k-Outerplanar Graphs, Planar Duality, and Low Stretch Spanning Trees

Yuval Emek*

Abstract

Low distortion probabilistic embedding of graphs into approximating trees is an extensively studied topic. Of particular interest is the case where the approximating trees are required to be (subgraph) spanning trees of the given graph (or multigraph), in which case, the focus is usually on the equivalent problem of finding a (single) tree with low average stretch. Among the classes of graphs that received special attention in this context are k-outerplanar graphs (for a fixed k): Chekuri, Gupta, Newman, Rabinovich, and Sinclair show that every k-outerplanar graph can be probabilistically embedded into approximating trees with constant distortion regardless of the size of the graph. The approximating trees in the technique of Chekuri et al. are not necessarily spanning trees, though.

In this paper it is shown that every k-outerplanar multigraph admits a spanning tree with constant average stretch. This immediately translates to a constant bound on the distortion of probabilistically embedding k-outerplanar graphs into their spanning trees. Moreover, an efficient randomized algorithm is presented for constructing such a low average stretch spanning tree. This algorithm relies on some new insights regarding the connection between low average stretch spanning trees and planar duality.

Keywords: planar graphs, outerplanarity, average stretch, planar dual.

*Microsoft Israel R&D Center, Herzelia, Israel and School of Electrical Engineering, Tel Aviv University, Tel Aviv, Israel. E-mail: `yuvale@eng.tau.ac.il`. Supported in part by the Israel Science Foundation, grants 221/07 and 664/05.

1 Introduction

The problem. Consider an n-vertex connected graph $G = (V(G), E(G))$ and let $\ell(e)$ be a positive *length* associated with every edge $e \in E(G)$. For any two vertices $u, v \in V(G)$, let $\delta_G(u, v)$ denote the *distance* between u and v in G, namely, the length, taken with respect to ℓ, of a shortest path connecting u and v in G. Given a spanning tree T of G and some edge $e \in E(G)$, the *stretch* of e in T is defined as $\mathrm{str}_T(e) = \delta_T(e)/\ell(e)$. Spanning trees with low stretch for all edges can be very useful in many applications. However, there exist some trivial graphs for which every spanning tree admits an edge with stretch $\Omega(n)$ (e.g., the n-cycle). This motivates Alon, Karp, Peleg, and West [2] to construct spanning trees with low *average stretch*, denoted by $\text{av-str}_G(T) = \frac{1}{|E(G)|} \sum_{e \in E(G)} \mathrm{str}_T(e)$.

The following related notion was introduced by Bartal [4]. Given a probability distribution $\mathcal{D}$ over a set $\mathcal{T}$ of spanning trees of G, we say that G is *probabilistically embedded* into $\mathcal{T}$ (under $\mathcal{D}$) with *distortion* α if $\mathbb{E}[\delta_T(u, v)] \leq \alpha \cdot \delta_G(u, v)$ for every two vertices $u, v \in V(G)$, where the expectation is with respect to $T \in_{\mathcal{D}} \mathcal{T}$. Alon *et al* [2] show that a graph G can be probabilistically embedded into its spanning trees with distortion α if and only if every multigraph obtained from G by replicating its edges has a spanning tree with average stretch α. Consequently, in the context of constructing low average stretch spanning trees, one usually considers multigraphs rather than simple graphs. (This can be viewed as taking a weighted average of the edge stretch factors.)

A tree T is called a *dominating tree* of the graph G if $V(T) \supseteq V(G)$ and $\delta_T(u, v) \geq \delta_G(u, v)$ for every two vertices $u, v \in V(G)$. Clearly, every spanning tree of G is also a dominating tree of G; the converse is not true as a dominating tree may have vertices and edges that do not exist in the original graph G, and hence it is not necessarily a subgraph of G. The notion of probabilistic embedding can be redefined by allowing the support $\mathcal{T}$ to contain dominating trees that are not subgraphs of G. For many applications and in particular, for those applications mentioned by Bartal [4, 5], this does not exhibit any obstacle. However, there exist some applications for which it is impossible to use non-subgraph dominating trees in the support of the probabilistic embedding, most notably in the context of networking, where G represents an existing physical graph (e.g., the *minimum communication spanning tree* problem introduced by Hu [14]).

k-outerplanar graphs. An *outerplanar* graph (or a 1-*outerplanar* graph) is a graph that can be drawn in the plane with all vertices lying on the unbounded face. A planar graph is said to be k-*outerplanar*, $k \geq 2$, if it can be drawn in the plane such that by removing the vertices on the unbounded face we obtain a $(k-1)$-outerplanar graph. A canonical example for a k-outerplanar graph is the $2k \times n$ grid (containing $2k$ rows of vertices with n vertices in each row) which also serves as a canonical example for a graph with tree width proportional to k. When referring to k-outerplanar graphs, we usually assume that k is fixed. However, every planar graph is k-outerplanar for some k (typically, much smaller than n) and this *outerplanarity factor* plays a key role in many polynomial time approximation schemes for NP-hard optimization problems on planar graphs (See Baker [3]).

Related work. The problem of constructing spanning trees with low average stretch was first studied by Alon *et al* [2] who proved that every n-vertex multigraph G admits a spanning tree T which satisfies $\text{av-str}_G(T) = e^{O(\sqrt{\ln n \ln \ln n})}$. They also show that there exist some graphs, the $\sqrt{n} \times \sqrt{n}$ grid being one of them, for which every spanning tree admits average stretch $\Omega(\log n)$ and conjectured that this lower bound is tight. The upper bound of Alon *et al* [2] was drastically improved by Elkin, Emek, Spielman, and Teng [8] who introduced a construction of spanning trees with average stretch $O(\log^2 n \log \log n)$. Recently, Abraham, Bartal, and Neiman [1] presented a further improvement by establishing an almost tight upper bound of $O(\log n \log \log n \log^3 \log \log n)$.

The notion of probabilistic embedding was explicitly introduced by Bartal [4] (although, it was implicitly used by Alon *et al* [2]). This initiated a series of papers that developed probabilistic embeddings of arbitrary graphs into non-subgraph dominating trees: in [4] Bartal shows that every n-vertex graph can be probabilistically embedded into dominating trees with distortion $O(\log^2 n)$, while some graphs must suffer a distortion of $\Omega(\log n)$; the upper bound was improved to $O(\log n \log \log n)$ by Bartal [5] and by Charikar, Chekuri, Goel, Guha, and Plotkin [6]; and a tight $O(\log n)$ upper bound is proved by Fakcharoenphol, Rao, and Talwar [10].

Some papers study probabilistic embeddings of specific graph classes. In [17] Konjevod, Ravi, and Salman show that every planar graph can be probabilistically embedded into its (non-subgraph) dominating trees with distortion $O(\log n)$ using the decomposition technique of Klein, Plotkin, and Rao [16] for graphs excluding small minors. This is generalized by Indyk and Sidiropoulos [15] to graphs of bounded genus by showing that such graphs can be probabilistically embedded with constant distortion into planar graphs. Gupta, Newman, Rabinovich, and Sinclair [12] prove that while series-parallel graphs can be embedded into ℓ_1 with constant distortion, there exist some unweighted series-parallel graphs that cannot be probabilistically embedded into dominating trees with distortion $o(\log n)$. This lower bound is matched by Emek and Peleg [9], showing that unweighted series-parallel graphs can be probabilistically embedded into their spanning trees with distortion $O(\log n)$.

It is also proved by Gupta *et al* [12] that every outerplanar graph can be probabilistically embedded into its spanning trees with constant distortion. The construction of Gupta *et al* for (1-)outerplanar graphs is (partially) generalized to k-outerplanar graphs by Chekuri, Gupta, Newman, Rabinovich, and Sinclair [7] who presented a probabilistic embedding with distortion exponential in k (but independent of n). Note however, that unlike the construction of Gupta *et al* [12], the technique of Chekuri *et al* [7] constructs (random) dominating trees which are not necessarily spanning trees of the original graph.

Contribution. In this paper we show that every k-outerplanar multigraph G admits a spanning tree T which satisfies $\text{av-str}_G(T) \leq c^k$, where c is an absolute constant. This immediately implies that every k-outerplanar graph can be probabilistically embedded into its spanning trees with distortion depending solely on k, thus enhancing the result of Chekuri *et al* [7]. Our proof is constructive: we present an efficient[1] randomized algorithm that constructs such a low average stretch spanning tree.

Techniques. The backbone of our algorithm is a rather standard peeling-an-onion decomposition (cf. Chekuri *et al* [7]): on input k-outerplanar graph G, we first peel off the vertices on the unbounded face to obtain a $(k-1)$-outerplanar graph G'; we then recursively construct a good spanning tree T' of G'; next, we insert the missing vertices of G back into T' to obtain the graph H; and finally, we construct a good spanning tree T of H. This framework is formally presented in Section 3. The novel ingredient of the algorithm lies in the last step: constructing a good spanning tree T of H.

As observed by Chekuri *et al* [7], the graph H is essentially a *Halin* graph, which can be viewed as a planar embedding of a tree merged with a cycle. Indeed, our main challenge is to construct low stretch spanning trees for (a generalization of) Halin graphs, as opposed to the non-subgraph dominating trees constructed in [7]. Our construction is completely different than the construction of Chekuri *et al* [7] and it relies on reducing the task of constructing a low stretch spanning tree for a planar graph to that of constructing a low stretch spanning tree for its planar dual (see Theorem 2.3). This reduction is employed in two distinct occasions within a series of graph manipulations presented in Section 4.

2 Preliminaries

Consider an n-vertex connected graph G. Let $V(G)$ and $E(G)$ denote the vertex and edge sets of G, respectively. Each edge $e \in E(G)$ is associated with some *length* $\ell(e) \in \mathbb{R}_{>0}$. The *length* of a path P in the graph is the sum of lengths of the edges in the path, denoted by $\ell(P) = \sum_{e \in E(P)} \ell(e)$. Given two vertices $u, v \in V(G)$, let $\delta_G(u, v)$ denote the *distance* between them in G, namely, the length of a shortest path from u to v. The *degree* of u, denoted $\deg(u)$, is defined as the number of edges incident on u in G. It will be convenient for us to define the reciprocal of the length of edge e as its *width*, denoted by $w(e) = 1/\ell(e)$.

In what follows we do not distinguish between graphs and multigraphs (namely, a graph may have edge multiplicities). We say that the graph G is *simple*[2] if G contains at most one edge with

[1] A careful implementation of our algorithm leads to an expected linear running time, however it requires some additional complications which are beyond the scope of this paper (that focuses on the existential angle). A polynomial time implementation for our algorithm is straightforward.

[2] Self loops are ignored in this paper.

endpoints u and v for every two vertices $u, v \in V(G)$. Edges that share both endpoints are called *replicas*. Replicas are usually assumed to have the same length (and width). Given two vertices $u, v \in V(G)$, the *multiplicity* of u and v in G, denoted by $\mu_G(u, v)$, is defined to be the number of (u, v)-replicas, i.e., the number of edges connecting u and v. The *skeleton* H of G is the graph obtained from G when all replicas $e_1, \ldots, e_m$ of an edge $(u, v) \in E(G)$, $m = \mu_G(u, v)$, are identified to a single edge $e_{u,v} \in E(H)$ with $w(e_{u,v}) = \sum_{i=1}^{m} w(e_i)$. Clearly, the skeleton H is a simple graph. Given a class $\mathcal{C}$ of graphs, and assuming that $\mathcal{C}$ is not closed under edge replication, the class *replicated-*$\mathcal{C}$ consists of every graph whose skeleton is in $\mathcal{C}$.

A path $\pi = (v_1, \ldots, v_k)$ in G is said to be *isolated* if $\deg(v_1), \deg(v_k) \neq 2$ and $\deg(v_i) = 2$ for every $1 < i < k$ (recall that $\deg(v)$ refers to the degree of v in G). The graph H obtained from G by contracting every isolated path π to a single edge e_π with $\ell(e_\pi) = \ell(\pi)$ is referred to as the *crux* of G. It is easy to verify that distances between vertices of degree different than 2 in H agree with those in G. Given a class $\mathcal{C}$ of graphs, and assuming that $\mathcal{C}$ is not closed under edge subdivision, the class *subdivided-*$\mathcal{C}$ consists of every graph whose crux is in $\mathcal{C}$.

A graph is called *biconnected* if the removal of any single vertex does not separate it. A *block* is a maximal biconnected subgraph. Clearly, every spanning tree of G can be edge-partitioned into spanning trees of the blocks of G.

Stretch and load. Consider some spanning tree T of G and let $e = (u, v)$ be some edge in $E(G)$. The *stretch* of e in T with respect to G is defined to be

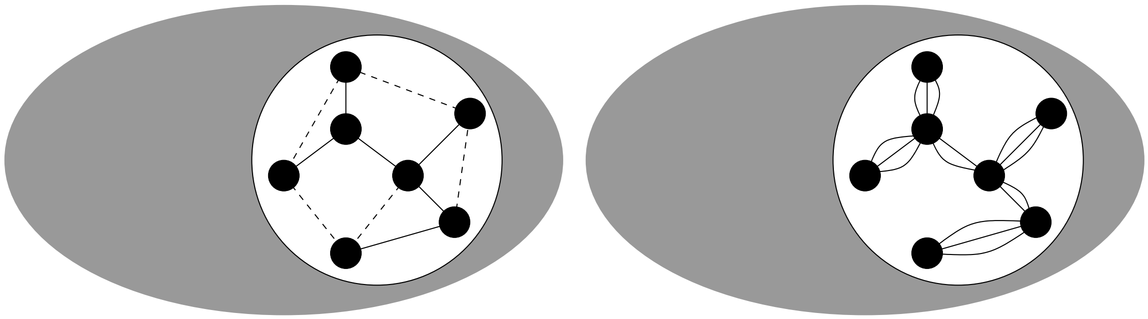
$$\mathrm{str}_{T,G}(e) = \delta_T(u, v)/\ell(e) \ .$$

(Observe that the stretch of e in the spanning tree T does not depend on the graph G, but our notation mentions G to recall that this is the graph that "hosts" T.) The *total stretch* of T with respect to G is denoted by $\text{tot-str}_G(T) = \sum_{e \in E(G)} \mathrm{str}_{T,G}(e)$ and the *average stretch* of T with respect to G is simply $\text{av-str}_G(T) = \text{tot-str}_G(T)/|E(G)|$.

Let $\mathrm{cut}_T(e) \subseteq V(G) \times V(G)$ be the set of all vertex pairs which are connected in T via e (if $e \notin E(T)$, then $\mathrm{cut}_T(e) = \emptyset$). The *load* of e in T with respect to G is defined to be

$$\mathrm{load}_{T,G}(e) = \sum_{e' \in E(G) \cap \mathrm{cut}_T(e)} w(e')/w(e) \ .$$

The *total load* of T with respect to G is denoted by $\text{tot-load}_G(T) = \sum_{e \in E(G)} \mathrm{load}_{T,G}(e)$ and the *average load* of T with respect to G is simply $\text{av-load}_G(T) = \text{tot-load}_G(T)/|E(G)|$. Since $\mathrm{load}_{T,G}(e) = 0$ for every edge $e \in E(G) - E(T)$, we can rewrite $\text{tot-load}_G(T) = \sum_{e \in E(T)} \mathrm{load}_{T,G}(e)$.



(a) G (b) $\check{G}$

Figure 1: The white circle depicts the vertex induced subgraph H in (a) and the load-replication $\widehat{T}$ in (b). The spanning tree T is captured by the solid bold lines in (a); the dashed lines represent edges in $E(H) - E(T)$. The gray zones depict the rest of the graph G.

By a simple change of summation, we obtain

$$\begin{aligned}\sum_{e \in E(G)} \mathrm{load}_{T,G}(e) &= \sum_{e \in E(T)} \sum_{e' \in E(G) \cap \mathrm{cut}_T(e)} w(e')/w(e) \\ &= \sum_{e' \in E(G)} \sum_{e \in E(T) : e' \in \mathrm{cut}_T(e)} \ell(e)/\ell(e') \\ &= \sum_{e' \in E(G)} \frac{1}{\ell(e')} \delta_T(e') \ ,\end{aligned}$$

thus establishing the following corollary which implies that our focus may be shifted from the construction of low average stretch spanning trees to that of low average load spanning trees.

Corollary 2.1. *For every graph G and spanning tree T of G, we have* $\text{tot-str}_G(T) = \text{tot-load}_G(T)$.

Consider some graph H and let T be a spanning tree of H. The *load-replication* $\widehat{T}$ of T under H is the graph obtained from T if each edge $e \in E(T)$ is replicated $\lceil \mathrm{load}_{T,H}(e) \rceil$ times, namely, $\mu_{\widehat{T}}(e) = \lceil \mathrm{load}_{T,H}(e) \rceil$. Clearly, the cardinality of the edge set of $\widehat{T}$ serves as an upper bound on (and a good estimation of) the total load of T under H. Taking load-replications of spanning trees is a fundamental step in our construction based on the following lemma (refer to Figure 1 for illustration).

Lemma 2.2. *Consider some graph G, a vertex induced subgraph H of G, and a spanning tree T of H. Let $\widehat{T}$ be the load-replication of T under H and let $\check{G}$ be the graph resulting from G if H is replaced by $\widehat{T}$, that is, $V(\check{G}) = V(G)$ and $E(\check{G}) = (E(G) - E(H)) \cup E(\widehat{T})$. Consider some spanning tree $\check{T}$ of $\check{G}$ (by definition, $\check{T}$ is also a spanning tree of G). Then* $\mathrm{load}_{\check{T},G}(e) \leq \mathrm{load}_{\check{T},\check{G}}(e)$ *for every edge $e \in E(\check{T})$.*

Proof. Consider some edges $e \in E(\check{T})$ and $f \in E(G)$. The assertion is established by showing that the (direct) contribution c of f to $\mathrm{load}_{\check{T},G}(e)$ is bounded from above by the (possibly indirect) contribution $\check{c}$ of f to $\mathrm{load}_{\check{T},\check{G}}(e)$. We may assume that $f \in \mathrm{cut}_{\check{T}}(e)$, and hence $c = w(f)/w(e)$, since otherwise, $c = 0$. Moreover, if $f \in E(G) - E(H)$, then f is also an edge of $\check{G}$ which means that $c = \check{c}$. Therefore we subsequently assume that $f \in E(H)$.

Indeed, let P be the unique path in T between the endpoints of f. We argue that $E(P) \cap \mathrm{cut}_{\check{T}}(e) \neq \emptyset$. Indeed, Let $\pi_{e'}$ be the unique path in $\check{T}$ between the endpoints of e' for every edge $e' \in E(P)$. If none of the edges in $E(P)$ is in $\mathrm{cut}_{\check{T}}(e)$, then we can take the union of the paths $\pi_{e'}$ for all $e' \in E(P)$ and obtain a path between the endpoints of f in $\check{T}$ that does not go via e, in contradiction to the assumption that $f \in \mathrm{cut}_{\check{T}}(e)$. Therefore there must exist some edge $e' \in E(P) \cap \mathrm{cut}_{\check{T}}(e)$.

Edge f accounts for at least $w(f)/w(e')$ (fractional) replicas of e' in $\widehat{T}$ and hence in $\check{G}$ (it accounts for slightly more if $\mathrm{load}_{T,H}(e')$ is not integral). Each of these replicas contributes $w(e')/w(e)$ to $\mathrm{load}_{\check{T},\check{G}}(e)$, which sums up to $\check{c} \geq \frac{w(f)}{w(e')} \cdot \frac{w(e')}{w(e)} = c$. The assertion follows. □

Lemma 2.2 essentially states that if we can construct a spanning tree T of H with low $\text{tot-load}_H(T)$, then for the sake of analysis, we can replace H in G by the load-replication of T under H (a replicated-tree) and continue from there.

Planar duality. Consider some planar graph G and fix some planar embedding η of G. The *planar dual* $\tilde{G}$ of G under η is the graph which has a vertex v_ϕ corresponding to each face ϕ in η and an edge $\tilde{e} = (v_\phi, v_{\phi'})$ corresponding to each edge $e \in E(G)$ on the boundary of the faces ϕ and ϕ' in η. The planar embedding η uniquely determines a *dual* planar embedding $\tilde{\eta}$ of $\tilde{G}$. It is well known that G is the planar dual of $\tilde{G}$ under $\tilde{\eta}$. We refer to the vertex $v_\phi \in V(\tilde{G})$ as the *dual* of the face ϕ and to the edge $\tilde{e} \in E(\tilde{G})$ as the *dual* of the edge $e \in E(G)$ with respect to the planar duality $\langle \eta, \tilde{\eta} \rangle$. Refer to Figure 2 for an illustration of a planar graph and its planar dual. We associate lengths (and widths) with the dual edges by setting $\ell(\tilde{e}) = w(e)$ (and $w(\tilde{e}) = \ell(e)$) for every $\tilde{e} \in E(\tilde{G})$. Clearly, this definition of dual edge lengths does not violate the bi-directionality of the planar duality $\langle \eta, \tilde{\eta} \rangle$, i.e., it is still true that if $\tilde{G}$ is the dual of G under η, then G is the dual of $\tilde{G}$ under $\tilde{\eta}$.

Consider some spanning tree T of G. The *dual* of T with respect to the planar duality $\langle \eta, \tilde{\eta} \rangle$ is the subgraph $\tilde{T}$ of $\tilde{G}$ defined by setting $V(\tilde{T}) = V(\tilde{G})$ and $E(\tilde{T}) = \{\tilde{e} \in E(\tilde{G}) \mid e \in E(G) - E(T)\}$. Whitney [18] proves that $\tilde{T}$ is a spanning tree of $\tilde{G}$. (Refer to Figure 2 for illustration.) Combined with the notion of load, we extend Whitney's proof to establish the following lemma.

Lemma 2.3. *The dual $\tilde{T}$ of T with respect to the planar duality $\langle \eta, \tilde{\eta} \rangle$ is a spanning tree of $\tilde{G}$. Moreover, $|\mathrm{str}_{T,G}(e) - \mathrm{load}_{\tilde{T},\tilde{G}}(\tilde{e})| \leq 1$ for every edge $e \in E(G)$, and therefore* $\text{av-load}_{\tilde{G}}(\tilde{T}) \leq \text{av-load}_G(T) + 1$.

Proof. The proof relies on the following notions from matroid theory (refer to Graham, Grötschel, and Lovász [11] for a more detailed exposition of the subject). A *matroid* M over the domain $\mathcal{D}$ is a collection of subsets of $\mathcal{D}$ that satisfies the following three properties: (1) $\emptyset \in M$; (2) if $X \in M$

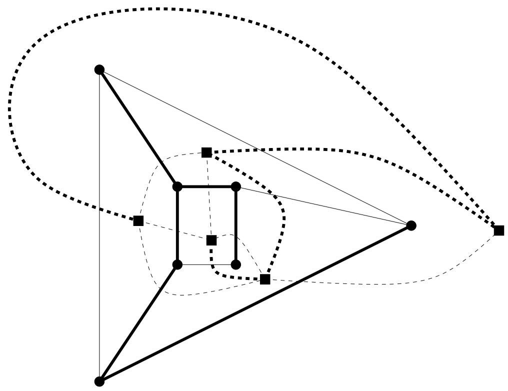


Figure 2: A planar primal graph (round vertices, solid edges) and its planar dual (square vertices, dashed edges). Spanning trees are represented by bold edges.

and $Y \subseteq X$, then $Y \in M$; and (3) if $X, Y \in M$ and $|X| > |Y|$, then there exists some element $e \in X - Y$ such that $Y \cup \{e\} \in M$. The subsets $X \in M$ are called *independent sets*; a maximal independent set is referred to as a *basis*. The *dual* M^* of M is the collection consisting of every subset $X \subseteq \mathcal{D}$ such that $\mathcal{D} - X$ contains a basis of M. One of the fundamental facts of matroid theory is that M^* is also a matroid. It is well known (and easy to verify) that the collection of all cycle-free edge subsets of a graph is a matroid, called the *forest matroid*.

In attempt to prove the assertion, we first argue that an edge set $C \subseteq E(G)$ induces a cycle on G if and only if the dual edge set $\tilde{C} = \{\tilde{e} \mid e \in C\}$ is a cut in $\tilde{G}$.[3] Indeed, if C induces a cycle on G, then it encloses some region R of the plane under η. Any path in $\tilde{G}$ from a vertex corresponding to a face in R to a vertex corresponding to a face outside R must go via an edge of $\tilde{C}$, thus $\tilde{C}$ is a cut in $\tilde{G}$. Conversely, if $\tilde{C}$ is a cut in $\tilde{G}$ separating between the vertices $v_\phi, v_{\phi'} \in V(\tilde{C})$, then the edges of C must enclose a region R in the plane under η such that ϕ is in R and ϕ' is outside R or vice versa. Thus C induces a cycle on G.

Let F be the forest matroid of G and let F^* be its dual matroid. By definition, $X \subseteq E(G)$ is an independent set of F^* if and only if $E(G) - X$ contains a basis of F, that is, a spanning tree of G. The latter holds if and only if X does not contain a cut in G, and this holds if and only if $\tilde{X} = \{\tilde{e} \mid e \in X\}$ does not contain a cycle of $\tilde{G}$, i.e., it is a forest of $\tilde{G}$ and an independent set of the forest matroid $\tilde{F}$ of $\tilde{G}$. Therefore the dual F^* of the forest matroid of G corresponds to the the forest matroid $\tilde{F}$ of the planar dual graph $\tilde{G}$. In particular, X is a basis of F^* (which means that $E(G) - X$ is a spanning tree of G) if and only if $\tilde{X}$ is a basis of $\tilde{F}$ (which means that $\tilde{X}$ is a spanning tree of $\tilde{G}$). It follows that $\tilde{T}$ is a spanning tree of $\tilde{G}$.

Now, consider some edge $e \in E(G)$. If $e \in E(T)$, then $\mathrm{str}_{T,G}(e) = 1$ and by definition, $\tilde{e} \notin E(\tilde{T})$, hence $\mathrm{load}_{\tilde{T},\tilde{G}}(\tilde{e}) = 0$. Assume that $e \notin E(T)$. Let $C \subseteq E(G)$ be the edge set of the cycle that e

[3]This is actually a well known fact. We establish it here for the sake of completeness.

closes when appended to T. We have

$$\begin{aligned}
\mathrm{str}_T(e) &= \delta_T(e)/\ell(e) \\
&= \sum_{e' \in C} \ell(e')/\ell(e) \; - 1 \\
&= \sum_{\tilde{e'} \in E(\tilde{G}) \cap \mathrm{cut}_{\tilde{T}}(\tilde{e})} w(\tilde{e'})/w(\tilde{e}) \; - 1 \\
&= \mathrm{load}_{\tilde{T},\tilde{G}}(\tilde{e}) - 1 \ ,
\end{aligned}$$

where the third equation follows from the observation that $\tilde{C} = \{\tilde{e'} \mid e' \in C\}$, which, as we already argued, is a cut in $\tilde{G}$, is exactly the cut obtained by removing $\tilde{e}$ (the dual edge of e) from $\tilde{T}$, i.e., $E(\tilde{G}) \cap \mathrm{cut}_{\tilde{T}}(\tilde{e})$. The assertion follows. $\square$

Graph classes. Given a planar embedding η of G, we say that η is *outerplanar* (or 1-*outerplanar*) if all vertices of G are incident on the unbounded (outer) face in η. Inductively, η is said to be k-*outerplanar*, $k \geq 2$, if by removing the vertices incident on the unbounded face (and the edges incident on these vertices), we obtain a $(k-1)$-outerplanar embedding of the remaining graph. The graph G is called k-*outerplanar* if it admits a k-outerplanar embedding. (An *outerplanar* graph is simply a 1-outerplanar graph.) Observe that outerplanar graphs are closed under edge replication and not closed under edge subdivision.

A *bush* H is a planar graph obtained by taking a planar embedding of a simple cycle C, embedding a forest T in the region enclosed by C (C and T are disjoint), and introducing some new edges, each one of them has at least one endpoint in C. In other words, the (planar) bush H is defined by taking $V(H) = V(T) \cup V(C)$, $V(C) \cap V(T) = \emptyset$, and $E(H) = E(C) \cup E(T) \cup D$, where $D \subseteq V(C) \times (V(C) \cup V(T))$. If each vertex of C has degree at most 3 in H (i.e., it is adjacent to at most one vertex other than its two neighbors in the cycle), then we say that the bush H is a *Halin* graph[4]. (Refer to Figure 3 for an illustration of a Halin graph.) Observe that bushes (and Halin graphs) are closed under edge subdivision and not closed under edge replication.

Consider some planar graph G. A vertex $u \in V(G)$ is said to be a *dominating* vertex if it is adjacent to all other vertices of G, that is, if $(u, v) \in E(G)$ for every vertex $v \in V(G) - \{u\}$. The graph is called a *dominated* graph if it has a dominating vertex. A vertex $u \in V(G)$ is said to be a *pivot* vertex if all simple cycles in G go via u. The graph is called a *pivot* graph if it has a pivot vertex. Observe that dominated graphs are closed under edge replication and not closed under edge subdivision. In contrast, pivot graphs are closed under edge subdivision and not closed under edge replication.

Suppose that G is a subdivided-dominated graph and let H be its crux. H is a dominated graph, thus it admits a dominating vertex v. Clearly, v is also a vertex of G. Moreover, v is

[4]Our definition of Halin graphs slightly generalizes the original definition of Halin [13].

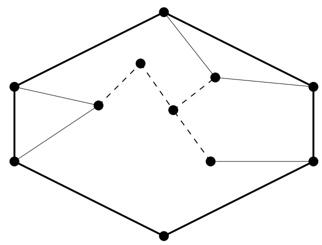


Figure 3: A Halin graph composed of a cycle (solid bold edges), a forest (dashed edges), and some edges connecting cycle vertices to tree vertices (solid thin edges).

connected by an isolated path (in G) to every vertex of degree different than 2 in $V(G) - \{v\}$. We refer to v as a *weak dominating vertex* of G.

Useful assumptions. Recall that the graph G may have arbitrary edge multiplicities. In particular, we cannot bound the number of edges $|E(G)|$ as a function of the number of vertices $n = |V(G)|$. Let m be the number of edges in the skeleton of G, that is, the number of (unordered) vertex pairs $(u, v) \in V(G) \times V(G)$ with $\mu_G(u, v) > 0$. The following lemma, which is essentially derived from Lemma 5.2 of Alon *et al* [2] due to Corollary 2.1, shows that it is sufficient to consider graphs that do not have "too many" edges.

Lemma 2.4 ([2])**.** *For every graph G, there exists some subgraph G' of G on the same vertex set such that (1) $|E(G')| \leq 2m$; and (2) $\text{av-load}_G(T) \leq 2 \cdot \text{av-load}_{G'}(T)$ for every spanning tree T of G'. Moreover, G' can be obtained from G in linear time.*

As G is planar, we know that $m \leq 3n - 6$. Therefore by employing Lemma 2.4, we can subsequently assume that $|E(G)| = O(n)$ at the price of losing a factor of 2 in the performance guarantee. Another assumption we will have to make is that each vertex in G is adjacent to at most three other vertices (although it may be incident on more than three edges due to edge multiplicities). In that case we say that G is *tri-adjacent*. For the purpose of making such an assumption, we introduce a linear time transformation (based on standard techniques), referred to as the *spreading* transformation. The spreading transformation depends on a real parameter $\tau > 0$ and its properties are stated in the following lemma.

Lemma 2.5. *Let G' be the outcome of the spreading transformation when applied to G with parameter τ. Then G' satisfies the following properties: (1) G' is tri-adjacent; (2) if G is k-outerplanar, then so is G'; and (3) every spanning tree T' of G' that satisfies $\text{av-load}_{G'}(T') \leq \tau$ can be* translated *in linear time back into a spanning tree T of G such that $\text{av-load}_G(T) \leq 3\tau$.*

Proof. The spreading transformation works as follows. Every vertex $u \in V(G)$ with neighbors $v_1, \ldots, v_j$ in G, where $j > 3$, is replaced by the simple path $\pi_u = (u_1, \ldots, u_j)$ in G'. For every $1 \leq i \leq j$, each edge $(u, v_i) \in E(G)$ is incident on u_i in G' (the length of the edge remains unchanged). We refer to the edges in the newly introduced paths π_u as *new* edges; all other edges in G' are referred to as *old* edges. Let $\lambda = \min\{\ell(e) \mid e \text{ is an old edge}\}$. We set the length of each new edge $e \in E(G')$ to be $\ell(e) = \lambda/(\tau \cdot |E(G')|)$.

Clearly, G' is tri-adjacent (every vertex has at most two neighbors in some newly introduced path and at most one "original" neighbor). Moreover, it is easy to verify that this transformation can be made to preserve k-outerplanarity (recursively make sure that the path π_u that replaces a vertex u incident on the original unbounded face, forms a part of the boundary of the new unbounded face). As for the last property, we first observe that we can charge every new edge in G' on some old edge such that no old edge is charged more than twice, hence $|E(G')| \leq 3|E(G)|$.

Now, consider some spanning tree T' of G' and suppose that $\text{tot-load}_{G'}(T') \leq \tau \cdot |E(G')|$. We argue that all new edges participate in T'. Indeed, if some new edge e is not in $E(T')$, then there must exist some old edge $f \in E(G')$ such that $e \in \text{cut}_{T'}(f)$. By the choice of $\ell(e)$, it follows that the contribution of e to the load of f in T' with respect to G' is $w(e)/w(f) = \ell(f)/\ell(e) \geq \tau \cdot |E(G')|$. The total load of T' with respect to G' is strictly greater than that, in contradiction to the assumption that $\text{tot-load}_{G'}(T') \leq \tau \cdot |E(G')|$.

We shall translate T' into a spanning tree T of G, simply by contracting the vertices $u_1, \dots, u_j \in V(G')$ (which form a connected component in T' since $(u_i, u_{i+1}) \in E(T')$ for every $1 \leq i < j$) into the original vertex $u \in V(G)$. The load of an old edge e in T with respect to G is equal to its load in T' with respect to G', thus $\text{tot-load}_G(T) \leq \text{tot-load}_{G'}(T') \leq \tau \cdot |E(G')| \leq 3\tau \cdot |E(G)|$. The assertion follows. $\square$

Assuming that the input graph G is tri-adjacent, we will construct in the remainder of the paper a spanning tree T of G that satisfies $\text{av-load}_G(T) \leq c^k$. Therefore by employing Lemma 2.5 with parameter $\tau = c^k$, we may subsequently make this assumption at the price of losing a factor of 3 in the performance guarantee.

3 The algorithm — peeling an onion

Our goal in this section (and in the whole paper) is to prove the following theorem.

Theorem 3.1. *For every k-outerplanar graph G, there exists a spanning tree T such that* $\text{av-load}_G(T) \leq c^k$*, where c is a universal constant (independent of k and G).*

The proof of Theorem 3.1 is constructive: we present an efficient randomized algorithm, referred to as the *onion peeling algorithm*, that given a k-outerplanar graph G with a realizing planar embedding η, constructs the desired spanning tree T of G. Recall our previous assumptions that $|E(G)| = O(n)$, where $n = |V(G)|$ (due to Lemma 2.4) and that G is tri-adjacent (due to Lemma 2.5). The onion peeling algorithm is based on a recursive process similar to that presented by Chekuri *et al* [7] (and essentially, to many other recursive processes on k-outerplanar graphs, cf. Baker [3]). However, the main building block of the onion peeling algorithm, namely, the construction of low stretch spanning trees for (replicated) Halin graphs, is entirely different (see Section 4). This also leads to a different type of analysis.

Informally, the onion peeling algorithm works as follows (a detailed description is provided later on): (i) remove the vertices on the unbounded face of G (and the edges incident on these vertices) to obtain a $(k-1)$-outerplanar graph G'; (ii) recursively construct a "good" spanning tree T' for G'; (iii) insert the vertices (and edges) that were removed in step (i) back into the planar embedding of T' to compose the graph H; and (iv) construct a "good" spanning tree T of H.

Our algorithm relies on two fundamental constructions. First (implicit in the above description), when the recursion reaches its halting condition on a 1-outerplanar graph G, we have to construct a "good" spanning tree T of G. This is done via the randomized construction of Gupta *et al* [12] that probabilistically embeds a given outerplanar graph G into its spanning trees with constant distortion. As we will see later on, this randomized construction of Gupta *et al* is employed by our algorithm in several occasions, and it is subsequently referred to as Procedure `GNRS`. Actually, we shall use a variant of Procedure `GNRS` (the procedure's name is kept, though) whose input may be a subdivided-outerplanar graph[5]. The performance guarantee of Procedure `GNRS` is stated in the following theorem.

Theorem 3.2 ([12])**.** *Procedure* `GNRS`*, when invoked on a subdivided-outerplanar graph G with a realizing planar embedding η, runs in expected polynomial time and returns a spanning tree T of G that satisfies* $\text{av-load}_G(T) \leq c_1$*, where c_1 is a universal constant (independent of G).*

The second fundamental construction on which the onion peeling algorithm relies is the construction of a "good" spanning tree T of H (step (iv)). A crucial observation in this context is that H is a replicated-Halin graph (actually, if G is simple, then H is strictly a Halin graph). This is due to the assumption that G is tri-adjacent (without which, H would have been a replicated-bush). The technique of Chekuri *et al* [7] probabilistically embeds a Halin graph H into a collection of dominating trees with constant distortion, but these dominating trees are not necessarily spanning trees of H. By contrast, we present a procedure, called Procedure `RH`, which guarantees that T is a spanning tree of H. The input of Procedure `RH` is not assumed to be a (simple) Halin graph, but rather a replicated-Halin graph (hence the name). The performance guarantee of Procedure `RH` is stated in the following theorem, proved in Section 4.

Theorem 3.3. *Procedure* `RH`*, when invoked on a replicated-Halin graph G with a realizing planar embedding η, runs in expected polynomial time and returns a spanning tree T of G that satisfies* $\text{av-load}_G(T) \leq c_2$*, where c_2 is a universal constant (independent of G).*

This leads to the question: what do we mean by a "good" spanning tree? In most of the previous works which considered graph composition based on replacing a subgraph H by a tree T (including the work of Chekuri *et al* [7]), the tree was chosen randomly according to some

[5]By employing a simple technique presented in [12], one can contract isolated paths at the price of increasing the distortion of the probabilistic embedding by at most 2.

Given a biconnected k-outerplanar graph G and a k-outerplanar embedding of G, do:

1. If $k = 1$, then invoke Procedure `GNRS` on G to produce the spanning tree T; return T together with its load-replication $\widehat{T}$ under G (the latter is required for the recursion to be properly defined);
2. Remove the vertices on the unbounded face of G (and the edges incident on these vertices) to obtain a $(k-1)$-outerplanar graph G'.
3. Let $G'_1, \ldots, G'_m$ be the blocks of G'.
4. For $i = 1, \ldots, m$, recursively invoke the onion peeling algorithm on G'_i to produce the spanning tree T'_i of G'_i and the replicated-tree $\widehat{T}'_i$.
5. Insert the vertices and edges that were removed in step 2 back into the planar embedding of $\widehat{T}'_1, \ldots, \widehat{T}'_m$ to compose the replicated-Halin graph H.
6. Invoke Procedure `RH` on H to produce the spanning tree T.
7. Return T together with its load-replication $\widehat{T}$ under H.

Table 1: The onion peeling algorithm in detail.

probability distribution (that may be supported on many trees) and the goal was to guarantee low distortion. In this work we use a different approach: we shall construct a single tree T and our goal is to guarantee low total load. (Corollary 2.1 stating that the total load is equal to the total stretch, implies that our approach can be viewed as a relaxation of the previous approach.)

Recall that Lemma 2.2 essentially implies that for the sake of analysis, we may replace the graph G' (the outcome of step (i)) with the load replication of T' (the outcome of step (ii)) under G' before inserting back the vertices and edges that were removed in step (i) and continue with the construction from there. The onion peeling process, whose detailed description is presented in Table 1, revolves around this phenomenon (refer to Figure 4 for illustration). For simplicity of the exposition, the description in Table 1 assumes that the input k-outerplanar graph G is biconnected. This assumption can be easily eliminated by generating a spanning tree for each block of G separately and returning the union of all generated trees.

Analysis. We now turn to analyze the onion peeling algorithm, starting with a brief outline. Theorem 3.1 is proved by induction on k. We first employ Theorems 3.2 (induction's base) and 3.3 (induction's step) to show that $|E(H)| \leq c^{k-1} \cdot |E(G)|$, where $c = c(c_1, c_2)$ is a universal constant. Next, we use Lemma 2.2 to argue that $\text{tot-load}_G(T) \leq \text{tot-load}_H(T)$. Finally, Theorem 3.3 guarantees that $\text{tot-load}_H(T) \leq c \cdot |E(H)|$, which completes the analysis as it implies that

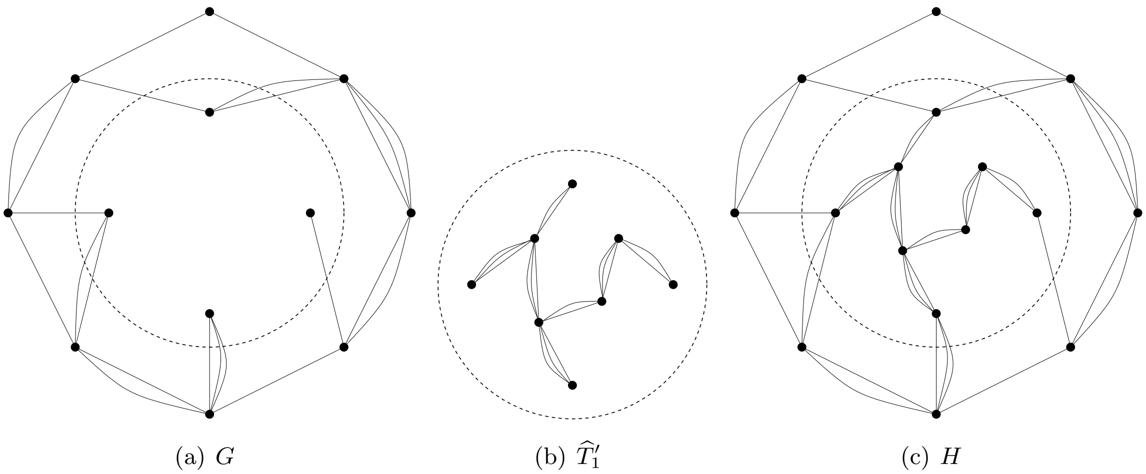


Figure 4: An application of the onion peeling algorithm on G. The vertices of the unbounded face and the edges incident on them are shown in (a), where G' is (hidden) in the dashed circle. The replicated-tree $\widehat{T}_1'$ is shown in (b). The replicated-Halin graph H composed by inserting the vertices and edges of (a) back into the planar embedding of (b) is shown in (c).

$\text{tot-load}_G(T) \leq c^k \cdot |E(G)|$.

For a more detailed (and formal) analysis of the recursive algorithm, we shall *label* each recursive invocation with a string in $\mathbb{Z}^*$. The labeling is done in an inductive manner. The top recursive invocation (on the graph originally input to the algorithm) is labeled with the empty string ω. Consider a recursive invocation labeled with the string σ on the graph G for some $\sigma \in \mathbb{Z}^*$ and assuming that G is not a 1-outerplanar graph, let $G_1', \ldots, G_m'$ be the blocks of G' (see step 3). Then the recursive invocation on the the graph G_i' is labeled with the string σi for every $1 \leq i \leq m$. We refer to the recursive invocation labeled with the string σ as the *σ-recursive invocation*. A string $\sigma \in \mathbb{Z}^*$ is called *valid* if there exists a σ-recursive invocation.

Consider some valid string σ. We denote the graph G input to the σ-recursive invocation by G_σ; the replicated-Halin graph H by H_σ (see step 5); the spanning tree T of H by T_σ (see step 6); and the load-replication $\widehat{T}$ of T under H by $\widehat{T}_\sigma$ (see step 7). We say that σ is *maximal* if G_σ is 1-outerplanar (in that case, all strings with proper prefix σ are not valid). For completeness, we define $H_\sigma = G_\sigma$ when σ is maximal (in which case, H is not defined in the algorithm). Observe that if G_σ is k'-outerplanar, where $1 \leq k' \leq k$, then $|\sigma| \leq k - k'$.

Fix[6] $\bar{c} = \max\{c_1, c_2\}$. Recall that Theorems 3.2 and 3.3 guarantee that $\text{tot-load}_{H_\sigma}(T_\sigma) \leq \bar{c} \cdot |E(H_\sigma)|$. We are now ready to prove the following two lemmas.

Lemma 3.4. *For every valid string σ, we have $|E(H_\sigma)| \leq (\bar{c}+1)^{k-|\sigma|-1} \cdot |E(G_\sigma)|$.*

Proof. The proof is by backward induction on the length of σ. The base case is trivial: if σ is

[6] As it turns out in Section 4, $c_2 > c_1$, so $\bar{c} = c_2$ and the separate notation is redundant.

maximal, and in particular, if $|\sigma| = k-1$, then $H_\sigma = G_\sigma$. Consider some valid string σ and assume by induction that the assertion holds for all valid strings of length greater than $|\sigma|$. Let $\sigma 1, \ldots, \sigma m$ be the valid strings of length $|\sigma|+1$, having σ as a prefix.

Since $\text{tot-load}_{H_{\sigma i}}(T_{\sigma i}) \leq \bar{c} \cdot |E(H_{\sigma i})|$, and by the definition of $\widehat{T}_{\sigma i}$, it follows that $|E(\widehat{T}_{\sigma i})| \leq (\bar{c}+1) \cdot |E(H_{\sigma i})|$ for every $1 \leq i \leq m$. Plugging it into the inductive hypothesis, we conclude that $|E(\widehat{T}_{\sigma i})| \leq (\bar{c}+1)(\bar{c}+1)^{k-|\sigma|-2} \cdot |E(G_{\sigma i})| = (\bar{c}+1)^{k-|\sigma|-1} \cdot |E(G_{\sigma i})|$. By the definition of H_σ, we have

$$
\begin{aligned}
|E(H_\sigma)| = & |E(G_\sigma)| + \sum_{i=1}^{m} |E(\widehat{T}_{\sigma i})| - |E(G_{\sigma i})| \\
\leq & |E(G_\sigma)| + \sum_{i=1}^{m} \Big((\bar{c}+1)^{k-|\sigma|-1} - 1 \Big) \cdot |E(G_{\sigma i})| \\
\leq & (\bar{c}+1)^{k-|\sigma|-1} \cdot |E(G_\sigma)| \ ,
\end{aligned}
$$

where the last inequality follows from the fact that $\sum_{i=1}^{m} |E(G_{\sigma i})| \leq |E(G_\sigma)|$. The assertion follows. $\square$

Lemma 3.5. *For every valid string σ and for every edge $e \in E(T_\sigma)$, we have* $\text{load}_{T_\sigma, G_\sigma}(e) \leq \text{load}_{T_\sigma, H_\sigma}(e)$.

Proof. The proof is by backward induction on the length of σ. The base case is trivial: if σ is maximal, and in particular, if $|\sigma| = k-1$, then $H_\sigma = G_\sigma$. Consider some valid string σ and assume by induction that the assertion holds for all valid strings of length greater then $|\sigma|$. Let $\sigma 1, \ldots, \sigma m$ be the valid strings of length $|\sigma|+1$, having σ as a prefix.

Let $\check{T}_{\sigma i}$ be the load-replication of $T_{\sigma i}$ under $G_{\sigma i}$ for every $1 \leq i \leq m$. (The replicated-tree $\check{T}_{\sigma i}$ should not be confused with $\widehat{T}_{\sigma i}$. The latter is the load-replication of $T_{\sigma i}$ under $H_{\sigma i}$.) Let $\check{H}_\sigma$ be the graph composed by inserting the vertices (and edges) that were removed in step 2 of the onion peeling algorithm in the σ-recursive invocation into the planar embedding of $\check{T}_{\sigma 1}, \ldots, \check{T}_{\sigma m}$ (just like step 5 but with the the replicated trees $\check{T}_{\sigma 1}, \ldots, \check{T}_{\sigma m}$). Lemma 2.2 implies that $\text{load}_{T_\sigma, G_\sigma}(e) \leq \text{load}_{T_\sigma, \check{H}_\sigma}(e)$ for every edge $e \in E(T_\sigma)$. By the inductive hypothesis, we have $\mu_{\check{T}_{\sigma i}}(u,v) \leq \mu_{\widehat{T}_{\sigma i}}(u,v)$ for every $1 \leq i \leq m$ and for every two vertices $u, v \in V(G_{\sigma i})$, thus $\mu_{\check{H}_\sigma}(u,v) \leq \mu_{H_\sigma}(u,v)$ for every two vertices $u, v \in V(G_\sigma)$. Therefore $\text{load}_{T_\sigma, \check{H}_\sigma}(e) \leq \text{load}_{T_\sigma, H_\sigma}(e)$ for every edge $e \in E(T_\sigma)$. The assertion follows. $\square$

Recall that $\text{tot-load}_{H_\omega}(T_\omega) \leq \bar{c} \cdot |E(H_\omega)|$. Lemma 3.5 guarantees that the load on each edge e of T_ω with respect to H_ω is at least the load on e with respect to G_ω, hence $\text{tot-load}_{G_\omega}(T_\omega) \leq \bar{c} \cdot |E(H_\omega)|$. By Lemma 3.4, we conclude that $\text{tot-load}_{G_\omega}(T_\omega) \leq \bar{c}(\bar{c}+1)^{k-1} \cdot |E(G_\omega)|$, which establishes Theorem 3.1.

4 Replicated-Halin graphs

In this section we present Procedure RH and prove Theorem 3.3. Recall that the input of Procedure RH is a replicated-Halin graph G with a realizing planar embedding η. The procedure returns a spanning tree T of G which satisfies $\text{av-load}_G(T) \leq c_2$, where c_2 is a universal constant. By Lemma 2.4, we may assume that $|E(G)| = O(n)$. (This assumption is essentially reflected in the constant c_2 which is twice as large as what we obtain in the remainder of this section.)

Overview. Procedure RH is described through a series of reductions presented in Section 4.2. These reductions relies on the structural observations presented in Section 4.1. The first principle of the procedure is that cast in Lemma 2.3: if you do not know how to construct a good spanning tree of G, try to construct a good spanning tree of its dual $\tilde{G}$. In particular, we take advantage of the fact that the planar dual of a Halin graph is a dominated graph and the planar dual of a (planar) pivot graph is an outerplanar graph (see Section 4.1).

The second principle of the procedure is that depicted in Lemma 2.2: a vertex induced subgraph H of G, for which we know how to construct a good spanning tree, can be replaced in G by the corresponding load-replication. In particular, Procedure GNRS is invoked (twice) on some designated vertex induced subgraphs that turn out to be outerplanar. For ease of reference, Figure 6 provides a schematic illustration of the various steps that constitute Procedure RH.

4.1 Taking planar duals.

Taking planar duals of some special classes of graphs is the main ingredient of our construction. Due to the sensitivity of the definition of load to edge multiplicities, we first want to understand how the operation of identifying two replicas in a planar graph affects its planar dual. To this end, suppose that some two replicas e and e' in the planar primal are identified to form a single edge of width $w(e) + w(e')$. In the planar dual this translates to the contraction of the simple path consisting of $\tilde{e}$ and $\tilde{e}'$ into a single edge of length $\ell(\tilde{e}) + \ell(\tilde{e}') = w(e) + w(e')$. The following observation is a direct consequence of this phenomenon.

Observation 4.1. *Let G and $\tilde{G}$ be two planar graphs with planar embeddings η and $\tilde{\eta}$, respectively. Let η' (respectively $\tilde{\eta}'$) be the planar embedding of the skeleton of G (resp., the crux of $\tilde{G}$), naturally derived from η (resp. $\tilde{\eta}$). If η and $\tilde{\eta}$ are duals, then so are η' and $\tilde{\eta}'$.*

We study planar dualities between some specific classes of (planar) graphs. Our insights are cast in the following lemma.

Lemma 4.2. *Consider a biconnected planar graph G with a planar embedding η and let $\tilde{G}$ be the planar dual of G under η.*

1. *If G is outerplanar with η being a realizing planar embedding, then $\tilde{G}$ is a pivot graph.*

2. If G is a pivot graph, then $\tilde{G}$ is an outerplanar graph.

3. If G is a Halin graph with η being a realizing planar embedding, then $\tilde{G}$ is a dominated graph.

4. If G is a dominated graph, then $\tilde{G}$ is a bush.

Proof. We prove the four claims separately.

1. Let ϕ be the unbounded face of G in η. By definition, all vertices of G are incident on ϕ, hence all faces of $\tilde{G}$ in the dual planar embedding $\tilde{\eta}$ are incident on the vertex $v_\phi \in V(\tilde{G})$. We prove that v_ϕ is a pivot vertex of $\tilde{G}$. Assume by way of contradiction that there exists a simple cycle C in $\tilde{G}$ that does not go via v_ϕ. There exist at least one face inside the region enclosed by C in $\tilde{\eta}$ and at least one face outside this region. The cycle C does not go via v_ϕ, thus in $\tilde{\eta}$, v_ϕ is either strictly inside C or strictly outside C. In any case, contradiction is derived since we identified at least one face in $\tilde{\eta}$ which is not incident on v_ϕ.

2. Let $v \in V(G)$ be a pivot vertex of G and let ϕ be the face of $\tilde{G}$ in the dual planar embedding $\tilde{\eta}$ which corresponds to v. By definition, all simple cycles of G go via v, hence in particular, v is incident on all faces of G in η. Therefore ϕ is incident on all vertices of $\tilde{G}$ in $\tilde{\eta}$, and by rearranging the planar embedding so that ϕ becomes the unbounded face, we conclude that $\tilde{G}$ is outerplanar.

3. Let ϕ be the unbounded face of G in η and let C be the simple cycle that corresponds to ϕ. We argue that C must share some vertices with any other simple cycle of G. Indeed, by the definition of Halin graphs, the graph obtained by the removal of the vertices of C is a forest. Another property of Halin graphs is that every vertex of C is adjacent to at most one vertex other than its too immediate neighbors in C, thus every simple cycle in G must also share at least one edge with C. In particular, every face shares at least one edge with ϕ in η. It follows that the vertex $v \in V(\tilde{G})$ which corresponds to ϕ is adjacent to any other vertex of $\tilde{G}$, hence it is a dominating vertex.

4. Let $v \in V(G)$ be a dominating vertex of G and let ϕ be the face of $\tilde{G}$ in the dual planar embedding $\tilde{\eta}$ which corresponds to v. Assume without loss of generality that ϕ is the unbounded face of $\tilde{\eta}$ and let C be the simple cycle which contains the edges of ϕ. By definition, v is adjacent to any other vertex in G, hence ϕ share some edges with any other face of $\tilde{G}$ in $\tilde{\eta}$. It follows that the graph resulting from the removal of C (including their vertices and all their adjacent edges) is a forest and $\tilde{G}$ is a bush.

The assertion follows. $\square$

There is one more structural insight we have to establish before we can present Procedure `RH`. Consider a subdivided-dominated graph G with some planar embedding η of G and let $v \in V(G)$

be a weak dominating vertex of G. Let G' be the graph obtained from G by the removal of v and let η' be the planar embedding of G', naturally derived from η. Let p be the point in the plane (under η) where v was positioned and let ϕ be the face of G' in η' that contains p.

We argue that ϕ is incident in η' on all vertices of degree different than 2 of G'. To see why this is true, recall that v is connected by an isolated path to every vertex of degree different than 2 in $V(G) - \{v\}$. In η this is reflected by a curve leading from p to the position of every vertex of degree different than 2 in $V(G) - \{v\}$. Since η is a planar embedding, and by the definition of an isolated path, it follows that these curves do not intersect with each other or with any other curve in η. Therefore in η' we can draw all these curves inside ϕ. Corollary 4.3 follows by taking ϕ to be the unbounded face of η'.

Corollary 4.3. *Each block of the graph obtained by removing a weak dominating vertex from a subdivided-dominated graph is a subdivided-outerplanar graph.*

4.2 Low load spanning trees for replicated-Halin graphs.

We now turn to describe the operation of Procedure `RH` on a replicated-Halin graph G with a realizing planar embedding η. As usual, we assume that G is biconnected (otherwise, we can break it and construct a separate spanning tree for each block). The procedure works in 8 steps. The outcome of step i is denoted by T^i if it is (surely) a tree; and by G^i if it is a graph that may contain cycles. (The superscript notation should not be confused with graph powers.) In this spirit, we denote the replicated-Halin graph G by G^0. The 8 steps of Procedure `RH` are as follows (refer to Figures 5 and 6 for graphical and schematic illustrations, respectively).

Step 1: Take the planar dual G^1 of G^0 under η. By definition, the skeleton of G^0 is a Halin graph, thus Observation 4.1 and Lemma 4.2 imply that G^1 is a subdivided-dominated graph. Let $v \in V(G^1)$ be a weak dominating vertex of G^1.

Step 2: Remove the vertex v and the edges incident on it from G^1 and let G^2 be the remaining graph. Let $G^2_1, \ldots, G^2_m$ be the blocks of G^2. By Corollary 4.3, G^2_i is a subdivided-outerplanar graph for every $1 \leq i \leq m$.

Step 3: For $i = 1, \ldots, m$, invoke Procedure `GNRS` on G^2_i to generate a spanning tree T^3_i. By Theorem 3.2, we have $\text{tot-load}_{G^2_i}(T^3_i) \leq c_1 \cdot |E(G^2_i)|$.

Step 4: For $i = 1, \ldots, m$, construct the load-replication $\widehat{T}^3_i$ of T^3_i under G^2_i. Insert the vertex and edges that were removed in step 2 back into the planar embedding of the replicated-trees $\widehat{T}^3_1, \ldots, \widehat{T}^3_m$ to compose the graph G^4. Note that $|E(G^4)| \leq (c_1 + 1) \cdot |E(G^1)|$. Since every simple cycle in the skeleton of G^4 must go via v, we conclude that G^4 is a replicated-pivot graph. Let $G^4_1, \ldots, G^4_{m'}$ be the blocks of G^4 (by definition, each of these blocks is also a replicated pivot graph).

Step 5: For $i = 1, \ldots, m'$, fix some arbitrary planar embedding η'_i of G^4_i and let G^5_i be the planar dual of G^4_i under η'_i. By Observation 4.1 and Lemma 4.2, G^5_i is a subdivided-outerplanar graph for every $1 \leq i \leq m'$.

Step 6: For $i = 1, \ldots, m'$, invoke Procedure `GNRS` on G_i^5 to generate a spanning tree T_i^6. By Theorem 3.2, we have $\text{tot-load}_{G_i^5}(T_i^6) \leq c_1 \cdot |E(G_i^5)|$ for every $1 \leq i \leq m'$.

Step 7: For $i = 1, \ldots, m'$, construct the dual T_i^7 of the spanning tree T_i^6 with respect to the planar duality $\langle \tilde{\eta}_i', \eta_i' \rangle$, where $\tilde{\eta}_i'$ is the dual planar embedding of η_i'. Lemma 2.3 guarantees that T_i^7 is a spanning tree of G_i^4 and by Lemma 2.3, we have $\text{tot-load}_{G_i^4}(T_i^7) \leq \text{tot-load}_{G_i^5}(T_i^6) + |E(G_i^5)| \leq (c_1+1) \cdot |E(G_i^5)| = (c_1+1) \cdot |E(G_i^4)|$ for every $1 \leq i \leq m'$. Let T^7 be the union of the trees $T_1^7, \ldots, T_m^7$. Note that T^7 is a spanning tree of G^4 and $\text{tot-load}_{G^4}(T^7) \leq (c_1 + 1) \cdot |E(G^4)|$. Since T^7 is also a spanning tree of G^1, we can apply Lemma 2.2 to deduce that $\text{tot-load}_{G^1}(T^7) \leq \text{tot-load}_{G^4}(T^7) \leq (c_1 + 1) \cdot |E(G^4)| \leq (c_1 + 1)^2 \cdot |E(G^1)|$.

Step 8: Construct the dual T^8 of the spanning tree T^7 with respect to the planar duality $\langle \tilde{\eta}, \eta \rangle$, where $\tilde{\eta}$ is the dual planar embedding of η. Lemma 2.3 guarantees that T^8 is a spanning tree of G^0 and by Lemma 2.3, we have $\text{tot-load}_{G^0}(T^8) \leq \text{tot-load}_{G^1}(T^7) + |E(G^1)| \leq ((c_1+1)^2+1) \cdot |E(G^1)| = ((c_1 + 1)^2 + 1) \cdot |E(G^0)|$.

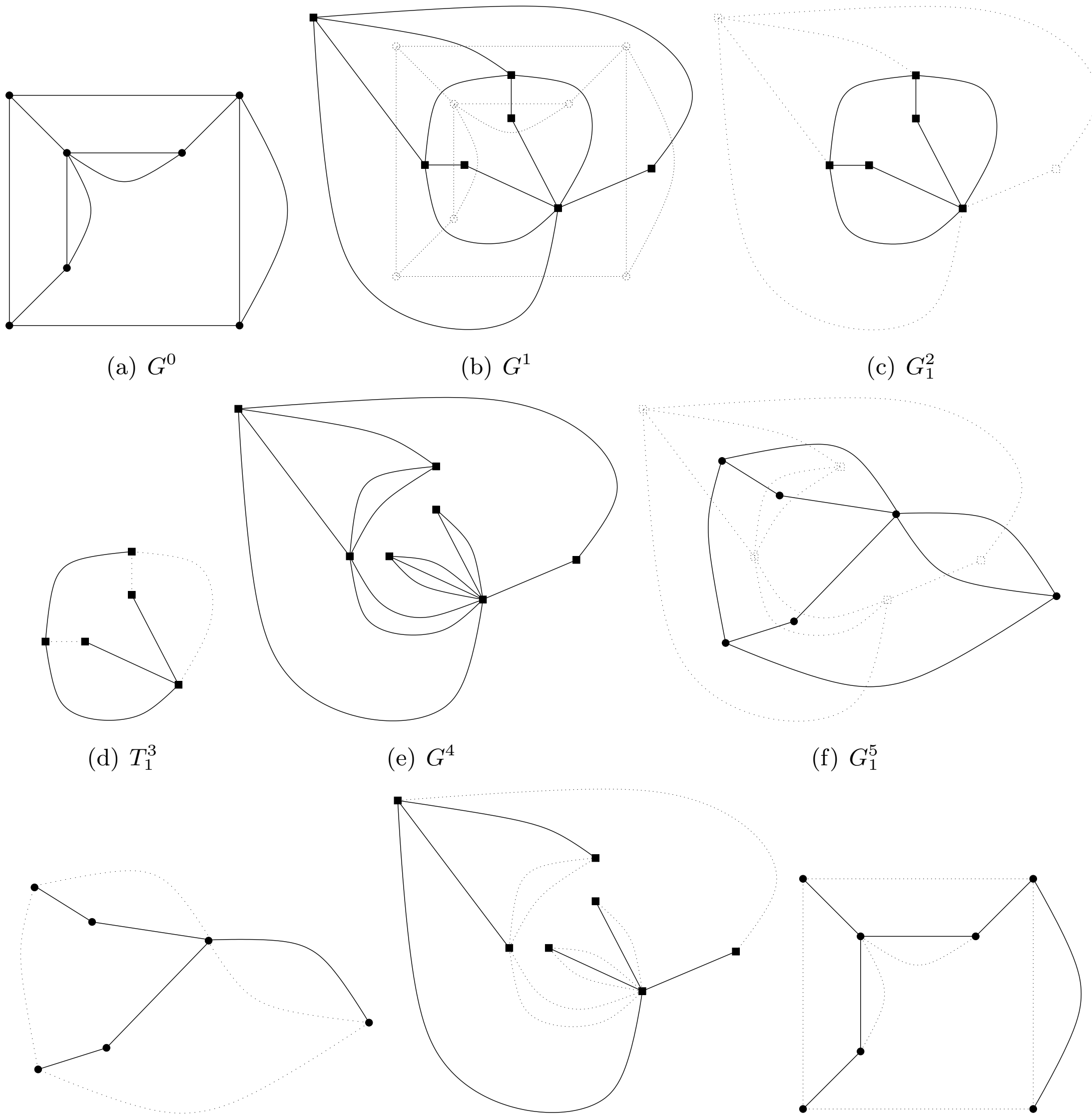
It follows that upon completion of step 8, we obtain a spanning tree $T = T^8$ which satisfies $\text{av-load}_G(T) \leq c_2$, where $c_2 = (c_1 + 1)^2 + 1$ is a universal constant. Theorem 3.3 follows.

5 Conclusions

We prove that every k-outerplanar graph G admits a spanning tree T such that $\text{av-load}_G(T) \leq c^k$, where c is an absolute constant. The same bound holds for the average stretch of T with respect to G based on the duality of load and stretch. We find it more convenient to bound the (total) load of the trees we construct, mainly due to the (fairly natural) load-replication representation which enables some sort of an iterative graph decomposition. (In previous works, similar approaches were based on probabilistic embeddings.) Planar duality plays a major role in our construction. We hope that some of the tools we develop here will prove useful in other types of embeddings of planar graphs (e.g., into L_1).

Acknowledgments

I would like to thank Robert Krauthgamer, Manor Mendel, and David Peleg for helpful discussions.



(a) G^0 (b) G^1 (c) G_1^2

(d) T_1^3 (e) G^4 (f) G_1^5

(g) T_1^6 (h) T^7 (i) T^8

Figure 5: The input replicated-Halin graph G^0 (a) and the products of the 8 steps of Procedure `RH` (b)–(i).

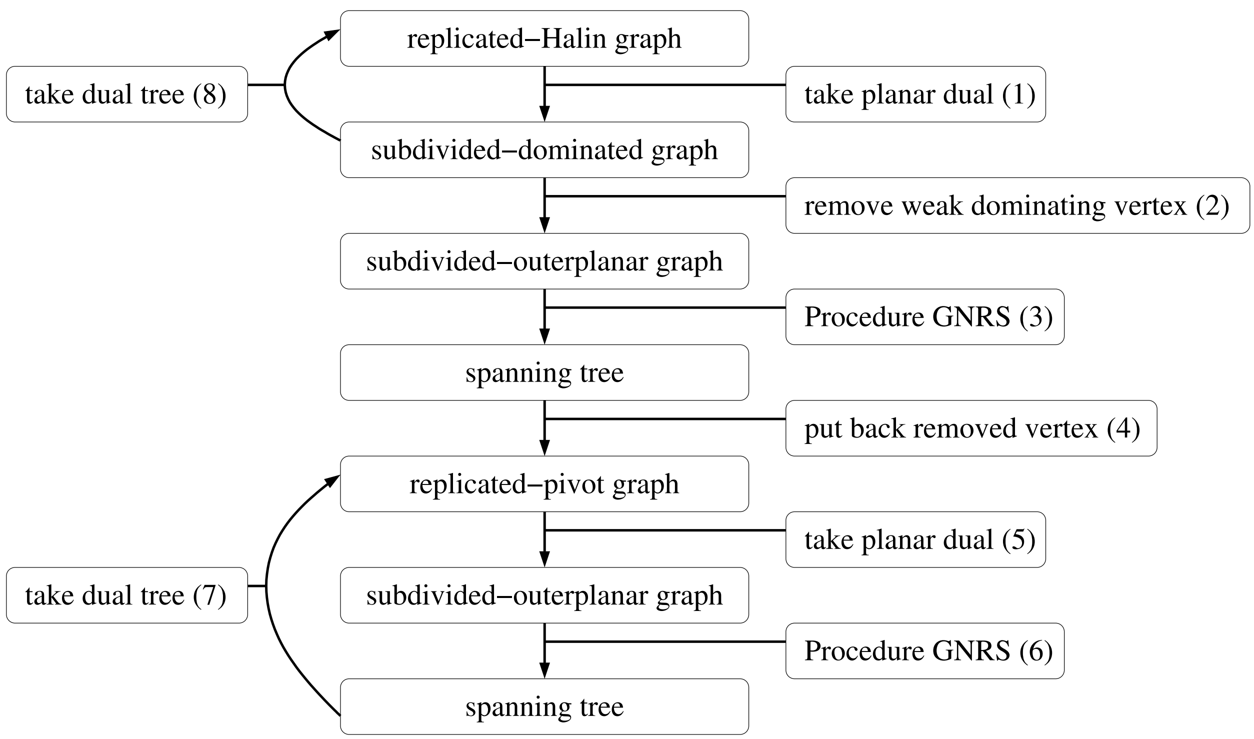


Figure 6: A schematic illustration of Procedure `RH`. The step numbers appear in parentheses.

References

[1] I. Abraham, Y. Bartal, and O. Neiman. Nearly tight low stretch spanning trees. In *Proc. 49th IEEE Symp. on Foundations of Computer Science (FOCS)*, 2008.

[2] 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.*, 24:78–100, 1995.

[3] B.S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994.

[4] Y. Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *Proc. 37th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 184–193, 1996.

[5] Y. Bartal. On approximating arbitrary metrics by tree metrics. In *Proc. 30th ACM Symp. on the Theory of Computing (STOC)*, pages 161–168, 1998.

[6] M. Charikar, C. Chekuri, A. Goel, S. Guha, and S.A. Plotkin. Approximating a finite metric by a small number of tree metrics. In *Proc. 39th Symp. on Foundations of Computer Science (FOCS)*, pages 379–388, 1998.

[7] C. Chekuri, A. Gupta, I. Newman, Y. Rabinovich, and A. Sinclair. Embedding k-outerplanar graphs into ℓ_1. *SIAM J. Discrete Math.*, 20(1):119–136, 2006.

[8] M. Elkin, Y. Emek, D.A. Spielman, and S.-H. Teng. Lower-stretch spanning trees. *SIAM J. Comput. (SICOMP)*, 38(2):608–628, 2008.

[9] Y. Emek and D. Peleg. A tight upper bound on the probabilistic embedding of series-parallel graphs. In *Proc. 17th ACM-SIAM Symp. on Discrete algorithm (SODA)*, pages 1045–1053, 2006.

[10] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci. (JCSS)*, 69(3):485–497, 2004.

[11] Editors: R.L. Graham, M. Grötschel, and L. Lovász. *Handbook of combinatorics (vol 1)*, MIT Press, 1995.

[12] A. Gupta, I. Newman, Y. Rabinovich, and A. Sinclair. Cuts, trees and ℓ_1-embeddings of graphs. *Combinatorica*, 24(2):233–269, 2004.

[13] R. Halin. Über simpliziale Zerfällungen beliebiger (endlicher oder unendlicher) Graphen. *Mathematische Annalen*, 156(3):216-225, 1964.

[14] T.C. Hu. Optimum communication spanning trees. *SIAM J. Comput.*, 3:188–195, 1974.

[15] P. Indyk and A. Sidiropoulos. Probabilistic embeddings of bounded genus graphs into planar graphs. In *Proc. 23rd ACM Symp. on Computational Geometry (SoCG)*, pages 204–209, 2007.

[16] P.N. Klein, S.A. Plotkin, and S. Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proc. 25th ACM Symp. on Theory of Computing (STOC)*, pages 682–690, 1993.

[17] G. Konjevod, R. Ravi, and F.S. Salman. On approximating planar metrics by tree metrics. *Inf. Process. Lett.*, 80(4):213–219, 2001.

[18] H. Whitney. On the abstract properties of linear dependence. *Amer. J. Math.*, 57:509–533, 1935.