Embedding k-Outerplanar Graphs into ℓ_1

Chandra Chekuri* Anupam Gupta† Ilan Newman‡ Yuri Rabinovich‡ Alistair Sinclair§

April 24, 2002

Abstract

We show that the shortest-path metric of any k-outerplanar graph, for any fixed k, can be approximated by a probability distribution over tree metrics with constant distortion, and hence also embedded into ℓ_1 with constant distortion. These graphs play a central role in polynomial time approximation schemes for many NP-hard optimization problems on general planar graphs, and include the family of weighted $k \times n$ planar grids.

This result implies a constant upper bound on the ratio between the sparsest cut and the maximum concurrent flow in multicommodity networks for k-outerplanar graphs, thus extending a classical theorem of Okamura and Seymour [26] for outerplanar graphs, and of Gupta et al. [17] for treewidth-2 graphs. In addition, we obtain improved approximation ratios for k-outerplanar graphs on various problems for which approximation algorithms are based on probabilistic tree embeddings. We also conjecture that our random tree embeddings for k-outerplanar graphs can serve as a building block for more powerful ℓ_1 embeddings in future.

*Lucent Bell Labs, 600 Mountain Avenue, Murray Hill NJ 07974. Email: chekuri@research.bell-labs.com

†Lucent Bell Labs, 600 Mountain Avenue, Murray Hill NJ 07974. Email: anupamg@research.bell-labs.com. Part of this work was done when the author was visiting the University of California, Berkeley, supported by a US-Israeli BSF grant #19999325.

‡Computer Science Department, University of Haifa, Haifa 31905, Israel. Email: {ilan, yuri}@cs.haifa.ac.il. Supported in part by a US-Israeli BSF grant #19999325.

§Computer Science Division, University of California, Berkeley CA 94720-1776. Email: sinclair@cs.berkeley.edu. Supported in part by NSF grants CCR-9820951 and CCR-0121555 and by a US-Israeli BSF grant #19999325.

1 Introduction

1.1 Background

Many optimization problems on graphs and related combinatorial objects involve some finite metric associated with the object. In particular, the shortest-path metric on the vertices of an undirected graph with nonnegative weights on the edges frequently plays an important role. While for general metric spaces such an optimization problem can be intractable, it is often possible to identify a subset of "nice" metrics for which the problem is easy. Thus, a natural approach to such problems — and one which has proved highly successful in many cases — is to *embed* the original metric into a nice metric, solve the problem for the nice metric, and finally translate the solution back to the original metric.

When the optimization problem is monotone and scalable in the associated metric (as is usually the case), it is natural to restrict one's attention to nice metrics which dominate the original metric, i.e., in which no distances are decreased. The maximum factor by which distances are stretched in the approximating metric is called the *distortion* of the embedding. Typically, the distortion translates more or less directly into the approximation factor that one has to pay in transforming the problem from one metric to the other, so obviously we seek an embedding with low distortion. The number of applications of this paradigm has exploded in the past few years, and it has become a versatile and standard part of the algorithm designer's toolkit: see the recent survey by Indyk [19] and the forthcoming book by Matoušek [23, Chapter 10]. These applications have also given impetus to the study of the underlying theory of finite metric spaces.

In this paper we will be concerned with embedding finite metric spaces into ℓ_1, i.e., real space endowed with the ℓ_1 (or Manhattan) metric. Low distortion embeddings into ℓ_1 have been recognized, along with embeddings into Euclidean space ℓ_2 and into low-dimensional ℓ_∞, to be of fundamental importance in applications of the above paradigm, as well as for the underlying theory. One of several compelling reasons for studying ℓ_1-embeddings comes from their intimate connection with the maxflow-mincut ratio in a multicommodity flow network. Namely, if every shortest-path metric on a given graph with arbitrary edge lengths can be embedded into ℓ_1 with distortion at most α, then the ratio between the sparsest cut and the maximum concurrent flow for any set of capacities and demands on the graph is bounded by α [22, 2, 17][1]. For more details on the sparsest cut problem, its relation to embeddings, and its application to the design of a host of divide-and-conquer algorithms, see the survey by Shmoys [31].

Equally important in algorithmic applications are certain special ℓ_1 embeddings known as embeddings into *random (dominating) trees*, whereby the given metric is approximated by a probability distribution over tree metrics. Since every tree metric can be embedded isometrically (i.e., exactly, or with distortion 1) into ℓ_1, approximating a metric by random trees with expected distortion α immediately yields an embedding into ℓ_1 with distortion α. As has been recognized in the work of Bartal and others [1, 5, 6], random tree embeddings have many additional applications to online and approximation algorithms that are not enjoyed by arbitrary ℓ_1 embeddings.

For *general* metrics the question of embeddability into ℓ_1 is essentially resolved: Bourgain [9] showed that any n-point metric can be embedded into ℓ_1 with $O(\log n)$ distortion, and a matching lower bound was established for the shortest-path metrics of unit-weighted expander graphs in [22]. For embeddings into random trees, a construction of Bartal [6] yields a distortion of $O(\log n \log\log n)$ for an arbitrary n-point metric.

However, tight bounds are still not known for many important classes of graphs, including planar graphs and graphs with bounded treewidth; many such restricted classes are conjectured to be embeddable with constant distortion. Indeed, the general question of how the topology of a graph affects its embeddability into ℓ_1, and into random trees, is one of the most important open issues in the area of metric embeddings. (See, e.g., the tutorial by Indyk in the last *FOCS* [19].) In addition to its inherent mathematical interest, this question impacts the design of approximation

[1] Indeed, the distortion of an optimal ℓ_1 embedding is *exactly* the worst cut-flow ratio for any choice of capacities and demands.

algorithms for many problems on restricted families of graphs and networks.

Some limited but interesting progress has been made on embedding restricted metrics into ℓ_1. Recently, Rao [27] showed that the shortest-path metric of any graph that excludes a $K_{r,r}$ is embeddable into ℓ_1 with distortion $O(r^3\sqrt{\log n})$. This beats the $\Omega(\log n)$ lower bound for general graphs for any constant r, and also gives $O(\sqrt{\log n})$ distortion embeddings for the classes of planar and bounded-treewidth graphs. However, Rao's approach (of first embedding these graphs into ℓ_2 and then using isometric embeddings of ℓ_2 into ℓ_1) was shown to be tight in [24], where a lower bound of $\Omega(\sqrt{\log n})$ distortion was shown for embedding even treewidth-2 (and hence also planar) graphs into ℓ_2.

Approaching the question from the other direction, a celebrated theorem of Okamura and Seymour [26] implies that any *outerplanar* metric can be embedded isometrically into ℓ_1.[2] However, it has been shown that outerplanar graphs are essentially the only graphs (with the exception of K_4) that are isometrically embeddable into ℓ_1 [25]. More recently, Gupta et al. [17] showed a constant distortion embedding into ℓ_1 for treewidth-2 graphs (which are essentially series-parallel graphs, and hence also planar). This was the first natural class of graphs shown to be embeddable with constant distortion strictly larger than 1. (For example, the graph $K_{2,3}$ has treewidth 2 but is not isometrically embeddable; see [12, Example 6.3.2] for a simple proof of this fact.)

Some, but not all of the above results carry over to the more restrictive setting of embedding into random trees. In [17] it is shown how to embed outerplanar graphs into random trees with small constant distortion (note that the isometric embedding of Okamura and Seymour is *not* a tree embedding); on the other hand, in the same paper it is shown that even series-parallel graphs incur a distortion $\Omega(\log n)$ for tree embeddings. Despite this limitation, it is worth pointing out that the random tree embeddings of outerplanar graphs played a key role in the development of constant distortion ℓ_1 embeddings of series-parallel graphs in [17]: the trick was to combine the special structure of the tree embeddings with judicious use of random cuts.

1.2 Results

In this paper, we extend the above line of research to a much wider class of planar graphs, namely k-outerplanar graphs for arbitary constant k. Informally, a planar graph is k*-outerplanar* if it has an embedding with disjoint cycles properly nested at most k deep. A formal definition is given in Section 2, while Figure 4.2 shows a simple example; a canonical example of a k-outerplanar family is the sequence of $k \times n$ rectangular grids. k-outerplanar graphs play a central role in polynomial time approximation schemes for many NP-hard optimization problems on general planar graphs (see, e.g., [4]). Our main result is the following:

Theorem *Any shortest-path metric of a k-outerplanar graph, for any fixed k, can be embedded into a probability distribution over trees, and hence into ℓ_1, with constant distortion. Moreover, such an embedding can be found in randomized polynomial time.*

Thus, not only do such graphs embed well into ℓ_1, but they even embed well into dominating trees. This is in contrast to the lower bound of $\Omega(\log n)$ for treewidth-2 graphs [17].

Our result immediately implies a constant maxflow-mincut ratio for arbitrary multicommodity flow problems on k-outerplanar graphs. This is the first progress in this direction in the two decades since the Okamura-Seymour result [26], which proves a ratio of 1 for 1-outerplanar graphs. Additionally, because our ℓ_1-embeddings are in fact random tree embeddings, we also obtain as an immediate byproduct improved approximation ratios for a number of algorithms for problems on k-outerplanar graphs, including the buy-at-bulk problem [3] and the group Steiner problem [15]. For any fixed k, the improvement in each case is by a $\Omega(\log n)$ factor; we defer the details to the full version of the paper.

[2] Their result deals more generally with the cut/flow ratio in planar networks where all terminals lie on a single face; this and other results where restrictions are placed on *both* the flow network *and* the demand structure can be found in surveys by Frank [14] and Schrijver [29].

We should also note that our result is the first demonstration of constant distortion ℓ_1 embeddings for a natural family of graphs with arbitrarily large (but bounded) treewidth[3]. Indeed, k-outerplanar graphs are a natural parameterized family of planar graphs having bounded treewidth. (Note that although all treewidth-2 graphs are planar, treewidth-3 graphs include non-planar examples such as $K_{3,3}$.)

Finally, recall that constant distortion random tree embeddings of 1-outerplanar graphs were a key ingredient in the construction of good ℓ_1 embeddings of series-parallel graphs in [17]. We are therefore optimistic that, with the addition of suitably chosen cuts, our new tree embeddings of k-outerplanar graphs may become a building block for constant distortion ℓ_1 embeddings of wider classes of graphs, such as bounded treewidth graphs or planar graphs.

1.3 Techniques

We start with the approach of trying to extend the random tree embeddings of outerplanar graphs [17] to 2-outerplanar graphs. We do not know a way to solve this problem directly. The first main idea in the paper is to identify a *sub-class* of 2-outerplanar graphs that are easier to embed, namely *Halin graphs*. Informally, a "Halin graph" is obtained by embedding a tree in the plane and attaching a cycle around the leaves. (The formal definition can be found in Section 2). Halin graphs are useful for the following reason. Given a 2-outerplanar graph, we can use the random embedding of [17] to embed the inner outerplanar graphs obtained by removing the outer face(s) into a random tree (in fact a forest). If we now add the outer face to this random tree we get a graph which is (very similar to) a Halin graph. Hence, if we can embed Halin graphs we can embed 2-outerplanar graphs. We are then able to extend this approach to embed any k-outerplanar graph by peeling off the outer layer and recursively embedding the inner layers.

The second main idea is a technique for embedding Halin graphs. We note that even this deceptively simple subclass of 2-outerplanar graphs had so far resisted attempts at constant distortion embeddings. This is derived by a subtle modification of the algorithm of Gupta [16] which showed how to remove *Steiner vertices*[4] from a tree metric with only a constant factor distortion in distances between the remaining vertices. Though seemingly unrelated to our problem (since we have no Steiner vertices), this algorithm can nonetheless be applied (with suitable modifications) to the tree in the Halin graph, with the effect of reducing the Halin graph to an outerplanar graph on its leaves. This we can once again embed into random trees using [17].

The rest of the paper is organized as follows. We first fix notation and give essential definitions in Section 2. In Section 3 we show how to embed Halin graphs into random trees with constant distortion. This is extended to obtain constant distortion embeddings for all k-outerplanar graphs in Section 4. In the interests of clarity of exposition, we make no attempt to optimize the constants that arise in the various steps of our procedure.

2 Notation and Preliminaries

Metrics: For general background on finite metrics and embeddings, see the book of Deza and Laurent [12]. Given two metric spaces, (V, ν) and (W, μ), and a map $f : V \to W$, define the following quantities:

$$\|f\| \;=\; \max_{x,y \in V} \frac{\mu(f(x), f(y))}{\nu(x,y)} \;; \quad \|f^{-1}\| \;=\; \max_{x,y \in V} \frac{\nu(x,y)}{\mu(f(x), f(y))}$$

We say that f has *contraction* $\|f^{-1}\|$, *expansion* $\|f\|$ and *distortion* $D(f) = \|f\| \cdot \|f^{-1}\|$. The *distortion between* μ *and* ν is at most r if there exists $f : V \to W$ with $D(f) \le r$. We often consider two metrics μ and ν over the same

[3]The maximum treewidth among k-outerplanar graphs is $\Theta(k)$.

[4]Given an induced metric defined on a subset of vertices of a graph, we call the vertices not belonging to this subset the *Steiner vertices*. Although we are interested only in the metric space induced on the non-Steiner vertices, the Steiner vertices might be necessary in order to define the distances between the non-Steiner vertices.

vertex set V; in such cases, we assume that f is the identity map. Metric μ is said to *dominate* ν if for all $x, y \in V$, $\mu(x,y) \geq \nu(x,y)$.

Let $G = (V, E)$ be an undirected graph. A metric (V, μ) is *supported on* (or *generated by*) G if it is the shortest-path metric of G w.r.t. some nonnegative weighting of the edges E. Given a graph G with edge weights $w(\cdot)$, d_G denotes the shortest path metric of G, and we assume that the edge weights satisfy $w(e) = d_G(x,y)$ for $e = \{x,y\} \in E$ unless otherwise stated.

For $S \subseteq V$, the *cut metric* $\delta_S(x,y)$ is defined to be 1 if $|S \cap \{x,y\}| = 1$, and 0 otherwise. It is known that a metric is embeddable into ℓ_1 iff it can be written as a non-negative linear combination of cut metrics [12].

A metric d_G supported on a graph G is *α-probabilistically approximated* by a distribution $\mathcal{D}$ over trees if **(1)** each tree T in the distribution $\mathcal{D}$ has $V(G) \subseteq V(T)$; **(2)** for all $x, y \in V(G)$ and T in the distribution, d_T *dominates* d_G, i.e., $d_G(x,y) \leq d_T(x,y)$; and **(3)** for all $x, y \in V(G)$, the expected distance $\mathrm{E}_{\mathcal{D}}[d_T(x,y)] \leq \alpha \cdot d_G(x,y)$. We shall also refer to this as an *embedding of G with distortion α into random trees*. (The fact that the distortion is only in expectation will often not be mentioned.) It is known that general graphs can be embedded into random trees with distortion $O(\log n \log\log n)$ [20, 1, 5, 6, 10, 11].

Graph-Theoretic Terms: Most graph-theory concepts which we use, such as treewidth, minors, and planarity, are covered in standard text-books (see, e.g., [13, 33]).

An embedding of a graph G is *outerplanar* (or *1-outerplanar*) if it is planar, and all vertices lie on the unbounded face. An embedding of a graph G is *k-outerplanar* if it is planar, and deleting all the vertices on the unbounded face leaves a $(k-1)$-outerplanar embedding of the remaining graph. A graph is k-outerplanar if it has a k-outerplanar embedding. It is known that a k-outerplanar graph has treewidth $\leq 3k-1$ [8, 28]; other properties of these graphs and related concepts can be found in [4, 8]. Given a planar graph, a k-outerplanar embedding for which k is minimal can be found in polynomial time [7].

A *Halin graph* [18] is obtained by taking a planar embedding of a tree $T = (V, E)$ and attaching a cycle $C = (U, E_c)$ around the leaves of the tree (in order). L denotes the set of leaves of T, and hence $V \cap U = L$. (Note that there may be vertices on the cycle that are not leaves of T.) A Halin graph $G = (V \cup U, E \uplus E_c)$ is 2-outerplanar and has treewidth 3. Many algorithmic problems can be solved efficiently on these graphs (see, e.g., [32] and the references therein).

3 Embedding a Halin Graph

Given a Halin graph, we will embed it into random trees thus: we first take the tree $T = (V, E)$ from the Halin graph and process it to give a random dominating tree $T^{(1)}$, which approximates distances in T to within a constant (in expectation). Furthermore, $T^{(1)}$ has a specific structure: it consists of a tree $T'' = (L, E'')$ on just the *leaves* L of the original tree T, and the rest of the vertices in $V \setminus L$ are attached to vertices in T''. Also, the tree T'' is a minor of T, and so attaching the cycle C back to the vertices in T'' gives us an outerplanar graph. This outerplanar graph is then embedded into a random tree using known techniques [17] to give the following theorem, which is the main result of this section:

Theorem 3.1 *Any metric generated by a Halin graph can be embedded into a distribution over dominating trees with constant distortion.*

3.1 Processing the tree

Let us assume that the tree T has a *root* vertex $r \in (V \setminus L)$, which imposes an ancestor-descendant relation between the vertices in V. Each vertex v naturally defines a tree $T(v)$, namely the subtree induced by the vertices that are descendents of v. For a vertex v, let $l(v)$ be the leaf in $T(v)$ *closest* to v, and $h(v)$ be the distance of v from $l(v)$ in T. Note that these functions are fixed given the input tree T. The processing algorithm works in two parts.

- The first step of the algorithm, given in Section 3.1.1, returns a tree $T^{(1/2)}$. This tree consists of a tree T' defined on the vertices of L and some extra (or *Steiner*) vertices, and the vertices of $V \setminus L$ hang off the vertices of T' in the form of (possibly several) subtrees. This is done incurring a constant expected distortion.
- Note that the previous step was almost what we wanted — we just have to get rid of the Steiner vertices. The second part, given in Section 3.1.2, eliminates the Steiner vertices of T' by contracting some of its edges, thus converting $T^{(1/2)}$ into $T^{(1)}$. This process is shown to incur a further distortion of a constant factor.

3.1.1 Processing I: Getting the tree $T^{(1/2)}$

In this section, we will show how to convert the tree T into the tree $T^{(1/2)}$ while incurring only a constant distortion. The algorithm **Process-Tree** to perform this processing cuts off a subtree $\hat{T}_0$ of T which contains the root but not the leaves, recursively acts on the subtrees thus created, makes a new root vertex and adds edges from it to the roots of each of the processed subtrees, and finally hangs $\hat{T}_0$ off this new root. (See Figure 3.1.)

Before we make **Process-Tree** concrete, we define the auxiliary procedure **Cut-Midway** which cuts a *random* set of edges to separate the root r from all the leaves of T. It returns a special tree $\hat{T}_0$ containing the root r of T and none of its leaves, and a set of subtrees T_i (for $1 \leq i \leq t$), each rooted at some vertex r_i. We say that an edge e is at a distance d from a vertex r if e is in the cut defined by the set of vertices whose distance from r is at most d.

Algorithm Cut-Midway(T)
    while there is a path from r to a leaf in T
        let $d \leftarrow$ distance to closest leaf
        let $S(d) \leftarrow$ set of leaves at distance $\in [d, 2d)$ from r in T
        let $T(d)$ be the union of paths from r to vertices in $S(d)$
        choose $D \in_R [d/2, 3d/4)$ uniformly
        $E(d) \leftarrow$ edges in $T(d)$ at distance D from r
        delete edges in $E(d)$ from T
    end while
    $\hat{T}_0 \leftarrow$ component of T containing root r but no leaves of T
    $T_1, T_2, \ldots, T_t \leftarrow$ other components of T
    let $d_i \leftarrow$ value of d when edge connecting r to T_i was cut.
    return $(\hat{T}_0; \langle T_1, d_1\rangle, \langle T_2, d_2\rangle, \ldots, \langle T_t, d_t\rangle)$.

Now we can formally state **Process-Tree**:

Algorithm Process-Tree (T)
    apply **Cut-Midway**(T) to get $(\hat{T}_0, \langle T_1, d_1\rangle, \langle T_2, d_2\rangle, \ldots, \langle T_t, d_t\rangle)$
    let r' be a new vertex, called the "Steiner twin" of r
    attach r' to r with edge of length $d_0 = h(r)$
    for $1 \leq i \leq t$ // *We don't have to work on* $\hat{T}_0$
        if T_i is just a single vertex x (hence $x \in L$) **then** $T_i^{(1/2)} \leftarrow T_i$
            else $T_i^{(1/2)} \leftarrow$ **Process-Tree** (T_i)
        let r_i' be root of $T_i^{(1/2)}$ // r_i' *is the Steiner twin of* r_i*, the root of* T_i
        add edge (r', r_i') with length $3d_i$
    end for
    return tree $T^{(1/2)}$ with r' as its root

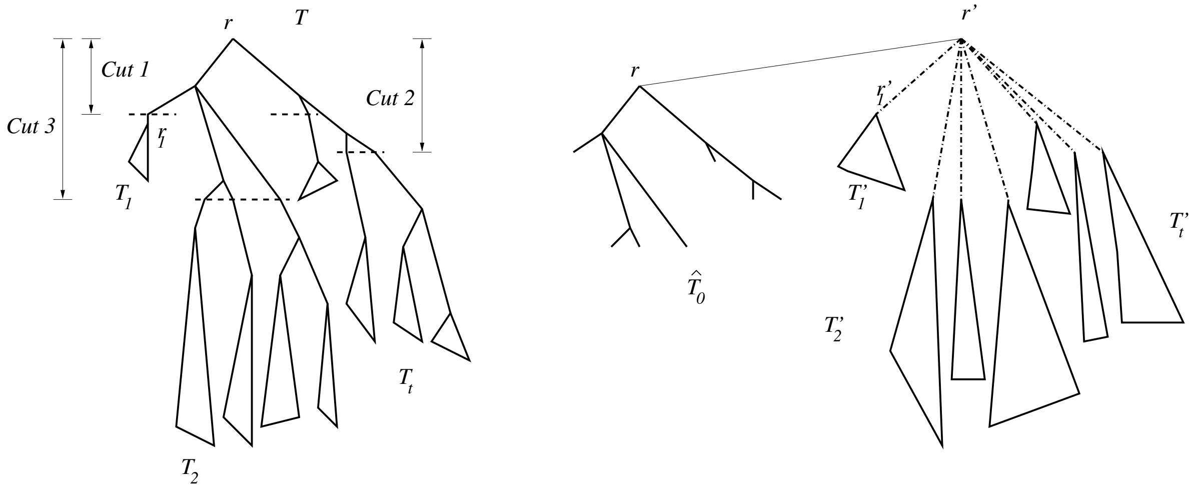

Figure 3.1: *One call of the procedure* **Process-Tree**.

Recall that we had mentioned that $T^{(1/2)}$ would have a portion called T'; this is formed by the new edges added between r' and r'_i (for $1 \le i \le t$) during the various recursive calls to **Process-Tree**. (Note that this does not include the edges added between r' and r, i.e., between the original roots and their Steiner twins.) Hence T' includes all the leaves of T, plus all the Steiner twins created. For an example, see Figure 3.1, where **Cut-Midway** performed three cuts, and **Process-Tree** resulted in the tree on the right. The solid edges belong to T, the dashed ones to T', and the edge (r, r') is shown as a faint line.

Let us call an edge a *candidate to be cut* at some step if it has a non-zero probability of being cut at that step. We can now show the following bound on the expected distortion incurred by the above procedure:

Theorem 3.2 *The (expected) distortion introduced by procedure* **Process-Tree** *is at most* 25.

Proof: Before we prove this, let us give a high-level sketch. It can be verified to see that distances are never contracted by **Process-Tree**, and hence it suffices to bound the expected expansion. We show this via two lemmas: firstly, Lemma 3.3 shows that an edge is a candidate to be cut on at most two (consecutive) occasions. Lemma 3.4 then shows that when an edge is a candidate to be cut, it suffers only a constant expected expansion. Combining these two results then gives us the result.

Lemma 3.3 *No edge is a candidate for cutting more than twice during the entire run of the algorithm* **Process-Tree**.

Proof: Let $e = (u, v)$ be an edge, where u is the ancestor of v. Consider the first instant when an edge e is a candidate to be cut in a call to **Cut-Midway**. Let r be the root at this point, and d^* be the value of the parameter d in the **while** loop of this call to **Cut-Midway**. In this call of **Cut-Midway**, it is clear that e cannot be considered again. Indeed, after the cut, e will not lie on any path from r to a leaf. A fact that will be useful later is that the portion of e that lies in the distance interval $[d^*/2, 3d^*/4)$ from r is $(\min(d_T(r, v), 3d^*/4) - \max(d_T(r, u), d^*/2))$, and that value multiplied by $4/d^*$ is the probability that e is cut at this time.

The edge e will never be considered again if the cut fell "below" v, or if it passed through e, so let us assume that the cut was above u and e lies in one of the trees T_i with root r_i. Clearly, it lies in some path from r_i to a leaf, and hence it will be part of the tree $T(d^{**})$ at some point in the call to **Cut-Midway** from r_i.

We claim that the cut made at this point will lie below u; i.e., $d^{**}/2 \ge d_T(r_i, u)$. Indeed, such a cut made from r_i at a distance at least $d^{**}/2 \ge h(r_i)/2$ from it, where $h(r_i) \ge d^* - d_T(r, r_i)$. Hence taking distances from r, this cut is at distance at least $d_T(r, r_i) + h(r_i)/2 \geq \frac{1}{2}(d_T(r, r_i) + d^*) \geq 3d^*/4$. But this distance is greater than $d_T(r, u)$, and hence u always lies above this next cut. Thus, when this next cut is made, either e will be deleted (if v fell below this cut), or the cut will fall below v and the edge e will never again be a candidate to be cut, proving the lemma.

Before we end, let us note that the portion of e that lies in distance interval $[d^{**}/2, 3d^{**}/4)$ is disjoint from the portion considered earlier, and has a length of at most $\max(d_T(r, v) - 3d^*/4, 0)$. As before, multiplying this by $4/d^{**}$ gives the probability that e is cut if it is considered a second time. ∎

Lemma 3.4 *If an edge e is cut by* **Cut-Midway** *with parameter d_i, its expected length is at most $6d_i - \ell_e$, where ℓ_e is the length of e.*

Proof: Consider an edge $e = (u, v)$ of length ℓ_e which is cut in some iteration of **Cut-Midway**, and let d_i be the value of the parameter d at this point. Consider the distance $d_{T^{1/2}}(u, v)$ between u and v in the resulting tree $T^{(1/2)}$.

The vertex u will be in $\hat{T}_0$ and the vertex v is the root of T_i for some i and hence will be in $T_i^{(1/2)}$ when T_i, rooted at $r_i = v$, is processed. From the description of **Process-Tree** we see that $d_{T^{(1/2)}}(u, v) = d_{T^{(1/2)}}(u, r_i)$ can be expressed as $d_T(u, r) + d_{T^{(1/2)}}(r, r') + d_{T^{(1/2)}}(r', r'_i) + d_{T^{(1/2)}}(r'_i, r_i)$. From our construction, $d_{T^{(1/2)}}(r, r') = h(r)$ and $d_{T^{(1/2)}}(r', r'_i) = 3d_i$ and $d_{T^{(1/2)}}(r'_i, r_i) = h(r_i)$. We observe that $h(r) \leq d_i$ for all i, and that $h(r_i) \leq 2d_i - d_T(r, r_i)$. The latter is true because for e to be cut, r_i is on the path from r to a leaf in T of length at most $2d_i$. Putting these observations together we obtain that

$$
\begin{aligned}
d_{T^{(1/2)}}(u, r_i) &= d_T(u, r) + d_{T^{(1/2)}}(r, r') + d_{T^{(1/2)}}(r', r'_i) + d_{T^{(1/2)}}(r'_i, r_i) \\
&\leq d_T(u, r) + h(r) + 3d_i + (2d_i - d_T(r, r_i)) \\
&\leq d_T(u, r) + d_i + 3d_i + (2d_i - d_T(r, r_i)) \\
&\leq d_T(u, r) - d_T(r, r_i) + 6d_i = 6d_i - \ell_e.
\end{aligned}
$$

∎

Now we complete the proof of Theorem 3.2: by Lemma 3.3, the edge $e = (u, v)$ is cut at most twice. The first time it is considered, it is cut with probability $p_1 = (\min(d_T(r, v), 3d^*/4) - \max(d_T(r, u), d^*/2)) \times 4/d^*$, and the expected length is at most $6d^* - \ell_e$. The second time the chance is $p_2 = (\max(d_T(r, v) - 3d^*/4, 0)) \times 4/d^{**}$, and the expected length is $6d^{**} - \ell_e$. Finally, with probability $(1 - p_1 - p_2)$, the length remains ℓ_e. Thus the expected distance between u and v is at most

$$
\begin{aligned}
& 6d^* \, p_1 + 6d^{**} \, p_2 + (1 - 2p_1 - 2p_2) l_e \;\leq\; 6(d^* \, p_1 + d^{**} \, p_2) + l_e \\
\leq\; & 24 \, [\min(d_T(r, v), 3d^*/4) - \max(d_T(r, u), d^*/2) + \max(d_T(r, v) - 3d^*/4, 0)] + l_e \\
\leq\; & 24 \, l_e + l_e \leq 25 \, l_e,
\end{aligned}
$$

which implies an expected distortion of 25, and proves the theorem. ∎

We close this subsection with a further observation about the tree T' constructed by the procedure Process-Tree.

Claim 3.5 *The tree T' constructed as above is a minor of the tree T.*

Proof: In each call to **Process-Tree**, we progressively construct T' by removing the tree T_0 and replacing it with a star connecting r' to the various r_i (for $1 \leq i \leq t$). But this star could equivalently be obtained by contracting all but the leaf edges of the tree T_0. (Of course, we are placing new lengths on these edges, but this does not affect the structure.) ∎

This claim also shows that the tree T' with the cycle around its leaves is still a Halin graph, since Halin graphs are closed under taking minors.

3.1.2 Processing II: Removing the Steiner vertices

In this section, we remove the Steiner vertices in the tree T' that were created during runs of **Process-Tree**, giving us a tree T''. (Since $T^{(1/2)}$ consists of T' with several subtrees attached to it via cut-edges, attaching those subtrees to T'' will give us a new tree $T^{(1)}$.) The argument in this section is similar in spirit to that in [16]. The Steiner twin vertices from $T^{(1/2)}$ are removed in the same order in which they were created. Consider r', the root of T'; it was created as the Steiner twin of vertex $r \in T$. We now identify all vertices on the path between r' and $l(r)$ with $l(r)$. This process is performed on each of the Steiner twin vertices in turn (in order of their creation), causing each of them to be identified with some vertex in $L \subseteq C$. Call the resulting tree $T^{(1)}$. This has the vertex set V, since we removed all the Steiner vertices we created in the previous section. The following lemma proves the main result of this section:

Lemma 3.6 *This edge-contraction procedure ensures that the distance between each pair of vertices of V in $T^{(1)}$ is no shorter than its distance in T.*

Proof: To show that there is no contraction, it suffices to check that no *edge* in $T^{(1)}$ is shorter than the distance between its endpoints in T. There are just three kinds of edges remaining in $T^{(1)}$: those which belong to the trees $\hat{T}_0$ in the various invocations of **Process-Tree**, those between some r and $l(r)$,[5] and those between $l(r)$ and $l(r_i)$. Note that the edges of this last type are the only edges that exist between $l(r_a)$ and $l(r_b)$, since such edges (w.l.o.g.) must be caused by r_a being the root at some invocation of **Process-Tree** and r_b being one of the r_i's created at this step, and r_a later being identified with $l(r_a)$.

Clearly, the edges in the trees $\hat{T}_0$ are not changed at all. Now consider an edge between a vertex $l(r)$ and r. (This edge is created since r' was identified with $l(r)$.) The length of this edge in T'' is just $h(r)$, which is also the distance between $l(r)$ and r in T. Finally, for an edge e between $l(r)$ and $l(r_i)$ in $T^{(1)}$, the length is just $6d_T(r, r_i)$. However, the distance between these points in T is at most $d_T(r, l(r)) + d_T(r, l_{T_i}(r_i))$ which we upper bound next. Let d^* be the value of d when r_i was separated from r in the procedure **Cut-Midway**. Then it follows that $d_T(r, l(r)) = h(r) \leq d^*$ and $d_T(r, l_{T_i}(r_i)) = d_T(r, r_i) + h_{T_i}(r_i) \leq 2d^*$. Hence the distance between $l(r)$ and $l(r_i)$ in T is at most $3d^*$, however $d_T(r, r_i) \geq d^*/2$, therefore the distance is at most $6d_T(r, r_i)$. ■

3.2 Wrapping it all up

Since the distances in $T^{(1)}$ are at least those in T, and at most 25 times those in T, this gives us a total (expected) distortion of 25. Furthermore, we can now add back the cycle C on the vertices of L, giving the graph $G^{(1)}$. This consists of an outerplanar graph on L, along with vertices of $T - L$ in the form of subtrees attached to vertices of L. To see that $T'' \cup C$ forms an outerplanar graph, note from Claim 3.5 that T' was a minor of T, and hence T'' obtained by contracting some edges in T' still leaves us with a minor of T, so the planar embedding of $T \cup C$ induces a planar embedding of $T'' \cup C$. Furthermore, T'' is defined on the vertices of $L \subseteq U$, so all the vertices lie on the outer face.

But now we can invoke the procedure of [17, Theorem 5.2] to get a random subtree of $G^{(1)}$ which approximates distances (in expectation) in $G^{(1)}$ to within a factor of 8, and hence those in G to within a factor of $8 \cdot 25 = 200$. This completes the proof of Theorem 3.1.

4 On to k-outerplanar graphs

In this section, we extend the construction of the previous section to k-outerplanar graphs. Recall that these are graphs embeddable in the plane such that removing the vertices on the outermost face k times deletes the graph. Before we begin, let us state two simple lemmas (whose proofs we omit) that allow us to replace a subgraph by its tree embedding, and to give embeddings of graphs in terms of their blocks.

[5] These edges were added between r and r', and the latter has been identified with $l(r)$.

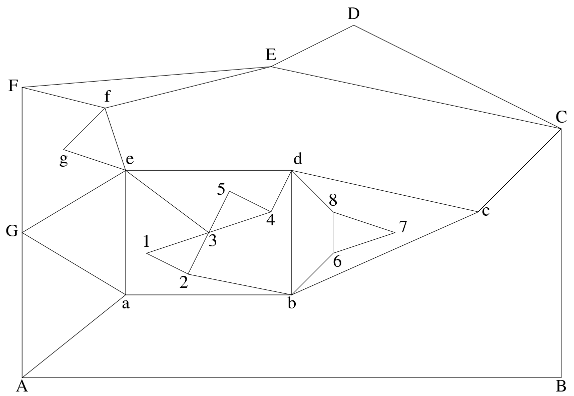

Figure 4.2: *A 3-outerplanar graph from [4]. The layers are* A-G*,* a-g*, and* 1-8*.*

Proposition 4.1 *Let $H = (V_H, E_H)$ be a subgraph of $G = (V, E)$. Let $H' = (V_H, E_{H'})$ be a graph on V_H such that $d_H(u, v) \leq d_{H'}(u, v) \leq \alpha \cdot d_H(u, v)$ for all $u, v \in V_H$. Then in the graph $G' = (V, E - E_H + E_{H'})$, $d_G(u, v) \leq d_{G'}(u, v) \leq \alpha \cdot d_G(u, v)$ for all $u, v \in V$.*

Proposition 4.2 *Let the graph G have a cut-edge whose removal results in a tree T and a graph H. If H can be probabilistically approximated by tree metrics with distortion α, then so can G.*

The main result of this section, and of the paper is the following:

Theorem 4.3 *There is a universal constant c such that any metric generated by a k-outerplanar graph can be embedded into random trees with distortion at most c^k.*

Proof: The proof is by induction on k; however, the induction hypothesis required is stronger than the statement of the theorem. We will assume that $G = (V, E)$ is given along with its k-outerplanar embedding, and $F_0(G)$ is the set of vertices on the outer face of G. (In the sequel, we will often abuse notation and blur the distinction between a face and the vertices that lie on it.)

Induction Hypothesis: Let $G = (V, E)$ be a connected k-outerplanar graph with $F_0(G)$ as the outer face in some k-outerplanar embedding of G. Then, the shortest-path metric of G can be probabilistically approximated by a collection of trees on V with expected distortion at most c^k such that each tree $T_i = (V, E_i)$ in the distribution has the following properties:

- **(i)** the subgraph of T_i induced by $F_0(G)$ is a minor of G; and
- **(ii)** the subgraph of T_i induced by $V(G) - F_0(G)$ is a forest, and each tree in the forest is connected to $F_0(G)$ by a single edge.

Informally, we will require that the random tree for G be embeddable in the plane even when the vertices on the outer face of G are "pinned down" to the plane. Clearly, if the trees T_i are subgraphs of G, then this is trivially satisfied. The reader can verify that the Halin graph embedding of Section 3 produces graphs which, though they are not subgraphs, nevertheless satisfy the above property.

The base case for the induction is $k = 1$ when G is an outerplanar graph. For outerplanar graphs [17, Theorem 5.2] shows an embedding of G into trees that are *subgraphs* of G with constant distortion. Hence the auxiliary conditions are trivially satisfied.

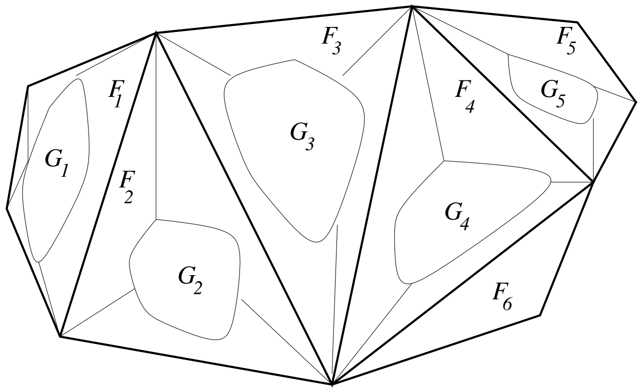

Figure 4.3: *Partitioning of a k-outerplanar graph G into $(k-1)$-outerplanar graphs $G_1, \ldots, G_5$. The bold lines indicate G_F, the graph induced by the outer face.*

For the induction step, let us assume that G is two vertex connected, else we can work with the blocks of G independently. Let G_F be the subgraph of G induced by $F_0(G)$, the vertices on its outer face; clearly G_F is an outerplanar graph. (See Figure 4.3.) Let $F_1, F_2, \ldots, F_\ell$ be the internal faces of G_F, V_i the subset of $V - F_0(G)$ lying inside the face F_i, and G_i the induced graph on V_i. We assume without loss of generality that G_i is connected for otherwise we can work with its connected components separately. We make the following assumption for technical reasons: for any vertex $v \in F_i$ there is at most one vertex $u \in V_i$ such that $(u, v) \in E(G)$. This is without loss of generality, since if this does not hold for a vertex $v \in F_i$, we can split v into a path of vertices (with the edges between them of length 0) and connect each one to a unique vertex of V_i without violating planarity. Note the following fact, which allows us to use the induction hypothesis:

Fact 4.4 *For $1 \leq i \leq \ell$, G_i is a $(k-1)$-outerplanar graph.*

Now applying the induction hypothesis, each G_i can be c^{k-1}-probabilistically approximated by trees satisfying conditions **(i)** and **(ii)**. We now give a procedure to extend the embeddings of the various G_i to an embedding of G. For $1 \leq i \leq \ell$, we pick a tree T_i from the distribution over tree metrics for G_i. Let G' be the graph obtained by adding the vertices of $F_0(G)$ and the edges incident to them (in G) to the trees $T_1, \ldots, T_\ell$. Proposition 4.1 implies that the metric induced by G' is within an expected c^{k-1} distortion of d_G, and hence approximating G' by tree metrics with an expected distortion of c will prove the induction hypothesis for G.

Let T_i' be the subtree of T_i induced by $F_0(G_i)$. The fact that it is a subtree is guaranteed by condition **(i)** of the hypothesis; in fact T_i' is a minor of G_i. Furthermore, $V_i - F_0(G_i)$, i.e., vertices of G_i not in T_i', induces a forest in T_i that is connected via cut-edges to T_i'. Also note that there are no edges between $F_0(G)$ and $V_i - F_0(G_i)$, since the graph is planar, and the layer $F_0(G_i)$ separates these two sets of vertices. Using Proposition 4.2, we can eliminate vertices in $V_i - F_0(G_i)$ (for $1 \leq i \leq \ell$) from G'. It now suffices to embed the resulting graph, which we call $\mathrm{core}(G')$, into trees with expected distortion at most c.

The key claim that essentially reduces this problem to the embeddings of Halin graphs given in the previous section is the following:

Claim 4.5 *Let G_i' be obtained by taking the tree T_i', and adding the vertices F_i and all the edges incident on F_i in $G[V_i \cup F_i]$. Then G_i' is a Halin graph.*

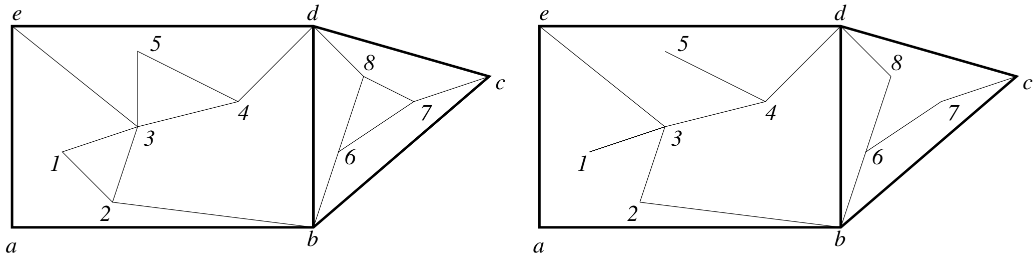

Figure 4.4: *Returning from the induction: the bold lines denote G_F.*

Proof: By the induction hypothesis, the tree T_i' is a minor of G_i, and hence the planar embedding of G_i induces a natural planar embedding of T_i'. Furthermore, from our earlier assumption, each vertex of F_i has at most one edge to T_i'; let E_i be the set of these edges. It follows that T_i' along with these edges E_i still forms a tree. Finally, the edges along the face F_i form a cycle around this tree, and hence G_i' is a Halin graph as claimed. ■

Note that our current graph $\mathrm{core}(G')$ is simply $\bigcup_i G_i'$. Since each T_i' is a minor of G_i, we obtain the following result.

Proposition 4.6 *The graph core(G') is a minor of G.*

Now since each G_i' is a Halin graph (with F_i as its outer face), we can apply the procedure of Section 3 to it. The resulting graph, which we call G_i'', will be an outerplanar graph on F_i, with the vertices of $G_i' - F_i$ inducing a forest, the trees of which are connected to vertices of F_i via cut-edges. Using Proposition 4.2 again, we can remove these hanging trees to obtain the graph $\mathrm{core}(G_i'')$.

Note that the procedure in Section 3 guarantees that $\mathrm{core}(G_i'')$ is a minor of G_i'. Furthermore, each $\mathrm{core}(G_i'')$ is an outerplanar graph on the face F_i of the outerplanar graph G_F. These two facts together imply that $H = \bigcup_i \mathrm{core}(G_i'')$ is also an outerplanar graph. We can embed H into subtrees of H with constant expected distortion following [17, Theorem 5.2]. (We assume that the distortion is at most c by choosing c sufficiently large.) This establishes that G can be embedded with expected distortion at most c^k.

It just remains to show that the conditions **(i)** and **(ii)** are met for the trees produced by this procedure. The final step is an embedding of H whose vertex set is $F_0(G)$. It can be seen that H is a minor of G; indeed, Proposition 4.6 shows that $\mathrm{core}(G') = \bigcup_i G_i'$ is a minor of G, and as observed above, each G_i'' is a minor of G_i'. Finally, using the procedure in [17] to embed H gives subtrees of H which are clearly minors of H, and thus of G. Hence our procedure guarantees that each random tree, when restricted to the vertices of $F_0(G)$, is a minor of G, thus establishing condition **(i)**. Finally, note that the procedure removes vertices only when Proposition 4.2 is applied, i.e., if the vertices induce a tree connected via a cut-edge to the rest of the graph. This implies condition **(ii)** of the induction hypothesis, thus completing the proof. ■

Acknowledgments

We would like to thank Amit Chakrabarti and Amit Kumar for useful discussions.

References

[1] Noga Alon, Richard M. Karp, David Peleg, and Douglas B. West. A graph-theoretic game and its applications to the k-server problem. *SIAM Journal on Computing*, 24(1):78–100, 1995.

[2] Yonatan Aumann and Yuval Rabani. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM Journal on Computing*, 27(1):291–301, February 1998.

[3] Baruch Awerbuch and Yossi Azar. Buy-at-bulk network design. In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science*, pages 542–547, 1997.

[4] Brenda S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. Assoc. Comput. Mach.*, 41(1):153–180, 1994.

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

[6] Yair Bartal. On approximating arbitrary metrics by tree metrics. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 161–168, 1998.

[7] Daniel Bienstock and Clyde L. Monma. On the complexity of embedding planar graphs to minimize certain distance measures. *Algorithmica*, 5(1):93–109, 1990.

[8] Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209:1–45, 1998.

[9] Jean Bourgain. On Lipschitz embeddings of finite metric spaces in Hilbert space. *Israel Journal of Mathematics*, 52(1-2):46–52, 1985.

[10] Moses Charikar, Chandra Chekuri, Ashish Goel, and Sudipto Guha. Rounding via trees: deterministic approximation algorithms for group Steiner trees and k median. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 114–123, 1998.

[11] Moses Charikar, Chandra Chekuri, Ashish Goel, Sudipto Guha, and Serge A. Plotkin. Approximating a finite metric by a small number of tree metrics. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 379–388, 1998.

[12] Michel Marie Deza and Monique Laurent. *Geometry of Cuts and Metrics*. Springer Verlag, 1997.

[13] Reinhard Diestel. *Graph Theory*. Springer-Verlag, New York, 1997.

[14] András Frank. Packing paths, circuits, and cuts – a survey. In Bernhard Korte, László Lovász, Hans Jürgen Prömel, and Alexander Schrijver, editors, *Paths, Flows and VLSI-Layout*, pages 47–100. Springer-Verlag, 1990.

[15] Naveen Garg, Goran Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. *Journal of Algorithms*, 37(1):66–84, 2000. (Preliminary version in: *9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 253–259, 1998).

[16] Anupam Gupta. Steiner nodes in tree metrics don't (really) help. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 220–227, 2001.

[17] Anupam Gupta, Ilan Newman, Yuri Rabinovich, and Alistair Sinclair. Cuts, trees and ℓ_1-embeddings. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 399–408, 1999.

[18] R. Halin. Studies on minimally n-connected graphs. In D.J.A. Welsh, editor, *Combinatorial Mathematics and its Applications (Proc. Conf., Oxford, 1969)*, pages 129–136. Academic Press, London, 1971.

[19] Piotr Indyk. Algorithmic aspects of geometric embeddings. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, 2001.

[20] Richard M. Karp. A $2k$-competitive algorithm for the circle. Manuscript, 1989.

[21] Sanjeev Khanna and Rajeev Motwani. Towards a syntactic characterization of PTAS. In *Proceedings of the Twenty-eighth Annual ACM Symposium on the Theory of Computing (Philadelphia, PA, 1996)*, pages 329–337, New York, 1996. ACM.

[22] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995. (Preliminary version in: *35th Annual Symposium on Foundations of Computer Science*, pages 577–591, 1994).

[23] Jiří Matoušek. *Lectures in Discrete Geometry*. Springer-Verlag, to appear.

[24] Ilan Newman and Yuri Rabinovich. Lower bounds on embedding planar graphs into ℓ_2. manuscript, 2001.

[25] Ilan Newman, Yuri Rabinovich, and Michael Saks. unpublished notes.

[26] Haruko Okamura and Paul D. Seymour. Multicommodity flows in planar graphs. *Journal of Combinatorial Theory Series B*, 31(1):75–81, 1981.

[27] Satish B. Rao. Small distortion and volume preserving embeddings for planar and Euclidean metrics. In *15th Annual ACM Symposium on Computational Geometry*, pages 300–306, 1999.

[28] Neil Robertson and Paul D. Seymour. Graph minors. III. Planar tree-width. *J. Combin. Theory Ser. B*, 36(1):49–64, 1984.

[29] Alexander Schrijver. Homotopic routing methods. In Bernhard Korte, László Lovász, Hans Jürgen Prömel, and Alexander Schrijver, editors, *Paths, Flows and VLSI-Layout*, pages 329–371. Springer-Verlag, 1990.

[30] Farhad Shahrokhi and D. W. Matula. The maximum concurrent flow problem. *J. Assoc. Comput. Mach.*, 37(2):318–334, 1990.

[31] David B. Shmoys. Cut problems and their application to divide-and-conquer. In Dorit S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, pages 192–235. PWS Publishing, 1997.

[32] Maciej M. Sysło and Andrzej Proskurowski. On Halin graphs. In *Graph theory (Łagów, 1981)*, pages 248–256. Springer, Berlin, 1983.

[33] Douglas B. West. *Introduction to graph theory*. Prentice Hall Inc., Upper Saddle River, NJ, 1996.