EMBEDDING *K*-OUTERPLANAR GRAPHS INTO ℓ_1*

CHANDRA CHEKURI[†], ANUPAM GUPTA[‡], ILAN NEWMAN[§], YURI RABINOVICH[‡] , AND ALISTAIR SINCLAIR[¶]

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 theorem of Okamura and Seymour (*J. Combin. Theory Ser. B*, 31(1), pp. 75–81, 1981) for outerplanar graphs, and a result of Gupta et al. (*Combinatorica*, 24(2), pp. 233–269, 2004) 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 conjecture that these embeddings for k-outerplanar graphs may serve as building blocks for ℓ_1 embeddings of more general metrics.

Key words. Metric embeddings, k-Outerplanar graphs, Planar graphs, Low-distortion Embeddings, Probabilistic Approximation, Metric Spaces

AMS subject classifications. 05C12, 05C78, 51F99, 54C25, 54E70, 68R10

1. Introduction. 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

*A preliminary version of this paper appeared in the Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, 2003

[†]Lucent Bell Labs, 600-700 Mountain Avenue, Murray Hill New Jersey 07974. Email: `chekuri@research.bell-labs.com`

[‡]Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213. Email: `anupamg@cs.cmu.edu`. This work was done while the author was with Lucent Bell Labs, and 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.

surveys [21, 22], or the book [25, Chapter 10] for more details. 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 [23, 4]. In fact, the connection is even stronger: if there is a metric on a graph G that incurs distortion α when optimally embedded into ℓ_1, then there is a setting of capacities and demands on the graph G that achieves a cut-flow ratio of α [19]. 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 [32].

A related and equally important tool in algorithmic applications is the notion of approximating a finite metric by a probability distribution over dominating tree metrics [7]. A metric M' *dominates* another metric M if, for every pair $u, v \in M$, the distance between u and v in M' is no smaller than their distance in M. If a metric M is approximated by a distribution over dominating metrics, then the distortion for pair u, v is the ratio of the expected distance between them in the metric chosen according to the distribution and their distance in M. The overall (expected) distortion is defined to be the maximum distortion over all pairs of points in M. We can view these probabilistic approximations as embeddings. We use the term *embedding into random trees* to mean that we approximate a metric by a distribution over dominating tree metrics. Since every tree metric can be embedded isometrically (i.e., exactly, or with distortion 1) into ℓ_1, embedding into random trees with expected distortion α immediately implies an embedding into ℓ_1 with distortion α. As has been recognized in the work of Bartal and others [1, 7], embeddings into random trees have many applications to online and approximation algorithms. Some of these applications are not enjoyed by arbitrary ℓ_1 embeddings.

For *general* metrics the question of embeddability into ℓ_1 is essentially resolved: Bourgain [11] showed that any n-point metric can be embedded into ℓ_1 with $O(\log n)$ distortion, and this result was made algorithmic by Linial, London and Rabinovich [23] and Aumann and Rabani [4]. A matching lower bound of $\Omega(\log n/p)$ distortion into ℓ_p-spaces was established in [23, 24] for the shortest-path metric of unit-weighted expander graphs. For the case of approximating distances by distributions over dominating trees, a line of work [1, 7, 8, 12, 15] culminated in showing that any n-point metric can be embedded into a distribution over dominating trees with distortion $O(\log n)$ [15]; the lower bound for embeddings into ℓ_1 shows that this is tight.

However, tight bounds on the distortion incurred when embedding into ℓ_1 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 [21, 22]. In addition to its inherent mathematical interest, this question impacts the design of approximation algorithms for many problems on restricted families of graphs and networks.

Some limited but interesting progress has been made on embedding restricted[1] metrics into ℓ_1. Rao [29] showed that the shortest-path metric of any graph that excludes $K_{r,r}$ 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 by Newman and Rabinovich [26], 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 [28] 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 [27]. More recently, Gupta et al. [19] 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 $K_{2,n}$ has treewidth 2 but is not isometrically embeddable into ℓ_1; see [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 [19] 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* an embedding into random trees. On the other hand, also in [19], it is shown that even series-parallel graphs incur a distortion $\Omega(\log n)$ when embedded into random trees. 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 [19]: the trick was to combine the special structure of the tree embeddings with judicious use of random cuts.

1.1. Results. In this paper, we extend the above line of research to a wider class of planar graphs, namely k-outerplanar graphs for arbitrary 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 4.1 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., the work of Baker [6]). Our main result is the following:

THEOREM 1.1. *There exists an absolute constant $c > 1$ such that any shortest-path metric of a k-outerplanar graph can be embedded into random trees, and hence into ℓ_1, with distortion c^k. 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

[1]We emphasize here that our focus is on constraints imposed on metrics by the *topological* properties of the graphs on which they are defined. Thus we exclude from our discussion the extensive recent progress on embedding other types of restricted metrics, such as "negative type metrics", into ℓ_1, as in [3] and related papers.

[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 [16] and Schrijver [31].

into random trees. This is in contrast to the lower bound of $\Omega(\log n)$ for treewidth-2 graphs [19] mentioned earlier.

Our result immediately implies a constant maxflow-mincut ratio for arbitrary multicommodity flow problems on k-outerplanar graphs. Additionally, because our ℓ_1-embeddings are in fact random tree embeddings, we also obtain as a byproduct improved approximation ratios for a number of algorithms for problems on k-outerplanar graphs, including the buy-at-bulk problem [5] and the group Steiner problem [17]. For any fixed k, the improvement in each case is by a $\Omega(\log n)$ factor.

We should also note that since the maximum treewidth among k-outerplanar graphs is $\Theta(k)$, our result is the first demonstration of ℓ_1-embeddings with constant distortion for a natural family of graphs with arbitrarily large (but bounded) treewidth. 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 [19]. 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.

1.2. Techniques. We start with the approach of trying to extend the random tree embeddings of outerplanar graphs [19] 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 *subclass* of 2-outerplanar graphs that are easier to embed, namely *Halin graphs* [20]. 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, if we remove the outer face we are left with a collection of outerplanar graphs. We can use the embedding of [19] to embed each of these outerplanar graphs into random trees with constant distortion. If we now add the outer face to this collection of trees, we obtain (essentially) a collection of Halin graphs. 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 for this deceptively simple subclass of 2-outerplanar graphs, it is apparently non-trivial to obtain constant distortion embeddings. To obtain an embedding, we resort to a subtle modification of the algorithm of Gupta [18] which showed how to remove *Steiner vertices*[3] 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 *a priori* 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 [19].

The rest of the paper is organized as follows. We first fix notation and give

[3]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.

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 [13] or [25, Chapter 15]. Given two metric spaces, (V, ν) and (W, μ), and a map $f : V \to W$, we define the quantities

$$\|f\| = \max_{x,y \in V} \frac{\mu(f(x), f(y))}{\nu(x,y)};$$
$$\|f^{\mathrm{inv}}\| = \max_{x,y \in V} \frac{\nu(x,y)}{\mu(f(x), f(y))}.$$

We say that f has *contraction* $\|f^{\mathrm{inv}}\|$, *expansion* $\|f\|$ and *distortion* $D(f) = \|f\| \cdot \|f^{\mathrm{inv}}\|$. 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 vertex set V; in such cases, we assume that f is the identity map. Metric μ is said to *dominate* ν if for all $x, y \in V$, $\mu(x, y) \ge \nu(x, y)$.

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 can be shown that a metric is isometrically embeddable into ℓ_1 iff it can be written as a non-negative linear combination of cut metrics [13].

A metric d_G supported on a graph G is α*-probabilistically approximated* by a distribution $\mathcal{D}$ over trees if the following conditions hold:

1. Each tree T in the distribution $\mathcal{D}$ has $V(G) \subseteq V(T)$.
2. For each tree T in the distribution, the metric d_T *dominates* the metric d_G, i.e., for all nodes $x, y \in V(G)$, $d_G(x, y) \le d_T(x, y)$.
3. For all $x, y \in V(G)$, the expected distance $\mathrm{E}_{\mathcal{D}}[d_T(x, y)] \le \alpha \cdot d_G(x, y)$.

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 $O(\log n)$ [7, 15].

We state two simple propositions (whose proofs we omit) which we will use extensively in the sequel. The first allows us to embed each block (maximal 2-vertex connected subgraph) of a graph separately; the second says that we may always replace a subgraph by its tree embedding without further loss.

PROPOSITION 2.1. *Suppose G has a cut-edge whose removal results in a tree T and a graph H. If H can be embedded into random trees with distortion α, then so can G.*

PROPOSITION 2.2. *Let $H = (V_H, E_H)$ be a subgraph of $G = (V, E)$. Let $H_1, H_2, \ldots, H_s$ be graphs on V_H such that $d_H(u, v) \le d_{H_i}(u, v) \le \alpha_i \cdot d_H(u, v)$ for all $u, v \in V_H$, $1 \le i \le s$. Then in the graph $G_i = (V, (E \setminus E_H) \cup E_{H_i})$, we have $d_G(u, v) \le d_{G_i}(u, v) \le \alpha_i \cdot d_G(u, v)$ for all $u, v \in V$. Moreover, consider a random variable X taking values in $\{1, 2, \ldots, s\}$, where $\mathbf{Pr}[X = i] = \mu_i$, and let*

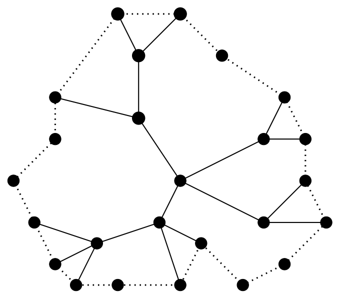

FIGURE 2.1. A Halin graph, with the tree $T = (V, E)$ in solid lines and the cycle $C = (U, E_C)$ in dashed lines.

$\alpha = \mathbf{E}[\,\alpha_X\,] = \sum_{i=1}^{s} \mu_i \alpha_i$. *Then, for any pair* $u, v \in V$*, the expected distance between* u *and* v *in the random graph* G_X *is at most* $\alpha d_G(u, v)$.

Graph-Theoretic Terms: A graph G' is a *minor* of G if G' is obtained from G by a sequence of edge deletions and contractions. A class of graphs is *closed* under taking minors if for every graph G in the class all its minors are also in the class. For example, planar graphs are minor-closed.

For a formal definition of *treewidth*, the reader is referred to standard graph theory texts such as [14, 34]. Informally, a graph has treewidth k if it can be decomposed recursively by vertex separators where the size of the vertex separator at each stage is at most k.

An embedding in the plane 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 at most $3k - 1$ [10, 30]; other properties of these graphs and related concepts can be found in [6, 10]. Given a planar graph, a k-outerplanar embedding for which k is minimal can be found in polynomial time [9].

A *Halin graph* [20] 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). If the set of leaves of T is denoted by L, then $V \cap U = L$; note that $U \setminus L$ may not be empty and hence there may be vertices on the cycle C that are not leaves of T. It is known that any 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., [33] and the references therein). We note that while Halin graphs (as defined here) are not minor-closed, we will not need this property in our algorithms.

3. Embedding a Halin graph.

The goal of this section is to prove the following theorem:

THEOREM 3.1. *The shortest path metric of a Halin graph can be embedded into random trees with distortion at most* 200.

Before embarking on the proof, we give a high level sketch of our strategy. Given a Halin graph consisting of a tree T and a cycle C, we first process the tree T to obtain a random dominating tree $T^{(1)}$, which approximates distances in T to within a constant factor (in expectation). Furthermore, the tree $T^{(1)}$ 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 $V \setminus L$ lie in subtrees that are attached to vertices in T''. Since we can ensure moreover that the tree T'' is a minor of T, attaching the cycle C back to the vertices in T'' gives us an outerplanar graph. Finally, this outerplanar graph is embedded into random trees with constant distortion using known techniques [19].

We will describe the tree processing procedure (which is the main content of the section) in Section 3.1, and in Section 3.2 we will explain how to use this to reduce to the outerplanar case.

3.1. Processing the tree. We assume that the tree T is rooted at a root vertex $r \in (V \setminus L)$. This imposes, in the usual manner, 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. We will use the following parameters extensively in the sequel.

DEFINITION 3.2. *For a vertex $v \in V$, let $l(v)$ be a leaf in $T(v)$* closest *to v, and $h(v)$ be the distance of v from $l(v)$ in T.*

Note that these functions $h(v)$ and $l(v)$ are fixed given the rooted tree T. Let us first give a brief overview of the processing algorithm, which has two conceptual 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 plus 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 while incurring a constant expected distortion.
- The second part of the processing, given in Section 3.1.2, eliminates the Steiner vertices of T' by contracting some of its edges to yield a tree T'' defined only on the leaves L. As a result, $T^{(1/2)}$ is converted into a tree $T^{(1)}$ with the properties claimed above. This part is shown to incur a further constant factor distortion.

3.1.1. Processing I: Constructing 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 procedure Process-Tree to perform this processing cuts off a subtree $\widehat{T}_0$ of T which contains the root but none of 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 $\widehat{T}_0$ off this new root. (See Figures 3.3 and 3.4.)

Before we make Process-Tree concrete, we define the auxiliary procedure Cut-Midway in Figure 3.1. This procedure takes as input a tree T which has root r and a set L of leaf nodes. It then cuts a *random* set of edges to separate r from all the leaves in L: in particular, it returns a special tree $\widehat{T}_0$ containing the root r and none of the nodes in L, and a set of subtrees T_i, $1 \leq i \leq t$, each rooted at some vertex r_i, which between them contain the leaves L. We say that an *edge $e = \{u, v\}$ is at 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; i.e., if $\mathbf{B}(r, d) \cap \{u, v\}$ has exactly one vertex. (Here $\mathbf{B}(r, d) = \{x \mid d_T(r, x) \leq d\}$ is the ball of radius d around the node r.) It should be noted that, in each interation of Cut-Midway, the set $\bar{L}$ decreases in size and the parameter d increases by at least a factor of 2.

The procedure Process-Tree, which outputs a tree $T^{(1/2)}$, is given in Figure 3.2. In this tree $T^{(1/2)}$, we denote by T' the portion formed by the new edges added between r' and r'_i (for $1 \leq i \leq t$) 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

while a path remains in T from the root r to a vertex in L
  let $\bar{L} \leftarrow$ vertices in L still reachable in T from r
  let $d \leftarrow$ distance in T to the closest vertex in $\bar{L}$
  let $S(d) \leftarrow \{x \in \bar{L} \mid d_T(r,x) \in [d, 2d)\}$
  let $T(d) \leftarrow$ union of the paths from r to vertices in $S(d)$
  choose $D \in_R [d/2, 3d/4)$ uniformly at random
  $E(d) \leftarrow$ edges in $T(d)$ at distance D from r
  delete the edges in $E(d)$ from T
end while
let $\widehat{T}_0 \leftarrow$ component of T containing root r but no leaves of T
let $T_1, T_2, \ldots, T_t \leftarrow$ other components of T
let $d_i \leftarrow$ value of d when edge connecting r to T_i was cut
return $(\widehat{T}_0; \langle T_1, d_1\rangle, \langle T_2, d_2\rangle, \ldots, \langle T_t, d_t\rangle)$

FIGURE 3.1. **Procedure** Cut-Midway(T)

apply Cut-Midway(T) to get
  $(\widehat{T}_0, \langle T_1, d_1\rangle, \langle T_2, d_2\rangle, \ldots, \langle T_t, d_t\rangle)$
let r' be a new vertex, called the "Steiner twin" of T's root r
attach r' to r with an edge of length $d_0 = h(r)$
for $1 \le i \le t$ *// We don't have to work on* $\widehat{T}_0$
  if T_i is just a single vertex x (hence $x \in L$) **then**
    let $T_i^{(1/2)} \leftarrow T_i$
  else
    let $T_i^{(1/2)} \leftarrow$ Process-Tree(T_i)
  let r_i' be the 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 3.2. **Procedure** Process-Tree(T)

and their Steiner twins.) For an example see Figure 3.3, where Cut-Midway performed three cuts, and Process-Tree resulted in the tree in Figure 3.4. The solid edges in the latter tree belong to T, the dashed ones to T', and the edge $\{r, r'\}$ is shown as a faint line. We remark that T' includes all the leaves of T, plus all the Steiner twins created during Process-Tree.

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 show the following bound on the expected distortion incurred by Process-Tree in passing from T to $T^{(1/2)}$.

THEOREM 3.3. *The (expected) distortion introduced by procedure* Process-Tree *is at most* 25.

Proof. We first give a high-level sketch. The construction of the tree $T^{(1/2)}$

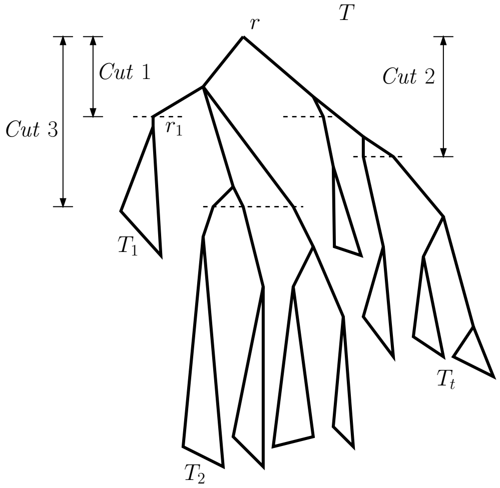

FIGURE 3.3. Cuts obtained by an invocation of Cut-Midway on a tree T

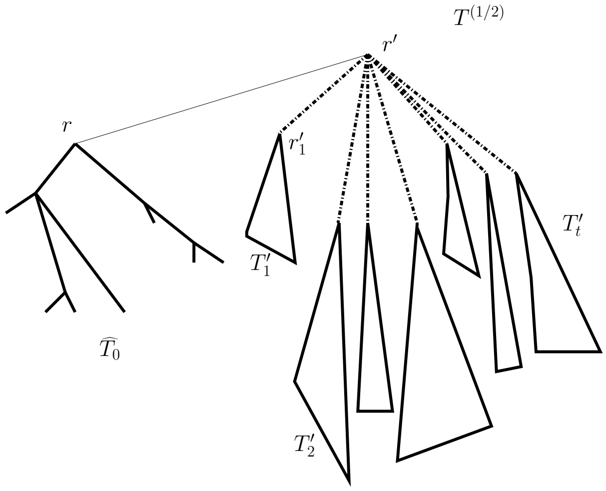

FIGURE 3.4. The tree $T^{(1/2)}$ output by Process-Tree on the tree T from Figure 3.3. The dotted lines indicate edges in T'.

ensures that distances are not contracted by Process-Tree; the algorithm explicitly ensures this in Process-Tree by the distances it chooses to connect the root r' to each r'_i. Hence it suffices to bound the expected expansion of distances. We do this via two lemmas: firstly, Lemma 3.4 shows that an edge is a candidate to be cut on at most two (consecutive) occasions. Lemma 3.5 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 Theorem 3.3.

LEMMA 3.4. *No edge is a candidate to be cut more than twice during the entire run of the procedure* Process-Tree.

Proof. Let $e = \{u, v\}$ be an edge with u being the parent of v. Consider the first instant in time when the edge e is a candidate to be cut in a call to Cut-Midway. Let r be the root at this time, and d^* be the value of the parameter d in the **while**

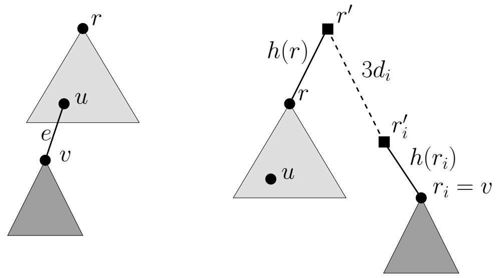

FIGURE 3.5. Illustration for Proof of Lemma 3.5.

loop of this call to Cut-Midway. In this call of Cut-Midway, it is clear that e cannot be a candidate 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 this value multiplied by $4/d^*$ is the probability that e is cut at this time.

The edge e will never be a candidate again if the cut fell "below" v, or if it passed through e, so let us assume that the cut was above u and thus e lies in one of the trees T_i with root r_i. In this case the tree T_i will be passed on to Cut-Midway by Process-Tree. Now e clearly lies on some path from r_i to a leaf, and hence it may be a candidate to be cut again. Let d^{**} be the value of the parameter d in Cut-Midway when this happens for the first time after T_i is formed.

We claim that the cut made at this point must fall below u; i.e., $d^{**}/2 \geq d_T(r_i, u)$. Indeed, such a cut is made at a distance at least $d^{**}/2 = h(r_i)/2$ from the root r_i, where $h(r_i) \geq d^* - d_T(r, r_i)$. 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 lies 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 a candidate a second time. ∎

LEMMA 3.5. *Let $e = \{u, v\}$ be an edge in G of length ℓ_e. If e is cut by* Cut-Midway *with parameter d_i, the expected distance between u and v in $T^{(1/2)}$ is at most $6d_i - \ell_e$.*

Proof. Consider an edge $e = \{u, v\}$ of length ℓ_e, with u the parent of v, which is cut in some iteration of Cut-Midway, and let d_i be the value of the parameter d at this time. 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 $\widehat{T}_0$ and the vertex v is the root r_i of T_i for some i, and hence will be in $T_i^{(1/2)}$ when T_i 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 $d_T(u,r) + d_{T^{(1/2)}}(r, r') + d_{T^{(1/2)}}(r', r'_i) + d_{T^{(1/2)}}(r'_i, r_i)$ (see Figure 3.5). From our construction, $d_{T^{(1/2)}}(r, r') = h(r)$, $d_{T^{(1/2)}}(r', r'_i) = 3d_i$, 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 inequality holds because for e to be cut, r_i must lie on the path from r to a leaf in T of length at most $2d_i$. Note that this calculation also holds in the special case that v is a leaf (when $0 = h(r_i) \leq 2d_i - d_T(r, r_i)$).

Putting all these observations together we obtain

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

□

Now we complete the proof of Theorem 3.3. By Lemma 3.4, the edge $e = \{u, v\}$ is a candidate to be cut at most twice. From the proof of Lemma 3.4, the first time it is a candidate 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, by Lemma 3.5, if it is cut the expected distance between u and v becomes at most $6d^* - \ell_e$. Similarly, the second time the chance of being cut is

$$p_2 = (\max(d_T(r,v) - 3d^*/4, 0)) \times 4/d^{**},$$

and the expected distance is $6d^{**} - \ell_e$. Finally, the distance remains unchanged at ℓ_e with the remaining probability $(1 - p_1 - p_2)$. Putting these together, we get that the expected distance between u and v after procedure Process-Tree is at most

$$\begin{aligned} & 6d^* p_1 + 6d^{**} p_2 + (1 - 2p_1 - 2p_2)\ell_e \\ & \qquad \leq 6(d^* p_1 + d^{**} p_2) + \ell_e \\ & \qquad \leq 24 \big(\min(d_T(r,v), 3d^*/4) - \max(d_T(r,u), d^*/2) \\ & \qquad\qquad + \max(d_T(r,v) - 3d^*/4, 0) \big) + \ell_e \qquad (3.1) \\ & \qquad \leq 24 \big(d_T(r,v) - \max(d_T(r,u), d^*/2) \big) + \ell_e \qquad (3.2) \\ & \qquad \leq 24 \big(d_T(r,v) - d_T(r,u) \big) + \ell_e \\ & \qquad \leq 24\, \ell_e + \ell_e = 25\, \ell_e, \end{aligned}$$

where we used the simplification $\min(x,y) + \max(x - y, 0) = x$ to obtain (3.2) from (3.1). Thus the expected distortion is at most 25, which proves the theorem. □

Recall that the tree $T^{(1/2)}$ constructed by the procedure Process-Tree includes a tree T' containing the leaves L of the original tree T; we close this subsection with a further observation about T'.

Claim 3.6. *The tree T' can be obtained from tree T by edge contractions.*

Proof. In each call to Process-Tree, we progressively construct T' by removing the tree $\widehat{T}_0$ 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 the edges of the tree $\widehat{T}_0$. (Of course, we are placing new lengths on the remaining edges, but this does not affect the topology.) □

Since L is also the set of leaves of T', and the edge contractions can be performed without changing the planar layout of the trees, adding the cycle C around the leaves of T' also gives us a Halin graph.

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 [18]. 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 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 3.7. *The edge-contraction procedure described above 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 $\widehat{T_0}$ in the various invocations of Process-Tree, those between some r and $l(r)$,[4] 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 $\widehat{T_0}$ are not changed at all. Now consider an edge between a vertex $l(r)$ and 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 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 $d_T(r, l(r)) = h(r) = d^*$; furthermore, the distance $d_T(r, l_{T_i}(r_i)) \leq 2d^*$, since r_i must lie on a root-leaf path of length at most $2d^*$. Hence the distance between $l(r)$ and $l(r_i)$ in T is at most $3d^*$. However, $d_T(r, r_i) \geq d^*/2$, so the distance is at most $6d_T(r, r_i)$ as required. ∎

3.2. Wrapping it all up. We now complete the proof of Theorem 3.1. Let G be the given Halin graph, consisting of a tree $T = (V, E)$ and a cycle $C = (U, E_c)$ around the leaves $L = V \cap U$ of T. We have seen how to transform T into a tree $T^{(1)}$ that consists of a tree $T'' = (L, E'')$ and a collection of trees $T_1, T_2, \ldots, T_j$ each of which is connected by an edge to a vertex in L. Every vertex in $V - L$ is contained in exactly one of $T_1, T_2, \ldots, T_j$. Moreover, the tree T'' is a minor of T. We have also seen that $T^{(1)}$ dominates T and that the expected expansion for any pair in T is at most 25. Now consider the graph $G^{(1)}$ obtained by adding the cycle C to the tree $T^{(1)}$. Let G' be the graph obtained by adding C to T''. (See Figure 3.6.) We claim that G' is an outerplanar graph. Assuming for the moment that this claim is true, we show how we can embed G into trees with the claimed distortion.

First, from Proposition 2.2, it follows that $G^{(1)}$ dominates G and for every pair $u, v \in V_G$, the expected distance in $G^{(1)}$ is at most $25d_G(u, v)$.

Next, note that $G^{(1)}$ consists of G' with the trees $T_1, T_2, \ldots, T_j$ connected to G'

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

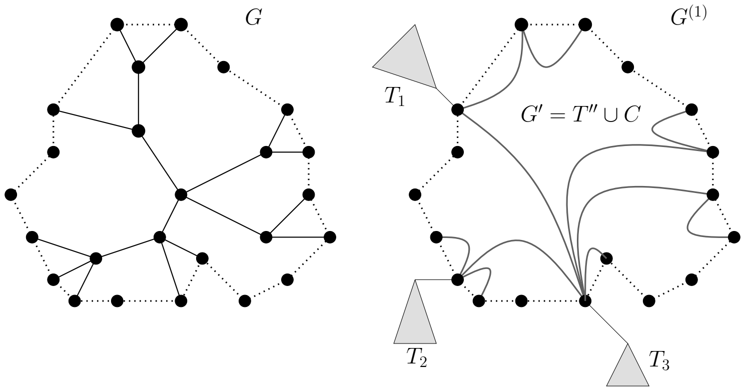

FIGURE 3.6. G is a Halin graph; G' is an outerplanar graph obtained from $T'' \cup C$, and $G^{(1)}$ is obtained by adding the trees T_i to G'.

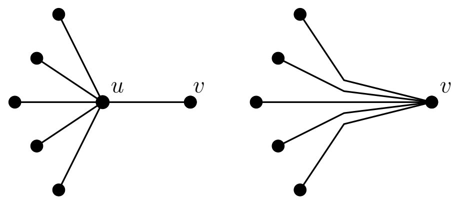

FIGURE 3.7. Contracting edge $\{u, v\}$ and removing u. Obtaining contours for new edges.

by cut-edges. From Proposition 2.1 it follows that embedding G' into random trees with distortion α produces an embedding of $G^{(1)}$ into random trees with distortion α. Since G' is an outerplanar graph, we can invoke the procedure of [19, Theorem 5.2] to get a random subtree of G' which approximates distances in G' (in expectation) to within a factor of 8. Thus $G^{(1)}$ can be embedded into random trees with distortion 8.

Finally, from Proposition 2.2 we see that embedding $G^{(1)}$ into random trees with distortion 8 implies that G can be embedded into random trees with distortion $8 \cdot 25 = 200$. This completes the proof of Theorem 3.1.

It remains to sketch the proof that G' is outerplanar, as was claimed above. From Claim 3.6, T' is a minor of T, and hence T'', which is obtained by contracting some edges in T', is also a minor of T. Moreover, since no two vertices of L are merged in obtaining T'', G' is a minor of G. Thus we can obtain G' from G by a sequence of edge deletions and contractions. This allows us to obtain an outerplanar embedding of G' from the given planar embedding of G as follows. First, remove any edges of G that are removed in obtaining G'. Then consider the first edge $\{u, v\}$ that is contracted in G. Vertices u and v cannot both be in L, so let u be the vertex outside of L. Let $u_1, u_2, \ldots, u_h$ be the neighbors of u that are not v. The edge $\{u_i, u\}$ is a contour in the planar embedding of G. When $\{u, v\}$ is contracted we remove u and extend the edge $\{u_i, u\}$ to $\{u_i, v\}$. By duplicating the contour of $\{u, v\}$ h times and shifting the resulting contours infinitesimally we can obtain new contours for the edges $\{u_1, v\}, \ldots, \{u_h, v\}$. (See Figure 3.7.) Thus we obtain a planar embedding of the graph with the edge $\{u, v\}$ contracted without changing the position of v. Thus all the vertices remain in their original positions and any edge $\{x, y\}$ that is not contracted or deleted has its contour intact. We can continue this process and obtain

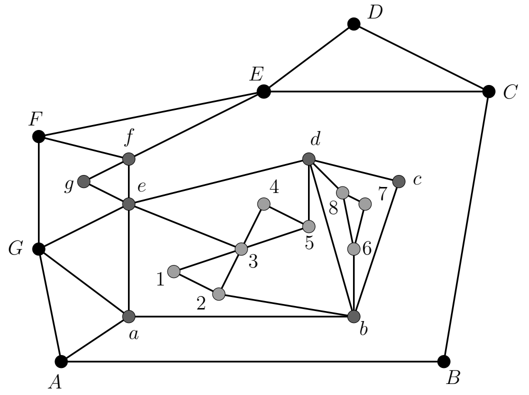

FIGURE 4.1. A 3-outerplanar graph from [6]. The three layers *A-G*, *a-g*, and *1-8* are shown using different shades of gray.

a planar embedding of G' such that the vertices $U \supseteq L$ and the contours of edges in E_c are unchanged from the planar embedding of G that we started with. Since all the vertices of G' are on the outer face C, it follows that we have an outerplanar embedding of G'.

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 which are dismantled by k repetitions of the process of removing the vertices on the outermost face. (See Figure 4.1 for an example.)

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

THEOREM 4.1. *There is a universal constant c such that the shortest path metric of a k-outerplanar graph can be embedded into random trees with distortion c^k.*

Proof. We begin with a high-level sketch of the proof, which proceeds by induction on k. Since G is k-outerplanar, removing the outer face of G decomposes it into a set of $(k-1)$-outerplanar subgraphs $G_1, \ldots, G_\ell$. Each G_i resides inside a face F_i of the graph induced by the vertices of the outer face of G. (See Figure 4.2.) By the induction hypothesis, each G_i can be embedded into random trees with distortion c^{k-1}; moreover, this can be done leaving the vertices on the outer face of G_i in their original positions. Replacing G_i by its corresponding tree T_i yields a *Halin* graph whose outer cycle is the face F_i (plus possibly some trees attached to internal nodes of T_i); see Figure 4.3. Now the procedure of Section 3 can be used to embed this Halin graph into an *outerplanar* graph on F_i (plus some attached trees) with constant distortion c_1. Finally, the union (over i) of all these outerplanar graphs is again outerplanar, and so by [19, Theorem 5.2] can be embedded into random trees with constant distortion c_2. The overall distortion incurred in this process is $c^{k-1} \cdot c_1 \cdot c_2 \leq c^k$ if we choose $c = c_1 c_2$.

Remark: The reader may recall from Section 3 that we can take $c_1 = 25$ and $c_2 = 8$ in the above. Hence Theorem 4.1 holds with the constant $c = 200$.

We now proceed to spell out the details of the above argument. We begin with the induction hypothesis, which needs to be slightly stronger than the statement of the theorem. We assume $G = (V, E)$ 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 embed-

ding. 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. Moreover, for each subtree T in this distribution, the vertices of the outer face $F_0(G)$ induce a (connected) subtree that is a minor of G.

The importance of the extra condition placed on the trees T is the following. Let T' be the subtree induced by the vertices of $F_0(G)$; note that the vertices of $V \setminus T'$ reside in subtrees hanging off T' by single edges. Since T' is a minor of G, we can construct it by edge deletions and contractions while leaving the vertices of $F_0(G)$ in their original positions, as explained in Section 3.2. This allows us in the induction to replace G by T' without disturbing the outer face $F_0(G)$.

The base case for the induction is $k = 1$, when G is an outerplanar graph. For outerplanar graphs, [19, Theorem 5.2] shows an embedding of G into trees that are *subgraphs* of G with constant distortion (at most 8). Being subgraphs these trees are certainly minors, so the extra condition in the induction is satisfied.

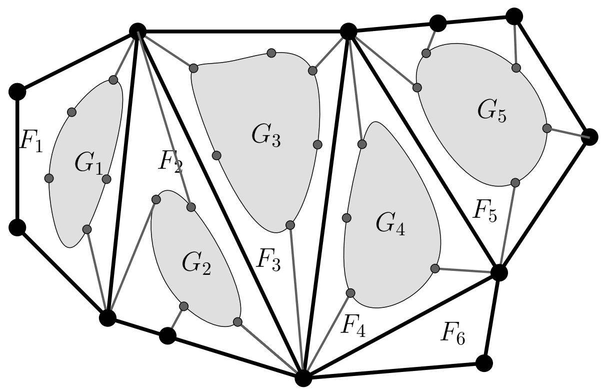

FIGURE 4.2. Partitioning of a k-outerplanar graph G into $(k-1)$-outerplanar graphs $G_1, \dots, G_5$. The bold lines indicate G_F, the graph induced by the outer face.

For the induction step, we may assume that G is 2-vertex connected; otherwise we can work with each block of G separately. 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 4.2.) Let $F_1, F_2, \dots, F_\ell$ be the internal faces of G_F, V_i the subset of $V \setminus F_0(G)$ 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, since 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 $\{u, v\} \in E$. This is without loss of generality, since if it 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 4.2. *For $1 \le i \le \ell$, G_i is a $(k-1)$-outerplanar graph.*

Thus, by the induction hypothesis, each G_i can be c^{k-1}-probabilistically approximated by trees satisfying the extra condition. We now give a procedure to extend the embeddings of the various G_i to an embedding of G. For $1 \le i \le \ell$, we independently 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

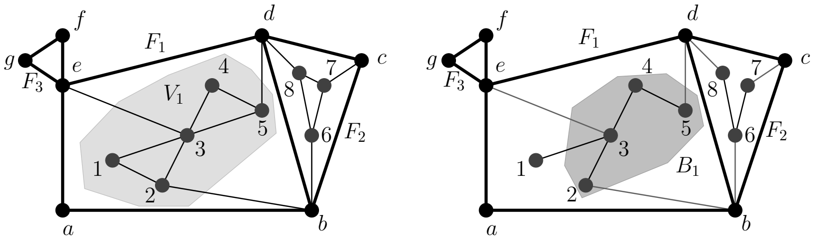

FIGURE 4.3. Returning from the induction: the bold lines denote G_F. The lightly shaded region on the left denotes the vertices V_1 corresponding to the face $F_1 = (abde)$, and the darker shaded region on the right denotes the set B_1. Note that $A_1 = \{2, 3, 5\}$, $B_1 \setminus A_1 = \{4\}$, and $V_1 \setminus B_1 = \{1\}$.

the trees $T_1, \ldots, T_\ell$. Proposition 2.2 implies that the metric induced by G' is within expected distortion c^{k-1} 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 that induced by $F_0(G_i)$; the fact that it is a tree is guaranteed by the extra condition in the induction hypothesis. Let A_i be the vertices in V_i that have an edge to some vertex in F_i; since G is planar, $A_i \subseteq F_0(G_i)$. Let T_i'' be the *minimal* connected subtree of T_i' that contains A_i. Let B_i the vertices in T_i''. (Note that B_i may contain vertices not in A_i but by minimality of T_i'', any vertex in $B_i \setminus A_i$ is an internal vertex of T_i''.) The remaining vertices, in $V_i \setminus B_i$, induce a forest in T_i that is connected via cut-edges to T_i''. (An example is given in Figure 4.3.) Using Proposition 2.1, we can eliminate the vertices in $V_i \setminus B_i$ (for $1 \le i \le \ell$) from G'. It now suffices to embed the resulting graph into trees with expected distortion at most c.

The key claim that reduces this problem to the embedding of Halin graphs given in the previous section is the following (see Figure 4.3):

CLAIM 4.3. *Let G_i' be obtained by adding to the tree T_i'' the vertices F_i and the edges in G connecting F_i to A_i. Then G_i' is a Halin graph with cycle F_i.*

Proof. By the induction hypothesis, the tree T_i' is a minor of G_i. Since T_i'' is a subtree of T_i' it is also a minor of G_i and hence, as in Section 3.2, the planar embedding of G_i induces a natural planar embedding of T_i''. Furthermore, by 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. Since T_i'' was chosen to be minimal, the leaves in T_i'' are a subset of A_i. Therefore the leaves in the tree formed by adding E_i to T_i'' are precisely the vertices of F_i incident to an edge in E_i. The edges along the face F_i form a cycle around these leaves, yielding a Halin graph. ∎

Now we can apply the procedure of Section 3 to G_i' (omitting the final step of embedding the outerplanar graph into trees). The resulting graph, which we call G_i'', will be an outerplanar graph on F_i, with the vertices of T_i'' attached as subtrees; the expected distortion will be at most 25. Using Proposition 2.1 again, we can remove these hanging subtrees to obtain the graph $\mathsf{core}(G_i'')$.

Note that the procedure in Section 3 guarantees that $\mathsf{core}(G_i'')$ is a minor of G_i'. Furthermore, each $\mathsf{core}(G_i'')$ is an outerplanar graph on the face F_i of the outerplanar graph G_F. These two facts together imply that $H = \bigcup_i \mathsf{core}(G_i'')$ is also an outerplanar graph. Thus we can use [19, Theorem 5.2] to embed H into random subtrees of H with expected distortion at most 8. Choosing $c = 25 \cdot 8 = 200$, we conclude that G' can be embedded into random trees with expected distortion at most c, and hence that G can be embeddded with expected distortion at most c^k, as required.

To complete the inductive proof, it remains to verify that the random trees produced by the above procedure satisfy the extra property stated in the induction hypothesis, namely that the vertices of $F_0(G)$ form a subtree that is a minor of G. The final step of the procedure constructs a subtree T_H of the graph H whose vertex set is exactly $F_0(G)$. Now observe that the procedure discards vertices only when they induce a subtree attached to the rest of the graph (invoking Proposition 2.1 on each occasion to ensure that this introduces no additional distortion). Thus the final tree consists of T_H with other subtrees hanging off it. To see that T_H is a minor of G, it suffices to show that H is a minor of G since T_H is a subtree (and hence a minor) of H. But $H = \bigcup_i \mathsf{core}(G_i'')$, and we already observed above that each $\mathsf{core}(G_i'')$ is a minor of G_i'. Furthermore, G_i' is formed by replacing G_i by the tree T_i'' inside the face F_i, and T_i'' is a subtree of T_i' and hence a minor of T_i'. And we know from the induction hypothesis that T_i' is a minor of G_i; hence G_i' is a minor of G_i. This implies that $\mathsf{core}(G_i'')$ is a minor of G_i, and hence that H is a minor of G, as required. This completes the inductive proof of Theorem 4.1. $\square$

Acknowledgments. We thank Amit Chakrabarti and Amit Kumar for useful discussions, and the referees for their suggestions which improved the presentation of the paper.

REFERENCES

[1] N. Alon, R. M. Karp, D. Peleg, and D. West, *A graph-theoretic game and its application to the k-server problem*, SIAM J. Comput., 24 (1995), pp. 78–100.

[2] A. Andoni, M. M. Deza, A. Gupta, P. Indyk, and S. Raskhodnikova, *Lower bounds for embedding edit distance into normed spaces*, in Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms (SODA), 2003, pp. 523–526.

[3] S. Arora, S. Rao, and U. Vazirani, *Expander flows, geometric embeddings, and graph partitionings*, in Proceedings of the 36th ACM Symposium on the Theory of Computing (STOC), 2004, pp. 222–231.

[4] Y. Aumann and Y. Rabani, *An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm*, SIAM J. Comput., 27 (1998), pp. 291–301.

[5] B. Awerbuch and Y. Azar, *Buy-at-bulk network design*, in Proceedings of the 38th Symposium on the Foundations of Computer Science (FOCS), 1997, pp. 542–547.

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

[7] Y. Bartal, *Probabilistic approximations of metric spaces and its algorithmic applications*, in Proceedings of the 37th Symposium on the Foundations of Computer Science (FOCS), 1996, pp. 184–193.

[8] ———, *On approximating arbitrary metrics by tree metrics*, in Proceedings of the 30th ACM Symposium on the Theory of Computing (STOC), 1998, pp. 161–168.

[9] D. Bienstock and C. L. Monma, *On the complexity of embedding planar graphs to minimize certain distance measures*, Algorithmica, 5 (1990), pp. 93–109.

[10] H. L. Bodlaender, *A partial k-arboretum of graphs with bounded treewidth*, Theoretical Computer Science, 209 (1998), pp. 1–45.

[11] J. Bourgain, *On Lipschitz embeddings of finite metric spaces in Hilbert space*, Israel Journal of Mathematics, 52 (1985), pp. 46–52.

[12] M. Charikar, C. Chekuri, A. Goel, S. Guha, and S. A. Plotkin, *Approximating a finite metric by a small number of tree metrics*, in Proceedings of the 39th Symposium on the Foundations of Computer Science (FOCS), 1998, pp. 379–388.

[13] M. M. Deza and M. Laurent, *Geometry of cuts and metrics*, vol. 15 of Algorithms and Combinatorics, Springer-Verlag, Berlin, 1997.

[14] R. Diestel, *Graph theory*, vol. 173 of Graduate Texts in Mathematics, Springer-Verlag, New York, second ed., 2000.

[15] J. Fakcharoenphol, S. Rao, and K. Talwar, *A tight bound on approximating arbitrary*

metrics by tree metrics, in Proceedings of the thirty-fifth ACM symposium on Theory of computing, ACM Press, 2003, pp. 448–455.

[16] A. Frank, *Packing paths, circuits, and cuts – A survey*, in Paths, Flows and VLSI-Layout, B. Korte, L. Lovász, H. J. Prömel, and A. Schrijver, eds., Springer-Verlag, 1990, pp. 47–100.

[17] N. Garg, G. Konjevod, and R. Ravi, *A polylogarithmic approximation algorithm for the group Steiner tree problem*, Journal of Algorithms, 37 (2000), pp. 66–84. (Preliminary version in *9th SODA*, pages 253–259, 1998).

[18] A. Gupta, *Steiner points in tree metrics don't (really) help*, in Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (Washington, DC, 2001), Philadelphia, PA, 2001, SIAM, pp. 220–227.

[19] A. Gupta, I. Newman, Y. Rabinovich, and A. Sinclair, *Cuts, trees and ℓ_1-embeddings of graphs*, Combinatorica, 24 (2004), pp. 233–269. (Preliminary version in 40th FOCS, 1999.).

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

[21] P. Indyk, *Algorithmic aspects of geometric embeddings*, in Proceedings of the 42nd Symposium on the Foundations of Computer Science (FOCS), 2001, pp. 10–33.

[22] N. Linial, *Finite metric-spaces—combinatorics, geometry and algorithms*, in Proceedings of the International Congress of Mathematicians, Vol. III (Beijing, 2002), Beijing, 2002, Higher Ed. Press, pp. 573–586.

[23] N. Linial, E. London, and Y. Rabinovich, *The geometry of graphs and some of its algorithmic applications*, Combinatorica, 15 (1995), pp. 215–245. (Preliminary version in *35th FOCS*, 1994).

[24] J. Matoušek, *On embedding expanders into l_p spaces*, Israel J. Math., 102 (1997), pp. 189–197.

[25] J. Matoušek, *Lectures on discrete geometry*, vol. 212 of Graduate Texts in Mathematics, Springer-Verlag, New York, 2002.

[26] I. Newman and Y. Rabinovich, *A lower bound on the distortion of embedding planar metrics into euclidean space*, in Proceedings of the eighteenth annual symposium on Computational geometry, ACM Press, 2002, pp. 94–96.

[27] I. Newman, Y. Rabinovich, and M. Saks, *Excluded minors for embeddings*. unpublished notes.

[28] H. Okamura and P. D. Seymour, *Multicommodity flows in planar graphs*, Journal of Combinatorial Theory Series B, 31 (1981), pp. 75–81.

[29] S. B. Rao, *Small distortion and volume preserving embeddings for planar and Euclidean metrics*, in 15th Annual ACM Symposium on Computational Geometry, 1999, pp. 300–306.

[30] N. Robertson and P. D. Seymour, *Graph minors. III. Planar tree-width*, J. Combin. Theory Ser. B, 36 (1984), pp. 49–64.

[31] A. Schrijver, *Homotopic routing methods*, in Paths, Flows and VLSI-Layout, B. Korte, L. Lovász, H. J. Prömel, and A. Schrijver, eds., Springer-Verlag, 1990, pp. 329–371.

[32] D. B. Shmoys, *Cut problems and their application to divide-and-conquer*, in Approximation Algorithms for NP-hard Problems, D. S. Hochbaum, ed., PWS Publishing, 1997, pp. 192–235.

[33] M. M. Sysło and A. Proskurowski, *On Halin graphs*, in Graph theory (Łagów, 1981), Springer, Berlin, 1983, pp. 248–256.

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