Chapter 21

Pathwidth and planar graph drawing

In this chapter, we study connections between the pathwidth and the height of planar graph drawings. Rather than using the previous definition of pathwidth, we will use in this chapter one that is easily seen to be equivalent: A graph has pathwidth $\leq k$ if and only if it has what we call a *vertex order of searchwidth at most* k: This is a vertex order $v_1, \ldots, v_n$ such that for any $1 \leq i \leq n$, there are at most k vertices among $\{v_1, \ldots, v_i\}$ that have neighbours in $\{v_{i+1}, \ldots, v_n\}$. We use $pw(G)$ to denote the pathwidth of a graph G.

21.1 From height to pathwidth

Surprisingly enough, the pathwidth of a planar graph G can serve as a lower bound on the size of a planar drawing of G.

Theorem 21.1 *[FLW03] Let G be a planar graph and presume it has a planar drawing with integer coordinates in a grid of height h. Then $h \geq pw(G)$.*

Rather than giving a direct proof, we give here a small detour (with a result that could be interesting in its own right.) Recall that a *visibility representation* represents every vertex as an axis-aligned box, and every edge is drawn as a vertical or horizontal line segment between the two boxes of its endpoints.[1] Since we only study planar graphs here, we always assume that such a visibility representation is planar as well, i.e., no two drawings of edges cross, and no edge crosses the box of a vertex.

Lemma 21.2 *If G has a planar straight-line drawing of height h, then G also has a planar visibiliy representation of height h.*

Proof: After possibly adding edges, we may assume that in the straight-line drawing of G, all inner faces are triangles.

[1]In fact, all the visibility representations we studied are *flat* visibility representations where the box of a vertex occupies only one row. But the claims in this section hold for any visibility representation.

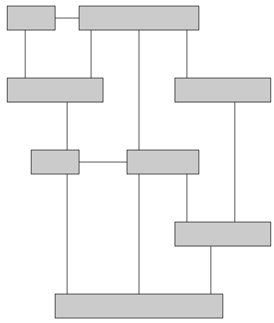


Figure 21.1: Example of a visibility representation.

Now we process the vertices in order of rows. In the topmost row, assign boxes to vertices in the same order in which they appear in the row. Any edge between two vertices in the top row is drawn horizontally between the corresponding boxes. Also, make the boxes of vertex v so wide that for each edge from v to a vertex below, we reserve a vertical line at the top. We maintain the same order among these incomplete edges as we had it in the planar straight-line drawing. See also Figure 21.2.

In any other row, we also assign boxes from left to right, but we also need to make sure that these boxes match the edges to vertices above. Since we triangulated the inner faces, we know that any vertex v in the row has a neighbour u that was in an earlier row. Hence, the box for u was already defined and it reserved a vertical line for the edge (u, v). To define the box of v, hence extend it to cover exactly the lines to all incomplete edges (u, v) with u in an earlier row. (Possibly the box of v is only a point.) Next, widen the box of v (by inserting new columns, if needed, next to it) so that it is wide enough that we can reserve a vertical line downwards for every edge (v, w) where w is on a later row.

During this transformation, we maintain within any row the order of the vertices and incomplete edges. Therefore, all horizontal edges in the straight-line drawing can simply be drawn as horizontal edges in the visibility representation, and the slanted or vertical edges become vertical edges in the visibility representation. Since we never add new rows, the height is maintained. □

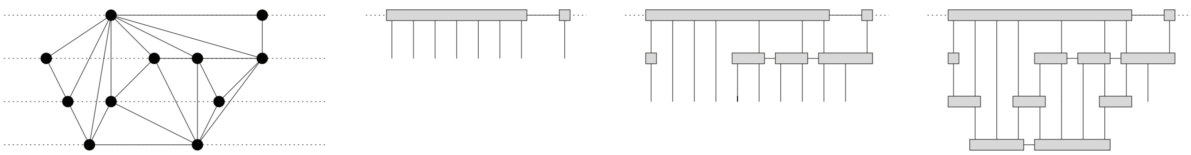


Figure 21.2: Converting a straight-line drawing to a visibility representation of the same height.

Lemma 21.3 *Let G be a graph that has a visibility representation of height h. Then G has pathwidth at most h.*

Proof: Given a visibility representation, we can naturally enumerate the vertices by the left endpoints of their representing segments. Here we break ties top-to-bottom. Let $v_1, \ldots, v_n$ be the resulting vertex order, and consider any $1 \leq i \leq n$. Let ℓ be the vertical line through the left endpoint of v_i. We consider the region "left of ℓ" to be everything that has either smaller x-coordinate than ℓ, or the same x-coordinate and is on or above v_i. Thus, vertices $v_1, \ldots, v_i$ are to the left of ℓ (at least partly), whereas vertices $v_{i+1}, \ldots, v_n$ are to the right of ℓ.

Let u be any vertex in $\{v_1, \ldots, v_i\}$ that has a neighbour w in $\{v_{i+1}, \ldots, v_n\}$. Since u is left of ℓ (at least partly) and w is right of ℓ, this is possible only if ℓ intersects the box of u or the drawing of edge (u, w) (and any such intersection must happen at a grid point since (u, w) is drawn horizontally or vertically.) Either way, we can find a grid point on ℓ (either the one that contains edge (u, w) or one that contains u) that can be assigned to u. Since no two edges or vertices overlap, we will assign every grid point on ℓ to at most one vertex. Therefore there are at most h vertices in $\{v_1, \ldots, v_i\}$ that have neighbours in the rest, which proves that G has pathwidth at most h. □

Putting the results together shows that any graph with a planar drawing of height h has pathwidth at most h.

21.2 Drawing trees with height $O(pw(T))$

Naturally one wonders whether the other direction holds. Given a graph of small pathwidth, can we draw it with small height? This turns out to be true for trees. In this section, we will review the algorithm by Suderman [Sud04] that shows that any tree T has a planar drawing of height at most $2pw(T)$. Since we just showed that any such drawing must have height at least $pw(T)$, we therefore get a 2-approximation algorithm for the height of drawing trees.

21.2.1 Main paths of trees

We first need to study the pathwidth of trees further. As it turns out (and was independently described multiple times [Sch90a, EST94]) the pathwidth is completely characterized by being able to split the tree apart at paths. We need a lemma:

Lemma 21.4 *Let T be a tree, and let v be a vertex of T. If $T - v$ has at least 3 subtrees that have pathwidth $\geq k$, then $pw(T) \geq k + 1$.*

Proof: Assume for contradiction that T has a vertex order with searchwidth at most k. Let T_1, T_2, T_3 be three subtrees of $T - v$ that have pathwidth k, then their induced vertex orders must have searchwidth exactly k. In particular, for each T_j there exists an index i_j such that k vertices of T_j with index $\leq i_j$ have a neighbour in T_j with index $> i_j$. We may assume after renaming that $i_1 \leq i_2 \leq i_3$.

We claim that no vertex v of T_3 has index $\leq i_2$. For if it did, then in the vertex order of T there are at least $k + 1$ vertices with index $\leq i_2$ (namely, k from T_2 and one from T_3)

that have a neighbour with index $> i_2$. But then the searchwidth of the order for T would be at least $k+1$, contradicting the assumption. By a symmetric argument no vertex of T_1 has index $> i_2$.

But now, where is v in this vertex order? If v's index is $\leq i_2$, then the edge from v to the root of T_3 crosses the cut at i_2. If v's index is $> i_2$, then the edge from v to the root of T_1 crosses the cut at i_2. So either way, there exists at least one vertex (either v or a vertex in T_1) that has index $\leq i_2$ but a neighbour has index $> i_2$. This, together with the k vertices in T_2, show that the searchwidth at index i_2 is at least $k+1$, a contradiction. □

This lemma can be used to characterize the pathwidth of trees.

Theorem 21.5 *A tree T has pathwidth at most $k > 0$ if and only if there exists a path P in T such that all subtrees of $T - P$ have pathwidth at most $k-1$.*

Proof: Presume first that T has pathwidth k, and let $v_1, \ldots, v_n$ be a vertex order of T that has searchwidth k. Let ℓ and r be the smallest and largest index where the searchwidth is attained, i.e., we know there are k vertices among $\{v_1, \ldots, v_\ell\}$ that have a neighbour in $\{v_{\ell+1}, \ldots, v_n\}$, and similarly for v_r.

Let P be the path from v_ℓ to v_r in T. We claim that all subtrees of $T-P$ have pathwidth at most $k-1$. To see this, use for any subtree T' of $T-P$ the vertex order induced by $v_1, \ldots, v_n$. If this had searchwidth $\geq k$, say at index i, then we would have $\ell \leq i \leq r$ by choice of ℓ and r. But then at least one edge of P would also cross the cut at index i. Since both endpoints of this edge belong to P (hence not to T'), therefore the vertex order of T would have searchwidth $\geq k+1$, a contradiction. So the pathwidth of T' is at most $k-1$.

Vice versa, assume there exists a path $P = w_1, \ldots, w_h$ such that any subtree has a vertex order with searchwidth at most $k-1$. Now create a vertex order for T as follows: Start with w_1, then list the vertex orders of each subtree T' of $T-P$ that is adjacent to w_1, then list w_2, then the vertex orders of the trees at w_2, and so on. In the resulting vertex order, at any time i we are at the vertex order of some subtree T' that is adjacent to (say) w_j, and the searchwidth is the searchwidth of the vertex order of T', plus one more for vertex w_j. Hence the searchwidth is at most k, which proves that T has pathwidth at most k. □

Call a path P a *main path* of tree T if any component of $T-P$ has smaller pathwidth than T. By the above any non-empty tree has a main path. Notice that it may have multiple main paths.

21.2.2 Drawing trees using main paths

The algorithm to draw trees is now by rooting the tree, and applying induction on the pathwidth. It will be crucial *where* the root of the tree is relative to a main path.

Lemma 21.6 *Let T be a rooted tree. Then T has a planar drawing of height $2pw(T)$ such that the root is drawn in the topmost row. Moreover, if the root belongs to a main path of T, then the drawing has height* $\max\{2pw(T)-1, 2\}$.

Proof: We proceed by induction on $pw(T)$. For $pw(T) = 1$, the tree is a caterpillar, and easily drawn on two rows, see Figure 21.3.

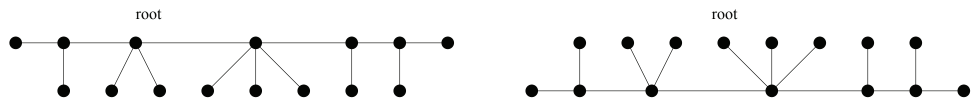
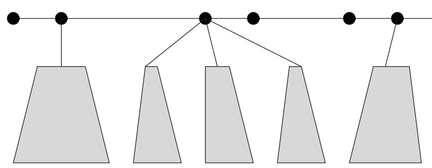


Figure 21.3: Drawing a caterpillar on two rows (with the root on the spine or the root a leg.)

Now let $pw(T) = k > 1$. As a first case, presume that the root of T belongs to a main path P. Then let $T_1, \ldots, T_\ell$ be the subtrees of $T - P$; by definition of a main path these have pathwidth at most $k-1$ and hence can be drawn with height at most $2k-2$ such that their roots are in the top row. To draw T, we draw the path P horizontally in the topmost row and then place the drawings of the subtrees in the $2k-2$ rows below. One easily verifies all conditions.



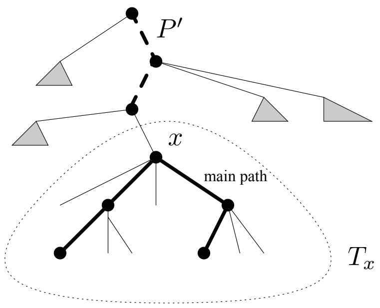


Figure 21.4: Drawing a tree by drawing a path and appending drawings of subtrees, and how to split the tree if the root is not on a main path.

In the second case, the root of T does not belong to a main path. Let x be the top-most vertex of a main path of T, and let P' be the path from the root to the parent of x. Now draw T as in the previous case, except use path P' instead of the main path P. We must argue that the height is correct.

Let T' be any subtree of $T - P'$. If T' is the subtree T_x rooted at x, then it can be drawn with height $2k-1$ (because T_x has pathwidth at most k, and its root is on a main path by definition of x.) In all other cases, T' is a subtree of $T - T_x$, which by definition of a main path means that it has pathwidth at most $k-1$ and hence it can be drawn with height at most $2k-2$. So any subtree T' of $T - P'$ can be drawn with height at most $2k-1$, yielding a height of $2k$ for the drawing of T. □

21.3 Drawing other graphs with height $O(pw(G))$?

So we know trees can be drawn with height $O(pw(T))$. Can we show this for any graph G? Unfortunately not. Recall the graph from Figure 21.5. This graph has outerplanarity $n/6$

(and if we fix this outer-face, it even has outerplanarity $n/3$), and hence by Theorem 19.1 requires $n/3$ height in any planar drawing ($2n/3$ height in any planar drawing that respects the outer-face.) But if we enumerate the vertices as illustrated in the figure, we get a vertex order where at any time at most 3 earlier vertices have neighbours later, so its pathwidth is at most 3.

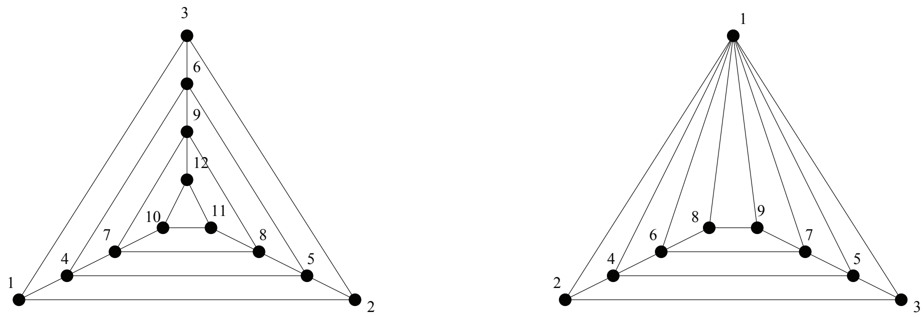


Figure 21.5: Example graphs with outerplanarity $n/3$ (in this planar embedding) and 2 that have constant pathwidth but require $\Omega(n)$ height in any planar embedding.

In fact, there even exist graphs that have *both* small outer-planarity and small pathwidth and still require a large height. See Figure 21.5 for the graph and [Bie11] for details.

However, some generalizations of the tree-result are possible: for maximal outerplanar graphs G, we can also achieve drawings of height $O(pw(G))$. Recall that a maximal outer-planar graph G is the same as an inner-triangulated n-cycle, and the dual of such a graph is a tree T, plus one vertex for the outer-face. One can show (this is not trivial) that for any maximal outerplanar graph G, the dual tree T satisfies $pw(T) \leq pw(G)$. Then, using the dual tree to guide how to break the graph apart and to draw, one can show that any maximal outerplanar graph G can be drawn with height at most $4pw(T)$. See [Bie13] for details.

This can even be generalized to all outer-planar graphs. This can be proved by showing that we can add edges to an outer-planar graph to make it maximal outerplanar without increasing the pathwidth too much. This is possible (but highly non-trivial) if we allow that pathwidth to increase by a factor of 16 [BBCR12]. Therefore, we can draw any outer-planar graph G with height at most $64pw(G)$.

However, this appears to be the end. There exists a series-parallel graph that requires $\Omega(2^{\sqrt{\log n}})$ height in any planar graph drawing [Fra10]. Since every series-parallel graph has pathwidth $O(\log n)$ (and $\log n \in o(2^{\sqrt{\log n}})$), therefore this series-parallel graph can not be drawn with height $O(pw(G))$. Likewise, the graph in Figure **??** is 2-outerplanar graph and has pathwidth 3 and requires height $\Omega(n)$ in any planar drawing [Bie11], hence 2-outerplanar graphs also cannot always be drawn with height $O(pw(G))$.

21.4 Testing drawability with height h

So it is not always possible to draw graphs with height $O(pw(G))$. But perhaps we can find an algorithm to test whether such drawings exist? Indeed this is the case.

Theorem 21.7 *For any planar graph G and any given h, testing whether G has a drawing of height h is fixed-parameter tractable in h.*

Proof: We only sketch the approach here, since it is quite complicated. First, run the algorithm from Corollary 20.7, which either proves that G has treewidth $> h$, or gives a tree decomposition of width at most $3h$. If G has treewidth $> h$, then of course G cannot have pathwidth at most h and so it certainly does not have a drawing of height h and we are done.

If G has treewidth at most $3h$, then we can test (with dynamic programming in the tree decomposition; we give no details of this) whether it has pathwidth at most h, in time that is fixed-parameter tractable in $3h$ (and hence also in h.) Again, if the answer is negative then we are done.

If the pathwidth is at most h, then the idea is to do dynamic programming in the path decomposition and to store for any subgraph whether it can be drawn in h rows. To be able to combine subgraphs, we need to specify how the $h+1$ vertices of each bag are drawn. Unfortunately, these vertices need not be drawn "nicely", with one vertex per row and the subgraph to the left. Instead, we must allow for the subgraph to be spread to both sides. Distinguishing now by the shape of the faces to the sides, one can show that there are $O(2^{32h^3})$ configurations near the $h+1$ vertices such that knowing the configuration of the drawing suffices to merge multiple drawings together efficiently, and hence allow dynamic programming. We refer to [DFK+08] for details. □

The resulting algorithm is theoretically interesting, but impractably slow. There are some faster algorithms for testing whether a graph can be drawn on 2 or 3 lines [CSW04].