UNIVERSITÀ DEGLI STUDI DI ROMA TRE
Dipartimento di Informatica e Automazione
Via della Vasca Navale, 79 – 00146 Roma, Italy

Packing and Squeezing Subgraphs into Planar Graphs

FABRIZIO FRATI[†], MARKUS GEYER[‡], AND MICHAEL KAUFMANN[‡]

RT-DIA-114-2007 **Aprile 2007**

[†] Dipartimento di Informatica e Automazione – Università di Roma Tre, Italy.
[‡] Wilhelm-Schickard-Institut für Informatik – Universität Tübingen, Germany.

The research for this paper was done when the first author was in visit at Tübingen university. The first author wishes to thank Michael Kaufmann for his hospitality.

ABSTRACT

We consider the following problem: Given a set S of graphs, each of n vertices, construct an n-vertex planar graph G containing all the graphs of S as subgraphs. We distinguish the variant in which any two graphs of S are required to have disjoint edges in G (known as 'packing') from the variant in which distinct graphs of S can share edges in G (called 'squeezing'). About the packing variant we show that an arbitrary tree and an arbitrary spider tree can always be packed in a planar graph, improving in this way partial results recently given on this problem. Concerning the squeezing variant, we establish which classes of graphs can generally be squeezed in a planar graph, and which classes cannot.

1 Introduction and Motivation

Motivated by several application issues, a number of graph algorithms require to find subgraphs satisfying certain properties in a larger graph. Moreover, some of the most studied and attracting topics in graph theory are strictly related to the problem of determining relationships between a graph and its subgraphs. The *subgraph isomorphism* problem asks for finding a subgraph H in a graph G [15, 7, 4]. The *graph thickness* problem asks for the minimum number of planar subgraphs in which the edges of a graph can be partitioned [12]. A famous conjecture by Erdős and Sós, already proved in some special cases, asserts that any arbitrary graph with n vertices and $\lfloor (k-1)n/2 \rfloor + 1$ edges contains any tree of order k as subgraph [1]. The *arboricity* problem is the one of determining the minimum number of forests in which a graph can be decomposed [3]. Every planar graph (maximal planar graph) can be decomposed in at most three forests (into three edge-disjoint trees [14]) and it has been recently proved [9] that every planar graph can be decomposed in two edge-disjoint outerplanar graphs.

The study of the relationships between a graph and its subgraphs can be also tackled from the opposite side: Given the n-vertex graphs $G_1, \ldots, G_k$, the requirement is to find a graph G satisfying certain properties and containing all the G_i's as subgraphs. This topic is present with different flavors in the computational geometry and graph drawing literature, motivated by visualization aims, like the display of evolving networks and the simultaneous visualization of relationships involving the same entities. In the *simultaneous embedding* problem [2, 8, 5] G is given and the goal is to draw it so that the drawing of each G_i is planar. The *simultaneous embedding without mapping* problem [2] is to find a graph G such that: (i) G contains all the G_i's as subgraphs, and (ii) G can be drawn with straight-line edges so that the drawing of each G_i is planar. The *packing* problem is the one of finding a graph G containing $G_1, \ldots, G_k$ as edge-disjoint subgraphs. Hedetniemi [10] showed that any two trees with diameter greater than 3 can be packed in a subgraph of K_n and Mateo et al. [11] gave a characterization of which triples of trees can be packed in K_n.

The *planar packing* problem is the variant of the packing problem in which G is required to be planar. García et al. in [6] conjectured that there exists a planar packing of any two non-star trees, that is of any two trees with diameter greater than 3. Notice that the hypothesis that each tree is different from a star is necessary, since any mapping between the vertices of a star and the vertices of an arbitrary tree leads to at least one common edge. García et al. proved the conjecture (1) if the trees are isomorphic or (2) if one of the trees is a path. Recently it has been proved in [13] that (3) there exists a planar packing of any two trees if one of them is a caterpillar. In [13] it is also proved the conjecture (4) if one of the trees is a spider with diameter at most 5. A *caterpillar* is a tree which becomes a path when all its leaves are deleted (see Fig. 1.a) and a *spider* is a tree with at most one vertex of degree greater than 2 (see Fig. 1.b). Such kinds of trees frequently arise in theory and in practice. For instance, the family of graphs that admits a 2-level planar drawing is a collection of caterpillars.

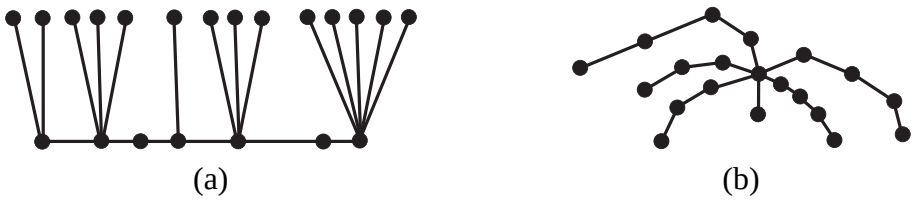


Figure 1: (a) A caterpillar. (b) A spider tree.

In this paper we contribute to the state of the art on the planar packing problem, by extending some of the results in [6] and [13]. Namely, in Section 3 we show that there exists a planar packing of any two trees of diameter greater than 3 if one of them is a spider tree. Notice that this result implies results (1) and (4) cited above. The study of the possibility of obtaining a planar packing of a spider tree and an arbitrary tree is motivated by the observation that a spider tree is a subdivision of a star, and hence spider trees are natural candidates for finding counter-examples of the above cited conjecture. In Section 4 we consider the relaxed version of the planar packing problem in which the subgraphs are not required to be edge-disjoint in the graph containing them. We call such a problem the *planar squeezing* problem and we formally define it as follows: Given the n-vertex graphs $G_1, \ldots, G_k$, find an n-vertex planar graph G containing all the G_i's as subgraphs. We consider the classes of graphs most commonly investigated in the computational geometry and planar graph drawing literature, and we fully determine which ones of them can be generally squeezed in a planar graph. Namely, we show that: (i) there exist a planar graph and a path (a planar graph and a star) that cannot be squeezed in a planar graph; (ii) every two outerplanar graphs (every two trees) can be squeezed in a planar graph; (iii) there exist three caterpillars (three trees) that cannot be squeezed in a planar graph; (iv) there exist two trees that cannot be squeezed in an outerplanar graph; and (v) any number of paths, stars and cycles can be squeezed in an outerplanar graph. Finally, in Section 5 we conclude and suggest some open problems.

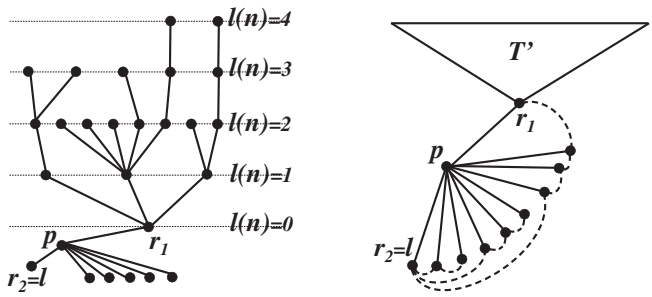
2 Definitions

A *drawing* of a graph is a mapping of each vertex to a distinct point in the plane and of each edge to a Jordan curve between the endpoints of the edge. A *planar drawing* is such that no two edges intersect but possibly at common endpoints. An *embedding* of a planar graph G is a circular ordering of the edges incident on each vertex of G. A *planar graph* is a graph that admits a planar drawing. An *outerplanar graph* is a planar graph that admits a planar drawing with all its vertices on the same face. An *outerplanar embedding* is such that all the vertices lie on the same face. The diameter of a tree is the length of the longest path in the tree. A *star* is a tree with diameter 3, that is a tree where every vertex, but for one, is a leaf, which is a vertex of degree one. A *caterpillar* is a tree such that the graph obtained by deleting its leaves is a path. A *spider* is a tree with at most one vertex, called *root*, of degree greater than 2. The paths starting at the root are called *legs* of the spider. Observe that by definition a star is also a spider and a caterpillar, a path is also a spider and a caterpillar, a caterpillar is also a tree, and a tree is also an outerplanar graph.

Given the n-vertex planar graphs $G_1, \ldots, G_k$, a *planar packing* of $G_1, \ldots, G_k$ is an n-vertex planar graph containing all the G_i's as edge-disjoint subgraphs (see also [6]). Given the n-vertex planar graphs $G_1, \ldots, G_k$, a *planar squeezing* of $G_1, \ldots, G_k$ is an n-vertex planar graph containing all the G_i's as subgraphs. In the following, unless otherwise specified, *packing* and *squeezing* will always stand for planar packing and planar squeezing, respectively.

3 Packing Trees in Planar Graphs

In this section we give an algorithm to pack any n-vertex non-star spider tree S and any n-vertex non-star tree T in a planar graph. Observe that we can suppose w.l.o.g. that the diameter of T is greater or equal than 5. In fact, since T is not a star its diameter is greater than 3. Moreover, if



(a) (b)

Figure 2: (a) A tree T embedded as described in the Preprocessing step. (b) Mapping the vertices of S to the neighbors of p. The dashed edges belong to S.

the diameter of T is 4 then T is a caterpillar, and the results of [13] imply that there is a planar packing of T and S.

The algorithm we present consists of a *Preprocessing step* and of an *Embedding step* that we sketch here and detail in the following. In the Preprocessing step we root the trees and we fix their embeddings. We also assign a level to each vertex of T. In the Embedding step we embed S on T to obtain a packing of the two trees. After having mapped the root of S to a vertex of T, the legs of S are embedded one at a time sorted by increasing length. For each leg its vertices are embedded one at a time in the order they appear on the leg starting from the nearest to the root and ending with the leaf of the leg. Let v_{cur} denote the vertex of S that has to be embedded. We call *active vertex* a the vertex of S that comes before v_{cur} in the leg of v_{cur}. By the order in which the vertices of S are embedded, a has been already mapped to a vertex of T when v_{cur} is embedded. At every step v_{cur} is mapped to an *'unchosen vertex'*, that is a vertex of T to which no vertex of S has been yet mapped. We also call *'chosen vertex'* a vertex of T to which a vertex of S has been already mapped. At every step of the algorithm T and the already embedded edges of S form an embedded graph $\mathcal{E}$. We call *active border* $\mathcal{F}$ the border of the outer face of $\mathcal{E}$. The same vertex of T can have several occurrences in $\mathcal{F}$, since $\mathcal{E}$ is generally a single-connected graph. We denote by $\mathcal{F}(\rightarrow, b, c)$ (by $\mathcal{F}(\leftarrow, b, c)$) the sequence of vertices occurrences that are encountered walking clockwise (resp. counter-clockwise) on $\mathcal{F}$ from an occurrence of a vertex b to an occurrence of a vertex c. When v_{cur} is embedded, edge (a, v_{cur}) is drawn inside the outer face of $\mathcal{E}$.

Preprocessing step. Pick a leaf l of T such that all the neighbors of the unique neighbor p of l are leaves, but for exactly one vertex r_1 (see Fig. 2.a). Note that such l always exists since T is different from a star. Let T' denote the tree obtained from T by deleting p and its adjacent leaves. We choose r_1 to be the root of T and the root of T' as well. The root r_2 of S is chosen as usually (see Section 2). Assign a level $l(n)$ to each vertex of T' so that the root is assigned level 0, all its children are assigned level 1, and so on. Embed T' so that for each vertex n the children of n are in clockwise order $n_1, n_2, \ldots, n_k$ such that $n_i < n_j$ implies that the subtree rooted at n_j contains a vertex v with $l(v) \geq l(u)$, for every vertex u in the subtree rooted at n_i. In the following we will suppose that the children of each node of T' are ordered in clockwise direction. Augment the embedding of T' into an embedding of T by inserting p before the first child of r_1 in T', and by ordering the neighbors of p in clockwise direction so that r_2 is the first vertex and r_1 is the second one (see Fig. 2.a). Map r_2 to l. Let r_2 be the first active vertex a.

Embedding step. This step is repeated until all the vertices and edges of S are embedded on the embedding of T constructed in the Preprocessing step. The legs of S are embedded one

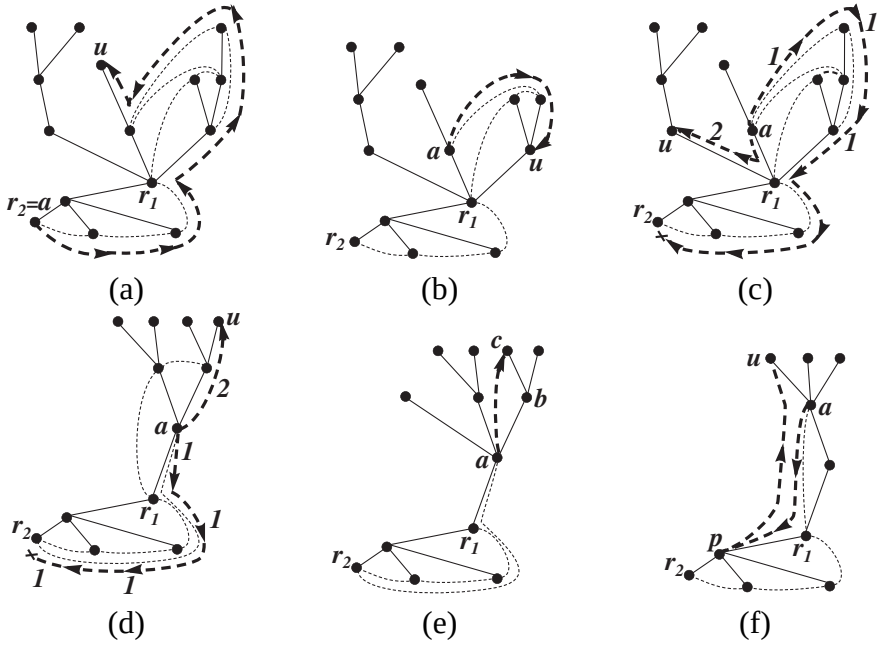


Figure 3: Illustrations for the different cases of the Embedding step. The dashed edges with arrows represent the searches for unchosen vertices that are done in $\mathcal{F}$ and the drawings of the edges (a, u), where u is the unchosen vertex for which it is set $u = v_{cur}$. (a) Case 1. (b) Case 2 (i). (c) Case 2 (ii): The search labelled by 1 corresponds to the clockwise search for unchosen vertices in $\mathcal{F}$, that does not succeed. The search labelled by 2 corresponds to the counter-clockwise search for unchosen vertices in $\mathcal{F}$, not considering the vertices in $T(a)$. Edge (a, u) will be drawn as the dashed edge labelled by 2. (d) Case 3.1 (ii): The search labelled by 1 corresponds to the clockwise search for unchosen vertices in $\mathcal{F}$, that does not succeed. The search labelled by 2 corresponds to the counter-clockwise search for unchosen vertices in $\mathcal{F}$, considering also the vertices in $T(a)$. Edge (a, u) will be drawn as the dashed edge labelled by 2. (e) Case 3.2, where the dashed edge represents the drawing of (a, c). (f) Case 3.3, where the dashed edges represent the drawing of (a, p) and the drawing of (p, u). Notice that the second edge is drawn only if p is the active vertex after setting $v_{cur} = p$.

at a time sorted by increasing length. For each leg its vertices are embedded one at a time in the order they appear on the leg starting from the nearest to r_2 and ending with the leaf of the leg. Let $p(v)$ denote the parent of a vertex v in T' and $T(v)$ denote the subtree of T' rooted at v. While p has unchosen neighbors, the algorithm will map v_{cur} to the first unchosen vertex in the counter-clockwise order of the neighbors of p starting at r_2. Hence, when v_{cur} is set equal to r_1, all the neighbors of p will be chosen vertices (see Fig. 2.b). Every time v_{cur} has to be embedded, do the following: (i) map v_{cur} to an unchosen vertex u of T; (ii) draw the edge between a and v_{cur} into the outer face of $\mathcal{E}$, and (iii) choose a new vertex of T to be the new active vertex. The choice of the new active vertex a is always done in the following way: If the next v_{cur} is on the same leg of the just embedded v_{cur}, then $a = u$, otherwise $a = r_2$. The choice of the vertex u to which v_{cur} is mapped and the drawing of edge (a, v_{cur}) vary according to several cases:

Case 1: (refer to Figure 3.a) If a coincides with r_2 or with any other neighbor of p not in T', then walk counter-clockwise on $\mathcal{F}$, starting from the only occurrence of r_2, until an unchosen vertex u is found. Map v_{cur} to u. Draw edge (a, v_{cur}) following the counter-clockwise walk done on $\mathcal{F}$.

Case 2: (refer to Figures 3.b and 3.c) If a does not coincide with r_2 and there is at least one unchosen vertex in the tree $T' \setminus T(a)$ that does not belong to the path from r_1 to a in T', then walk on $\mathcal{F}(\rightarrow, a, r_2)$ not considering the vertices in $T(a)$.

(i) If an unchosen vertex $u \neq p(a)$ has been encountered then map v_{cur} to u and draw the edge (a, v_{cur}) following the clockwise walk done on $\mathcal{F}$.

(ii) If the last occurence of $p(a)$ in $\mathcal{F}(\rightarrow, a, r_2)$ has been encountered and $p(a)$ is not yet chosen or if no unchosen vertex has been found in $\mathcal{F}(\rightarrow, a, r_2)$, then reverse the search direction and map v_{cur} to the first unchosen vertex u in $\mathcal{F}(\leftarrow, a, p)$, not considering the vertices in $T(a)$. Draw the edge (a, u) following the counter-clockwise walk done on $\mathcal{F}$.

Case 3: If a does not coincide with r_2, and if there are no unchosen vertices in $T' \setminus T(a)$, but eventually for those in the path from r_1 to a in T', we distinguish three subcases:

Case 3.1: (refer to Figure 3.d) If no unchosen child of a exists and if $T(a)$ contains unchosen vertices of level $l(a) + 2$ or higher, then search in $\mathcal{F}(\rightarrow, a, r_2)$ not considering the vertices in $T(a)$.

(i) If an unchosen vertex $u \neq p(a)$ has been reached then map v_{cur} to u and draw edge (a, v_{cur}) following the clockwise walk done on $\mathcal{F}$.

(ii) If the last occurence of $p(a)$ in $\mathcal{F}(\rightarrow, a, r_2)$ has been reached and $p(a)$ is not yet chosen or if no unchosen vertex has been found in $\mathcal{F}(\rightarrow, a, r_2)$, then reverse the search direction and map v_{cur} to the first unchosen vertex u in $\mathcal{F}(\leftarrow, a, p)$ starting from the first occurrence of a in $\mathcal{F}(\leftarrow, r_2, p)$. In this case the vertices of $T(a)$ are considered first. Draw edge (a, v_{cur}) following the counter-clockwise walk done on $\mathcal{F}$.

Case 3.2: (refer to Figure 3.e) If there are unchosen children of a and if $T(a)$ contains unchosen vertices of level $l(a) + 2$ or higher, then consider the last child b of a. Select the first child c of b. We will prove later that c is an unchosen vertex. Map v_{cur} to c and draw edge (a, v_{cur}) passing just before edge (a, b) in the clockwise order of the children of a.

Case 3.3: (refer to Figure 3.f) If $T(a)$ does not contain unchosen vertices of level $l(a) + 2$ or higher, then we are in the final phase of our algorithm. Notice that the only unchosen vertices in T', but for p, are either at distance one from a or lie on the path from a to r_1. We will prove later that all the unchosen vertices on such a path are pairwise non-adjacent. Map v_{cur} to p draw edge (a, v_{cur}) by walking counter-clockwise on $\mathcal{F}$ starting from the first occurrence of a in $\mathcal{F}(\rightarrow, p, r_2)$. After that, if $a = p$ then search in $\mathcal{F}(\rightarrow, a, r_2)$ until an unchosen vertex u is found. Map v_{cur} to u and draw edge (a, v_{cur}) following the clockwise walk done on $\mathcal{F}$. At this point, or if it was $a = r_2$, only Cases 1, 2, and 3.1 will be applied, until all the remaining vertices of S are mapped to unchosen vertices of T. Notice that Case 3.3 is applied exactly once in one application of the algorithm.

In the following we give some lemmas that will be helpful to prove that the described algorithm constructs a planar packing of S and T. The proofs of such lemmas are in the Appendix, for reasons of space.

Lemma 1 *Let v and $p(v)$ be unchosen vertices in T'. Then all vertices in $T(v)$ are unchosen.*

Proof. Assume the contrary. Consider the first step s^* of the algorithm in which a vertex $w \neq v, p(v)$ of S was mapped to a vertex in $T(v) \cup p(v)$. Clearly, before s^* all the vertices of $T(v)$ were unchosen. If one of the Cases 1, 2, 3.1, and 3.3 was applied in s^*, then $p(v)$ would have been chosen instead of w, since $p(v)$ would have been encountered before w in any clockwise or counter-clockwise visit of $\mathcal{F}$ starting at a chosen vertex of T (see Fig. 4.a). Otherwise, suppose Case 3.2 was applied in s^*. If at the beginning of s^* the active vertex a was $p(p(p(v)))$ or was $p(p(v))$, then $p(v)$ or v would have been chosen instead of w, respectively. If at the beginning of s^* the active vertex a was in $T(v) \cup p(v)$, then s^* would not have been the first step in which a vertex of $T(v) \cup p(v)$ was chosen. Finally, if at the beginning of s^* the active vertex a was not $p(p(p(v)))$, was not $p(p(v))$ and was not a vertex in $T(v) \cup p(v)$, then the vertex chosen in s^* would not have been in $T(v) \cup p(v)$, contradicting the assumption that in s^* a vertex in $T(v) \cup p(v)$ is chosen. Hence, such a w cannot exist. □

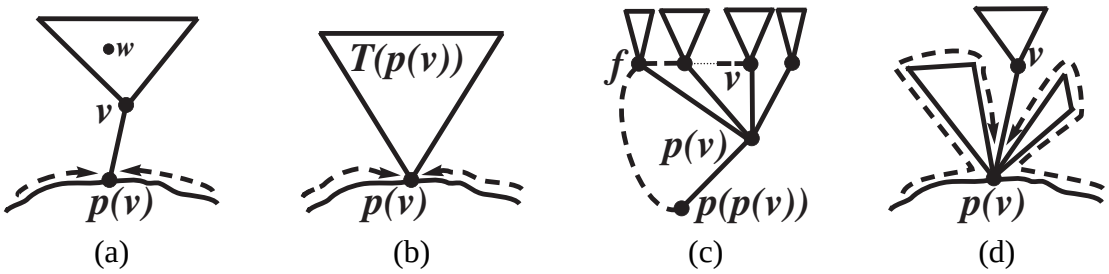


Figure 4: (a) Illustration for the proof of Lemma 1: If one of the Cases 1, 2, 3.1, and 3.3 was applied at step s^* of the Embedding step, then $p(v)$ would have been chosen instead of w. (b) Illustration for the proof of Lemma 2: If one of the Cases 1, 2, 3.1, and 3.3 was applied at step s^* of the Embedding step, then $p(v)$ would have been chosen instead of w. (c) Illustration for the proof of Lemma 2: an application of Case 3.2 followed by $j-1$ applications of Case 2 (i) of the Embedding step. (d) Illustration for the proof of Lemma 2: if Case 1 was applied at least once in the $j-1$ steps after s^*, then $p(v)$ would have been chosen before v.

Corollary 1 *Let $v \in T'$ be a chosen vertex and let $\mathcal{P} = (r_1 = v_1, v_2, \ldots, v_{l-1}, v_l = v)$ be the path connecting r_1 and v in T', with $l \geq 2$. There exist no two consecutive unchosen vertices v_i and v_{i+1} in $\mathcal{P}$.*

Lemma 2 *If v is the j-th child of $p(v)$ in T', if $p(v)$ is unchosen, and if v_{cur} has been mapped to v in the current step of the algorithm, then during the last j steps of the application of the algorithm Case 3.2 was applied once to draw an edge from $p(p(v))$ to the first child f of $p(v)$ and Case 2 (i) was applied in the following $j-1$ steps to draw $j-1$ edges connecting the first j children of $p(v)$.*

Proof. Consider the first step s^* of the algorithm in which a vertex w of $T(p(v))$ is chosen. If one of the Cases 1, 2, 3.1, and 3.3 was applied in s^*, then $p(v)$ would have been chosen instead of w, since $p(v)$ would have been encountered before w in any clockwise or counter-clockwise visit of $\mathcal{F}$ starting at a chosen vertex of T (see Fig. 4.b). Hence, Case 3.2 was applied. Since before s^* all the vertices in $T(p(v))$ were unchosen and since $p(v)$ is unchosen even after s^*, then Case 3.2 was applied to draw the edge $(p(p(v)), f)$. Further, in the $j-1$ steps after s^* Case 2 (i) was applied (see Fig. 4.c). In fact, denoting by u the current active vertex, if u is a child of $p(v)$ there are unchosen vertices in $T' \setminus T(u)$ (v is one of such unchosen vertices) and searching

in clockwise direction from u not considering $T(u)$ the next unchosen vertex encountered will be the child x of $p(v)$ after u. Notice that there will be other occurrences of $p(v)$ in $\mathcal{F}(\rightarrow, x, p)$. Finally, in the $j-2$ steps after s^* the active vertex is always the next chosen vertex x child of $p(v)$, otherwise Case 1 would have been applied and $p(v)$ would have been chosen before v and before every other vertex x' in $T(v)$, since $p(v)$ would have been encountered before v and x' in any clockwise or counter-clockwise visit of $\mathcal{F}$ starting at a chosen vertex of T (see Fig. 4.d). □

Corollary 2 *Let a be an active vertex in T' and let $p(a)$ be unchosen. There exists no occurrence of $p(a)$ in $\mathcal{F}(\leftarrow, a, p)$.*

Lemma 3 *Let $v \in T'$ be an occurrence of a vertex in $\mathcal{F}$. In $\mathcal{F}(\rightarrow, v, r_2)$ there exists at least one occurrence of every unchosen vertex belonging to the path connecting r_1 and v in T'. Moreover, all the unchosen vertices of T appear at least once in $\mathcal{F}$.*

Proof. The statement is trivially true when vertices of S have been mapped to all neighbors of p and $v_{cur} = r_1$. Suppose that the statement is true before step s^* of the application of the algorithm. We show that the statement holds true whichever case of the Embedding step is applied in s^*. Notice that if no occurrence of any unchosen vertex is deleted from $\mathcal{F}$ after s^*, but for the ones of the vertex chosen in s^*, then the statement holds after s^*. Conversely, if some occurrences of an unchosen vertex u are deleted from $\mathcal{F}$ after s^*, we have to show that after s^* there are still occurrences of u on $\mathcal{F}$ and that such occurrences come in $\mathcal{F}(\rightarrow, w, r_2)$ after every occurrence of each vertex $w \in T(u)$.

If Case 1 is applied in s^*, it is set $v_{cur} = u$ for the first unchosen vertex u found in $\mathcal{F}(\leftarrow, r_2, p)$. Hence, all the occurrences of unchosen vertices are still on $\mathcal{F}$ and the statement holds after s^*.

Denoting by a the active vertex in s^*, if one of the Cases 2 (i) and 3.1 (i) is applied in s^* and if $p(a)$ is already chosen, then the first unchosen vertex u found in $\mathcal{F}(\rightarrow, a, r_2)$ is chosen and the statement holds after s^*. If one of the Cases 2 (i) and 3.1 (i) is applied in s^* and if $p(a)$ is not yet chosen, then by the hypothesis of Cases 2 (i) and 3.1 (i) the last occurrence $o(p(a))$ of $p(a)$ in $\mathcal{F}(\rightarrow, p, r_2)$ is not deleted after s^*. Before s^* all the occurrences of the vertices in $T(p(a))$ came before $o(p(a))$ in $\mathcal{F}(\rightarrow, p, r_2)$ and hence, after s^*, $o(p(a))$ still come in $\mathcal{F}(\rightarrow, p, r_2)$ after all the occurrences of the vertices in $T(p(a))$.

If one of the Cases 2 (ii) and 3.1 (ii) is applied in s^* then the first unchosen vertex u found in $\mathcal{F}(\leftarrow, a, p)$ is chosen and the statement holds after s^*. Notice that, by Corollary 2 $u \neq p(a)$.

If Case 3.2 is applied in s^*, consider the last child b of a. If b was chosen in a step of the algorithm in which a was still unchosen, then, by Lemma 2, all the children of a are chosen and then Case 3.2 would not be applied in s^*. Hence b is unchosen. Consider the step s^*_{-1} before s^* in which it is set $v_{cur} = a$. Before s^*_{-1} both a and b are unchosen. By Lemma 1 it follows that all the vertices in $T(b)$ are unchosen. Turning again to step s^*, edge (a, c) deletes the first occurrence of b in $\mathcal{F}(\rightarrow, p, r_2)$. However, since all the vertices in $T(b)$, but for c, are still unchosen after s^* there is still an occurrence of b coming after every vertex of $T(b)$ in $\mathcal{F}(\rightarrow, p, r_2)$ and the statement holds.

Finally, suppose Case 3.3 is applied in s^*. If all the children of a are unchosen, by Lemma 1 all the vertices in $T(a)$ are unchosen, since before setting $v_{cur} = a$, a and all its children were unchosen. In this case, the first occurrence of a in $\mathcal{F}(\rightarrow, p, r_2)$ comes before every vertex in $T(a)$. Moreover, since there are no unchosen vertices in $T \setminus T(a)$ but for p and for those in the path from r_1 to a in T' and since there are occurrences of all the vertices in the path from r_1 to

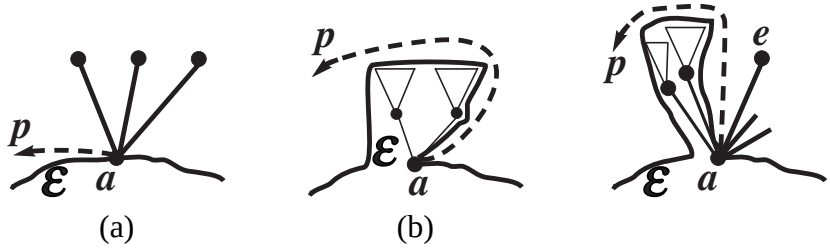


Figure 5: Illustrations for the proof of Lemma 3, supposing that Case 3.3 was applied in s^*. (a) If all the children of a are unchosen, edge (a, p) passes before the edge between a and the first child of a. (b) If all the children of a are chosen, edge (a, p) can deletes the vertices of $T(a)$, but for a, from $\mathcal{F}$. (c) If some children of a are unchosen and some are chosen, then edge (a, p) passes just before the edge between a and the first unchosen child e of a.

a in T' that come after a in $\mathcal{F}(\rightarrow, p, r_2)$, then after drawing edge (a, p) the statement still holds (see Fig. 5.a). If all the children of a are chosen, by the hypothesis of Case 3.3 all the vertices in $T(a)$ are chosen. Moreover, since there are no unchosen vertices in $T \setminus T(a)$ but for p and for those in the path from r_1 to a in T' and since there are occurrences of all the vertices in the path from r_1 to a in T' that come after a in $\mathcal{F}(\rightarrow, p, r_2)$, then after drawing edge (a, p) the statement still holds (see Fig. 5.b). Now suppose that some children of a are already chosen before s^* and some not (see Fig. 5.c). Consider the first unchosen child e of a for which all the children of a before e are already chosen before s^*. Notice that $T(e)$ contains only vertex e. For converse, suppose that $T(e)$ contains other vertices. By the hypothesis of Case 3.3 all the vertices in $T(e)$, but for e, must be chosen. Consider the first vertex $w \neq e$ that was chosen in $T(e)$. w could not be chosen by applying Case 1, Case 2, or Case 3.1, since e would have been encountered before w in any clockwise or counter-clockwise visit of $\mathcal{F}$. Moreover w could not be chosen by applying Case 3.2, since the parent of its parent either is a (that is active vertex in Case 3.3 and not in Case 3.2) either is an other vertex of $T(e)$, contradicting the hypothesis that w is the first chosen vertex of $T(e)$. Finally w could not be chosen by applying Case 3.3 that must be applied in s^* and cannot be applied twice. Notice that all the vertices in the subtrees $T(f)$, for every child f of a that comes before e in the order of the children of a, are already chosen by the hypotheses of Case 3.3. Hence, since edge (a, p) passes just before edge (a, e) in the order of the edges around a, no occurrences of vertices in $T(a)$ are deleted from $\mathcal{F}$. Moreover, since there are no unchosen vertices in $T \setminus T(a)$ but for p and for those in the path from r_1 to a in T' and since there are occurrences of all the vertices in the path from r_1 to a in T' that come after a in $\mathcal{F}(\rightarrow, p, r_2)$, then after drawing edge (a, p) the statement still holds. □

Theorem 1 *There exists an algorithm that in polynomial time constructs a planar packing of any n-vertex non-star spider tree S and any n-vertex non-star tree T.*

Proof. Apply the algorithm described in this section to T and S. First, notice that the algorithm can be easily implemented to run in polynomial time. We claim that the constructed embedding $\mathcal{E}$ is a planar packing of T and S. More precisely, we will prove that: (1) $\mathcal{E}$ is planar, (2) every two vertices of S are mapped to distinct vertices of T, (3) there are no common edges between S and T, and finally (4) all the vertices of S are mapped to vertices of T.

(1): The planarity of $\mathcal{E}$ follows from the fact that at every step all the unchosen vertices are incident to the outer face (by Lemma 3) and that by construction every inserted edge is placed inside the outer face of $\mathcal{E}$.

(2): When one of the Cases 1, 2, 3.1, and 3.3 of the Embedding step is applied, by the description of the algorithm v_{cur} is mapped to an unchosen vertex of T. Hence, we have only to show that when Case 3.2 has to be applied in a step s^* of the algorithm the first child c of the last child b of a is unchosen. If before s^* vertex b is chosen, then by Lemma 2 all the children of a were already chosen when it was set $v_{cur} = b$, hence Case 3.2 would not be applied in s^*. Otherwise, prior to the choice of a just before step $s*$, both b and a were unchosen and so, by Lemma 1, all the vertices in $T(b)$, including c, are unchosen at the beginning of s^*.

(3): Consider the different cases of the Embedding step. In Case 1 a common edge is inserted only if it is set $v_{cur} = p$. If $a \neq r_2$ then setting $v_{cur} = p$ would imply that T is a star, contradicting the hypoteses. Notice that vertices of S are mapped to all the neighbors of p by applications of Case 1 before any other case of the Embedding step is applied. If $a = r_2$ then, since p is the last vertex in $\mathcal{F}(\leftarrow, r_2, p)$, setting $v_{cur} = p$ implies that no other vertex of $\mathcal{F}$ is unchosen and, by Lemma 3, that no other vertex of T is unchosen. Hence, the current leg of S is the last one. Since this leg should have length 1 and since the legs of S are ordered by increasing length, S would be a star, contradicting the hypoteses. In Case 2, the only neighbor of a in T' that belongs to $T' \setminus T(a)$ is $p(a)$. However, in Case 2 (i) it is clear that $v_{cur} = u$ is chosen for a vertex $u \neq p(a)$. In Case 2 (ii) the algorithm chooses for v_{cur} the first unchosen vertex u in $\mathcal{F}(\leftarrow, a, p)$. By Corollary 2 there exists no occurrence of $p(a)$ in such a visit and so $u \neq p(a)$. In Case 3.1 (i) the same considerations done for Case 2 (i) hold. In Case 3.1 (ii) for v_{cur} is a vertex u chosen, that belongs to $T(a)$. Since all the children of a are already chosen, then no common edge is inserted. In Case 3.2 a and c are not neighbors in T. Finally, consider Case 3.3. Since a belongs to T', then a and p are neighbors only if $a = r_1$. However, if $a = r_1$ and there are no unchosen vertices in T', but for the children of a, then the diameter of T would be at most 4, contradicting the hypoteses. Concerning edge (p, u) all the neighbors of p are already chosen before applying Case 3.3, so u cannot be a neighbor of p.

(4) We have to prove that while there are unchosen vertices in T the algorithm applies one of the cases in the Embedding step to map v_{cur} to a vertex of T. All the neighbors of p are chosen at the beginning of the Embedding step by applications of Case 1. After that phase only p and the vertices in $T' \setminus r_1$ are still unchosen. Now let a be the current active vertex. Suppose Case 1 has to be applied. By Lemma 3 at every step of the algorithm all the unchosen vertices are on $\mathcal{F}$, so Case 1 finds an unchosen vertex u to set $v_{cur} = u$. Suppose Case 2 has to be applied. If there are occurrences of unchosen vertices in $\mathcal{F}(\leftarrow, a, r_2)$ not belonging to $T(a)$ or to the path connecting r_1 and a in T', then even if Case 2 (i) fails, then Case 2 (ii) would find such occurrences. Otherwise, suppose that the only unchosen vertices not belonging to $T(a)$ or to the path connecting a and r_1 in T' appear before a in $\mathcal{F}(\leftarrow, r_2, p)$. If $p(a)$ is already chosen, then Case 2 (i) would always succeed. If $p(a)$ is unchosen and if a is the j-th child of $p(a)$, then $T(p(p(a)))$ contains the only unchosen vertices remaining, but for p and for the vertices in the path from r_1 to $p(p(a))$ in T', since Case 3.2 was applied j steps before the current one when $p(p(a))$ was the active vertex (by Lemma 2). Since $p(a)$ is the last child of $p(p(a))$, then the only vertices that can have occurrences before a in $\mathcal{F}(\leftarrow, r_2, p)$ are the vertices in $T(p(a))$. Such occurrences are clearly encountered before the last occurrence of $p(a)$ in $\mathcal{F}(\rightarrow, a, p)$, hence Case 2 (i) finds them and succeeds. Suppose Case 3.1 has to be applied. Then either Case 3.1 (i) succeeds, or Case 3.1 (ii) finds an unchosen vertex in $T(a)$. Such vertex exists by the hypoteses of Case 3.1. Suppose Case 3.2 has to be applied. We have already shown in part (2) of the proof, that if vertex c exists, then it is unchosen. Now we only have to prove the existence of such a vertex. By the construction of the embedding of T' the children of a are clockwise ordered by increasing depth of the subtrees rooted at them; observing that in $T(a)$

there are vertices of level $l(a) + 2$ or higher, then vertex c exists. Finally if Case 3.3 has to be applied, then no problem arises, since p is unchosen and it is on $\mathcal{F}$ before the only application of Case 3.3. Notice that by Corollary 1 and Lemma 3 after Case 3.3 is applied all the remaining unchosen vertices of T are disconnected and are on $\mathcal{F}$. Therefore, Cases 1, 2, and 3.1 can be applied until all the vertices of S are mapped to vertices of T. □

4 Squeezing Planar Graphs in Planar Graphs

When dealing with the planar packing problem, it can be easily observed that two sufficiently dense planar graphs cannot be packed in the same planar graph. For instance, two maximal outerplanar graphs have $2n - 3$ edges each, and a packing of them contains $4n - 6$ edges, that are more than the ones that a planar graph can have.

If you want to obtain planar squeezings of planar graphs, edges of different graphs can overlap, and so edge-counting arguments do not work. However, the following two results are just slightly more than trivial:

Theorem 2 *There exist a planar graph G and a path P that cannot be squeezed in a planar graph.*

Proof. Let G be an n-vertex triangulated planar graph that does not contain any Hamiltonian path, and let P be an n-vertex path. Observe that since G is maximal no edge can be added to it without violating its planarity. However, when squeezing G and P, at least one edge of P is not common to an edge of G, otherwise G would contain an Hamiltonian path. □

Theorem 3 *There exist a planar graph G and a star S that cannot be squeezed in a planar graph.*

Proof. Let G be an n-vertex triangulated planar graph that does not contain a vertex of degree greater than $n - 2$, and let S be an n-vertex star. Since G is maximal no edge can be added to it without violating its planarity. However, when squeezing G and S, at least one edge of S is not common to an edge of G, otherwise G would contain a vertex of degree $n - 1$. □

Turning the attention from planar to outerplanar graphs, we have:

Theorem 4 *Any two outerplanar graphs can be squeezed in a planar graph.*

Proof. Let O_1 and O_2 be two outerplanar graphs. Assume w.l.o.g. that both O_1 and O_2 are biconnected. Hence O_1 and O_2 contain Hamiltonian cycles, say C_1 and C_2, respectively. Now map the vertices of O_1 and O_2 so that C_1 and C_2 are coincident. Furthermore, embed the edges of O_1 that do not belong to C_1 inside C_1, and embed the edges of O_2 that do not belong to C_2 and that are not common to edges of O_1 outside C_1. By the outerplanarity of O_1 (of O_2) there are no intersections between edges inside C_1 (resp. outside C_1). Further, there are no intersections between edges inside C_1 and edges outside C_1, since they are separated by C_1. □

Since trees are outerplanar graphs, the following holds:

Corollary 3 *Any two trees can be squeezed in a planar graph.*

Corollary 3 shows that the problem of determining whether for any two trees there exists a planar graph containing them as subgraphs, that has been tackled in [6], in [13] and in Section 3, is easily solvable if common edges are allowed.

However, if one augments to more than two the number of trees that must be squeezed, then a planar squeezing is not generally possible. Namely, in the following we provide three caterpillars that cannot be squeezed in the same planar graph.

Theorem 5 *There exist three caterpillars that cannot be squeezed in the same planar graph.*

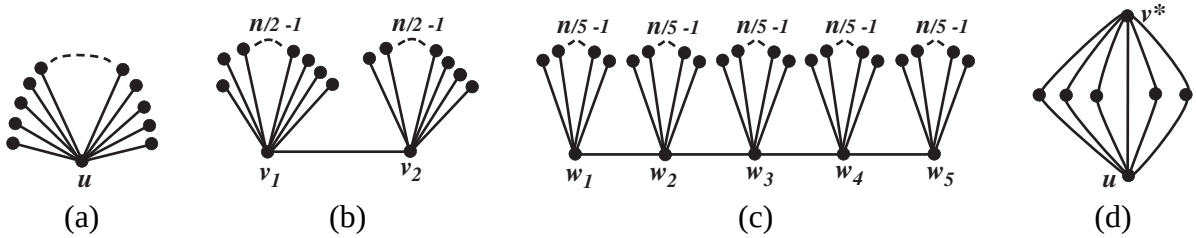


Figure 6: (a) Star C_1. (b) Caterpillar C_2. (c) Caterpillar C_3. (d) Embedding $\mathcal{E}_A$.

Proof. Let C_1 be a star with center u and $n-1$ leaves (see Fig. 6.a), let C_2 be a caterpillar with two adjacent vertices v_1 and v_2 of degree $n/2$ and $n-2$ leaves (see Fig. 6.b), and let C_3 be a caterpillar with five vertices $w_1, \ldots, w_5$ of degree at most $n/5+1$ forming a path and with $n-5$ leaves. Each vertex w_i has $n/5-1$ adjacent leaves (see Fig. 6.c).

We will try to construct a planar embedding that contains embeddings of C_1, C_2, and C_3 and we will show that this goal is not achievable. Observe that C_1 has a unique embedding $\mathcal{E}_1$ up to a relabelling of its leaves. First, construct a planar embedding $\mathcal{E}_2$ by embedding C_2 on $\mathcal{E}_1$ in any way. Let G_2 denote the planar graph obtained by such a squeezing. Notice that there exists one out of v_1 and v_2, say v^*, that has not been mapped to u and that shares with u exactly $n/2-1$ common neighbors. In fact, if vertex v_1 (vertex v_2) has been mapped to u, then v_2 (resp. v_1) has been mapped to a leaf of C_1 and all the $n/2-1$ leaves adjacent to v_2 (resp. to v_1) have been mapped to leaves of C_1, that are neighbors of u. Otherwise, if both vertices v_1 and v_2 have been mapped to leaves of C_1, then v_1 (or v_2) has exactly $n/2-2$ adjacent leaves that have been mapped to leaves of C_1, that are neighbors of u, and v_2 (resp. v_1) is a neighbor of both u and v_1 (resp. of both u and v_2). Consider the set A of vertices that are neighbors of both u and v^*. Vertex u, vertex v^*, and the vertices in A induce an embedded subgraph $\mathcal{E}_A$ of $\mathcal{E}_2$ that is done by at least one and at most two nested triangles sequences, all sharing edge (u, v^*) (see Fig. 6.d).

Now consider any embedding $\mathcal{E}_3$ of C_3 on $\mathcal{E}_2$. Let us discuss how many vertices of C_3 can be mapped to vertices in A, while preserving the planarity of $\mathcal{E}_3$. Since the degree of each vertex w_i is at most $n/5+1$, at most $2n/5+2$ vertices of A could be neighbors of u and v^* in C_3. Vertices $w_1, \ldots, w_5$ of C_3 that are not mapped to u and v^* can have at most two vertices of A as adjacent leaves. In fact, if vertex w_i is mapped to a vertex of A, then it is incident to two adjacent faces of $\mathcal{E}_A$ that have at most two vertices distinct from u, from v^*, and from w_i itself. If vertex w_i is mapped to a vertex not in A and inside any face of $\mathcal{E}_A$, then it can be a neighbor of the at most two vertices of that face that are in A. Hence, for every vertex w_i three vertices internal to $\mathcal{E}_A$ can have a mapping, two with leaves adjacent to w_i and one with w_i itself. Hence less than $2n/5+2+3\cdot 5 = 2n/5+17$ vertices of A can have a mapping with a vertex of C_3 while preserving the planarity of $\mathcal{E}_3$. Choosing $|A| = n/2-1 > 2n/5+17$ (i.e. choosing $n > 180$)

implies that the vertices of C_3 cannot be mapped to all the vertices in A while preserving the planarity of $\mathcal{E}_3$ and hence that there is no planar squeezing of C_1, C_2, and C_3 $\square$

Since caterpillars are trees we have the following:

Corollary 4 *There exist three trees that cannot be squeezed in the same planar graph.*

If one wants to squeeze trees in trees, trees in outerplanar graphs, or outerplanar graphs in outerplanar graphs, then very few is allowed. Namely, we show that there exist two caterpillars that cannot be squeezed in the same outerplanar graph. Let C_1 be a star with center u and seven leaves and let C_2 be a caterpillar consisting of two vertices of degree four and six leaves. We claim that C_1 and C_2 cannot be squeezed in the same outerplanar graph. This is proved by showing that any planar embedding of an 8-vertex planar graph that contains embeddings of C_1 and C_2 cannot be an outerplanar embedding. First, observe that C_1 has a unique embedding up to a relabelling of its leaves. So consider it as embedded. Now embed C_2. Since there is just one non-leaf vertex in C_1, at least one of the vertices of C_2 with degree 4 must be mapped to a leaf of C_1. Let v be such a vertex. Again, since there is one non-leaf vertex in C_1, at least three of the neighbors of v must be mapped in leaves of C_1. This implies that in any embedding containing embeddings of C_1 and C_2 there is a cycle formed by v, u and a neighbor of v enclosing a neighbor of v. Hence there exists no outerplanar embedding containing embeddings of C_1 and C_2 and so there exists no outerplanar graph containing C_1 and C_2.

Theorem 6 *There exist two caterpillars that cannot be squeezed in the same outerplanar graph.*

Since caterpillars form a subclass of trees we have the following:

Corollary 5 *There exist two trees that cannot be squeezed in the same outerplanar graph.*

We conclude this section observing that any number of paths, cycles and stars can be squeezed in an outerplanar graph having one vertex of degree $n-1$.

5 Conclusions and Open Problems

We have considered the problem of packing and squeezing subgraphs in planar graphs. Concerning the planar packing problem, the previous works on this topic [6, 13] contain algorithms that construct embeddings of the trees by observing the 'separation principle', i.e. by separating the edges of the two trees in two different portions of the embedding plane, established in advance. This allows to mind only to the presence of common edges for obtaining a planar packing. As far as we know, our algorithm is the first one that does not bind the embeddings of the trees to be separated as described. Also the tree embeddings produced by our algorithm could not be separated in different parts of the plane, since there are vertices with a sequence $[T_1, T_2, T_1, T_2]$ of consecutive edges, where T_1 (T_2) indicates an edge belonging to the first (resp. the second) tree.

Problem 1 *Does a planar packing of any two non-star trees with the further constraint of having an embedding where the two trees can be separated by a simple line that intersects the embedding only at vertices of the graph exist?*

Concerning the squeezing problem, we showed which are the most interesting classes of planar graphs that can generally be squeezed and which cannot. However, the following open problem is worth of interest:

Problem 2 *Which is the time complexity of determining if two planar graphs can be squeezed in a planar graph?*

The last question seems to be strictly related to some of the most important problems in graph theory, like *graph isomorphism* and *subgraph isomorphism*.

References

[1] B. Bollabás and V. Nikiforov. Extremal graph theory. *Handbook of Graph Theory*, pages 788–816, 2004.

[2] P. Braß, E. Cenek, C. A. Duncan, A. Efrat, C. Erten, D. Ismailescu, S. G. Kobourov, A. Lubiw, and J. S. B. Mitchell. On simultaneous planar graph embeddings. In *WADS*, pages 243–255, 2003.

[3] D. Eppstein. Arboricity and bipartite subgraph listing algorithms. *Information Processing Letters*, 51(4):207–211, 1994.

[4] D. Eppstein. Subgraph isomorphism in planar graphs and related problems. *J. Graph Algorithms & Applications*, 3(3):1–27, 1999.

[5] F. Frati. Embedding graphs simultaneously with fixed edges. In *Graph Drawing*, pages 108–113, 2006.

[6] A. García, C. Hernando, F. Hurtado, M. Noy, and J. Tejel. Packing trees into planar graphs. *J. Graph Theory*, pages 172–181, 2002.

[7] M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979.

[8] M. Geyer, M. Kaufmann, and I. Vrtó. Two trees which are self-intersecting when drawn simultaneously. In *Graph Drawing*, pages 201–210, 2005.

[9] Daniel Gonçalves. Edge partition of planar graphs into two outerplanar graphs. In *STOC*, pages 504–512, 2005.

[10] S.M. Hedetniemi, S.T. Hedetniemi, and P.J. Slater. A note on packing two trees into k_n. *Ars Combinatorica*, pages 149–153, 1981.

[11] M. Maheo, J.F. Saclé, and M. Woźniak. Edge-disjoint placement of three trees. *Ars Combinatorica*, pages 543–563, 1996.

[12] P. Mutzel, T. Odenthal, and M. Scharbrodt. The thickness of graphs: A survey. *Graphs and Combinatorics*, 14(1):59–73, 1998.

[13] Y. Oda and K. Ota. Tight planar packings of two trees. In *European Workshop on Computational Geometry*, pages 215–216, 2006.

[14] W. Schnyder. Embedding planar graphs on the grid. In *Proc. 1st ACM-SIAM Sympos. Discr. Alg.*, pages 138–148, 1990.

[15] J. R. Ullmann. An algorithm for subgraph isomorphism. *J. ACM*, 23(1):31–42, 1976.