# An Improved Fixed-Parameter Algorithm for One-Page Crossing Minimization

**Yasuaki Kobayashi**[1], **Hiromu Ohtsuka**[2], **and Hisao Tamaki**[3]

1 **Kyoto University, Kyoto, Japan**
`kobayashi@iip.ist.i.kyoto-u.ac.jp`
2 **Meiji University, Kanagawa, Japan**
`ohtsuka_yumecs.meiji.ac.jp`
3 **Meiji University, Kanagawa, Japan**
`tamaki@cs.meiji.ac.jp`

**Abstract**

*Book embedding* is one of the most well-known graph drawing models and is extensively studied in the literature. The special case where the number of pages is one is of particular interest: an embedding in this case has a natural circular representation useful for visualization and graphs that can be embedded in one page without crossings form an important graph class, namely that of outerplanar graphs.

In this paper, we consider the problem of minimizing the number of crossings in a one-page book embedding, which we call *one-page crossing minimization*. Here, we are given a graph $G$ with $n$ vertices together with a non-negative integer $k$ and are asked whether $G$ can be embedded into a single page with at most $k$ crossings. Bannister and Eppstein (GD 2014) showed that this problem is fixed-parameter tractable. Their algorithm is derived through the application of Courcelle's theorem (on graph properties definable in the monadic second-order logic of graphs) and runs in $f(L)n$ time, where $L = 2^{O(k^2)}$ is the length of the formula defining the property that the one-page crossing number is at most $k$ and $f$ is a computable function without any known upper bound expressible as an elementary function. We give an explicit dynamic programming algorithm with a drastically improved running time of $2^{O(k \log k)}n$.

**1998 ACM Subject Classification** F.2.2 Nonnumerical Algorithms and Problems, G.2.2 Graph Theory

**Keywords and phrases** Book Embedding, Fixed-Parameter Tractability, Graph Drawing, Tree-width

**Digital Object Identifier** 10.4230/LIPIcs.IPEC.2017.25

## 1 Introduction

In *book embeddings*, a graph is drawn in such a way that the vertices are aligned on a straight line, called the *spine*, as distinct points and each edge is drawn as a semicircle in a half plane defined by the spine. We call this half plane a *page*. In general, multiple pages are required to draw a graph without introducing any edge crossings, where a *crossing* is defined by a pair of edges that has a non-empty intersection distinct from their end vertices. The minimum number of pages we need to draw a graph without edge crossings, called *page number* or *book thickness*, is extensively studied in the literature (e.g. [3, 23]). The problem of computing page number is known to be NP-hard. More precisely, deciding if a given graph can be drawn in two pages without any crossing is NP-complete [7].

An optimization problem with an objective function different from the crossing number has also been studied. In this problem, given a graph $G$ and a small integer $p$, the objective

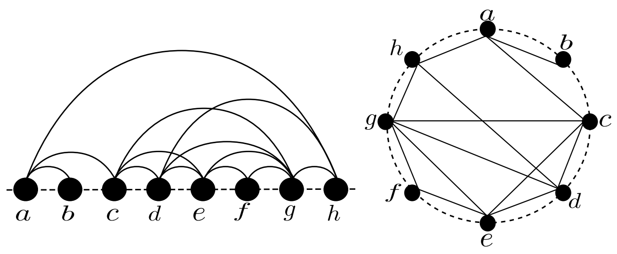

**Figure 1** A one-page drawing and a circular drawing of a graph.

is to minimize the total number of crossings in a drawing of $G$ with at most $p$ pages. This optimization problem is known as *p-page crossing minimization* and is introduced by Shahrokhi et al. [20]. Since the problem of deciding whether the page number of a graph is at most two is NP-complete [7], two-page crossing minimization is NP-hard. Indeed, the simplest case of one-page crossing minimization is already interesting. One-page crossing number is studied under various names such as *circular crossing number*, *convex crossing number*, and *outerplanar crossing number*. See [15] for example. In circular drawings, each vertex is placed on the circumference of a circular disk and each edge is drawn inside of the disk as a straight line segment. A one-page drawing may be turned into a circular drawing by topologically mapping the spine into a circle, identifying the positive and negative infinities, and mapping the half-plane into the disc enclosed by the circle. (See Figure 1, for an example). This mapping clearly preserves crossings.

One-page drawing (and hence circular drawing) with fewer crossings is important in several fields. This drawing style is frequently studied in the graph drawing community. Some well-known graph drawing software such as Graphviz[1] and yFiles[2] can produce good circular drawings. Blin et al. [4] suggested to use one-page crossing number for computing a similarity of mRNA sequences that have some secondary structures. In their paper, they asked for a fast algorithm to compute a one-page drawing with the fewest crossings.

Unfortunately, the problem of computing one-page crossing number is NP-hard as shown by Masuda et al. [18]. There are some results for special graph classes [12, 13] and some heuristics [2, 17, 21]. Bannister and Eppstein [1] tackled this problem from the perspective of parameterized complexity. They showed that the treewidth of graphs with at most $k$ crossings is $O(\sqrt{k})$ and that the graph property of having at most $k$ crossings can be defined by a formula of length $L = 2^{O(k^2)}$ in monadic second-order logic of graphs. These results are sufficient for the application of Courcelle's theorem [8, 9] to obtain an $f(L)n$ time algorithm for deciding if the one-page crossing number of a given graph is at most $k$, where $n$ is the number of vertices and $f$ is a computable function without any known upper bound expressible as an elementary function [11].

This situation is in contrast to that in the related research area of layered graph drawings, where vertices are placed on $h$ parallel lines and each edge is drawn as a straight line segment between two adjacent parallel lines. Dujmović et al. [10] gave an explicit $2^{O(h+k)^3}n$ time dynamic programming algorithm based on path-decompositions to decide if a given graph has $h$-layer drawing with at most $k$ crossings.

In this paper, we give an explicit tree decomposition based dynamic programming algorithm for one-page crossing minimization.

[1] `http://www.graphviz.org/`
[2] `https://www.yworks.com/`

▶ **Theorem 1.** *There is an algorithm which, given a graph $G$ and a non-negative integer $k$, decides whether $G$ has a one-page drawing with at most $k$ crossings in $2^{O(k \log k)}n$ time, where $n$ is the number of vertices of $G$. Moreover, if the answer is affirmative, the algorithm produces an optimal one-page drawing within the same running time.*

We would like to mention that this and the algorithm of Bannister and Eppstein [1] run in linear time, which generalize linear time algorithms recognizing and drawing outerplanar graphs [19, 22].

We borrow two tools from Bannister and Eppstein [1]. One is the upper bound on the treewidth of a graph of one-page crossing number $k$ mentioned above. The other is the concept of "crossing diagrams" which are used to classify YES-instances of the decision problem. For each such diagram, they construct a formula of length $k^{O(1)}$ for recognizing YES-instances conforming to the diagram and then take a disjunction of the formulas for all the diagrams. There are $2^{O(k^2)}$ crossing diagrams and therefore the total formula length is $2^{O(k^2)}$. We use a similar structure, which we call a "sketch", in our dynamic programming algorithm and obtain an upper bound of $2^{O(k \log k)}$ on the number of sketches through a similar but indeed somewhat finer analysis (see the proof of Lemma 14). We, however, remark that there is a fundamental difference between crossing diagrams and sketches. A crossing diagram represents a "type" of YES-instances and, for each fixed type, we need to examine each given instance for acceptance by a formula or an algorithm. On the other hand, a sketch is a succinct summary of a drawing of a subgraph. We define the "validity" of a sketch in such a way that a valid sketch of the entire graph is an immediate certificate for the positive answer to the instance and the validity of sketches can be efficiently determined by dynamic programming on a tree decomposition. We also remark that such a succinct representation is made possible by our observation on one-page drawings of biconnected graphs, which we call the chain lemma (see Section 2).

## 2 Preliminaries

Let $G$ be a graph. The set of vertices of $G$ is denoted by $V(G)$ and the set of edges of $G$ by $E(G)$. For each $v \in V(G)$, $N_G(v)$ denotes the set of neighbors of $v$ in $G$: $N_G(v) = \{u \in V(G) \mid \{u, v\} \in E(G)\}$. For $X \subseteq V(G)$, $G[X]$ denotes the subgraph of $G$ induced by $X$ and $N_G(X) = (\bigcup_{v \in X} N_G(v))$ denotes the set of neighbors of $X$.

As we have mentioned, a one-page drawing and a circular drawing are equivalent for our purposes. Therefore, we will work on circular drawings, and whenever we refer to drawings, we always refer to circular drawings. For a drawing $D$ of $G$, we write $\mathrm{cr}(D)$ to denote the number of crossings in $D$. The *one-page crossing number* $\mathrm{cr}_1(G)$ of $G$ is the minimum integer $k$ such that $G$ has a circular drawing of $k$ crossings.

Let $D$ be a drawing of $G$. We write $V(D)$ and $E(D)$ to denote $V(G)$ and $E(G)$, respectively. We denote by $\mathrm{cycle}(D)$ the cycle on $V(D)$ induced by the circle on which the vertices are drawn: two vertices are adjacent to each other in $\mathrm{cycle}(D)$ if they are consecutively placed on this circle. As special cases, if $V(D)$ is empty then $\mathrm{cycle}(D)$ is an empty graph; if $V(D)$ is a singleton, then $\mathrm{cycle}(D)$ is a self-loop; if $|V(D)| = 2$, then $\mathrm{cycle}(D)$ is a multigraph consisting of two parallel edges between the two vertices. Note that $D$ is essentially determined by $V(D)$, $E(D)$, and $\mathrm{cycle}(D)$. For $X \subseteq V(D)$, we denote by $D|X$ the subdrawing of $D$ induced by $X$, that is, the drawing obtained from $D$ by deleting all vertices in $V(D) \setminus X$. An *extension* of $D$ is a drawing obtained by adding some vertices and edges to $D$.

▶ **Definition 2.** A *tree decomposition* of $G$ is a tree $T$ where each $t \in V(T)$ is associated with $X_t \subseteq V(G)$, called a *bag*, such that

- $\bigcup_{t \in V(T)} X_t = V(G)$,
- for each $\{u, v\} \in E(G)$, there is $t \in V(T)$ with $\{u, v\} \subseteq X_t$, and
- for each $u \in V(G)$, the subgraph of $T$ induced by $\{t \in V(T) : u \in X_t\}$ is connected.

The *width* of a tree decomposition $(T, \{X_t : t \in V(T)\})$ is the maximum size of a bag minus one. The *treewidth* of $G$ is the minimum integer $w$ such that $G$ has a tree decomposition of width $w$.

To distinguish between vertices of $G$ and those of $T$, we call the vertices of $T$ *nodes.* We assume, in the rest of the paper, that tree decompositions are rooted. For a node $t \in V(T)$, we define $V_t = \bigcup_{t' \in V(T_t)} X_{t'}$, where $T_t$ is the subtree of $T$ rooted at $t$.

▶ **Definition 3.** A tree decomposition $T$ is *nice* if for each node $t$ of $T$, exactly one of the following conditions is satisfied.

- $t$ is a leaf of $T$ with $X_t = \emptyset$,
- $t$ has exactly one child $t'$ with $X_t = X_{t'} \setminus \{v\}$ for some $v \in X_{t'}$,
- $t$ has exactly one child $t'$ with $X_t = X_{t'} \cup \{v\}$ for some $v \notin X_{t'}$, or
- $t$ has exactly two children $t_1$ and $t_2$ with $X_t = X_{t_1} = X_{t_2}$.

We respectively call nodes that satisfy the above conditions, *leaf nodes*, *forget nodes*, *introduce nodes*, and *join nodes*.

▶ **Lemma 4** ([16]). *Suppose we are given a graph $G$ and its tree decomposition of width $w$. Then, there is a nice tree decomposition of $G$ of width at most $w$ such that it has $O(wn)$ nodes, where $n$ is the number of vertices of $G$. Moreover, such a nice tree decomposition can be computed in $O(w^2 n)$ time.*

The following lemma is a well-known characterization of crossing-free drawings.

▶ **Lemma 5** (Theorem 2.5 in [3]). *For every graph $G$, $\mathrm{cr}_1(G) = 0$ if and only if $G$ is outerplanar.*

Since every outerplanar graph has treewidth at most 2, we immediately have an upper bound on the treewidth with respect to its one-page crossing number: $\mathrm{tw}(G) = O(\mathrm{cr}_1(G))$. Bannister and Eppstein [1] gave a tighter bound.

▶ **Lemma 6** (Lemma 5 in [1]). *For every graph $G$, $\mathrm{tw}(G) = O(\sqrt{\mathrm{cr}_1(G)})$.*

The following simple lemma is used in some previous results (see [1], for example).

▶ **Lemma 7.** *Let $G_1, G_2, \ldots, G_t$ be the biconnected components of $G$. Then, $\mathrm{cr}_1(G) = \sum_{1 \le i \le t} \mathrm{cr}_1(G_i)$.*

Owing to this lemma, we will henceforth assume that the given graph is biconnected. We prove below a lemma crucial in exploiting the biconnectivity in our algorithm. This lemma generalizes the Hamiltonicity of biconnected outerplanar graphs with at least three vertices [6].

Let $D$ be a drawing of $G$. A path in $G$ is a *chain* in $D$ if it is also a path in cycle($D$). We say that a vertex is *chained* in $D$ if it is an internal vertex of a chain.

▶ **Lemma 8** (Chain lemma). *Let $G$ be a biconnected graph with at least three vertices and let $D$ be a drawing of $G$. Then every vertex not incident to any crossing edge in $D$ is chained in $D$.*

**Proof.** Let $u$ be a vertex of $G$ not incident to any crossing edge. Let $v_1$ and $v_2$ be the two neighbors of $u$ in cycle$(D)$. As $G$ is biconnected, $u$ has at least two neighbors in $G$. If $u$ has no neighbor distinct from both $v_1$ and $v_2$, we are done. Suppose otherwise: $u$ has a neighbor $x$ with $x \notin \{v_1, v_2\}$. Let $P_1$ and $P_2$ be internally disjoint paths between $v_1$ and $v_2$, which exist since $G$ is biconnected. Since $\{u, x\}$ is not a crossing edge, one of the path, say $P_1$, contains $u$ and the other one, say $P_2$, contains $x$. Suppose $P_1$ does not go through edge $\{v_1, u\}$. Let $y$ be the vertex adjacent to $u$ on the subpath of $P_1$ between $v_1$ and $u$. Then, the edge $\{y, u\}$ must cross an edge in $P_2$, contradicting the assumption that $u$ is not incident to a crossing edge. Hence $P_1$ must go through edge $\{v_1, u\}$. A symmetric argument shows that $P_1$ also goes through $\{v_2, u\}$. Therefore, $u$ is chained in $D$. ◀

## 3 Colored drawings and sketches

In our dynamic programming algorithm, for each node $t$ of a given tree decomposition, we enumerate structures we call "sketches" which succinctly describe drawings of $G[V_t]$. To establish recurrences among sketches, it turns out necessary for the drawings described by sketches to carry some information on our plan on how to extend those drawings to the final drawing of $G$.

We use colors on vertices, black, white, and gray, to represent this information. A *colored drawing* of graph $H$ is a triple $(D, B, W)$ where $B$ and $W$ are disjoint subsets of $V(H)$ and $D$ is a drawing of $H$, such that every vertex incident to a crossing edge in $D$ belongs to $B$. We call the vertices in $B$ *black*, those in $W$ *white*, and all others *gray*. If $C$ is the colored drawing $(D, B, W)$, then we write $V(C)$, $E(C)$, and cycle$(C)$ to denote $V(D)$, $E(D)$, and cycle$(D)$, respectively. We use $B(C)$ and $W(C)$ to denote the set of black vertices and the set of white vertices of $C$, respectively. For $U \subseteq V(C)$, we write $C|U$ to denote the colored drawing $(D|U, B \cap U, W \cap U)$.

In the above definition, both ends of a crossing edge must be black in a colored drawing, but not vice versa: a black vertex is not necessarily incident to any crossing edge. A vertex being black rather indicates our plan that it can be incident to any crossing edges in the extension of the drawing of a subgraph of $G$ into the drawing of the entire $G$. We need some more definitions to explain the intention of the other two colors.

Let $C$ be a colored drawing. We say that $C$ *respects* $X \subseteq V(C)$ if $V(C) = B(C) \cup W(C) \cup X$ and $W(C) \cap X = \emptyset$. Note that if $C$ respects $X$ then every vertex in $X$ is either black or gray, while every vertex not in $X$ is either black or white. We will often consider a colored drawing of an induced subgraph $G[U]$ of the given graph $G$, where $U$ is accompanied with a *boundary* $X$ of $U$, a subset of $U$ such that $N_G(U \setminus X) \subseteq X$. In those situations, we will require each colored drawing of $U$ to respect $X$. This ensures that there is no edge between white vertices in $U$ and the vertices in $V(G) \setminus U$. Thus, white vertices will never be adjacent to vertices introduced in the extensions.

Let $C$ and $C'$ be colored drawings of graphs $H$ and $H'$ respectively, such that $|V(H)| = |V(H')|$. A bijection $\phi : V(H) \to V(H')$ is an *isomorphism* from $C$ to $C'$ if it is an isomorphism from $H$ to $H'$ ($u, v \in V(H)$ are adjacent to each other in $H$ if and only if $\phi(u)$ and $\phi(v)$ are adjacent to each other in $H'$), is an isomorphism of cycle$(C)$ to cycle$(C')$, and preserves the coloring ($\phi(B) = B'$, and $\phi(W) = W'$). Suppose $V(H)$ and $V(H')$ intersect each other and let $X \subseteq V(H) \cap V(H')$. We call an isomorphism $\phi$ from $C$ to $C'$ an *$X$-isomorphism* if $\phi$ fixes each vertex of $x \in X$, that is, $\phi(x) = x$. If there is an isomorphism ($X$-isomorphism) from $C$ to $C'$, then we say $C$ and $C'$ are *isomorphic* (*$X$-isomorphic*) to each other and that each of them is an *isomorph* (*$X$-isomorph*) of each other.

Let $C$ be a colored drawing. We say that $C$ is *well-formed* if every white vertex of $C$ is chained in $C$. It will turn out in Section 4 that to inductively construct drawings of $G$ (which we are assuming is biconnected), it suffices to consider only well-formed drawings.

We now define "sketches" which describe colored drawings. Let $C$ be a well-formed colored drawing. A chain in $C$ is *white* if every vertex in the chain is white. The *contraction* of $C$ is the colored drawing obtained from $C$ by contracting each maximal white chain into a single white vertex (which inherits all neighbors from the vertices of the chain). Let us note that the contraction of $C$ is unique up to isomorphism. The contraction of $C$ is well-formed since we are assuming that $C$ is well-formed. Moreover, the contraction operation preserves crossings: it does not introduce any new crossings or, since the vertices contracted are white (and hence are not incident any crossing edges), does not remove any crossings. We say that $C$ is *contracted* if it does not contain any white chain with at least two vertices and hence is the contraction of itself.

Let $X$ be a vertex set. A *sketch on* $X$ is a well-formed and contracted colored drawing that respects $X$. Let $C$ be a well-formed colored drawing of graph $H$ that respects some vertex set $X \subseteq V(H)$. Let $S$ be a sketch on $X$. We say that $S$ *describes* $C$ if $S$ is $X$-isomorphic to the contraction of $C$.

## 4 Recurrences

In this section, we fix a nice tree decomposition $T$ of $G$, and use the notation $X_t$ and $V_t$, where $t$ is a node of $T$, introduced in Section 2.

We say that a sketch $S$ on $X_t$ is *valid for* $V_t$ if there is a well-formed colored drawing $C$ of $G[V_t]$ that respects $X_t$ and is described by $S$. For brevity, we say a sketch on $t$ to mean a sketch on $X_t$ and also say that a sketch is valid on $t$, or simply valid if $t$ is clear from the context, to mean that it is valid for $V_t$.

Our dynamic programming algorithm enumerates, for each $t$, all valid sketches on $t$ that are small in the sense defined later. In the following lemmas, we establish recurrences that can be used to inductively decide if a given sketch on $t$ is valid, based on the validity of sketches on $t'$ for child nodes $t'$ of $t$.

### Leaf node

▶ **Observation 9.** *Let $t$ be a leaf node of $T$. Then, the empty sketch $(D_\emptyset, \emptyset, \emptyset)$ on $X_t = \emptyset$, where $D_\emptyset$ is an empty drawing, is valid.*

### Forget node

Let $t$ be a forget node in $T$ and $t'$ the unique child of $t$. Let $v \in V_{t'}$ be the vertex forgotten: $X_t = X_{t'} \setminus \{v\}$. Let $S$ be a sketch on $X_t$ and $S'$ a sketch on $X_{t'}$. We say that $S$ is the *parent* of $S'$ if either

**F1** $v \in B(S')$ and $S = S'$ or

**F2** $v \notin B(S')$, $v$ is chained in $S'$, and $S$ is the contraction of the colored drawing obtained from $S'$ by changing the color of $v$ from gray to white (which is well-formed since $v$ is chained).

We say that $S'$ is a *child* of $S$ if $S$ is the parent of $S'$. Note that the parent of a sketch on $X_{t'}$ is unique if one exists, while a sketch on $X_t$ may have any number of children: zero, one, or more.

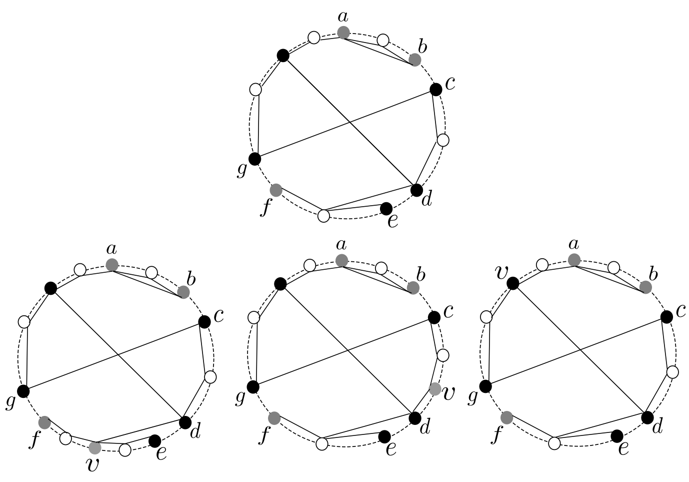

**Figure 2** The figure shows an example of a sketch on a forget node $t$. The top figure depicts a sketch on $X_t$ and the bottom figures depict some of its children. Here, $X_t = \{a, b, c, d, e, f, g\}$ and $v \in X_{t'} \setminus X_t$ is the vertex forgotten.

▶ **Lemma 10.** *Let $t$ be a forget node and $t'$ its unique child node. Then, a sketch on $t$ is valid if and only if it has a child that is valid on $t'$.*

**Proof.** Let $v$ the vertex forgotten at node $t$: $X_t = X_{t'} \setminus \{v\}$.

Let $S$ be a valid sketch on $t$. Let $C$ denote the well-formed colored drawing of $G[V_t]$ that respects $X_t$ and is described by $S$. Since $v \in V_t \setminus X_t$ and $C$ respects $X_t$, we have either $v \in B(C)$ or $v \in W(C)$.

Suppose first that $v \in B(C)$. Then, since sketch $S$ describes $C$, we have $v \in B(S)$. Then, $S$ is a sketch on $t'$ and, by Case F1, it is a child of $S$ itself. Moreover, as $W(C) \cap X_{t'} = W(C) \cap (X_t \cup \{v\}) = W(C) \cap X_t = \emptyset$, $C$ respects $X_{t'}$ as well. Since $S$ is a sketch on $t'$ and describes a colored drawing $C$ for $G[V_{t'}] = G[V_t]$ that respects $X_{t'}$, it is valid on $t'$. Therefore, we are done in this case.

Suppose next that $v \in W(C)$. Let $C'$ be the colored drawing of $G[V_{t'}] = G[V_t]$ that is identical to $C$ except that the color of $v$ is gray instead of white. Since every white vertex of $C'$ is chained as it is chained in $C$, $C'$ is well-formed. Since $W(C') \cap X_{t'} = (W(C) \setminus \{v\}) \cap (X_t \cup \{v\}) = W(C) \cap X_t = \emptyset$, $C'$ respects $X_{t'}$. Let $S'$ be the sketch on $X_{t'}$ that describes $C'$, which is unique up to $X_{t'}$-isomorphisms. Then, as $v \notin B(C')$ and the $X_{t'}$-isomorphism from $S'$ to the contraction of $C'$ fixes $v \in X_{t'}$, we have $v \notin B(S')$. Since $v$ is white in $C$ and $C$ is well-formed, $v$ is chained in $C$ and hence in $C'$. As $v$ is gray in $C'$ the contraction of $C'$ keeps $v$ chained in $S'$. Therefore, Case F2 applies and we obtain the parent $S''$ of $S'$ by turning the gray vertex $v$ white and then contracting the result. We may view the relationship between $C$ and $S''$ as follows: we first turn the color of $v$ in $C$ from white to gray and contract the result $C'$ to obtain $S'$ through an $X_{t'}$-isomorphism; then, we turn $v$ white and further contract the result into $S''$. The contraction of $C$ in one step gives us an $X_t$-isomorph of $S''$ which, since $S$ describes $C$ by assumption, is an $X_t$-isomorph of $S$ as well. As $S''$ is the parent of $S'$ and is $X_t$-isomorphic to $S$, $S$ has a child that is $X_{t'}$-isomorphic to $S'$. Since $S'$ describes $C'$, this child of $S$ is valid and we are done in this case as well.

For the converse, suppose that $S$ has a valid child $S'$. Let $C'$ be a well-formed colored drawing of $G[V_{t'}]$ that respects $X_{t'}$ and is described by $S'$. First suppose $v \in B(C')$. Since $v \in X_{t'}$, the $X_{t'}$-isomorphism from $S'$ to the contraction of $C'$ fixes $v$. Therefore, we have $v \in B(S')$. Case F1 applies and we have $S = S'$. Since $S$ describes $C'$ and $C'$ respects $X_t \subseteq X_{t'}$, $S$ is valid on $t$ and we are done in this case.

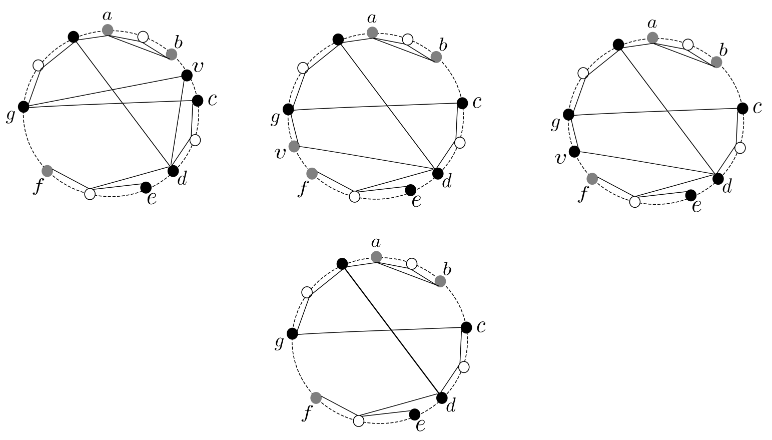

**Figure 3** The figures show an example of sketches on an introduce node $t$. The bottom figure depicts a sketch on $X_{t'}$ and the top figures depict some of its parents. Here, $X_t = \{a, b, c, d, e, f, g, v\}$ and $v \in X_t \setminus X_{t'}$ is the vertex introduced.

So suppose that $v \notin B(C')$. Then we have $v \notin B(S')$ and Case F2 must apply since $S'$ is a child of $S$. Let $C$ be the colored drawing of $G[V_t] = G[V_{t'}]$ which is identical to $C'$ except that the color of $v$ is turned white from gray. Since $v$ is chained in $S'$ by the condition of Case F2 and the $X_{t'}$-isomorphism from $S'$ to the contraction of $C'$ fixes $v$, $v$ is chained in $C'$ and hence in $C$. We may obtain an $X_t$-isomorph of $S$ from $C$ by turning the color of $v$ gray, contracting the result $C'$ into an $X_{t'}$-isomorph of $S'$, turning the color of $v$ back to white, and then contracting the result. Since $v$ is chained in $C$, we may perform the contraction in one step, which shows that $S$ describes $C$ and hence is valid on $t$. ◀

## Introduce node

Let $t$ be an introduce node in $T$ and $t'$ the unique child of $t$. Let $v \in X_t$ be the vertex introduced: $X_t = X_{t'} \cup \{v\}$. Let $S$ be a sketch on $X_t$ and $S'$ a sketch on $X_{t'}$. We say that $S'$ is the *child* of $S$ if either

1. $|X_t| = 1$ and $S'$ is an unique empty sketch, or
2. $|X_t| > 1$, $S' = S|V_{t'}$, and neither of the two vertices in $N_{\text{cycle}(S)}(v)$, which are not necessarily distinct, are white in $S$.

Note that $V_{t'} = V_t \setminus \{v\}$ and hence $S'$ is obtained from $S$ by removing $v$. We say that $S$ is a parent of $S'$ if $S'$ is the child of $S$. Note that the child of a sketch on $X_t$ is unique up to isomorphism if one exists, while a sketch on $X_{t'}$ may have more than one parent in general.

▶ **Lemma 11.** *Let $t$ be an introduce node and $t'$ a unique child of $t$. A sketch $S$ on $t$ is valid if and only if it has a child that is valid on $t'$.*

**Proof.** Let $v$ be the vertex introduced in $t$: $X_t = X_{t'} \cup \{v\}$. We only prove the case 2, since the case 1 is straightforward.

Suppose that $S$ is a valid sketch on $t$ such that $|X_t| > 1$. Let $C$ be a well-formed colored drawing of $G[V_t]$ that respects $X_t$ and is described by $S$. Since $v \in X_t$ and $C$ respects $X_t$, $v$ is either black or gray in $C$. Let $C' = C|V_{t'}$. Since $V_t = V_{t'} \cup \{v\}$, $C'$ is obtained from $C$ by removing $v$. Let $u$ be a vertex adjacent to $v$ in cycle$(C)$. If $u \in W(C)$ then $u$ must be chained in $C$ since $C$ is well-formed. But this is a contradiction, since $v \notin V_{t'}$ has no neighbor in $W(C) = W(C') \subseteq V_{t'} \setminus X_{t'}$. Therefore, we conclude that neither of the two vertices in $N_{\text{cycle}(C)}(v)$ are white in $C$. Since the contraction of $C$ does not change this local structure around $v$, the vertices in $N_{\mathrm{cycle}(C)}(v)$ correspond to those in $N_{\mathrm{cycle}(S)}(v)$ through the $X_t$-isomorphism from $S$ to the contraction of $C$. Therefore, $S$ satisfies the condition for having a child. Let $S'$ be the child of $S$. Since $C$ and $C'$ have the same set of maximal white chains, the $X_t$-isomorphism from $S$ to the contraction of $C$ gives an $X_{t'}$-isomorphism from $S'$ to the contraction of $C'$, when restricted to $V(S') = V(S) \setminus \{v\}$. Therefore, $S'$ is valid.

For the converse, suppose $S$ has a child $S'$ that is valid on $t'$. Let $C'$ be a well-formed colored drawing of $G[V_{t'}]$ that respects $X_{t'}$ and is described by $S'$. Let $v_1$ and $v_2$ be the two vertices in $N_{\mathrm{cycle}(S)}(v)$. Each of $v_1$ and $v_2$ is either black or gray in $S$, since $S$ has a child. Since $S$ respects $X_t$ and $v \in X_t$, $v$ is also either black or gray. Since $v_1$ and $v_2$ are adjacent to each other in cycle$(S')$, the $X_{t'}$-isomorphism from $S'$ to the contraction of $C'$ maps $v_1$ and $v_2$ to vertices $v_1'$ and $v_2'$ of $C'$ that are either black or gray in $C'$ and are adjacent to each other in cycle$(C')$. Let $D$ be the drawing of $G[V_t]$ obtained from the drawing of $C'$ by adding $v$ between $v_1'$ and $v_2'$. Let $C = (D, B, W(C'))$, where $B = B(C') \cup \{v\}$ if $v \in B(S)$ and $B = B(C')$ otherwise.

We claim that $C$ is a colored drawing. To see this, let $e$ and $f$ be edges of $G[V_t]$ that cross each other in $D$. If neither $e$ nor $f$ is incident to $v$, then the crossing is in $C'$ and hence all the ends of $e$ and $f$ are black in $C'$ and hence in $C$. So suppose one of $e$ and $f$, say $e$, is incident with $v$. Let $u_1$ be the other end of $e$ and let $u_2$ and $u_3$ be the ends of $f$. For $i = 1, 2, 3$, $u_i$ has a vertex $u_i'$ in $S'$ corresponding to $u_i$: the $X_{t'}$-isomorphism from $S'$ to the contraction of $C'$ maps $u_i'$ to either $u_i$ or the contraction of a white maximal chain containing $u_i$. Since $e$ and $f$ cross each other in $C$, the edge between $v$ and $u_1'$ and the edge between $u_2'$ and $u_3'$ must cross each other in $S$. As sketch $S$ is a colored drawing, $v$, $u_1'$, $u_2'$, and $u_3'$ are black in $S$. By the definition of $C$, $v$ is in $C$. Moreover, from the definition of the contraction, we see that $u_1$, $u_2$, and $u_3$ must be black in $C'$ and hence in $C$. Therefore, $C$ satisfies the condition for being a colored drawing. $C$ is well-formed, since every white vertex in $C$ is a white vertex in $C'$ and therefore is chained in $C'$ and hence in $C$.

Since $C'$ is obtained from $C$ by removing $v$ and neither of the two vertices in $N_{\mathrm{cycle}(C)}(v)$ are white, $C$ is contracted in the same way as $C'$ is contracted into an $X_{t'}$-isomorph of $S'$, resulting in an $X_t$-isomorph of $S$. Therefore $S$ describes $C$ and hence is valid. ◀

## Join Node

Let $t$ be a join node with child nodes $t_1$ and $t_2$. From the definition of join node, we have $X_t = X_{t_1} = X_{t_2}$ and $(V_{t_1} \setminus X_t) \cap (V_{t_2} \setminus X_t) = \emptyset$. We may assume that neither $V_{t_1} \setminus X_t$ nor $V_{t_2} \setminus X_t$ is empty and hence $|V_t| \geq 3$.

Let $S$ be a sketch on $t$. Let $S_1$ and $S_2$ be sketches on $t_1$ and $t_2$, respectively. We say that $S$ is the parent of the pair $(S_1, S_2)$ if

1. $V(S_1) \setminus X_t$ and $V(S_2) \setminus X_t$ partition $V(S) \setminus X_t$,
2. there is no edge in $E(S)$ between $V(S_1) \setminus X_t$ and $V(S_2) \setminus X_t$,
3. $S_1 = S|V(S_1)$, and
4. $S_2 = S|V(S_2)$.

▶ **Lemma 12.** *Let $t$ be a join node with child nodes $t_1$ and $t_2$. A sketch $S$ on $t$ is valid if and only if there are valid sketches $S_1$ on $t_1$ and $S_2$ on $t_2$ such that $S$ is the parent of the pair $(S_1, S_2)$.*

**Proof.** Suppose first that $S$ is a valid sketch on $t$. Let $C$ be a well-formed colored drawing of $G[V_t]$ that respects $X_t$ and is described by $S$. Let $C_i = C|V_{t_i}$ for $i = 1, 2$. Then, for $i = 1, 2$, $C_i$ is well-formed, since each white vertex $v \in C_i$ is chained in $C$ and, since $v \in V_{t_i} \setminus X_t$, $N_G(v)$

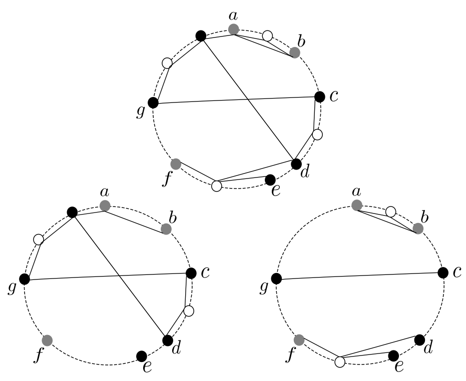

**Figure 4** The figures show an example of a sketch on a join node $t$. The top figure depicts a sketch on $X_t$ and the bottom figures depict a pair of sketches whose parent is the sketch in the top figure. Here, $X_t = \{a, b, c, d, e, f, g\}$.

are in $V_{t_i}$, is chained in $C_i$ as well. Moreover, for $i = 1, 2$, $C_i$ respects $X_{t_i}$. To confirm this, fix $i$. Since $W(C) \cap X_t = \emptyset$ as $C$ respects $X_t$, we have $W(C_i) \cap X_{t_i} = (W(C) \cap V_{t_i}) \cap X_t = \emptyset$. We also have $V(C_i) = B(C_i) \cup W(C_i) \cup X_{t_i}$, since $V(C_i) = V(C) \cap V_{t_i}$, $B(C_i) = B(C) \cap V_{t_i}$, and $W(C_i) = W(C) \cap V_{t_i}$. Therefore, $C_i$ respects $X_{t_i}$.

Observe that each chain in $C$ is either entirely contained in $C_1$ or entirely contained in $C_2$, as there is no edge of $G$ between $V_{t_1} \setminus X_t$ and $V_{t_2} \setminus X_t$. Therefore, the partition $(V_{t_1} \setminus X_t, V_{t_2} \setminus X_t)$ of $V_t \setminus X_t$ induces a partition $(R_1, R_2)$ of $V(S) \setminus X_t$ through the $X_t$-isomorphism from $S$ to the contraction of $C$. There is no edge in $E(S)$ between $R_1$ and $R_2$, since such an edge would correspond to an edge between $V_{t_1} \setminus X_t$ and $V_{t_2} \setminus X_t$, contradicting the definition of a join node. Therefore, $S$ is the parent of the pair $(S_1, S_2)$, where $S_i$ for $i = 1, 2$ is defined by $S_i = S|(R_i \cup X_t)$. The contraction of $C$ to an $X_t$-isomorph of $S$ induces the contraction of $C_i$ to an $X_t$-isomorph of $S_i$, for $i = 1, 2$. Therefore, $S_i$ describes $C_i$ and hence is valid, for $i = 1, 2$.

For the converse, suppose $S$ is the parent of a pair $(S_1, S_2)$ where $S_i$ is a valid sketch on $t_i$, for $i = 1, 2$. For $i = 1, 2$, let $C_i$ be a well-formed colored drawing of $G[V_{t_i}]$ that respects $X_{t_i}$ and is described by $S_i$. We combine $C_1$ and $C_2$ into a colored drawing $C$ of $G[V_t]$ as follows. We take the sketch $S$ and replace each vertex $v \in V(S)$ as follows. If $v \in X_t$ then we keep $v$ as it is. If $v \in B(S_1) \setminus X_t$, then we replace $v$ by the vertex of $C_1$ to which $v$ is mapped by the $X_t$-isomorphism from $S_1$ to the contraction of $C_1$; the case $v \in B(S_2) \setminus X_t$ is similar. If $v \in W(S_1)$, then we replace $v$ by the maximal white chain of $C_1$, the contraction of which $v$ is mapped to by the $X_t$-isomorphism from $S_1$ to the contraction of $C_1$; the case $v \in W(S_2)$ is similar. We let the resulting colored drawing be $C$. All edges of the drawing except for those in the maximal white chains correspond to edges in $S$ and edges in white chains are not crossing. Therefore, every vertex incident to a crossing edge is colored black in $C$ as it is in $S$ and hence $C$ is indeed a colored drawing. That $C$ respects $X_t$ is trivial. To see that $C$ is well-formed, let $w$ be a white vertex of $C$. Then, either $w$ corresponds to a white vertex of $S$ or $w$ is in a white chain that corresponds to a white vertex of $S$. Since this white vertex in $S$ is chained since $S$ is well-formed, $w$ is chained in $C$. Finally, combining the contractions of $C_i$ into $X_t$-isomorphs of $S_i$ for $i = 1, 2$, we get the contraction of $C$ into an $X_t$-isomorph of $S$. Therefore, $S$ describes $C$ and hence is valid. ◀

## 5 Algorithm

By Lemma 7, we can solve our problem independently for each biconnected component of $G$. Moreover, the biconnected components can be computed in linear time [14]. When the treewidth of $G$ is larger than $c\sqrt{k}$ for some constant $c > 0$, by Lemma 6, we can conclude $\mathrm{cr}_1(G) > k$. Thus, in the following, we can assume that the given graph $G$ is biconnected and its treewidth is at most $c\sqrt{k}$. Applying the algorithm of Bodlaender et al. [5], we can obtain a tree decomposition of $G$ whose width is $O(\sqrt{k})$ in $2^{O(\sqrt{k})}n$ time and its nice tree decomposition by Lemma 4.

We say that a sketch is *small* if it has at most $4k$ black vertices and contains at most $k$ crossings. Our dynamic programming algorithms inductively enumerates the set of all valid sketches on $t$ that are small, for each node $t$ of the nice tree decomposition, using the recurrences established in the previous section. The dynamic programming table for $t$ contains one representative sketch from each $X_t$-isomorphism class. It is straightforward to verify that, to decide if a small sketch on $t$ is valid or not, only small sketches on child node(s) of $t$ need to be examined. Thus, the computation gives us the set of all small and valid sketches on the root of the tree decomposition. Therefore, the proof of the correctness of our algorithm lies in showing the following lemma.

▶ **Lemma 13.** *$G$ has a drawing with at most $k$ crossings if and only if there is a small sketch on $\emptyset$ that is valid for $V(G)$.*

**Proof.** Suppose first that there is a small sketch $S$ on $\emptyset$ that is valid for $V(G)$. Then, since $S$ is valid, there is a colored drawing $C$ of $G$ described by $S$. The number of crossings of $C$ is equal to the number of crossings in $S$ and is at most $k$.

For the converse, suppose that $G$ has a drawing with at most $k$ crossings. Turn this drawing into a colored drawing $C$ of $G$ by making the ends of all crossing edges black and all other vertices white. We have at most $4k$ black vertices and $C$ respects $\emptyset$. Moreover, $C$ is well-formed by the chain lemma (Lemma 8). Contracting $C$ (with respect to boundary $\emptyset$), we obtain a small sketch on $\emptyset$ that describes $C$ and hence is valid for $V(G)$. ◀

For the running time, we prove the following:

▶ **Lemma 14.** *For each $t \in V(T)$, the number of small sketches on $t$, counting one from each $X_t$-isomorphism class, is $2^{O(k \log k)}$.*

**Proof.** The number of non-isomorphic graphs with $p$ vertices and $q$ edges is upper bounded by $p^{2q}$. For such a graph, the number of different colorings is at most $3^p$, and the number of different drawings is at most $(p-1)!$. Observe that every small sketch of $p$ vertices has $2p + k - 3$ edges. This follows from the fact that we can obtain an outerplanar drawing by removing at most $k$ edges from an arbitrary small sketch. Since the color of the two vertices in $N_{\mathrm{cycle}(S)}(v)$ for every white vertex $v$ in an arbitrary sketch $S$ is either black or gray, the number of white vertices is at most $4k + O(\sqrt{k})$. Therefore, each small sketch has at most $8k + O(\sqrt{k})$ vertices and at most $17k + O(\sqrt{k})$ edges, and hence the number of such drawings is $2^{O(k \log k)}$. ◀

## References

1 M. J. Bannister and D. Eppstein. Crossing minimization for 1-page and 2-page drawings of graphs with bounded treewidth. In *Proc. of GD 2014*, pages 210–221. Springer, 2014.

2 M. Baur and U. Brandes. Crossing reduction in circular layouts. *WG*, 3353:332–343, 2004.

3 F. Bernhart and P. C. Kainen. The book thickness of a graph. *Journal of Combinatorial Theory, Series B*, 27(3):320–331, 1979.

4 G. Blin, G. Fertin, D. Hermelin, and S. Vialette. Fixed-parameter algorithms for protein similarity search under mRNA structure constraints. *Journal of Discrete Algorithms*, 6(4):618–626, 2008.

5 H. L. Bodlaender, P. G. Drange, M. S. Dregi, F. V. Fomin, D. Lokshtanov, and M. Pilipczuk. A $c^k n$ 5-Approximation Algorithm for Treewidth. *SIAM Journal on Computing*, 45(2):317–378, 2016.

6 G. Chartrand and F. Harary. Planar Permutation Graphs. *Annales de l'I.H.P. Probabilités et statistiques*, 3(4):433–438, 1967.

7 F. R. K. Chung, F. Thomson Leighton, and A. L. Rosenberg. Embedding graphs in books: a layout problem with applications to VLSI design. *SIAM Journal on Algebraic Discrete Methods*, 8(1):33–58, 1987.

8 B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and computation*, 85(1):12–75, 1990.

9 B. Courcelle and M. Mosbah. Monadic second-order evaluations on tree-decomposable graphs. *Theoretical Computer Science*, 109(1):49–82, 1993.

10 V. Dujmović, M. R. Fellows, M. Kitching, G. Liotta, C. McCartin, N. Nishimura, P. Ragde, F. Rosamond, S. Whitesides, and D. R. Wood. On the parameterized complexity of layered graph drawing. *Algorithmica*, 52(2):267–292, 2008.

11 M. Frick and M. Grohe. The complexity of first-order and monadic second-order logic revisited. *Annals of Pure and Applied Logic*, 130(1):3–31, 2004.

12 R. Fulek, H. He, O. Sỳkora, and I. Vrt'o. Outerplanar crossing numbers of 3-row meshes, Halin graphs and complete $p$-partite graphs. *SOFSEM 2005: Theory and Practice of Computer Science*, pages 376–379, 2005.

13 H. He, A. Sălăgean, and E. Mäkinen. One-and two-page crossing numbers for some types of graphs. *International Journal of Computer Mathematics*, 87(8):1667–1679, 2010.

14 J. Hopcroft and R. Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, 1973.

15 P. C. Kainen. The book thickness of a graph II. *Congressus Numerantium*, 71:121–132, 1990.

16 T. Kloks. *Treewidth: computations and approximations*, volume 842. Springer Science & Business Media, 1994.

17 E. Mäkinen. On circular layouts. *International Journal of Computer Mathematics*, 24(1):29–37, 1988.

18 S. Masuda, T. Kashiwabara, K. Nakajima, and T. Fujisawa. On the NP-completeness of a computer network layout problem. In *Proceedings of the 1987 IEEE International Symp. on Circuits and Systems*, pages 292–295, 1987.

19 S. L. Mitchell. Linear algorithms to recognize outerplanar and maximal outerplanar graphs. *Information Processing Letters*, 9(5):229–232, 1979.

20 F. Shahrokhi, O. Sỳkora, L. A. Székely, and I. Vrt'o. Book embeddings and crossing numbers. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 256–268. Springer, 1994.

21 J. M. Six and I. G. Tollis. Circular drawings of biconnected graphs. In *ALENEX*, volume 99, pages 57–73. Springer, 1999.

22 M. M. Sysło. Characterizations of outerplanar graphs. *Discrete Mathematics*, 26(1):47–53, 1979.

23 M. Yannakakis. Embedding planar graphs in four pages. *Journal of Computer and System Sciences*, 38(1):36–67, 1989.