Crossing Number for Graphs with Bounded Pathwidth[*,†]

Therese Biedl[‡1], Markus Chimani[§2], Martin Derka[¶3], and Petra Mutzel[‖4]

1 **Dept. of Computer Science, University of Waterloo, Canada**
`biedl@uwaterloo.ca`
2 **Dept. of Computer Science, Universität Osnabrück, Germany**
`markus.chimani@uni-osnabrueck.de`
3 **Dept. of Computer Science, University of Waterloo, Canada**
`mderka@uwaterloo.ca`
4 **Dept. of Computer Science, Technische Universität Dortmund, Germany**
`petra.mutzel@cs.tu-dortmund.de`

Abstract

The crossing number is the smallest number of pairwise edge crossings when drawing a graph into the plane. There are only very few graph classes for which the exact crossing number is known or for which there at least exist constant approximation ratios. Furthermore, up to now, general crossing number computations have never been successfully tackled using bounded width of graph decompositions, like treewidth or pathwidth.

In this paper, we for the first time show that crossing number is tractable (even in linear time) for maximal graphs of bounded pathwidth 3. The technique also shows that the crossing number and the rectilinear (a.k.a. straight-line) crossing number are identical for this graph class, and that we require only an $O(n) \times O(n)$-grid to achieve such a drawing.

Our techniques can further be extended to devise a 2-approximation for general graphs with pathwidth 3, and a $4\mathbf{w}^3$-approximation for maximal graphs of pathwidth $\mathbf{w}$. This is a constant approximation for bounded pathwidth graphs.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Crossing Number, Graphs with Bounded Pathwidth

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2017.13

1 Introduction

The crossing number $cr(G)$ is the smallest number of pairwise edge-crossings over all possible drawings of a graph G into the plane. Despite decades of lively research, see e.g. [25, 26], even most seemingly simple questions, such as the crossing number of complete or complete bipartite graphs, are still open, cf. [23]. There are only very few graph classes, e.g., Petersen graphs $P(3, n)$ or Cartesian products of small graphs with paths or trees, see [4, 20, 24], for which the crossing number is known or can be efficiently computed.

* This work has been started at the Crossing Number Workshop 2016 in Strobl (Austria)
† A full version of the paper is available at [1], `https://arxiv.org/abs/1612.03854`.
‡ Research of T.B. supported by NSERC.
§ Research of M.C. partially supported by the German Science Foundation (DFG), project CH 897/2-1.
¶ Research of M.D. supported by NSERC Vanier CSG.
‖ Research of P.M. partially supported by the DFG within the SFB 876 (project A6).

Considering approximations, we know that computing $cr(G)$ is APX-hard [5], i.e., there does not exist a PTAS (unless P = NP). The best known approximation ratio for general graphs with bounded maximum degree is $\tilde{O}(n^{0.9})$ [10]. We only know constant approximation ratios for special graph classes. In fact, all known constant approximation ratios are based on one of three concepts: *Topology-based* approximations require that G can be embedded without crossings on a surface of some fixed or bounded genus [13, 16, 17]. *Insertion-based* approximations assume that there is only a small (i.e., bounded size) subset of graph elements whose removal leaves a planar graph [6–9]. In either case, the ratios are constant only if we further assume bounded maximum degree. Finally, some approximations for the crossing number exist if the graph is *dense* [12].

While treewidth and pathwidth have been very successful tools in many graph algorithm scenarios, they have only very rarely been applied to crossing number: Since general crossing number seems not to be describable with second order monadic logic, Courcelle's result [11] regarding treewidth-based tractability can only be applied if cr itself is bounded [14, 18]. The related strategy of "planar decompositions" lead to linear crossing number bounds [27].

Contribution. In this paper, we for the first time show that such graph decompositions, in our case pathwidth, *can* be used for computing crossing number. We show for maximal graphs G of pathwidth 3 (see Section 3):

- We can compute the *exact* crossing number $cr(G)$ in linear time.
- The topological $cr(G)$ equals the *rectilinear* crossing number $\overline{cr}(G)$, i.e., the crossing number under the restriction that all edges need to be drawn as straight lines.
- We can compute a drawing realizing $\overline{cr}(G)$ on an $O(n) \times O(n)$-grid.

We then generalize these techniques to show:

- A 2-approximation for $cr(G)$ and $\overline{cr}(G)$ for general graphs of pathwidth 3, see Section 4.
- A $4\mathbf{w}^3$-approximation for $cr(G)$ for maximal graphs of pathwidth $\mathbf{w}$, see Section 5. This can be achieved by placing vertices and bend points on a $4n \times \mathbf{w}n$ grid.

Observe that in contrast to most previous results, these approximation ratios are *not* dependent on the graph's maximum degree. As a complementary side note, we show (in the full version of the paper, see [1]) that the *weighted* (possibly rectilinear) crossing number is weakly NP-hard already for maximal graphs with pathwidth 4.

Focusing on graphs with bounded pathwidth may seem very restrictive, but in some sense these are the most interesting graphs for crossing minimization because Hliněný showed that crossing-number critical graphs have bounded pathwidth [15].

2 Preliminaries

We always consider a simple undirected graph G with n vertices as our input. A drawing of G is a mapping φ of vertices and edges to points and simple curves in the plane, respectively. The curve $\varphi(e)$ of an edge $e = (u, v)$ does not pass through any point $\varphi(w)$, $w \in V(G)$, but has its ends at $\varphi(u)$ and $\varphi(v)$. When asking for a crossing minimum drawing of G, we can restrict ourselves to *good* drawings, which means that adjacent edges do not cross, non-adjacent edges cross at most once, and no three edges cross at the same point of the drawing. For other drawings, straightforward redrawing arguments, see e.g. [25], show that the crossing number can never increase when establishing these properties.

A *clique* is a complete graph and a *biclique* is a complete bipartite graph. While the exact crossing number is unknown for general cliques and bicliques, there are upper bound constructions, conjectured to attain the optimal value. In particular the old construction due to Zarankiewicz, attaining $\lfloor\frac{n_1}{2}\rfloor\lfloor\frac{n_1-1}{2}\rfloor\lfloor\frac{n_2}{2}\rfloor\lfloor\frac{n_2-1}{2}\rfloor$ crossings for K_{n_1,n_2}, is known to give the optimum for $n_1 \le 6$ [19].

A prominent variant of the traditional ("topological") crossing number $cr(G)$ is the *rectilinear* crossing number $\overline{cr}(G) \ge cr(G)$, sometimes also known as geometric or straight-line crossing number. Thereby, edges are required to be drawn as straight line segments without any bends. Interestingly, while we know $\overline{cr}(G) > cr(G)$ in general (e.g., already for complete graphs), Zarankiewicz's construction is a straight-line drawing, suggesting that maybe $cr(G) = \overline{cr}(G)$ for bicliques.

Alternating path decompositions and clusters. There are several equivalent definitions of pathwidth; we use here the one based on tree decompositions, see e.g. [21]. A *path decomposition* $\mathcal{P}$ of a connected graph G consists of a finite set of *bags* $\{X_i \mid 1 \le i \le \xi \in \mathbb{N}\}$, where each bag is a subset of the vertices of G, such that for every edge (v, w) at least one bag contains both v and w, and for every vertex v of G the set of bags containing v forms an interval (i.e., the underlying graph formed by the bags is a path). The indexing of the bags gives a total ordering and we may speak of *first*, *last*, *preceding*, and *succeeding* bags. The *width* of a path decomposition is the maximum cardinality of a bag minus one, i.e., $\max_{1\le i\le\xi} |X_i|-1$. The *pathwidth* $\mathbf{w} := \mathbf{w}(G)$ of G is the smallest width that can be achieved by a path decomposition of G. A *maximal pathwidth-*$\mathbf{w}$ *graph* is a graph of pathwidth $\mathbf{w}$ for which adding any edge increases its pathwidth. In particular, this implies that the vertices in each bag form a clique. We assume that $n > \mathbf{w} + 1$; otherwise G is a clique and the crossing number is 0 for $\mathbf{w} = 3$ and easily approximated within a factor of $O(1)$ for bigger $\mathbf{w}$ (e.g., via the crossing lemma [22]).

Several additional constraints can be imposed on the bags and the path decomposition without affecting the required width. We use a variant of a *nice* path decomposition that we call an *alternating* path decomposition (see Fig. 1); one can easily show that such a decomposition exists:

- There are exactly $\xi = 2n - 2\mathbf{w} - 1$ bags.
- $|X_i| = \mathbf{w} + 1$ if i is odd and $|X_i| = \mathbf{w}$ if i is even.
- For any even $1 < i < \xi$, we have $X_{i-1} \supset X_i \subset X_{i+1}$.

Note that for any odd i there is exactly one vertex v that is in X_i but not in bag X_{i+1}. We say that v is *forgotten* by bag X_{i+1}. Similarly, bag X_i contains exactly one vertex v that was not in bag X_{i-1}. We say that v is *introduced* by bag X_i. We define the *age-order* $\{v_1, \ldots, v_n\}$ of the vertices of G as follows: v_1 is forgotten by X_2; $v_2, \ldots, v_{\mathbf{w}+1}$ are the other vertices of bag X_1 in arbitrary order. The order of the remaining vertices corresponds to the order of the bags by which they are introduced. We say that v_i is *older* than v_j if $i < j$, so the three oldest vertices are v_1, v_2, v_3. Note that we can choose v_2, v_3 arbitrarily among $X_1 - \{v_1\}$. In particular, if two vertices $p, q \in X_1$ are specified, then we can ensure that they are among the three oldest; this will be exploited in Section 4.2.

In our algorithms and proofs, we will work with special subsets of bags called *clusters*. Let G be a connected graph of pathwidth 3 with an alternating path decomposition $\mathcal{P} = \{X_i\}_{1\le i\le\xi}$. Consider a set of three vertices Y that constitute at least one bag (this bag has an even index). There can be several such bags with exactly those vertices, but all bags containing Y are consecutive. For any such Y, we define a *cluster* C as the maximal consecutive set of bags that all contain Y. We say that $T(C) := Y$ is the *anchor-triplet* of C. Any cluster has at least 3 bags. They alternate between size 4 and 3, starting and ending with size-4 bags. Two consecutive clusters overlap in exactly one bag (which consequently has size 4). The order of the bags induces a unique order of the clusters $\{C_1, \ldots, C_\kappa\} =: \mathcal{C}$.

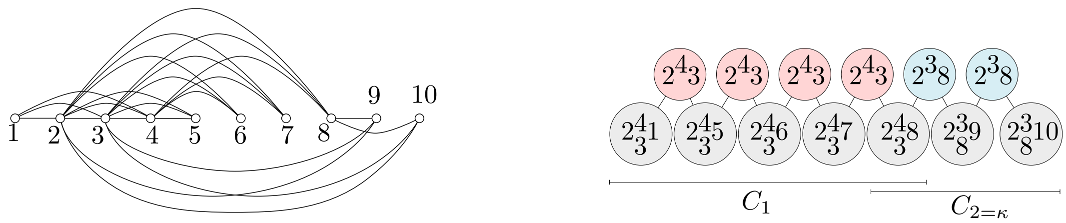


Figure 1 (left) A graph, with vertices in age order according to $\mathcal{P}$. (right) Its alternating path decomposition $\mathcal{P}$ of width 3, with two clusters: C_1 has $T(C_1) = \{2, 3, 4\}$, and consists of all bags containing this anchor-triplet. Analogously, we have $T(C_2) = \{2, 3, 8\}$. In C_1, the lost vertex is $x_1^- = 1$ and the emerging vertex is $x_1^+ = 8$.

Note that a cluster C can be described as a set of bags, or by its anchor-triplet. Denote the vertices that appear in the union of bags of C by $V(C)$, and let $n(C) := |V(C)|$. The following observation is trivial (because any vertex of the anchor-triplet of C belongs to all bags of C) but crucial for our analysis.

▶ **Observation 1.** *Let G be a maximal pathwidth-3 graph and let C be a cluster. Then the graph induced by $V(C)$ consists of the triangle induced by $T(C)$ and (edge-disjoint) a biclique $K_{3,n(C)-3}$ with one partition being $T(C)$.*

We define the *emerging vertex* of C_i, denoted by x_i^+, as the vertex introduced by the last bag of C_i. Note that x_i^+ belongs to the anchor-triplet of the next cluster C_{i+1} if $i < \kappa$. We define the *lost vertex* of C_i, denoted by x_i^-, as the vertex that was forgotten by the second bag of C_i. Note that x_i^- belongs to the anchor-triplet of the previous cluster C_{i-1} if $i > 1$, but not to the anchor-triplet of C_i. Observe that $x_1^- = v_1$, $x_\kappa^+ = v_n$, $x_{i-1}^+ \neq x_i^-$ and $T(C_i) = T(C_{i-1}) \cup \{x_{i-1}^+\} \setminus \{x_i^-\}$ for all $2 \le i \le \kappa$. For notational simplicity, we define $x_0^+ := v_2$. Any vertex x that belongs to C_i but is not in $T(C_i) \cup \{x_i^+, x_i^-\}$ is called a *singleton* of C_i. Vertex x belongs to a “middle” bag of C_i and only appears in this bag; it belongs to no cluster other than C_i. See Fig. 1 for an example.

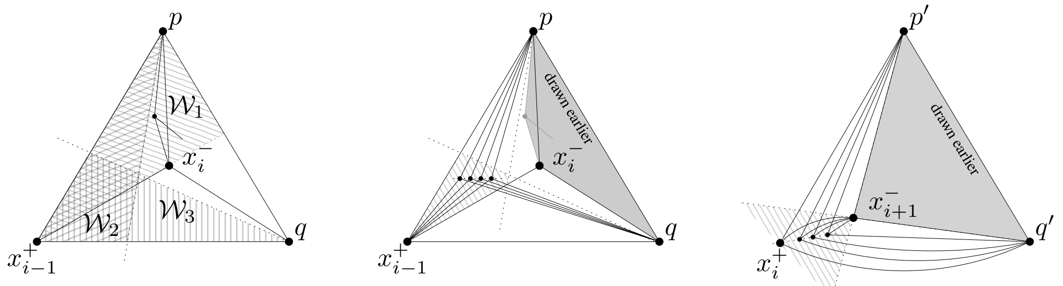
3 Exact Algorithm for Maximal Pathwidth-3 Graphs

Let G be a maximal pathwidth-3 graph and fix an alternating path decomposition of width 3. By maximality, all bags form cliques, and in particular, each anchor-triplet induces a triangle in the graph, called *anchor triangle* consisting of *anchor edges*.

The general idea to draw G is to iterate through the clusters $C_1, \ldots, C_\kappa$. When considering cluster C_i, its first bag will already be drawn and the anchor triangle will form the outer face of the current drawing. About half of the vertices introduced by C_i will be drawn inside the anchor triangle while the other half will be drawn outside, mimicking Zarankiewicz' construction locally. The number of crossings that these vertices add will be exactly the minimum number of crossings needed to draw the biclique $K_{3,n(C_i)-3}$ of cluster C_i, hence leading to an optimal drawing.

We start with drawing bag $X_1 = \{v_1, v_2, v_3, v_4\}$ as a planar drawing of K_4 with the vertices $T(C_1) = X_2 = \{v_2, v_3, v_4\}$ on the outer face. Now we iterate over all clusters C_i, $1 \le i \le \kappa$, drawing their bags with the following invariants:

- The drawing is good and straight-line.
- Before drawing C_i, the outer face contains the three vertices $T(C_i)$.
- For any $j \le i$, the anchor edges of C_j are drawn without crossings.



(a) Wedges $\mathcal{W}_1$, $\mathcal{W}_2$, and $\mathcal{W}_3$. **(b)** Adding ℓ_1 vertices inside. **(c)** Adding ℓ_2 vertices outside.

Figure 2 Drawing maximal pathwidth-3 graphs. For ease of legibility we draw some edges in (c) slightly curved. Dotted lines mark boundaries of the regions defined in the text.

Let ℓ be the number of singleton vertices in C_i (possibly $\ell = 0$). We need to place the ℓ singletons and the emerging vertex x_i^+. We will add $\ell_1 := \lfloor(\ell+1)/2\rfloor \leq \ell$ vertices into an inner face of the current drawing and $\ell_2 = \lceil(\ell+1)/2\rceil \geq 1$ vertices on the outside. Note that $\ell_1 + \ell_2 = \ell + 1$.

Placement on the inside. By the invariant the outer face consists of the edges connecting $T(C_i) = \{x_{i-1}^+, p, q\}$ for some p, q. W.l.o.g. assume that x_{i-1}^+, p, and q occur in clockwise order walking along the outer face. By maximality, and because x_{i-1}^+ has just been introduced, x_{i-1}^+ has degree 3 in the current graph, and its neighbors are p, q, x_i^-.

Let $\mathcal{R}$ be the open region obtained by the intersection of three open "wedges" $\mathcal{W}_1, \mathcal{W}_2, \mathcal{W}_3$ defined as follows: Wedge $\mathcal{W}_1$ emanates from x_{i-1}^+ between edges (x_{i-1}^+, p) and (x_{i-1}^+, x_i^-) in the interior of the triangle induced by $T(C_i)$. Wedge $\mathcal{W}_2$ ($\mathcal{W}_3$) emanates at p (q) inside of $T(C_i)$ and runs along edge (p, x_{i-1}^+) ((q, x_{i-1}^+), respectively) with a sufficiently small angle such that it crosses only edges incident to x_{i-1}^+. Any point inside $\mathcal{R}$ can be connected to all of p, q, x_{i-1}^+ with straight lines and a single crossing (with edge (x_{i-1}^+, x_i^-)).

Consider a straight line s through $\mathcal{R}$ but not through any of p, q, x_{i-1}^+. Place ℓ_1 vertices (for ℓ_1 singletons of C_i) along s within $\mathcal{R}$, and connect each of them to all of p, q, x_{i-1}^+. All generated crossings are with edge (x_{i-1}^+, x_i^-) or among the added edges. The drawing is straight-line and good (no three edges cross in a point), and the number of added crossings is $\ell_1 + \binom{\ell_1}{2} = \frac{1}{2}\ell_1(\ell_1+1)$.

Placement on the outside. The outer face of the drawing is still formed by the edges connecting $T(C_i)$, since all vertices from the paragraph above were added inside $\mathcal{R}$ and thus in the interior of $T(C_i)$. We know that the vertex x_{i+1}^- in $T(C_i)$ will be lost in the next cluster C_{i+1} (if there is any); it will play a prominent role now. Since we may or may not have $x_{i+1}^- = x_{i-1}^+$, we label the vertices of $T(C_i)$ afresh as $\{x_{i+1}^-, p', q'\}$.

Define an open wedge $\mathcal{W}$ in the exterior of $T(C_i)$ emanating from x_{i+1}^- between the extensions of the edges (p', x_{i+1}^-) and (q', x_{i+1}^-) beyond x_{i+1}^-. Any point inside $\mathcal{W}$ can be connected via straight lines to all of p', q', x_{i+1}^- without any crossings. Consider a straight line s' through $\mathcal{W}$, not through any of x_{i+1}^-, p', q', and crossing (p', q'). Now place ℓ_2 vertices along s' within $\mathcal{W}$, and connect all of them to all of x_{i+1}^-, p', q' via straight lines. All generated crossings are among the added edges. The drawing is still straight-line and good, and the number of added crossings is $\binom{\ell_2}{2}$. The outer face of the resulting drawing is again a triangle with two corners being p' and q' and the third corner being a vertex that was added on s'. We assign this latter vertex the role of the emerging vertex x_i^+; the other inserted vertices are the necessary singletons. With this, the invariant holds since $T(C_{i+1}) = T(C_i) \cup \{x_i^+\} \setminus \{x_{i+1}^-\}$.

This finishes the description of the drawing algorithm. We claim that the final drawing has the minimum possible number of crossings: We first give an upper bound on the number of crossings that we achieve, and then show that any drawing requires this number.

▶ **Lemma 2.** *The above algorithm produces at most* $\sum_{i=1}^{\kappa} \lfloor \frac{1}{2}(n(C_i) - 3) \rfloor \lfloor \frac{1}{2}(n(C_i) - 4) \rfloor$ *crossings.*

Proof. The algorithm started with a planar drawing of K_4. We argued above that the i-th iteration (drawing C_i, which contains ℓ singletons) added

$$\frac{1}{2}\ell_1(\ell_1 + 1) + \frac{1}{2}\ell_2(\ell_2 - 1) = \lfloor \frac{1}{2}(\ell + 1) \rfloor \lfloor \frac{1}{2}(\ell + 2) \rfloor$$

crossings, where $\ell_1 = \lfloor (\ell + 1)/2 \rfloor$ and $\ell_2 = \lceil (\ell + 1)/2 \rceil$. Finally, observe that $\ell = n(C_i) - 5$ since all vertices of C_i except $T(C_i) \cup \{x_i^+, x_i^-\}$ are singletons. ◀

▶ **Lemma 3.** *Any good drawing of G requires at least* $\sum_{i=1}^{\kappa} \lfloor \frac{1}{2}(n(C_i) - 3) \rfloor \lfloor \frac{1}{2}(n(C_i) - 4) \rfloor$ *crossings.*

Proof. From Observation 1 we know that each cluster C_i contains a biclique $B(C_i) := K_{3,n(C_i)-3}$. By Zarankiewicz' formula, $K_{3,m}$ needs $\lfloor m/2 \rfloor \lfloor (m-1)/2 \rfloor$ crossings in any drawing. Thus, within each cluster we only introduce the optimal number of crossings.

However, we must argue that it is impossible for one crossing to belong to two or more clusters in an optimal drawing. This holds because nearly all of $V(C_i)$ does not belong to other clusters. More precisely, assume some other cluster C_j shares vertices with C_i; we may assume $j < i$. Then all common vertices must appear in the first bag $X = T(C_i) \cup \{x_i^-\}$ of C_i. However, only three edges of those induced by X are in $B(C_i)$, and all three of them are incident to x_i^-. Since adjacent edges do not cross in a good drawing, no crossing can be shared between $B(C_i)$ and $B(C_j)$. ◀

▶ **Theorem 4.** *There is a linear time algorithm to compute the exact crossing number $cr(G)$ of any maximal pathwidth-3 graph G. Furthermore, $cr(G) = \overline{cr}(G)$, and the algorithm gives rise to a straight-line drawing where the anchor edges are not crossed.*

Proof. Optimality follows from Lemmas 2 and 3. The second part of the claim follows from the first and third invariant in the above algorithmic description. It remains to argue linear running time. Computing a path decomposition of width 3 (if it exists) can be done in linear time [2, 3]. This path decomposition can be turned into an alternating path decomposition in linear time as well. On it we compute $cr(G)$ as the sum in Lemma 2 in linear time. ◀

Assume we are interested in the drawing achieving this solution. The drawing algorithm uses $O(n)$ operations, but this does not immediately imply linear time, since coordinates may become very small. We also cannot list all crossings, as there can be $\Theta(n^2)$ many. If, however, we are careful about how to place anchor-triplets, then singletons can be inserted while keeping all vertices at grid-points of an $O(n) \times O(n)$-grid, and thus we require only linear time to compute and output the drawing. Details are given in the full version of the paper [1, Appendix B]. We summarize:

▶ **Theorem 5.** *Every maximal pathwidth-3 graph on n vertices has a crossing-minimum drawing that is good, straight-line, and lies on a $28n \times 29n$-grid. It can be found in $O(n)$ time.*

4 Approximation Algorithm for Pathwidth-3 Graphs

We now give an algorithm that draws graphs of pathwidth 3 (not necessarily maximal) such that the number of crossings is within a factor of 2 of the optimum. Roughly speaking, if the graph is 3-connected (technically, we will define a slightly weaker assumption 3-*traceable*), then the algorithm for maximal pathwidth-3 graphs is applied, and the number of crossings is within a factor of 2. If the graph is not 3-traceable, then it can be split and the arising subdrawings can be "glued" together without increasing the approximation ratio.

4.1 3-traceable graphs

We first analyze graphs that satisfy a condition that is weaker than 3-connectivity. Define a *non-anchor vertex* to be a vertex that occurs in exactly one bag. Those are exactly v_1, v_n, and all the singletons defined earlier.

▶ **Definition 6** (3-traceable graph). A graph G with an alternating path decomposition $\mathcal{P}$ of width 3 is 3-*traceable* if every non-anchor vertex has degree at least 3, and for all $1 \leq i \leq \kappa$, edge (x^+_{i-1}, x^-_i) exists.

Assume we are given a 3-traceable graph G with an alternating path decomposition $\mathcal{P}$ of width 3. We can first maximize G (obtaining G') by adding all edges that have both ends in one bag, but are not in G' yet. We then apply the algorithm described in Section 3 to G', and finally delete the temporarily added edges again. We will show:

▶ **Lemma 7.** *Let G be a 3-traceable graph. Then the algorithm of Theorem 4 gives a drawing of G with at most $2cr(G)$ crossings.*

We first give a sketch of the proof. The main challenge is that a cluster C now does not necessarily contain a biclique $K_{3,n(C)-3}$. However, we can argue that G contains a subdivision of $K_{3,n(C)-3}$ that uses mostly vertices of C, but "borrows" a non-anchor vertex each (to play the role of x^-_i and x^+_i) from the nearest preceding and succeeding cluster that has such vertices. This subdivided $K_{3,n(C)-3}$ requires $cr(K_{3,n(C)-3})$ crossings. The main work is then in arguing that these subdivided bicliques cannot overlap much, or more precisely, that any crossing can belong to at most 2 of them. Lemma 7 then follows by applying the upper bound given in Lemma 2.

As before, let $C_1, \ldots, C_\kappa$ be the clusters of G with anchor-triplets $T(C_1), \ldots, T(C_\kappa)$, and recall that we have an age-order $\{v_1, \ldots, v_n\}$.

There are three types of edges in G. Type I are edges that are incident to non-anchor vertices. Type II are edges that have the form (x^+_{i-1}, x^-_i) for some $2 \leq i \leq \kappa$. Finally, Type III are the remaining edges (they connect vertices of some anchor-triplet $T(C_i)$, $1 \leq i \leq \kappa$).

▶ **Observation 8.** *Consider a 3-traceable graph. For any $1 \leq i < j \leq \kappa$, there are three vertex-disjoint paths $\Pi_{i,j}$ from $T(C_i)$ to $T(C_j)$ that are either single vertices or consist exactly of the Type II edges (x^+_{k-1}, x^-_k) for $i < k \leq j$. Every non-anchor vertex attaches to the three different paths $\Pi := \Pi_{1,\kappa}$.*

Proof. For any $1 \leq i < \kappa$, we have $T(C_{i+1}) = T(C_i) \cup \{x^+_i\} \setminus \{x^-_{i+1}\}$. By 3-traceability of G, edge (x^-_{i+1}, x^+_i) exists and $\Pi_{i,i+1}$ consists of two paths of length 0 (the common vertices of the triplets) and the third path being this edge. We obtain arbitrary $\Pi_{i,j}$ by extending $\Pi_{i,i+1}$ via $\Pi_{i+1,j}$. Since G is 3-traceable, the non-anchor vertices have degree 3 and are adjacent to the vertices of the anchor-triplet of their unique cluster; those lie on distinct paths of Π. ◀

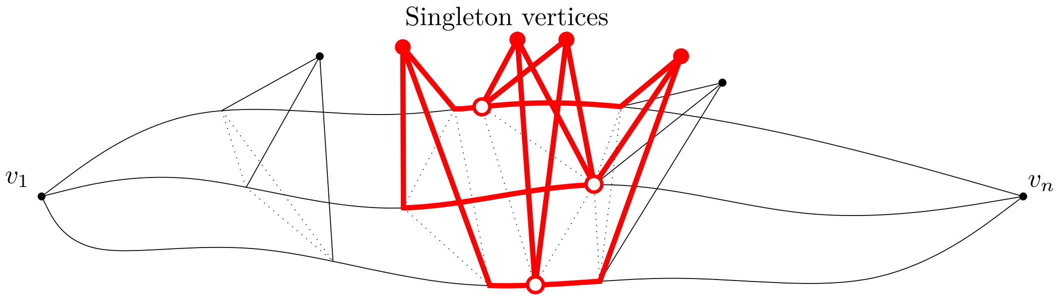


Figure 3 The structure of a 3-traceable graph. Dotted triangles mark anchor-triples with at least one adjacent singleton. In bold, we show one cluster biclique: the anchor vertices depicted as circles form one partition side. The left- and rightmost bold singleton is "borrowed" from the preceding and succeeding singleton-containing cluster, respectively.

This shows that G has $K_{3,n'}$ as a minor, where n' is the number of non-anchor vertices. Unfortunately this is not sufficient for crossing number arguments as contracting edges may increase the crossing number. Instead, we will use the above structure to extract a subdivision of $K_{3,n(C)-3}$ for each cluster C in such a way that these bicliques do not overlap "much."

▶ **Definition 9.** Let C_i, $1 \leq i \leq \kappa$, be a cluster with at least one singleton. The *cluster biclique of* C_i, denoted $\mathcal{B}(C_i)$, is a subdivision of $K_{3,n(C_i)-3}$ obtained as follows, cf. Fig. 3:

(a) The 3-side is formed by the three vertices of $T(C_i)$.

(b) Every singleton w that belongs to C_i (there are $n(C_i) - 5$ of them) is one of the vertices on the side that will have $n(C_i) - 3$ vertices. We know that $\deg(w) = 3$ by 3-traceability, and it is adjacent to all of $T(C_i)$ as required for the biclique.

(c) Let $i_- < i$ ($i_+ > i$) be maximal (minimal) such that cluster C_{i_-} (C_{i_+}, respectively) has a non-anchor vertex; among its non-anchor vertices, let w_- (w_+) be the youngest (oldest, respectively). If $i = 1$, we simply set $w_- := v_1$; if $i = \kappa$, we set $w_+ := v_n$. By Observation 8, we can establish three disjoint paths from w_- and w_+ to $T(C_i)$. Hence, add w_- and w_+ to the "big" side of $\mathcal{B}(C_i)$. Observe that in either case, w_- and w_+ are distinct from the the singletons of C_i and their paths to $T(C_i)$.

▶ **Lemma 10.** *Let e_1, e_2 be two edges of G without common endpoint. There are at most two cluster bicliques that contain both e_1 and e_2.*

Proof. We are done if at least one of e_1 and e_2 is of Type III, because then it belongs to no cluster biclique at all. Assume that one of e_1 and e_2 is of Type II, say $e_1 = (x^+_{i-1}, x^-_i)$ for some $2 \leq i \leq \kappa$. Edge e_1 may be used only for the cluster bicliques $\mathcal{B}(C_{j^-})$ and $\mathcal{B}(C_{j^+})$ where $j^- < i$ ($j^+ \geq i$) is the maximal (minimal) index such that cluster C_{j^-} (C_{j^+}, respectively) has singletons. The fact that e_1 belongs to at most two cluster bicliques proves the claim.

Finally, assume that both e_1 and e_2 are of Type I, i.e., incident to distinct non-anchor vertices, say $y_1 \in C_i$ and $y_2 \in C_{i'}$. Let $\mathcal{C}' \subseteq \mathcal{C}$ be the ordered subsequence of clusters that have at least one non-anchor vertex. A non-anchor vertex x can belong to at most three cluster bicliques, refer to Definition 9: the one of its "own" cluster $C \in \mathcal{C}'$, and those of the directly preceding and succeeding cluster in $\mathcal{C}'$. Assume that y_1 and y_2 are in three cluster bicliques. If $i = i'$, y_1 and y_2 are singletons of different age in C_i, and the two clusters directly preceding and succeeding C_i would have chosen distinct singletons of C_i, a contradiction. If $i \neq i'$, any overlap of three-element subsequences of $\mathcal{C}'$ with distinct middle clusters has size at most 2, a contradiction. ◀

Proof of Lemma 7. We know from Lemma 2 that the algorithm of Theorem 4 gives a drawing with at most $\sum_{C\in\mathcal{C}}\lfloor\frac{1}{2}(n(C)-3)\rfloor\lfloor\frac{1}{2}(n(C)-4)\rfloor$ crossings. We need to consider only clusters C that have at least one singleton; for any other cluster we have $n(C)=5$ and therefore its summand is 0. For any cluster C that has a singleton, we have $\mathcal{B}(C)$, a subdivision of $K_{3,n(C)-3}$, which requires at least $\lfloor\frac{1}{2}(n(C)-3)\rfloor\lfloor\frac{1}{2}(n(C)-4)\rfloor$ crossings in any good drawing $\mathcal{D}$ of G. Any crossing in $\mathcal{D}$ is created by two edges without common endpoints, and by Lemma 10, any such pair belongs to at most two cluster bicliques. Hence any drawing of G has at least $\frac{1}{2}\sum_{C\in\mathcal{C}}\lfloor\frac{1}{2}(n(C)-3)\rfloor\lfloor\frac{1}{2}(n(C)-4)\rfloor$ crossings, yielding the 2-approximation. ◀

4.2 General pathwidth-3 graphs

A pair of vertices $\{u,v\}$ of a 2-connected graph G is called a *separation pair* if $G-\{u,v\}$ is not connected. Assume that the pathwidth-3 graph G is 2-connected but not 3-traceable. We will show that we can split the graph at separation pairs within anchor-triplets, draw the cut-components recursively, and merge them without introducing additional crossings. We start with a more general auxiliary statement whose proof is in [1, Appendix C].

▶ **Lemma 11.** *Let G be a 2-connected graph with a separation pair $\{u,v\}$. Consider a partition of G into two edge-disjoint connected subgraphs H_1,H_2 with $H_1\cap H_2=\{u,v\}$. Define $H_i^+=H_i\cup\{(u,v)\}$ for $i=1,2$. Then $cr(H_1^+)+cr(H_2^+)\leq cr(G)$.*

We will draw cut-components inside triangles bounded by their three oldest vertices.

▶ **Lemma 12.** *Let G be a 2-connected graph with an alternating path decomposition $\mathcal{P}$ of width 3. Then there exists an algorithm to create a straight-line drawing of G with at most $2cr(G)$ crossings. All anchor-edges are drawn without crossings, and the three oldest vertices $\{v_1,v_2,v_3\}$ form the corners of the triangular convex hull of the drawing.*

Proof. We prove the result by induction on the structure and size of the graph.

Base case: G is 3-traceable or a K_4. If $G=K_4$, the claim is obvious. Otherwise, we apply Lemma 7. However, the algorithm of Theorem 4 used therein grows the drawing "outwards", while we would now like the oldest vertices to form the outer triangle. Thus we apply the algorithm for the reverse path decomposition; this makes (by suitably placing the last vertex) $T(C_1)=\{v_1,v_2,v_3\}$ the outer face and draws it as a triangle.

Induction Step: G is neither 3-traceable nor a K_4. For every non-anchor vertex $w\neq v_1$ of degree 2, let p_w,q_w be its adjacent anchor vertices. We can temporarily remove w from G, ensure that the reduced graph contains edge (p_w,q_w), draw the reduced graph, and—since (p_w,q_w) will be drawn crossing free by the induction hypothesis—reinsert each w with $(p_w,w),(w,q_w)$ crossing-free close to the drawing of (p_w,q_w). Similarly, we can remove v_1 if it has degree 2: We can choose an age-order of the reduced graph G' such that the neighbors of v_1 are among the three oldest vertices of G' and hence draw G' such that the neighbors of v_1 are on the outer-triangle; then v_1 can be reinserted on the outside to form the desired outer triangle. If the graph became 3-traceable by these operations, we are done (base case). Otherwise, we can now assume that all non-anchor vertices have degree 3.

Since G is not 3-traceable, $(x_{i-1}^+,x_i^-)\notin G$ for some $2\leq i\leq\kappa$. There exists a unique bag X_j, the common bag of C_{i-1} and C_i, that contains both x_{i-1}^+ and x_i^-. Let p,q be the two other vertices in this bag, and observe that $T(C_{i-1})=\{p,q,x_i^-\}$ while $T(C_i)=\{p,q,x_{i-1}^+\}$. Let G_ℓ be the graph induced by all vertices that appear in bags $\mathcal{P}_\ell:=[X_1,X_{j-2}]$, and let G_r be the graph induced by all vertices that appear in bags

$\mathcal{P}_r := [X_{j+2}, X_\xi]$. Any edge of G appears in G_ℓ or G_r, since $\{x_i^-, x_{i-1}^+\}$ is the only vertex-pair that existed in bags of $\mathcal{P}$, but neither of $\mathcal{P}_\ell$ nor $\mathcal{P}_r$. Clearly, $\{p, q\}$ is a separation pair with $G_\ell \cap G_r = \{p, q\}$.
Define $G_\ell^+ = G_\ell \cup \{(p, q)\}$ and $G_r^+ = G_r \cup \{(p, q)\}$. By the addition of edge (p, q) (if it did not already exist), both graphs are 2-connected. Apply induction to G_r^+ (with path decomposition $\mathcal{P}_r$) and G_ℓ^+ (with the path decomposition $\mathcal{P}_\ell$). Since p, q belong to the first bag of $\mathcal{P}_r$, we can ensure that they are among the three oldest vertices of G_r^+. We obtain two drawings $\mathcal{D}_1^+, \mathcal{D}_2^+$ in both of which (p, q) is not crossed. We can insert (affinely transformed) $\mathcal{D}_2^+$, which has (p, q) on its bounding triangle, along (p, q) in $\mathcal{D}_1^+$ without additional crossings. Finally, we remove edge (p, q) from the resulting drawing if $(p, q) \notin E(G)$.
By induction hypothesis, $cr(\mathcal{D}_\ell^+) \leq 2cr(G_\ell^+)$ and $cr(\mathcal{D}_r^+) \leq 2cr(G_r^+)$. By Lemma 11, $cr(G_\ell^+) + cr(G_r^+) \leq cr(G)$ and since the gluing gave no new crossings, the claim follows. ◀

We are now ready to establish the theorem for general pathwidth-3 graphs.

▶ **Theorem 13.** *Let G be any pathwidth-3 graph. We have $\overline{cr}(G) \leq 2cr(G)$, and a linear time algorithm to create a good straight-line drawing of G with at most $2cr(G)$ crossings.*

Proof. (Sketch) If G is 2-connected, then the result holds by Lemma 12. It is well known that $cr(G)$ is additive over the 2-connected components of G. When gluing at cut-vertices, the cut-vertex must be on the outer face of the drawing to be inserted into the other. We can achieve this while maintaining a straight-line drawing by choosing appropriate path decompositions; see [1, Appendix D]. The running time follows as in Theorem 4. ◀

5 Approximation Algorithm for Graphs of Higher Pathwidth

We now study the crossing number of graphs that have pathwidth $\mathbf{w} \geq 4$, and are maximal within this class. We give an algorithm to draw such graphs, and show that the number of crossings in the resulting drawing is within a factor of $4\mathbf{w}^3$ of the crossing number. As opposed to Section 3, the drawings we create here are not straight-line drawings.

As before we assume that we have an alternating path decomposition $\mathcal{P} = \{X_i\}_{1 \leq i \leq \xi}$ of width $\mathbf{w}$. We again use the *age-order* $\{v_1, \ldots, v_n\}$ of the vertices of G. Define G_i to be the graph induced by vertices $v_1, \ldots, v_i$, and use $\deg_{G_i}(v)$ to denote the number of neighbors that v has within graph G_i. For any $1 \leq i \leq n$, let the *predecessors* of vertex v_i be those neighbors that are older. We will only use this concept for $i \geq \mathbf{w} + 1$, which implies that v_i has exactly $\mathbf{w}$ predecessors by maximality of G. We enumerate them as $\{p_1^i, \ldots, p_{\mathbf{w}}^i\}$ in age-order, with p_1^i the oldest.

Drawing algorithm. We create a drawing of G by starting with $G_{\mathbf{w}+1}$ (the graph induced by $v_1, \ldots, v_{\mathbf{w}+1}$) and then iteratively adding vertex v_i. We maintain the following invariants for the drawing of G_i (see also Figure 4):

- Vertex v_j is drawn at $(j, 0)$ for all $1 \leq j \leq i$.
- The drawing is contained in the half-space $\{(x, y) : x \leq i\}$.
- All vertices w in the bag introducing v_i are *bottom-visible*, i.e., the vertical ray downward from w does not intersect any edge.

We start by placing $v_1, \ldots, v_{\mathbf{w}+1}$ at their specified coordinates, and draw the edges between them as half-circles above the x-axis. This satisfies the above invariants and gives rise to $\binom{\mathbf{w}+1}{4}$ crossings since crossings are in 1-to-1-correspondence with subsets of 4 vertices.

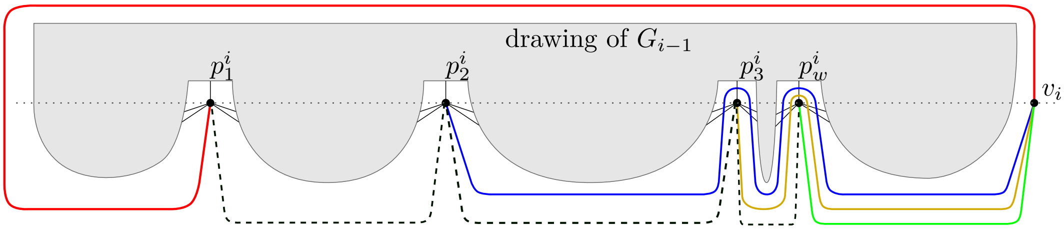


Figure 4 The construction for higher pathwidth: edge routings when adding vertex v_i.

Assume G_{i-1} is drawn and consider v_i, for $i \geq \mathbf{w}+2$. Place v_i as specified, i.e., to the right of all previous vertices and edges. Let $p_1^i, \dots, p_{\mathbf{w}}^i$ be the predecessors of v_i, all of which are bottom-visible by the invariant. We draw the edges to them using two different methods (and then redraw previous edges as a third step for each i). See also Figure 4.

- The edge to p_1^i (the oldest predecessor) is routed counterclockwise around the drawing of G_{i-1} until it is below but slightly to the left of p_1^i, from where it connects to p_1^i. We need no crossings, and all predecessors remain bottom-visible.
- All other $\mathbf{w}-1$ edges incident to v_i are routed together as a bundle from v_i leftward below the drawing of G_{i-1}. This allows v_i to be bottom-visible. Whenever the bundle is slightly to the right of some p_k^i, $\mathbf{w} \geq k \geq 2$, one of the bundle's lines (the lowest one) connects to p_k^i. The remaining bundle lines go counterclockwise around p_k^i, in its direct vicinity, until they are to the left of p_k^i and below G_{i-1}. The bundle hence crosses every edge incident to p_k^i in G_{i-1}, but no other edges, and p_k^i remains bottom-visible. This drawing scheme continues until the last bundle line connects to p_2^i.
- Finally, we redraw the edges (p_{k-1}^i, p_k^i) for $3 \leq k \leq \mathbf{w}$; they exist by maximality. Both ends of any such edge are bottom-visible, so we can redraw it without crossing below the entire drawing, including the newly drawn edges from v_i. We remove the previous drawings of these edges and retain bottom-visibility of the vertices in the current bag.

In the full paper [1, Appendix E] we analyze of the number of crossings and obtain:

▶ **Theorem 14.** *Let G be a maximal graph of pathwidth $\mathbf{w} \geq 4$. The described algorithm runs in linear time and finds a drawing of G with at most $2(\mathbf{w}-1)(\mathbf{w}-2)(2\mathbf{w}-4)cr(G) \leq 4\mathbf{w}^3 cr(G)$ crossings. In particular, for any constant pathwidth $\mathbf{w}$, we have an $O(1)$-approximation of the crossing number. The drawing is poly-line on a $4n \times \mathbf{w}n$ grid.*

6 Conclusions and Open Questions

We have shown that the path decomposition of a graph can be used to efficiently compute or bound the crossing number of a graph. This is the first successful use of such graph decomposition for crossing numbers (besides the use of a tree decomposition in the special case that $cr(G)$ is bounded by a constant [14, 18]). Several interesting questions remain:

- Can we attain stronger approximation results for general pathwidth-3 graphs? The proven ratio of 2 may simply be due to a too weak lower bound, and we, in fact, do currently not know an instance where the algorithm does not obtain the optimum.
- Can we approximate $cr(G)$ for arbitrary (not maximal) pathwidth-$\mathbf{w}$-graphs?
- In [1] we only showed weak NP-completeness for the weighted crossing number version on pathwidth-restricted graphs. Can this be strengthened to unweighted graphs?

Finally, there is of course the question whether we can use the stronger tool of tree decompositions, instead of path decompositions, to achieve crossing number results.

References

1 T. Biedl, M. Chimani, M. Derka, and P. Mutzel. Crossing number for graphs with bounded pathwidth. *CoRR*, abs/1612.03854, 2016. URL: http://arxiv.org/abs/1612.03854.

2 H.L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996. doi:10.1137/S0097539793251219.

3 H.L. Bodlaender and T. Kloks. Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *J. Algorithms*, 21(2):358–402, 1996. doi:10.1006/jagm.1996.0049.

4 D. Bokal. On the crossing numbers of cartesian products with paths. *J. Comb. Theory Ser. B*, 97(3):381–384, May 2007. doi:10.1016/j.jctb.2006.06.003.

5 S. Cabello. Hardness of approximation for crossing number. *Discrete & Computational Geometry*, 49(2):348–358, 2013. doi:10.1007/s00454-012-9440-6.

6 S. Cabello and B. Mohar. Crossing number and weighted crossing number of near-planar graphs. *Algorithmica*, 60(3):484–504, 2011. doi:10.1007/s00453-009-9357-5.

7 M. Chimani and P. Hliněný. A tighter insertion-based approximation of the crossing number. *Journal of Combinatorial Optimization*, pages 1–43, 2016. doi:10.1007/s10878-016-0030-z.

8 M. Chimani and P. Hliněný. Inserting multiple edges into a planar graph. In *SoCG 2016*, pages 30:1–30:15. LIPIcs, 2016. doi:10.4230/LIPIcs.SoCG.2016.30.

9 M. Chimani, P. Hliněný, and P. Mutzel. Vertex insertion approximates the crossing number for apex graphs. *European Journal of Combinatorics*, 33:326–335, 2012.

10 J. Chuzhoy. An algorithm for the graph crossing number problem. In *STOC '11*, pages 303–312. ACM, 2011.

11 B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

12 J. Fox, J. Pach, and A. Suk. Approximating the rectilinear crossing number. In *GD 2016*, LNCS 9801, pages 413–426. Springer, 2016. doi:10.1007/978-3-319-50106-2_32.

13 I. Gitler, P. Hliněný, J. Leanos, and G. Salazar. The crossing number of a projective graph is quadratic in the face-width. *Electronic Journal of Combinatorics*, 15(1):#R46, 2008.

14 M. Grohe. Computing crossing numbers in quadratic time. *J. Comput. Syst. Sci.*, 68(2):285–302, 2004.

15 P. Hliněný. Crossing-number critical graphs have bounded path-width. *J. Comb. Theory, Ser. B*, 88(2):347–367, 2003. doi:10.1016/S0095-8956(03)00037-6.

16 P. Hliněný and M. Chimani. Approximating the crossing number of graphs embeddable in any orientable surface. In *SODA '10*, pages 918–927, 2010.

17 P. Hliněný and G. Salazar. Approximating the crossing number of toroidal graphs. In *ISAAC '07*, LNCS 4835, pages 148–159. Springer, 2007.

18 K-I. Kawarabayashi and B. Reed. Computing crossing number in linear time. In *STOC '07*, pages 382–390, 2007.

19 D.J. Kleitman. The crossing number of $K_{5,n}$. *J. of Comb. Theory*, 9(4):315–323, 1970. doi:10.1016/S0021-9800(70)80087-4.

20 M. Klešč and J. Petrillová. The crossing numbers of products of path with graphs of order six. *Discussiones Mathematicae Graph Theory*, 33(3):571–582, 2013.

21 T. Kloks. *Treewidth, Computations and Approximations.* LNCS 842. Springer, 1994. doi:10.1007/BFb0045375.

22 F.T. Leighton. *Complexity Issues in VLSI: Optimal Layouts for the Shuffle-exchange Graph and Other Networks.* MIT Press, Cambridge, MA, USA, 1983.

23 S. Pan and R.B. Richter. The crossing number of K_{11} is 100. *Journal of Graph Theory*, 56(2):128–134, 2007. doi:10.1002/jgt.20249.

24 R.B. Richter and G. Salazar. The crossing number of $P(N,3)$. *Graphs and Combinatorics*, 18(2):381–394, 2002. doi:10.1007/s003730200028.

25 M. Schaefer. The graph crossing number and its variants: A survey. *Electronic Journal of Combinatorics*, #DS21, May 15, 2014.

26 I. Vrt'o. Crossing numbers of graphs: A bibliography. `ftp://ftp.ifi.savba.sk/pub/imrich/crobib.pdf`, 2014.

27 D.R. Wood and J.A. Telle. Planar decompositions and the crossing number of graphs with an excluded minor. *New York J. Math.*, 13:117–146, 2007.