h-Quasi Planar Drawings of Bounded Treewidth Graphs in Linear Area*

Emilio Di Giacomo, Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani

Dip. di Ingegneria Elettronica e dell'Informazione, Università degli Studi di Perugia
{digiacomo,didimo,liotta,montecchiani}@diei.unipg.it

Abstract. We study the problem of computing h-quasi planar drawings in linear area; in an h-quasi planar drawing the number of mutually crossing edges is at most $h-1$. We prove that every n-vertex partial k-tree admits a straight-line h-quasi planar drawing in $O(n)$ area, where h depends on k but not on n. For specific sub-families of partial k-trees, we present ad-hoc algorithms that compute h-quasi planar drawings in linear area, such that h is significantly reduced with respect to the general result. Finally, we compare the notion of h-quasi planarity with the notion of h-planarity, where each edge is allowed to be crossed at most h times.

1 Introduction

Area requirement of graph layouts is a widely studied topic in Graph Drawing and Geometric Graph Theory. Many asymptotic bounds have been proven for a variety of graph families and drawing styles. One of the most fundamental results in this scenario establishes that every planar graph admits a planar straight-line grid drawing in $O(n^2)$ area and that this bound is worst-case optimal [8]. This has motivated lot of work devoted to discover sub-families of planar graphs that admit planar straight-line drawings in $o(n^2)$ area. Unfortunately, sub-quadratic upper bounds are known only for trees [7] and outerplanar graphs [9], while super-linear lower bounds are known for series-parallel graphs [19]. Bounds for planar poly-line drawings are also known [3,4].

Although planarity is one of the most desirable properties when drawing a graph, many real-world graphs are in fact non-planar. Furthermore, planarity often imposes severe limitations on the optimization of the drawing area, which may sometimes be overcome by allowing either "few" edge crossings or specific types of edge crossings that do not affect too much the drawing readability. So far, only a few papers have focused on computing non-planar layouts in sub-quadratic area. Wood proved that every k-colorable graph admits a non-planar straight-line grid drawing in linear area [22], which implies that planar graphs admit such a drawing. However, the technique by Wood does not provide any guarantee on the type and number of edge crossings. More recently, Angelini *et al.* provided techniques for constructing poly-line *large angle crossing drawings* (*LAC drawings*) of planar graphs in sub-quadratic area [1]. We recall that the study of drawings with large angle crossings started in [13].

* Research supported in part by the MIUR project AlgoDEEP prot. 2008TFBWL4.

M.C. Golumbic et al. (Eds.): WG 2012, LNCS 7551, pp. 91–102, 2012.

In this paper we study the problem of computing linear area straight-line drawings of graphs with controlled *crossing complexity*, i.e., drawings where some types of edge crossings are forbidden. We study *h-quasi planar drawings*, i.e., drawings with no h mutually crossing edges; this measure of crossing complexity can be regarded as a sort of planarity relaxation. The combinatorial properties of h-quasi planar drawings have been widely investigated [18,21]. The contributions of the paper are as follows: (i) We prove that every n-vertex partial k-tree (i.e., any graph with bounded treewidth) admits a straight-line h-quasi planar drawing in $O(n)$ area, where h depends on k but not on n (Section 3). (ii) For specific sub-families of partial k-trees (outerplanar graphs, flat series-parallel graphs, and proper simply-nested graphs), we provide ad-hoc algorithms that compute h-quasi planar drawings in $O(n)$ area with values of h significantly smaller than those obtained with the general technique (Section 4). (iii) We compare the notion of h-quasi planarity with that of *h-planarity*, which allows every edge to be crossed at most h times. We prove that h-quasi planarity is, in some cases, less restrictive than h-planarity in terms of area requirement. Namely, while linear area h-quasi planar drawings exist for series-parallel graphs (i.e. partial 2-trees) with $h = 11$, we prove that for any given constant h there exists a family of series-parallel graphs that do not admit a linear area straight-line h-planar drawing (Section 5). For reasons of space, many proofs are omitted in this extended abstract.

2 Preliminaries

A *drawing Γ* of a graph G maps each vertex v of G to a point p_v on the plane, and each edge $e = (u, v)$ to a Jordan arc connecting p_u and p_v not passing through any other vertex; furthermore, any two edges have at most one point in common. If all edges are mapped to straight-line segments, Γ is a *straight-line drawing* of G. If all vertices are mapped to points with integer coordinates, Γ is a *grid drawing* of G. The *bounding box* of a straight-line grid drawing Γ is the minimum axis-aligned box containing the drawing. If the bounding box has side lengths $X - 1$ and $Y - 1$, then we say that Γ is a drawing with *area* $X \times Y$. A drawing Γ is *h-quasi planar* if it has no h mutually crossing edges. A 3-quasi planar drawing is also called a *quasi planar drawing*.

We recall now definitions about track layouts which have been introduced and studied by Dujmović, Pór and Wood [15]. A *vertex coloring* $\{V_i : i \in I\}$ of a graph G is a partition of the vertices of G such that no edge has both endvertices in the same partition set V_i ($i \in I$). The elements of I are *colors* and each set V_i is a *color class*. A *t-track assignment* of G consists of a vertex coloring with t colors and a total ordering $<_i$ of the vertices in each color class V_i. Each pair $(V_i, <_i)$ is a *track* and will be denoted as τ_i. An *X-crossing* in a track assignment consists of two edges (u, v) and (w, z) such that $u, w \in V_i$, $v, z \in V_j$, $u <_i w$ and $z <_j v$, for $i \neq j$. An *edge c-coloring* of G is a partition of the edges of G into c sets, each set called a *color*. A *(c, t)-track layout of* G consists of a t-track assignment of G and an edge c-coloring of G such that no two edges of the same color form an X-crossing. The minimum t such that a graph G admits a (c, t)-track layout is denoted by $tn_c(G)$. A $(1, t)$-track layout is called a *t-track layout*. The *track-number* of G is $tn_1(G)$, simply denoted by $tn(G)$.

A *k-tree*, $k \in \mathbb{N}$, is defined as follows. The clique of size k is a k-tree; the graph obtained from a k-tree by adding a new vertex adjacent to each vertex of a clique of size k is also a k-tree. A *partial k-tree* is a subgraph of a k-tree. A graph has bounded *treewidth* if and only if it is a partial k-tree [5].

3 Compact h-Quasi Planar Drawings of Partial k-Trees

In this section we first describe a general technique to "transform" a (c,t)-track layout into an h-quasi planar drawing in linear area with $h = c(t-1)+1$. We then describe how to compute a $(2,t)$-track layout of a k-tree where t depends on k but not on n. The two results imply that every partial k-tree admits an h-quasi planar drawing in linear area, where h depends on k but not on n.

Lemma 1. *Let G be a graph with n vertices. If G admits a (c,t)-track layout, then G admits an h-quasi planar grid drawing in $O(t^3 n)$ area, where $h = c(t-1)+1$.*

Proof. We describe how to use a (c,t)-track layout γ of G to compute an h-quasi planar grid drawing in $O(t^3 n)$ area, where $h = c(t-1)+1$. The vertices of each track τ_i $(i = 0, \ldots, t-1)$ are drawn as points of a horizontal segment s_i whose y-coordinate is $-i$. The idea is to place the t segments s_i on a parabola in such a way that no connection between two segments crosses a third. We place the vertices on s_i from left to right according to $<_i$. As a consequence, no two edges whose endvertices belong to two tracks τ_i and τ_j can cross in the drawing unless they form an X-crossing in γ. We will use this fact to bound the number of mutually crossing edges. More precisely, the vertices are placed on s_i from left to right according to $<_i$, with unit distance between any two consecutive vertices. Each segment has length n^*, where $n^* = \max_i\{|\tau_i|\} - 1$ (thus, the length of s_i is sufficient to host all vertices of τ_i). We denote by p_i and q_i the leftmost and the rightmost point of s_i, respectively. Also, we denote by x_i the x-coordinate of p_i. We place each segment s_i in such a way that $x_i = x_{i-1} + n^* + \Delta_i + 1$, where $\Delta_i = 2(i-1)n^* + i$ (see Figure 1 for an example).

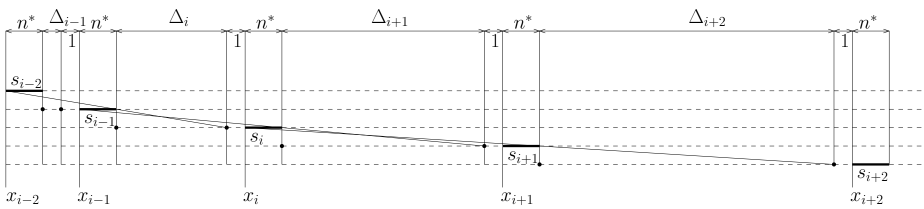

Fig. 1. Illustration of the construction described in Lemma 1

We prove now that the computed drawing Γ is an h-quasi planar grid drawing of G with $h = c(t-1)+1$. First of all we prove that no edge in the drawing passes through a vertex in Γ. Let (u,v) be an edge with $u \in \tau_i$ and $v \in \tau_j$, with $i < j$. If (u,v) passed through a vertex w, then w would belong to a track τ_l with $i < l < j$. We prove that segment s_l is in fact completely to the left of the segment $\overline{p_i p_j}$ and therefore w is to the left of (u,v). The proof is by induction on $j-i$. The base case is when $j-i = 2$ (and $l = i+1$). In this case s_{i+1} is to the left of $\overline{p_i p_{i+2}}$ by construction; namely, the slope of the segment $\overline{p_i q_{i+1}}$ is $-\frac{1}{x_{i+1}+n^*-x_i} = -\frac{1}{2(i+1)n^*+i+2}$, while the slope of the segment $\overline{p_i p_{i+2}}$ is $-\frac{2}{x_{i+2}-x_i} = -\frac{1}{2(i+1)n^*+i+2.5}$, which implies that the whole segment s_{i+1} is to the left of $\overline{p_i p_{i+2}}$. Assume now that $j-i>2$. All segments s_r with $i<r<j-1$ are to the left of segment $\overline{p_i p_{j-1}}$ by induction; s_{j-1} is to the left of $\overline{p_{j-2} p_j}$ also by induction. It follows that all segments s_r with $i<r<j-1$ are to the left of $\overline{p_i p_j}$.

Every edge (u,v) with u drawn on s_i and v drawn on s_j is completely contained in the parallelogram $\Pi_{i,j}$ whose corners are p_i, q_i, p_j, and q_j $(0 \leq i,j \leq t-1)$. By definition of (c,t)-track layout there are at most c mutually crossing edges inside each parallelogram. We will show that at most $t-1$ parallelograms $\Pi_{i,j}$ mutually overlap, which implies that there are at most $c(t-1)$ mutually crossing edges in our drawing. Consider two parallelograms $\Pi_{i,j}$ and $\Pi_{r,l}$ and assume without loss of generality that $i<j$ and $r<l$. It is easy to see that $\Pi_{i,j}$ and $\Pi_{r,l}$ overlap if and only if one of the following three conditions hold: (α) $i<r<j<l$; (β) $i=r$; (γ) $j=l$. The proof that at most $t-1$ parallelogram mutually overlap in Γ is by induction on t. If $t=2$, there is a single parallelogram and the statement trivially holds. Assume now that $t>2$. We denote by Γ_i $(0 \leq i \leq t-1)$ the subdrawing of Γ induced by the vertices drawn on the segments $s_0,\dots,s_i$. Suppose, as a contradiction, that there is a set S of at least t mutually overlapping parallelograms in Γ_{t-1}. Partition S into two subsets P and R defined as follows. $P=\{\Pi_{i_1,t-1}, \Pi_{i_2,t-1}, \dots, \Pi_{i_{|P|},t-1}\}$ is the set of parallelograms having s_{t-1} as rightmost side and $R=S\setminus P$. Since, by induction, there are at most $t-2$ mutually overlapping parallelograms in Γ_{t-2}, P contains at least two parallelograms, i.e., $|P| \geq 2$. Observe that, by conditions α,β,γ, the parallelograms in R have a side s_j with $0 \leq j \leq i_1$ and a side s_l with $i_{|P|}+1 \leq l \leq t-2$. Also, all these parallelograms are present in Γ_{t-2}. By our assumption that S contains at least t parallelograms, it follows that $|R| \geq t-|P|$. Let l be the greatest index among the segments in R $(i_{|P|}+1 \leq l \leq t-2)$; we have that each parallelogram in the set $Q=\{\Pi_{i_2,l},\dots,\Pi_{i_{|P|},l}\}$ and all the parallelograms in R mutually overlap. Thus, they form a bundle of mutually overlapping parallelograms of size $|R|+|Q| \geq t-|P|+|P|-1=t-1$ in Γ_{t-2}, a contradiction.

We conclude the proof by showing that the area of the computed drawing is $O(t^3 n)$. We have $x_i = x_{i-1}+(2i-1)n^*+i+1$. We show by induction that $x_i = x_0 + i^2(n^*+1) - \frac{i(i-3)}{2}$. This is true for $i=0$; assume it is true for $i-1$, we have $x_i = (x_0+(i-1)^2(n^*+1)-\frac{(i-1)(4-i)}{2})+(2i-1)n^*+i+1 = x_0+i^2(n^*+1)-\frac{i(i-3)}{2}$. The width of the drawing is $x_{t-1}+n^*-x_0$ which is $(t-1)^2(n^*+1)-\frac{(t-1)(t-4)}{2}+n^* = O(t^2 n^*) = O(t^2 n)$. Since the height is $O(t)$ the statement follows. □

Lemma 1 implies that every graph with constant track number admits an h-quasi planar grid drawing in linear area with h being a constant. Since it is known that partial k-trees have track number that is constant in n (although depending on k) [14], this implies that every partial k-tree admit an h-quasi planar grid drawing in linear area where the value of h does not depend on n. The current best upper bound on the track number of k-trees is given in [12]. Thus, every k-tree has an h_k-quasi planar drawing in $O(n)$ area with $h_k \in O(1)$. In what follows we will improve this result by presenting a technique that gives better values for h_k.

Now we describe an algorithm, called `kTreeLayouter`, that computes a $(2, t)$-track layout of a k-tree where t depends on k but not on n. We start by recalling a decomposition technique introduced by Dujmović, Morin, and Wood [14] and by giving some further definitions that will be used to prove our results. Let $G = (V(G), E(G))$ be a graph and let $T = (V(T), E(T))$ be a rooted tree. Let $\{T_\mu \subseteq V(G) \mid \mu \in V(T)\}$ be a set of subsets of $V(G)$ indexed by the nodes of T. The pair $(T, \{T_\mu \mid \mu \in V(T)\})$ is a *tree partition* of G if: (i) $\forall \mu, \nu \in V(T)$, if $\mu \neq \nu$ then $T_\mu \cap T_\nu = \emptyset$; (ii) $\forall (u, v) \in E(G)$, either $\exists$ a node $\mu \in V(T)$ with $u, v \in T_\mu$, or $\exists$ an edge $(\mu, \nu) \in E(T)$ such that $u \in T_\mu$ and $v \in T_\nu$. Let μ be an element of $V(T)$ of a tree partition of G. The *pertinent graph* of μ is the subgraph of G induced by the vertices in T_μ; the pertinent graph of μ is denoted as G_μ. The following result about tree-partitions of k-trees is proved in [14].

Theorem 1. [14] *Let G be a k-tree. There exists a tree-partition $(T, \{T_\mu \mid \mu \in V(T)\})$ of G such that for every node μ of T: (i) The pertinent graph G_μ is a connected partial $(k-1)$-tree. (ii) If μ is a non-root node of T and ν is the parent of μ in T, then the set of vertices in T_ν with a neighbour in T_μ induce a clique of size k in G.*

The clique induced by the vertices in T_ν with a neighbour in T_μ is called the *parent clique* of μ. From now on, we shall only consider tree partitions with the properties of Theorem 1. For reasons of brevity, we shall often use T rather than $(T, \{T_\mu \mid \mu \in V(T)\})$ to denote a tree partition. Let (μ, ν) be an edge of T such that μ is the parent of ν. Let $e = (u, v)$ be an edge of G such that $u \in T_\mu$ and $v \in T_\nu$. Edge e is a *jumping edge*, vertex u is the *parent vertex* of e, and vertex v is the *child vertex* of e. We call *r-prism* a group of r tracks ($r > 1$) and i-clique a clique of size i ($i > 0$). Let G be a k-tree, let $\gamma(G)$ be a (c, t)-track layout of G and let Θ be a subset of $k+1$ tracks of $\gamma(G)$. Let C be an $(k+1)$-clique of G. *C covers Θ* if C has one vertex in each track of Θ. Let C_0 and C_1 be two $(k+1)$-cliques of G. C_0 and C_1 are of the same *category* if they cover the same subset of tracks in $\gamma(G)$. The number of distinct categories of $\gamma(G)$ is called the *a-number* of $\gamma(G)$. Let C_0 and C_1 be two $(k+1)$-cliques of G of the same category. C_0 and C_1 have the same *color* if the vertices of one of them (say C_0) precede (or possibly coincide with) the vertices of the other one (i.e., C_1) on all the tracks covered by the two cliques. This means that no two edges of the two cliques form an X-crossing. Notice that, given two cliques of the same color it is possible to order them according to the order of their vertices on the tracks that they cover. The maximum number of colors over all categories of $\gamma(G)$ is called the *b-number* of $\gamma(G)$. Two $(k+1)$-cliques of G are of the same *type* if they are of the same category and have the same color. The number of distinct types (which is at most $a \cdot b$) is called the *c-number* of $\gamma(G)$. Since the cliques of the same color can be totally ordered, the cliques of the same type can be totally ordered accordingly. We denote such an ordering as $\prec_c$.

Let G be a k-tree, an *equipped tree partition* T of G is a tree partition such that each node μ is equipped with a (g_μ, t_μ)-track layout $\gamma(G_\mu)$ of its pertinent graph G_μ. Let a_μ, b_μ, and c_μ be the a-number, the b-number, and the c-number of $\gamma(G_\mu)$, respectively. We denote by t_T the value $\max_{\mu \in V(T)} t_\mu$. Analogously, we set $a_T = \max_{\mu \in V(T)} a_\mu$, $b_T = \max_{\mu \in V(T)} b_\mu$, and $c_T = \max_{\mu \in V(T)} c_\mu$. In order to compute a $(2, t)$-track layout of a k-tree G, we use a recursive technique based on an equipped tree partition of G. The pertinent graph G_μ of any node μ of the tree-partition T is a partial $(k-1)$-tree. G_μ is augmented to a k-tree and a $(2, t_\mu)$-track layout $\gamma(G_\mu)$ of G_μ with at most

t_T tracks is recursively computed. The maximum number of types of cliques in any $\gamma(G_\mu)$ is c_T. For each type of clique in G_μ the (at most) t_T-tracks of the $(2, t_\mu)$-track layout of each node whose parent clique is of that type are identified with the t_T tracks of a different t_T-prism. We define a total order $\prec_T$ of the nodes of the equipped tree partition T of G. To this aim we first define a total order $\prec_n$ of the children of each node λ of T. The children of λ are first ordered according to the categories of their parent cliques (the categories are ordered arbitrarily), within the same category they are ordered according to their parent clique color (the colors are ordered arbitrarily), within the same type they are ordered according to the order $\prec_c$ of their parent cliques; if they have the same parent clique they are ordered arbitrarily. The total order $\prec_T$ of the nodes of T is the order given by a preorder visit of T where the children of each node are visited according to $\prec_n$.

We are now ready to describe the algorithm `kTreeLayouter` to compute a $(2, (c_T + 1)t_T)$-track layout of G. We will use $(c_T + 1)$ t_T-prisms denoted as $P_0, \ldots, P_{c_T}$. This results in a number of tracks equal to $(c_T + 1)t_T$. The tracks of prism P_h $(0 \leq h \leq c_T)$ are denoted as $\tau_{h \cdot t_T + i}$ $(0 \leq i \leq t_T - 1)$. The nodes of T are processed one per time according to the total ordering $\prec_T$. Let G_μ be the pertinent graph of the current node μ, let λ be the parent of μ and let P_h be the t_T-prism whose tracks contain the vertices of G_λ. Let C be the parent clique of μ and let $\chi_{i,j}$ be its type $(0 \leq i \leq a_T - 1$, $0 \leq j \leq b_T - 1)$, the (at most) t_T-tracks of the $(2, t_\mu)$-track layout $\gamma(G_\mu)$ of G_μ are identified with the t_T tracks of the t_T-prism $P_{h'}$ with $h' = (h + b_T \cdot i + j + 1) \mod (c_T + 1)$. Notice that, $h + 1 \leq b_T \cdot i + j + 1 \leq h + c_T$, which means that the t_T-prism $P_{h'}$ is different from P_h. Consider now a vertex v of G_μ and suppose that v belongs to a track τ_l $(0 \leq l \leq t_T - 1)$ in $\gamma(G_\mu)$; v is assigned to the track $\tau_{h' \cdot t_T + l}$ of $P_{h'}$. Moreover, the vertices of G_μ are ordered in the tracks of $P_{h'}$ in such a way that: (i) their relative order is the same as the one they have in $\gamma(G_\mu)$; (ii) they follow the vertices on their track that belong to the pertinent graph $G_{\mu'}$ of any node μ' of T that has been processed before μ by the algorithm. It is easy to see that the algorithm `kTreeLayouter` computes a $((c_T + 1)t_T)$-track assignment $\gamma(G)$. Namely, the edges of each G_μ do not have both endvertices in the same track because $\gamma(G_\mu)$ is a $(2, t_\mu)$-track layout; the jumping edges have endvertices in different tracks because they are in different t_T prisms. To prove that $\gamma(G)$ is a $(2, (c_T + 1)t_T)$-track layout of G, we give a preliminary lemma.

Lemma 2. *Let G be a k-tree and let $\gamma(G)$ be the track assignment computed by algorithm `kTreeLayouter`. Let τ_h and τ_l ($0 \leq h, l \leq (c_T + 1)t_T - 1$) be two tracks of $\gamma(G)$. Let $e_0 = (u_0, v_0)$ and $e_1 = (u_1, v_1)$ be two jumping edges of G such that $u_0, u_1 \in \tau_h$, $v_0, v_1 \in \tau_l$, u_0 is the parent vertex of e_0 and u_1 is the parent vertex of e_1. Then e_0 and e_1 do not form an X-crossing.*

Lemma 3. *Let G be a k-tree. The algorithm `kTreeLayouter` correctly computes a $(2, (c_T + 1)t_T)$-track layout $\gamma(G)$ of G.*

Proof. In [15] it has been shown that, given a t-track assignment γ, it is possible to color the edges with c distinct colors so that no two edges of the same color form an X-crossing (i.e., to compute a (c, t)-track layout) if and only if γ has no crossing $(c+1)$-tuple. A set S of $c + 1$ edges in a track assignment γ is called a *crossing* $(c + 1)$*-tuple* if each pair of edges in S form an X-crossing in γ. Thus, to prove our statement it sufficient to show that there is no crossing 3-tuple in $\gamma(G)$. Consider any three edges $e_0 = (u_0, v_0)$, $e_1 = (u_1, v_1)$, and $e_2 = (u_2, v_2)$ such that u_0, u_1, and u_2 are in the same track τ_h and v_0, v_1, and v_2 are in the same track τ_l ($0 \le h, l \le (c_T+1)t_T - 1$). Assume first that τ_h and τ_l belong to the same t_T-prism. If e_0, e_1, and e_2 are edges of the same pertinent graph G_μ, then they do not form a crossing 3-tuple because otherwise there would be a crossing 3-tuple in the $(2, t_T)$-track layout $\gamma(G_\mu)$ of G_μ. If e_0, e_1, and e_2 are edges of different pertinent graphs, then at least two of them do not form an X-crossing (give two distinct pertinent graphs on the same t_T-prism, the vertices of one of them follow the vertices of the other one) and therefore they cannot form a crossing 3-tuple.

Assume now that τ_h and τ_l belong to different t_T-prisms (e_0, e_1, and e_2 are jumping edges). At least two among u_0, u_1, and u_2 are either parent vertices or child vertices of their jumping edges. By Lemma 2, at least two among e_0, e_1 and e_2 do not cross. □

The proof of the upper bound to the value $(c_T + 1)t_T$ is omitted. We can prove that the values of h_k given in Theorem 2 are smaller than those obtained by using the track number upper bound in [12].

Theorem 2. *Every partial k-tree with n vertices admits an h_k-quasi planar grid drawing in $O(t_k^3 n)$ area, where $h_k = 2t_k - 1$ and t_k is given by the following recursive equation:*

$$t_k = (c_{k-1,k} + 1)t_{k-1}$$

$$c_{k,i} = (c_{k-1,k} + 1)\Big(c_{k-1,i} + \frac{c_{k-1,k}}{4}\sum_{j=1}^{i-1} c_{k-1,j} \cdot c_{k-1,i-j}\Big) \quad (i = 1, \dots, k+1) \quad (1)$$

$$c_{k,k+2} = 0$$

with $t_1 = 2$ and $c_{1,1} = 4$ and $c_{1,2} = 2$.

By Theorem 2, every partial 2-tree admits an 11-quasi planar drawing in $O(n)$ area. Partial 2-trees are SP-graphs [5], which will be further investigated in the next sections.

4 Improved Bounds for Specific Families of Planar Partial k-Trees

According to Theorem 2, every n-vertex partial k-tree admits an h-quasi planar drawing in $O(n)$ area with $h \in O(1)$. In this section we describe some ad-hoc drawing techniques that, still producing drawing in linear area, reduce the value of h for some sub-families of partial k-trees.

Outerplanar graphs. A graph G is *outerplanar* if it admits a planar embedding such that all vertices are on the external face (i.e., an *outerplanar embedding*). It is known that outerplanar graphs are partial 2-trees [5]. Thus, from Theorem 2 they admit an 11-quasi planar drawing in $O(n)$ area. We prove that the value of h can be reduced from 11 to 3, describing an algorithm `OuterplanarDrawer`, which takes as input an n-vertex outerplanar graph G with a given outerplanar embedding and returns a quasi planar grid drawing of G. The algorithm uses an approach similar to the one described in [17]. It can be divided in two main steps. In the first step it computes a drawing Γ^* of G as follows. Perform a breadth-first-search of G (starting from any vertex) and assign to each vertex v of G two numbers: $level(v)$ which is the depth of v in the BFS tree, and $order(v)$ which is the progressive number of v in the BFS order. For each vertex v of G set $x^*(v) = order(v)$ and $y^*(v) = level(v)$, where $x^*(v)$ and $y^*(v)$ are the x- and y-coordinates of v in Γ^*, respectively. In the second step, it "wraps" the drawing Γ^* of G on two levels, producing the final drawing Γ. For each vertex v of G, it sets $x(v) = x^*(v)$ and $y(v) = y^*(v) \mod 2$, where $x(v)$ and $y(v)$ are the x- and y-coordinates of v in Γ, respectively.

Theorem 3. *Every outerplanar graph with n vertices admits a quasi planar grid drawing in $O(n)$ area.*

Flat series-parallel graphs. A series-parallel graph, or SP-graph, is *flat* if it does not contain two nested parallel components. For an exact definition of flat SP-graphs and decomposition tree see [11]. Flat SP-graphs are a meaningful subfamily of SP-graphs, previously studied in [11]. We lower the value of h for flat SP-graphs from 11 to 5.

Let G be a flat SP-graph and let T be its decomposition tree. We assign to each node ν of T a number, denoted as $level(\nu)$, computed as follows. The root ρ of T has $level(\rho) = 0$. For each non-root node ν, if ν is an S-node then $level(\nu) = level(parent(\nu)) + 1$, else $level(\nu) = level(parent(\nu))$. Using the level numbering of T we assign a number $level(v)$ to each vertex v of G, which is the minimum among the levels of all the nodes of T having v as a pole. We call *jumping edges* those edges whose end-vertices are assigned to different levels. Notice that the level numbering is such that the level number changes in correspondence of the S-nodes. In [11] it has been proved that the leftmost child and the rightmost child of an S-node are both Q-nodes and the edges associated with them are both jumping edges.

We can now describe the drawing algorithm `FlatSPDrawer`, which takes as input a flat SP-graph G and its decomposition tree T and returns a 5-quasi planar grid drawing of G. Also in this case the algorithm has two main steps. In the first step we produce a preliminary drawing Γ^* of G. For each vertex v of G, we set $y^*(v) = level(v)$ and compute x^* as follows. We perform a breadth first search of T, initializing a counter $i = 0$ before starting the visit. For each node ν of T in the BFS order, if ν is a P-node or a Q-node we process its two poles s and t: if the x-coordinate of the source s has not yet been assigned we set $x^*(s) = i$ and increment i by one unit; if the x-coordinate of the sink t has not yet been assigned we set $x^*(t) = i$ and increment i. Notice that if both the poles of ν have not been processed before considering ν (i.e., ν does not share them with its parent), then they receive consecutive x-coordinates. Again, Γ is obtained from Γ^* by setting $x(v) = x^*(v)$ and $y(v) = y^*(v) \mod 2$ for each vertex v of G.

Lemma 4. *Let G be a flat SP-graph and let T be its decomposition tree. Let $l \leq n$ be the number of levels assigned to the nodes of T by the algorithm* `LevelNumbering`*. The first step of the algorithm* `FlatSPDrawer` *produces a drawing Γ^* of G on l levels, such that: (i) for every edge e of G either e connects two vertices on the same level, or e is a jumping edge connecting vertices between two consecutive levels; (ii) there are no overlaps among edges; (iii) there are no three mutually crossing edges.*

Proof. By definition a non-jumping edge $e = (u, v)$ has $|level(u) - level(v)| = 0$, and therefore e connects two vertices on the same level. For a jumping edge $e = (u, v)$, we have, by definition, $|level(u) - level(v)| > 0$. Let e be a jumping edge. As already said, the Q node representing e in T is either the leftmost or the rightmost child of a non-root S-node ν. As a consequence, one end vertex of e, say u, is a pole shared by ν and its parent (a P-node); the other end vertex v is a pole shared by two consecutive children of ν (one of which is the leftmost or rightmost child) and not shared with ν. Since the level number changes only in correspondence of the S-nodes, $level(u) = level(parent(\nu))$ and $level(v) = level(\nu)$, i.e., $|level(u) - level(v)| = 1$.

Since every vertex v has a different x-coordinate, there can be only two kinds of crossings: an overlap between two non jumping edges, or a proper crossing between two jumping edges. We prove now that the first case never happen. Let ν be a P-node or a Q-node of T such that $level(\nu) = j$, let s and t be its two poles and let μ be its parent node. We have the following cases: **(1)** ν is a P-node and μ is an S-node. In this case ν and μ do not share a pole (the leftmost and rightmost child of μ are Q-nodes), thus the two poles of ν have consecutive x-coordinates, i.e., $x^*(s) - x^*(t) = 1$ and $y^*(s) = y^*(t)$. **(2)** ν is a Q-node and μ is a P-node. In this case ν represents a transitive edge connecting the two poles of μ which have already been processed when μ was considered; by case 1 we have $x^*(s) - x^*(t) = 1$ and $y^*(s) = y^*(t)$. **(3)** ν is a Q-node and μ is an S-node. If ν is the leftmost/rightmost child of μ, its associated edge is a jumping edge and the two poles have distinct y-coordinates. If ν is not the leftmost/rightmost child of μ, then ν and μ do not share a pole. Also in this case the two poles of ν have consecutive x-coordinates, i.e., $x^*(s) - x^*(t) = 1$ and $y^*(s) = y^*(t)$. If e is a non-jumping edge, then either Case 2 or 3 holds for its corresponding Q-node. In both cases the endvertices of e have consecutive x-coordinates. It follows that there can not be an overlap between two non-jumping edges.

Now we prove that there are no more than 2 mutually crossing jumping edges. We assign to a jumping edge the red color if its corresponding Q-node is the leftmost child of its parent and the blue color if its corresponding Q-node is the rightmost child of its parent. Let $e = (u, v)$ and $e' = (w, z)$ be two jumping edges of the same color. If $level(u) \neq level(w)$ or $level(v) \neq level(z)$ then it is immediate to see that e and e' do not cross. Assume then $level(u) = level(w) = j$ and $level(v) = level(z) = j + 1$. If e and e' share an end vertex they obviously cannot cross. If e and e' do not share an end vertex, u and w are two poles of two different S-nodes ν_u and ν_w. Assume that $x^*(u) < x^*(w)$, which means that ν_u is visited before ν_w in the BFS visit of T. This implies that the Q-node of e is visited before the Q-node of e'. Thus, $x^*(v) < x^*(z)$ and e and e' cannot cross. Hence, there cannot be three mutually crossing edges because red edges can cross only blue edges and vice versa. □

Theorem 4. *Every flat SP-graph with n vertices admits a 5-quasi planar grid drawing in $O(n)$ area.*

Proper simply-nested graphs. A graph is *k-outerplanar* ($k > 1$) if it admits a planar embedding such that the graph remaining after removing all vertices on the external face is a $(k - 1)$-outerplanar graph. A graph is 1-*outerplanar* if it is outerplanar. In other words a graph is k-outerplanar if it admits a planar embedding such that it can be made empty by removing the vertices on the external face k times. The vertices that are on the external face after i ($0 \leq i \leq k-1$) removals are called vertices of level $i+1$. A *simply-nested graph* is a k-outerplanar graph such that the vertices of levels from 1 to $k-1$ are chordless cycles and level k is either a cycle or a tree. Simply-nested graphs have been widely studied in the literature (see, e.g., [2]). We say that a simply-nested graph is *proper* if level k is a chordless cycle. It is known that k-outerplanar graphs have treewidth at most $3k-1$ [5]. By using the technique of Section 3 we would obtain an h-quasi planar drawing in linear area with h given by Equation 1. Notice that h would be a function of the number of levels k. We show that for simply-nested graphs h can be reduced to 3 (independent of the number of levels). We remark that proper simply-nested graphs may require quadratic area if we want a planar drawing; they include the classical examples used to prove the quadratic area lower bound of planar graphs.

Theorem 5. *Every proper simply-nested graph with n vertices admits a quasi planar grid drawing in $O(n)$ area.*

Sketch of Proof: Let G be a proper simply-nested graph. We describe an algorithm to compute a quasi planar grid drawing of G. Let $C_1, C_2, \ldots, C_k$ be the cycles of levels $1, 2, \ldots, k$, respectively. We assume that all the internal faces of G except possibly the one delimited by C_k are triangles. If this is not the case, we can add edges to guarantee this property. For each cycle C_i we choose a vertex, denoted as v_i, called the *reference vertex* of C_i. The references vertices are chose in such a way that v_i is adjacent to v_{i-1}.

We draw the vertices of each cycle C_i on an isosceles triangle T_i whose basis has length $2(n_i - 2)$ and height 2, where $n_i = |C_i|$. The y-coordinate of the apex of T_i is 3 if i is odd or 0 if i is even. The y-coordinate of the basis of T_i is 1 if i is odd or 2 if i is even. All the vertices of C_i, except v_i, are placed on the basis of T_i on grid points with even x-coordinates so that their left-to-right order coincides with (is opposite to) their counter-clockwise order along C_i if i is odd (if i is even). Vertex v_i is drawn at the apex of T_i. Denote by x_i the x-coordinate of v_i ($i = 1, \ldots, k$) and let $m_i = \max\{n_{i-1}, n_i\}$ ($i = 2, \ldots, k$). The triangles are placed so that $x_1 = n_1 - 2$ and $x_i = x_{i-1} + \lceil \frac{3}{2}(m_i - 1) \rceil$. □

5 Comparing h-Quasi Planarity and h-Planarity

Other definition of crossing complexity are possible, for example *h-planarity* [20]. A drawing of a graph is *h-planar* if no edge has more than h crossings. Straight-line 1-planar drawings are studied in [16]. A natural question deriving from the results of Sections 3 and 4 is whether analogous results also hold for h-planar drawings. Theorem 6 shows that, for every constant h, $\omega(n)$ area is required for SP-graphs, while Theorem 2 implies that every SP-graph admits an 11-quasi planar drawing in $O(n)$ area.

Let G be a graph, we define the h^**-extension* of G as a graph G^*, constructed by attaching h^* paths of length 2 to each edge of G.

Lemma 5. *Let h be a positive integer, and let G be a planar graph. In any h-planar drawing of the $3h$-extension G^* of G, there are no two edges of G that cross each other.*

Theorem 6. *Let h be a positive integer, for every $n > 0$ there exist a $\Theta(n)$-vertex series-parallel graph such that any h-planar straight-line or poly-line grid drawing requires $\Omega(n2^{\sqrt{\log n}})$ area.*

Proof. Let G be an n-vertex graph of the family defined by Frati in [19], which requires $\Omega(n2^{\sqrt{\log n}})$ area in any planar straight-line or poly-line drawing. By Lemma 5 there exists an n^*-vertex planar graph G^*, with $n^* = \Theta(n)$, such that in any h-planar drawing Γ of G^* the underlying graph G must be drawn planar. Since G^* is still a SP-graph (the $3h$-extension preserves the property of being a SP-graph) the statement follows. □

With the same argument, we can prove the following theorem for general planar graphs. Notice that it states that quadratic area is necessary if we impose $h(n) \in O(1)$.

Theorem 7. *Let ε be given such that $0 \leq \varepsilon \leq 0.5$ and let $h(n) : \mathbb{N} \to \mathbb{N}$ be a function such that $h(n) \leq n^{0.5-\varepsilon}$ $\forall n \in \mathbb{N}$. For every $n > 0$ there exists an $O(n)$-vertex graph G such that any $h(n)$-planar straight-line grid drawing of G requires $\Omega(n^{1+2\varepsilon})$ area.*

6 Concluding Remarks and Open Problems

In this paper we studied the problem of computing compact h-quasi planar drawings of partial k-trees. Indeed, our algorithms can be regarded as drawing techniques that produce drawings with optimal area and with bounded crossing complexity. This point of view is particularly interesting in the case of planar graphs. As recalled in the introduction, planar graphs can be drawn with either optimal crossing complexity (i.e., in a planar way), in which case they may require $\Omega(n^2)$ area [8], or with optimal $\Theta(n)$ area but without any guarantee on the crossing complexity [22]. These two extremal results naturally raise the following question: is it possible to compute a drawing of a planar graph "controlling" both the area and the crossing complexity? In particular, it is possible to compute an h-quasi planar drawing of a planar graph in $o(n^2)$ area and $h \in o(n)$? In Section 4 we showed that $O(n)$ area and $h \in O(1)$ can be simultaneously achieved for some families of planar graphs. In fact our results imply a positive answer to the above question even for general planar graphs.

Theorem 8. *Every planar graph with n vertices admits a $O(\log^{16} n)$-quasi planar grid drawing in $O(n \log^{48} n)$ area.*

Proof. Let G be a n-vertex graph with acyclic chromatic number $\chi_a(G) \leq c$ and queue number $qn(G) \leq q$, then G has track-number $tn(G) \leq c(2q)^{c-1}$ [14]. If G is planar then $qn(G) \in O(\log^4 n)$ [10] and $\chi_a(G) = 5$ [6]. Thus, every planar graph has $tn(G) \in O(\log^{16} n)$ and by Lemma 1 the statement follows. □

The results in this paper give rise to several interesting open problems. Among them: (1) Reducing the value of h_k given by Equation 1 for other sub-families of partial k-trees. (2) Studying whether planar graphs admits h-quasi planar drawings in $O(n)$ area with $h \in o(n)$, possibly $h \in O(1)$. (3) Studying h-quasi planar drawings in linear area and aspect ratio $o(n)$.

References

1. Angelini, P., Di Battista, G., Didimo, W., Frati, F., Hong, S.-H., Kaufmann, M., Liotta, G., Lubiw, A.: RAC and LAC drawings of planar graphs in subquadratic area. In: ECG 2011, pp. 125–128 (2011)
2. Angelini, P., Di Battista, G., Kaufmann, M., Mchedlidze, T., Roselli, V., Squarcella, C.: Small Point Sets for Simply-Nested Planar Graphs. In: Speckmann, B. (ed.) GD 2011. LNCS, vol. 7034, pp. 75–85. Springer, Heidelberg (2011)
3. Biedl, T.C.: Drawing Outer-Planar Graphs in $O(n \log n)$ Area. In: Goodrich, M.T., Kobourov, S.G. (eds.) GD 2002. LNCS, vol. 2528, pp. 133–148. Springer, Heidelberg (2002)
4. Biedl, T.C.: Small Drawings of Series-Parallel Graphs and Other Subclasses of Planar Graphs. In: Eppstein, D., Gansner, E.R. (eds.) GD 2009. LNCS, vol. 5849, pp. 280–291. Springer, Heidelberg (2010)
5. Bodlaender, H.L.: A partial k-arboretum of graphs with bounded treewidth. TCS 209(1-2), 1–45 (1998)
6. Borodin, O.: On acyclic colorings of planar graphs. Discr. Math. 25(3), 211–236 (1979)
7. Crescenzi, P., Di Battista, G., Piperno, A.: A note on optimal area algorithms for upward drawings of binary trees. CGTA 2, 187–200 (1992)
8. de Fraysseix, H., Pach, J., Pollack, R.: How to draw a planar graph on a grid. Combinatorica 10, 41–51 (1990)
9. Di Battista, G., Frati, F.: Small area drawings of outerplanar graphs. Algorithmica 54, 25–53 (2009)
10. Di Battista, G., Frati, F., Pach, J.: On the queue number of planar graphs. In: Proc. of FOCS 2010, pp. 365–374 (2010)
11. Di Giacomo, E.: Drawing Series-Parallel Graphs on Restricted Integer 3D Grids. In: Liotta, G. (ed.) GD 2003. LNCS, vol. 2912, pp. 238–246. Springer, Heidelberg (2004)
12. Di Giacomo, E., Liotta, G., Meijer, H.: Computing straight-line 3D grid drawings of graphs in linear volume. CGTA 32(1), 26–58 (2005)
13. Didimo, W., Eades, P., Liotta, G.: Drawing graphs with right angle crossings. TCS 412(39), 5156–5166 (2011)
14. Dujmović, V., Morin, P., Wood, D.R.: Layout of graphs with bounded tree-width. SIAM J. on Comp. 34(3), 553–579 (2005)
15. Dujmović, V., Pór, A., Wood, D.R.: Track layouts of graphs. DMTCS 6(2), 497–522 (2004)
16. Hong, S.-H., Eades, P., Liotta, G., Poon, S.: Fáry's Theorem for 1-Planar Graphs. In: Gudmundsson, J., Mestre, J., Viglas, T. (eds.) COCOON 2012. LNCS, vol. 7434, pp. 335–346. Springer, Heidelberg (2012)
17. Felsner, S., Liotta, G., Wismath, S.K.: Straight-Line Drawings on Restricted Integer Grids in Two and Three Dimensions. In: Mutzel, P., Jünger, M., Leipert, S. (eds.) GD 2001. LNCS, vol. 2265, pp. 328–342. Springer, Heidelberg (2002)
18. Fox, J., Pach, J.: Coloring k_k-free intersection graphs of geometric objects in the plane. In: Proc. of SCG 2008, pp. 346–354. ACM (2008)
19. Frati, F.: Lower bounds on the area requirements of series-parallel graphs. DMTCS 12(5), 139–174 (2010)
20. Pach, J., Tóth, G.: Graphs drawn with few crossings per edge. Combinatorica 17, 427–439 (1997)
21. Suk, A.: k-Quasi-Planar Graphs. In: Speckmann, B. (ed.) GD 2011. LNCS, vol. 7034, pp. 266–277. Springer, Heidelberg (2011)
22. Wood, D.R.: Grid drawings of k-colourable graphs. CGTA 30(1), 25–28 (2005)