in S_i$  and  $v \in S_j$  and  $(i, j) \in F$ .
5. if  $c$  has parent  $p$  in  $T$ , then  $\forall v \in S_c, \exists u \in S_p$  such that  $d(u, v) \leq k$ .

Note that if graph  $G$  has tree-bandwidth  $k$ , we can assume that  $d(v, p(v)) \leq 1$  while incurring only  $O(k)$  distortion.

We will now show a relationship between *bandwidth* and *tree-bandwidth*. In particular, we will show that any of bandwidth  $k$  embeds into a graph of tree-bandwidth  $k$  with low distortion. This will enable us to focus our efforts on embedding graphs of bounded tree-bandwidth into  $\ell_p$ . We observe that the inverse relationship is not true since trees have tree-bandwidth one but do not embed with low distortion into bounded bandwidth graphs. The proof of the following theorem will be given in appendix A.

**Theorem 2.3.** *Any graph with bandwidth  $k$  can be embedded with distortion  $O(k^4)$  into a graph with tree-bandwidth  $k$ .*

### 2.1 Relating Tree-Bandwidth to Treewidth

A major conjecture in metric embedding states that any graph with treewidth  $k$  can be embedded into  $\ell_1$  with distortion dependent only upon  $k$ . Of course, no similar conjecture can hold for  $\ell_2$ ; even for graphs of treewidth 2 we have a lower bound of  $\Omega(\sqrt{\log n})$  [17]. While this paper will not resolve the conjecture, there are interesting relationships between tree-bandwidth (which we will show how to embed into  $\ell_1$  with distortion polynomial in  $k$ ) and treewidth. We will explore these relationships here.

We will first make some definitions regarding DFS-trees, and relate them to treewidth and tree-bandwidth.

**Definition 2.4.** (i) Given a connected graph  $G = (V, E)$ , a *DFS-tree* is a rooted spanning subtree  $T = (V, F \subset E)$  such that for each edge  $(u, v) \in E$ ,  $v$  is an ancestor of  $u$  or  $u$  is an ancestor of  $v$  in  $T$ .

(ii) The *value* of DFS-tree  $T$  is the maximum over all  $v \in V$  of the number of ancestors that are adjacent to  $v$  or a descendent of  $v$ .

(iii) The *edge stretch* of DFS-tree  $T$  is the the maximum over all  $v, w \in V$  of the distance  $d(v, w)$  where  $w$  is an ancestor of  $v$  and  $w$  is adjacent to  $v$  or a descendent of  $v$ .

We use the following definition of treewidth due to T. Kloks and related in a paper of Bodlaender [2]:

**Definition 2.5.** Given a connected graph  $G = (V, E)$ , the *treewidth* of  $G$  is the minimum value of a DFS-tree of a supergraph  $G' = (V, E')$  of  $G$  where  $E \subset E'$ .

The following proposition follows immediately from the definition of tree-bandwidth:

**Proposition 2.6.** *Given a connected graph  $G = (V, E)$ , the tree-bandwidth of  $G$  is the minimum edge stretch of a DFS-tree of  $G$ .*

Thus, treewidth and tree-bandwidth appear to be related in much the same way that cutwidth and bandwidth are related (see [2] for instance). The close relationship between treewidth and tree-bandwidth is cemented by the following observation:

**Lemma 2.7.** *Any metric supported on a weighted graph  $G = (V, E)$  of treewidth- $k$  can be embedded with distortion 4 into a weighted graph with tree-bandwidth- $O(k)$*

Thus, a technique for embedding *weighted* tree-bandwidth- $k$  graphs into  $\ell_1$  with  $O(f(k))$  distortion would immediately result in constant distortion  $\ell_1$ -embeddings of weighted treewidth- $k$  graphs.

## 2.2 Bounded Bandwidth Example

We give an example of a graph of bandwidth at most  $2k - 1$  which is not planar. This implies that the constant distortion  $\ell_1$ -embedding techniques of [18, 11, 8] will not apply.

Construct  $G$  in the following manner:

1. Consider  $k$  points connected in an arbitrary way.
2. Add  $k$  new points connected to each other and the previous  $k$  points in an arbitrary way.
3. Repeat step 2 an arbitrary number of times.

Clearly the graph  $G$  generated in this way has bandwidth  $\leq 2k - 1$ . However, note that if  $k \geq 3$  and some set of  $2k$  consecutively added points contains  $K_{3,3}$  or  $K_5$  then  $G$  is not planar.

## 2.3 Bounded Tree-Bandwidth Example

To show that bounded tree-bandwidth graphs form a broader class than the bounded bandwidth graphs consider the following example. Let  $G = (V', E')$  consist of  $k$  copies of an arbitrary tree  $T = (V, E)$ . Construct  $G'$  from  $G$  as follows:

1. For  $x \in V$ , let  $\{x_1, \dots, x_k\}$  be the  $k$  copies of  $x$  in  $V'$ .
2. For each  $x \in V$ , connect  $\{x_1, \dots, x_k\}$  in an arbitrary way.

While the resulting graph  $G'$  clearly has tree-bandwidth  $k$ , a complete binary tree of depth  $d$  has bandwidth  $\Omega(d)$  [9], thus  $G'$  may have bandwidth  $\Omega(\log n)$ .

Note again that if  $k \geq 5$  and  $G'$  contains  $K_{3,3}$  or  $K_5$  then  $G'$  is not planar and thus previous constant distortion  $\ell_1$ -embedding techniques cannot be applied.

Also note that there are trees  $T$  with  $|V| = n$  such that any  $\ell_2$ -embedding of  $T$  has distortion  $\Omega(\sqrt{\log \log n})$  [5]. Since Rao's technique embeds first into  $\ell_2$  this gives a lower bound of  $\Omega(\sqrt{\log \log n})$  on the distortion achievable using Rao's technique to embed  $G'$  into  $\ell_1$ . The technique presented in this paper embeds these examples into  $\ell_1$  with distortion depending only on  $k$ .

Apart from being interesting from a technical viewpoint, bounded tree-bandwidth graphs may also be a good model for phylogenetic networks with limited introgression/reticulation [16]. This is a fruitful connection to explore, though it is outside the scope of this paper.

### 3 Local Embedding

Given a graph  $G$  of tree-bandwidth  $k$ , and a tree-bandwidth decomposition  $(T = (I, F), \{X_i | i \in I\})$  of  $G$ , we will construct a collection of clusterings for  $X_i$  which approximately preserves the metric of  $G$  restricted to  $X_i$ . Furthermore, these clusterings will not be constructed independently; they will be built in a way that is sensitive to the structure of the decomposition  $(T = (I, F), \{X_i | i \in I\})$  of  $G$ . We will match coordinates to each cluster and assign a nonzero value to the coordinate if and only if a point is a member of the cluster. These coordinates can be thought of as providing an embedding into  $\ell_p$  for any  $p \geq 1$ .

#### 3.1 Metric Sensitive Clustering

Given any set  $X_i$  from the tree-bandwidth decomposition of  $G$ , the shortest path metric of  $G$  induces a metric on the points of  $X_i$ . We will refer to this metric as the metric of  $X_i$ .

Let  $n = |V|$ . Let  $\delta = \log_2 n$ .

Let  $C$  be an ordered collection of  $\delta$  clusterings of the points of  $X_i$ . We will use  $C_j$  to denote the  $j$ th clustering of  $C$ . We will call  $C_j$  *valid* for  $X_i$  if the following properties hold:

1. if  $d(x, y) > 2^j$  then  $C_j$  separates  $x, y$
2. if  $d(x, y) \leq \frac{1}{k}2^j$  then  $C_j$  clusters  $x, y$

We will call  $C_j$  *almost-valid* for  $X_i$  if the following properties hold:

1. if  $d(x, y) > 2^{j+1}$  then  $C_j$  separates  $x, y$
2. if  $d(x, y) \leq \frac{1}{k}2^{j-1}$  then  $C_j$  clusters  $x, y$

We will say the ordered collection  $C$  is *almost-valid* for  $X_i$  if  $C_j$  is *almost-valid* for  $X_i$  for every  $j$ . On occasion we will omit the  $X_i$  and simply say that  $C$  is *valid* or *almost-valid* when the set  $X_i$  is clear from context.

**Theorem 3.1.** *A valid clustering  $C_j$  can be constructed for any set  $X_i$ .*

*Proof.* To build the clustering at scale  $j$  connect all pairs of points  $x, y$  where  $d(x, y) \leq \frac{1}{k}2^j$ . Let clusters correspond to the resulting connected components.

Since there are at most  $k$  nodes in  $X_i$ , each connected component has diameter  $\leq 2^j$ . Furthermore, distinct clusters are separated by distance  $\geq \frac{1}{k}2^j$ .  $\square$

We can use an almost-valid clustering to construct an embedding of the set  $X_i$  into  $\ell_p$ . This is done by associating a weight with each clustering  $C_j$ . We then associate a coordinate with each cluster in  $C_j$ , such that points not belonging to that cluster have value zero in the associate coordinate, and points in that cluster have value equal to the weight.

**Theorem 3.2.** *If we apply weight  $2^{j+2}$  to clustering  $C_j$  from an almost-valid collection of clusterings  $C$ , then the resulting  $\ell_p$  embedding of  $X_i$  does not contract any distances and expands distances by at most a factor of  $32k$ .*

*Proof.* Consider some  $x, y$  with  $2^j < d(x, y) \leq 2^{j+1}$ .

We observe that  $x, y$  will be clustered by  $C_{j+2+\log k}$  since  $d(x, y) \leq 2^{j+1} \leq \frac{1}{k}2^{j+1+\log k}$ . Thus the embedded distance between  $x$  and  $y$  will be at most  $2^{\sum_{m \leq j+1+\log k} 2^{m+2}} \leq k2^{j+5} \leq 32kd(x, y)$ , which bounds the expansion.

On the other hand,  $x, y$  will be separated by  $C_{j-1}$  (and  $C_m$  for  $m < j-1$ ) since  $d(x, y) > 2^j$ . Thus the embedded distance between  $x$  and  $y$  will be at least  $2^{j+1} \geq d(x, y)$ , implying no contraction.  $\square$

## 4 Global Embedding

### 4.1 Inheriting Clusterings and Timers

We would now like to use the bounded tree-bandwidth of  $G$  to embed the entire graph into  $\ell_p$  with minimal distortion. We will use the proximity of points in  $X_i$  to points in  $X_{p(i)}$  to show that we can force agreement between valid clusterings of sets which are adjacent in the tree-bandwidth decomposition.

For each scale  $j$  with  $0 \leq j \leq \delta$  we will have a timer  $\tau_j$ . When we embed the root of the tree-bandwidth decomposition, we will initialize  $\tau_j$  to a value selected uniformly from  $[0, \frac{2^{j-2}}{k} - 1]$  and we will compute a *valid* collection of clusterings for the root. Whenever we embed some super-node  $X_i$  in the tree-bandwidth decomposition, we will copy the values of  $\tau_j$  from its parent. We then increment the value of  $\tau_j$  for every  $j$  such that there exist two points  $x, y$  in the current super-node whose distance apart satisfies  $\frac{1}{k}2^{j-1} \leq d(x, y) \leq 2^{j+1}$ . For every  $j$  such that  $\tau_j < \frac{2^{j-2}}{k}$  we inherit clustering  $C_j$  from the parent. Each node will belong to the same cluster in this clustering that its parent node belonged to. On the other hand, if  $\tau_j = \frac{2^{j-2}}{k}$  then we will define a new *valid* clustering  $C_j$  for the points of  $X_i$ , and then reset  $\tau_j$  to be zero.

We now prove a sequence of lemmata which bound the distance between parent and child nodes, as well as the distance between two nodes in the same super-node  $X_i$ .

**Lemma 4.1.** *The clusterings we use for  $X_i$  are almost-valid.*

*Proof.* Consider some clustering  $C_j$  that is not *almost-valid*. It follows that there is some pair  $x, y \in X_i$  such that  $d(x, y) \leq \frac{1}{k}2^{j-1}$  but where  $x, y$  are in different clusters, or that there is some pair  $x, y \in X_i$  such that  $d(x, y) > 2^{j+1}$  but where  $x, y$  are in the same cluster. We will consider the first case (the proof for of contradiction the second case is similar). We backtrack to the most recent time when we replaced cluster  $C_j$ . At that time, there were some ancestors of  $x, y$  which we will call  $a(x), a(y)$ . Suppose that  $d(a(x), a(y)) > \frac{1}{k}2^j$ . Since at each step the parent of a node is adjacent to the child, we conclude that there is a sequence of nodes connecting  $a(x)$  to  $x$  and similarly  $a(y)$  to  $y$ . There must have been more than  $\frac{2^{j-2}}{k}$  super-nodes in the intervening time that would have updated the timer  $\tau_j$ . This contradicts the manner in which the timers work. On the other hand, suppose  $d(a(x), a(y)) \leq \frac{1}{k}2^j$ . In this case when we produced clustering  $C_j$  it was valid, so  $a(x)$  and  $a(y)$  were in the same cluster. But then  $x, y$  inherit the clustering of their ancestors, so they too would be in the same cluster, contradicting the assumption that they are in

different clusters. Since in any case we have a contradiction, we conclude that in fact the clusterings are almost-valid.  $\square$

**Lemma 4.2.** *When we embed  $X_i$ , at most  $O(k \log k)$  timers will be incremented.*

*Proof.* We first observe that a single distance can increment at most  $O(\log k)$  timers. Consider building the minimum spanning tree on the nodes of  $X_i$ , where the weight of an edge equals the distance between its endpoints. There are  $k - 1$  edges in this spanning tree. Now consider any pair of points  $x, y$  in  $X_i$ . Consider the path  $P(x, y)$  through the spanning tree. If the distance between  $x, y$  is less than the length of the longest spanning tree edge in  $P(x, y)$ , then we could produce a better spanning tree by removing that longest edge and replacing it with  $(x, y)$ . On the other hand, the distance between  $x, y$  cannot exceed the length of  $P(x, y)$  which is at most  $(k - 1)$  times the length of the longest spanning tree edge in  $P(x, y)$ . Thus every distance is within a factor of  $k$  of the length of some spanning tree edge. Distances within a factor of  $k$  of a particular distance can increment only  $O(\log k)$  timers, so we get at most  $O(k \log k)$  timers incremented in total.  $\square$

**Lemma 4.3.** *For any timer  $\tau_j$  which is incremented, the probability of defining a new clustering  $C_j$  for  $X_i$  is exactly  $\frac{k}{2^{j-2}}$ ; for any timer  $\tau_j$  which is not incremented, the probability of defining a new clustering  $C_j$  for  $X_i$  is zero.*

*Proof.* We always maintain that timers have value at most  $\frac{2^{j-2}}{k} - 1$ , so we can only define a new clustering  $C_j$  if we increment timer  $\tau_j$ . The number of times  $\tau_j$  has been incremented since the root node is deterministic, so there is exactly one initial value for  $\tau_j$  such that we will define a new clustering  $C_j$ . Since the initial values were determined uniformly at random, the lemma follows.  $\square$

**Theorem 4.4.** *The coordinates used for the clusterings guarantee that the expected embedded distance between  $x \in X_i$  and  $p(x) \in X_{p(i)}$  is no more than  $O(k^2 \log k)$ .*

*Proof.* Inherited clusterings don't create any distance between parent and child – the same coordinate will be given the same values. The coordinates where  $x$  and  $p(x)$  differ are those corresponding to clusterings  $C_j$  where we defined a new clustering. Whenever this happens, we will have  $x$  and  $p(x)$  in different clusters, so we will have a pair of coordinates one of which is  $2^{j+2}$  and the other of which is zero for  $x$ , with  $p(x)$  being exactly the opposite. For any  $C_j$  where we incremented the timer, the probability of this mismatch will be  $\frac{k}{2^{j-1}}$ . Applying linearity of expectations, we can conclude that the expected  $\ell_1$  distance between parent and child is at most  $O(k^2 \log k)$ . Since the  $\ell_p$  distance can never exceed the  $\ell_1$  distance (for  $p > 1$ ) we have the same bound for  $\ell_p$ .  $\square$

## 5 Bounding Overall Distortion

We will add a few new coordinates to our metric. In particular, we compute an embedding of the *tree decomposition* of the graph into  $\ell_p$ . This embedding will be non-contracting, and will guarantee an expansion of at most  $\alpha$ . In general we can guarantee that  $\alpha \leq O(\sqrt{\log \log n})$  because the tree decomposition is a tree. For  $\ell_1$ , we have  $\alpha = 1$ . Of course, for some special tree decompositions (such as those arising from a low bandwidth graph) we will be able to better refine our value of  $\alpha$ . We will now bound the  $\ell_p$  distortion of our embedding.

**Theorem 5.1.** *Our algorithm embeds tree-bandwidth  $k$  graphs into  $\ell_p$  with distortion  $O(k^3 \log k + k\alpha)$  where  $\alpha$  is the distortion for embedding the tree decomposition into  $\ell_p$ .*

*Proof.* We will lose a factor of  $k$  because of our assumption that parent and child nodes are always adjacent (distance one) instead of distance at most  $k$ . Now consider two points  $x, y$  in the original graph. We consider three cases:

1. If  $x, y$  are in the same super-node in the tree decomposition, then we simply combine theorem 3.2 with lemma 4.1 to see that their embedded distance via the clusterings does not contract the real distance and does not expand the real distance by more than  $O(k)$ . We observe that the coordinates corresponding to the tree decomposition will be identical for  $x$  and  $y$  and will therefore have no effect.
2. If  $x, y$  are in adjacent super-nodes, then assume without loss of generality that  $y$  is in the parent super-node and  $x$  in the child. Let  $p(x)$  be the parent of  $x$ . Now  $E[d_E(y, x)] \leq E[d_E(y, p(x))] + E[d_E(p(x), x)]$  by applying triangle inequality and linearity of expectation to the embedded distances. Since  $y$  and  $p(x)$  are in the same super-node, we can apply the previous case to this distance. For the distance between  $x$  and its parent, we apply theorem 4.4. Combining these yields  $E[d_E(y, x)] \leq O(\alpha + k^2 \log k)$ , where the additional  $\alpha$  comes from the coordinates corresponding to the tree decomposition. On the other hand, we also guarantee that  $d_E(y, x) \geq (d_E(y, p(x))^p + \Delta^p)^{1/p}$  where  $\Delta \geq 1$  is the embedded distance in the tree decomposition. This gives us  $d_E(y, x) \geq \frac{1}{2}d(y, x)$  and bounds the contraction.
3. If  $x, y$  are in distinct, non-adjacent super-nodes, then there is some path  $Q$  through the tree-bandwidth decomposition separating their super-nodes. We will prove our bound by induction on the length of this path, with the base case being covered in the previous case where the super-nodes are adjacent. Inductively, we find the shortest path in the original metric between  $x$  and  $y$ , and observe that it must visit some node  $z$  in each super-node  $Z$  lying on the path  $Q$ . Since  $z$  is closer in the tree-bandwidth decomposition to both  $x$  and  $y$ , we can inductively write  $d_E(x, z) \leq O(k^2 \log k)d(x, z)$  and similarly for  $d_E(x, y)$ . Applying triangle inequality along with the fact that  $d(x, y) = d(x, z) + d(y, z)$  completes the induction and gives us the desired bound on the expansion.

We now need to bound the contraction. We let  $Q$  be the path through the tree decomposition between the super-nodes containing  $x$  and  $y$ . We consider  $Z$  to be the common ancestor of the super-nodes containing  $x$  and  $y$ , and let  $x', y'$  be closest nodes in  $Z$  to  $x$  and  $y$  respectively. The coordinates representing clusters guarantee that  $d_E(x, y) \geq d_E(x', y')$  since  $x$  and  $y$  inherit clusterings from their ancestors. The tree decomposition coordinates give  $d_E(x, y) \geq |Q|$ . Since  $x', y'$  are in the same super-node we have  $d_E(x', y') \geq d(x', y')$ , and because of the parent-child distance of one we can guarantee that  $d(x', y') \geq d(x, y) - |Q|$ . Combining these yields:

$$d_E(x, y) \geq \max[d(x, y) - |Q|, |Q|] \geq \frac{1}{2}d(x, y)$$

Thus we have expansion by at most  $O(k^2 \log k + \alpha)$  and contraction by at most a factor of two. Of course, this was assuming parent and child are adjacent, and removing that assumption gives us an additional  $k$ . We can eliminate contraction by simply doubling the values of all coordinates (which increases expansion by a factor of two).  $\square$

In general we will have  $\alpha \leq O(\sqrt{\log \log n})$  as this is the bound for embedding trees into  $\ell_p$ . However, in the case of graphs with bounded bandwidth we should be able to do better.

**Theorem 5.2.** *If graph  $G$  has bandwidth  $k$ , then we can embed  $G$  into  $\ell_p$  with distortion  $O(k^7 \log k)$ .*



*Proof.* We make use of our result in theorem 2.3 to embed the original metric into a graph metric of bounded tree bandwidth. The distortion of this embedding is  $O(k^4)$ . We then embed that graph into  $\ell_p$ . Consider the tree decomposition produced in the proof of theorem 2.3. There is a single “spine” of the graph, along which we have potentially many branches. Each branch is recursively constructed with bandwidth  $k - 1$ . If we use one coordinate to represent the position on each branch, this yields an isometric  $\ell_1$  embedding for the tree. In  $\ell_p$ , we observe that any point has at most  $k$  nonzero coordinates in this system, which implies that the  $\ell_p$  distance is within a factor of  $k$  of the  $\ell_1$  distance. Thus we have  $\alpha \leq k$  for this particular tree bandwidth decomposition, which completes the proof.  $\square$

We can additionally de-randomize the embedding of bounded tree-bandwidth graphs into  $\ell_p$ . Note that this does not immediately de-randomize the embedding of bounded bandwidth into  $\ell_p$  because the embedding of bandwidth into tree-bandwidth was also randomized.

**Theorem 5.3.** *We can de-randomize the tree-bandwidth embedding into  $\ell_p$  with only constant factor loss in the distortion bounds.*

## References

- [1] M. Badoiu, K. Dhamdhere, A. Gupta, Y. Rabinovich, H. Raecke, R. Ravi, and A. Sidiropoulos, “Approximation Algorithms for Low-Distortion Embeddings Into Low-Dimensional Spaces”, In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2005, pp. 119-128.
- [2] H. Bodlaender, “A partial k-arboretum of graphs with bounded treewidth”, *Theoretical Computer Science*, 209 (1998), pp. 1-45.
- [3] H. Bodlaender, “A Note On Domino Treewidth”, *Discrete Mathematics and Theoretical Computer Science*, 3 (1999), pp. 144-150.
- [4] J. Bourgain, “On Lipschitz Embeddings of Finite Metric Spaces in Hilbert Space.”, *Israel Journal of Mathematics*, 52 (1985), pp. 46-52.
- [5] J. Bourgain, “The Metrical Interpretation of Superreflexivity in Banach Spaces.”, *Israel Journal of Mathematics*, 56 (1986), pp. 222-230.
- [6] D. E. Carroll and A. Goel, “Lower Bounds for Embedding into Distributions over Excluded Minor Graph Families”, In *Proceedings of the 12th European Symposium on Algorithms*, 2004, pp. 146-156.
- [7] D. E. Carroll, A. Goel, A. Meyerson, “Embedding Bounded Bandwidth Graphs into  $\ell_1$ ”, In *Proceedings of the 33rd Annual ICALP*, 2006, pp. 27-37.
- [8] C. Chekuri, A. Gupta, I. Newman, Y. Rabinovich, and A. Sinclair, “Embedding  $k$ -Outerplanar Graphs into  $\ell_1$ ”, In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2003, pp. 527-536.
- [9] F. Chung, P. Seymour, “Graphs with Small Bandwidth and Cutwidth”, *Discrete Mathematics*, 75 (1989), pp. 113-119.
- [10] U. Feige, “Approximating the bandwidth via volume respecting embeddings”, In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, 1998, pp. 90-99.

- [11] A. Gupta, I. Newman, Y. Rabinovich, and A. Sinclair, “Cuts, trees and  $\ell_1$ -embeddings.”, In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, 1999, pp. 399-408.
- [12] P. Indyk, “Algorithmic Aspects of Geometric Embeddings.”, In *Proceedings of the 42th Annual IEEE Symposium on Foundations of Computer Science*, 2001, pp. 10-33.
- [13] R. Krauthgamer, J. Lee, M. Mendel, A. Naor, “Measured descent: A new embedding method for finite metrics”, In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, 2004, pp. 434-443.
- [14] N. Linial, E. London, and Y. Rabinovich, “The geometry of graphs and some of its algorithmic applications”, *Combinatorica*, 15 (1995), pp. 215-245.
- [15] J. Matousek, “On embedding trees into uniformly convex Banach spaces.”, *Israel Journal of Mathematics*, 114 (1999), pp. 221-237.
- [16] L. Nakhleh, T. Warnow, C. R. Linder, and K. St. John, “Reconstructing reticulate evolution in species - theory and practice.”, *Journal of Computational Biology*, (To Appear).
- [17] Ilan Newman, Yuri Rabinovich, “A lower bound on the distortion of embedding planar metrics into Euclidean space.”, In *Proceedings of the Symposium on Computational Geometry*, 2002, pp. 94-96.
- [18] H. Okamura, P. Seymour, “Multicommodity Flows in Planar Graphs”, *Journal of Combinatorial Theory Series B*, 31 (1981), pp. 75-81.
- [19] Y. Rabinovich, “On Average Distortion of Embedding Metrics into the Line and into  $\ell_1$ ”, In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, 2003, pp. 456-462.
- [20] S. Rao, “Small distortion and volume preserving embeddings for Planar and Euclidean metrics”, In *Proceedings of the 15th Annual Symposium on Computational Geometry*, 1999, pp. 300-306.
- [21] N. Robertson, P. Seymour, “Graph Minors II. Algorithmic Aspects of Tree-Width”, *Journal of Algorithms*, 7 (1986), pp. 309-322.

## A Proof of Theorem 2.3

### Definition A.1.

- A *far point*  $f$  is a point that has no leftward edge in the bandwidth ordering.
- A *connecting path* of a far point  $f$  is a path beginning at  $f$  and ending at  $v$  where  $\phi(v) < \phi(f)$ .
- The *minimum connecting path* of  $f$  is the connecting path with the smallest maximum index.
- The *magnitude* of a far point  $f$  is the length of the minimum connecting path.
- The max index of the minimum connecting path of  $f$  is at least  $magnitude(f) + index(f) - 1$

---

**Input:** Assume we are given graph  $G = (V, E)$  with bandwidth- $k$ . We will refer to the nodes of  $G$  according to the ordering.

1.  $G' \leftarrow G$
  2.  $root \leftarrow v \in V$  such that  $\phi(v) = 1$
  3. give each edge weight 1
  4. For each far point  $f_i$  do:
    - (a)  $l_i \leftarrow$  endpoint of minimum connecting path of  $f_i$
    - (b)  $m_i \leftarrow magnitude(f_i)$
    - (c) Randomly choose  $c_i \in [\phi(f_i), \phi(f_i) + m_i - 1]$
  5. Consider the indices  $c_i$  in increasing order and do:
    - (a)  $\delta \leftarrow m_i$
    - (b) for each edge  $e$  that crosses  $c_i$  do:
      - i.  $weight(e) \leftarrow weight(e) + 2k\delta$
    - (c) consider the connected components that would result from cutting all edges crossing  $c_i$
    - (d) let  $current$  be the component that contains  $f_i$
    - (e)  $star \leftarrow \{ \}$ ;  $target \leftarrow l_i$ ;  $threshold \leftarrow m_i$
    - (f) while  $root, target \notin current$  and  $\delta \geq threshold$  do:
      - i.  $star \leftarrow Boundary(current) \cup \{connector\}$
      - ii. let  $C$  be the component containing  $target$
      - iii.  $current \leftarrow current \cup C$
      - iv. if far point  $f_j$  is the far point with the largest magnitude in  $current$  do
        - A.  $target \leftarrow l_j$
        - B.  $threshold \leftarrow m_j$
        - C.  $connector \leftarrow$  the last edge in the minimum connecting path of  $f_j$  that crosses  $c_i$
    - (g) Identify the midpoints of all edges in  $star$  - this will create new point  $p$  in  $G'$
    - (h) if  $root \in current$ , we call  $p$  a spine point otherwise we call it a connecting point
    - (i) the connected components to the left of  $c_i$  formed by removing  $p$  are called the hanging components of  $p$
- 

Figure 1: Algorithm REMOVE-FAR-POINTS

Our embedding is as follows: We apply algorithm REMOVE-FAR-POINTS. This gives us a graph  $G'$ . Let  $G_S$  be  $G'$  with the hanging components of all spine points removed. First note that  $G_S$  has bandwidth- $k$  and has the property that every  $v \in G_S$  has a leftward edge. Next note that each hanging component  $H$  of a spine point  $p$  has bandwidth  $\leq k-1$ . We will then apply the algorithm REMOVE-FAR-POINTS recursively to each hanging component  $H$  with the corresponding spine point as the root of  $H$ .

**Lemma A.2.** *After one application of REMOVE-FAR-POINTS, each far point belongs to a hanging component of exactly one spine point.*

*Proof.* Basically, the idea is that once edges get clustered, they do not get split up.

Same component before cut implies same component after.

Every component must eventually merge into the root component.

Essentially, when component  $A$  merges into component  $B$ , all edges of  $A$  merge into  $B$ .

**Claim A.3.** *The minimum connecting distance of a component is greater than its diameter.*

*Proof.* The proof is by induction on the number of far points in the connected component. If there is only one far point in component  $C$ , then the magnitude of the shortest minimum connecting path is  $m_i$  and the diameter is  $\leq m_i$ .

Assume  $C$  has  $t$  far points. Assume statement is true for components with  $< t$  far points. Assume  $C$  has far point  $f_j$  with a shortest minimum path which does not lie in  $C$ , and largest far point  $f_i$  which has magnitude  $m_i > m_j$ . Consider when the components  $C_i, C_j$  of  $f_i$  and  $f_j$  merged.  $f_i$  was the largest far point in  $C_i$ , so if  $c_i$  had connected  $C_i, C_j$ , then the minimum connecting path of  $f_j$  would also have been included in  $C$  which is a contradiction. Thus, it must have been  $c_j$  that connected  $C_i, C_j$ . □

□

**Lemma A.4.**  *$G_S$  has bandwidth  $\leq k$  and has the property that every  $v \in G_S$  has a leftward edge.*

*Proof.* Clearly replacing edges by a star does not increase the bandwidth. By the previous lemma, all far points have been removed and our algorithm adds no new far points. □

**Lemma A.5.** *For any index  $i$ , there are at most  $k$  far points whose minimum connecting paths can cross  $i$ .*

*Proof.* Fix index  $i$ . No far point with index greater than  $i$  can have a minimum connecting path which crosses  $i$ . Assume  $x, y$  are far points with indices less than  $i$ . Assume WLOG that  $x$  appears before  $y$  in the ordering. Let  $P(x), P(y)$  be the minimum connecting paths of  $x, y$  respectively. Assume  $P(x), P(y)$  both cross index  $i$ . Let  $x', y'$  be the first points on the paths  $P(x), P(y)$  respectively which have edges crossing index  $i$ . If  $x' = y'$ , then there is a path from  $y$  to  $x$  which does not cross index  $i$ . This contradicts the definition of  $P(y)$  as the minimum connecting path. Thus, distinct far points must have distinct first nodes which have edges crossing  $i$ . Since the bandwidth is  $k$ , there be at most  $k$  such distinct nodes which have rightward edges crossing  $i$ . Thus, at most  $k$  far points have minimum connecting paths that cross  $i$ . □

□

**Theorem A.6.** *For an unweighted graph  $G$  of bandwidth- $k$  the distortion of our embedding is  $O(k^4)$ .*

*Proof.* The theorem is an immediate consequence of the following lemmas:

**Lemma A.7.** *REMOVE-FAR-POINTS does not contract any distance.*

*Proof.* We never decrease the weight of any edge, so contraction can only occur when we replace a set of edges crossing a cut with a star.

Consider the star formed at index  $c_i$ . Cutting  $c_i$  would create at most  $k$  connected components to the left of  $c_i$ . We only cluster components together when they are less than  $m_i$  apart.

**Claim A.8.** *Every component that does not contain root has a far point with magnitude greater than the diameter of the component.*

*Proof.* Fix component  $C$  which does not contain *root*. Let  $v$  be the minimum index point in  $C$ , thus it is a far point. The minimum connecting path of  $v$  must connect it to a point outside of  $C$ . The maximum index on this path exceeds  $c_i$ . Thus, the magnitude of  $v$  must be greater than the number of points in  $C$ .  $\square$

Thus, every component with the exception of the last component has diameter  $\leq m_i$ . Thus, the maximum distance between the endpoints of any pair of edges that is replaced by a star is  $(k-1)2m_i + 2$  which is less than  $2km_i$  which is the length of the shortest path through the new star.  $\square$

**Lemma A.9.** *The expected expansion of embedding REMOVE-FAR-POINTS is  $O(k^3)$ .*

*Proof.* By linearity of expectation, it is enough to bound the stretch of arbitrary edges.

Fix edge  $e = (u, v)$ . Since  $G$  has bandwidth- $k$ ,  $e$  can only cross  $k-1$  indices. The expected stretch of  $e$  is at most the sum of the expected stretches of these  $k-1$  indices. Let us bound the expected stretch of edges across a given index.

Assume the minimum connecting path of far point  $f_i$  crosses index  $t$ . The chance that  $f_i$  causes the edges across  $t$  to be stretched is  $1/(m_i-1)$ . The stretch of edge  $e$  that crosses index  $t$  is  $2k \cdot m_i$ .

A critical point here is that the minimum connecting path of a far point  $f_i$  never crosses an edge of weight  $> 1$ . If such a path did pass through a stretched edge, it would have to pass through some connecting point  $p$ . But this would mean that  $f_i$  already has a spine point. Thus, minimum connecting paths in a graph that has already been processed are no longer than they were in the original graph.

So, for a particular edge there are at most  $k-1$  indices at which a far point can stretch that edge. By Lemma A.5 there are at most  $k$  far points whose minimum connecting paths can cross any index  $t$ . Finally, the expected stretch of edges across an index caused by a particular far point is  $\leq 2k \cdot m_i$ . Thus, the expected stretch of an edge is  $\leq k \cdot (k-1) \cdot 2k \cdot m_i \frac{1}{m_i-1} = O(k^3)$

$\square$

As noted in the lemma, the distortion generated by each application of REMOVE-FAR-POINTS is additive and since there are at most  $k-1$  applications that affect a given edge, the expected distortion is  $O(k^4)$ .

$\square$

## B Proof of Lemma 2.7

**Lemma B.1.** *Assume we are given weighted graph  $G = (V, E)$  with treewidth- $k$ . There exists an embedding with distortion  $\leq 2$  into the shortest path metric of a weighted graph  $G' = (V', E')$  with max degree 3 and treewidth- $k+1$ .*

*Proof.* Assume WLOG that all weights are greater than 1. Multiply the weight of each edge by  $n^2$ . Consider a width- $k$  tree decomposition  $(T, \{S_i\})$  of  $G$ . Expand each treenode  $S_i$  into a sequence of  $\binom{k+1}{2}$  treenodes so that each treenode has at most 1 edge.

Assume  $(x, y) \in S_i$ . Assume  $x$  is incident on 2 edges in ancestors of  $S_i$ . Replace  $S_i$  with 2 treenodes  $S_{i,1}, S_{i,2}$  as follows: Let  $S_{i,1}$  contain new node  $x'$  in addition to all the nodes of  $S_i$  and weight 1 edge  $(x, x')$ . Let  $S_{i,2}$  contain all the nodes of  $S_i$  except that  $x'$  replaces  $x$  and edge  $(x', y)$  replaces  $(x, y)$ . Now replace  $x$  with  $x'$  in all the original descendants of  $S_i$  which are now descendants of  $S_{i,2}$ . Repeat process for  $y$  if necessary. Repeat for descendent treenodes.

This process does not contract any distance and does not expand any distance by more than  $\frac{n^2+n}{n^2} \leq 2$  □

**Theorem B.2.** (From [3]) Given graph  $G = (V, E)$  with treewidth- $k$  and max degree  $d$ . Then  $G$  has domino treewidth  $\leq (9k + 7)d(d + 1) - 1$ .

**Lemma 2.7.** Any metric supported on a weighted graph  $G = (V, E)$  of treewidth- $k$  can be embedded with distortion 4 into a weighted graph with tree-bandwidth- $O(k)$ .

*Proof.* Given graph  $G = (V, E)$  with treewidth- $k$ , we can apply Lemma B.1 to get graph  $G'$  which by Theorem B.2 has domino treewidth  $\leq 108k + 191$ . This means that it has a tree decomposition  $(T, \{S_i\})$  of width  $\leq 108k + 191$  in which no node appears more than twice.

We will then take each node and replace it by 2 nodes connected by an edge so that no node appears more than once. The resulting graph has bounded tree-bandwidth except that it violates property 5. We can ensure that the graph has bounded tree-bandwidth by adding an edge between each violating point and a point in the parent treenode. If we then assign a weight equal to the distance in graph  $G'$  then adding these additional edges does not affect the shortest path distances in the graph. The resulting graph  $G''$  satisfies the properties claimed. □