Constrained Outer-String Representations

Therese Biedl
David R. Cheriton School of Computer Science, University of Waterloo, Canada

Sabine Cornelsen
Department of Computer and Information Science, University of Konstanz, Germany

Jan Kratochvíl
Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

Ignaz Rutter
Faculty of Computer Science and Mathematics, University of Passau, Germany

Abstract

An *outer-string representation* of a graph is an intersection representation in which each vertex is represented by a curve that is contained in the unit disk and has at least one endpoint on the boundary of the unit disk. In an *outer-1-string representation* the curves representing any two vertices are in addition allowed to intersect at most once.

In this paper, we consider the following constrained version: Given a graph G plus a cyclic order $v_1, \ldots, v_n$ of the vertices in G, test whether G has an outer-string or an outer-1-string representation in which the curves representing $v_1, \ldots, v_n$ intersect the boundary of the unit disk in this order. We first show that a graph has an outer-string representation for all possible cyclic orders of the vertices if and only if the graph is the complement of a chordal graph. Then we turn towards the situation where one particular cyclic order of the vertices is fixed.

We characterize the chordal graphs admitting a constrained outer-string representation and the trees and cycles admitting a constrained outer-1-string representation. The characterizations yield polynomial-time recognition and construction algorithms; in the case of outer-1-string representations the run time is linear. We also show how to decide in polynomial time whether an arbitrary graph admits a constrained *L-shaped* outer-1-string representation. In an L-shaped representation the curves are 1-bend orthogonal polylines anchored on a horizontal line, and they are contained in the half-plane below that line. However, not even all paths with a constrained outer-1-string representation admit one with L-shapes. We show that 2-bend orthogonal polylines are sufficient for trees and cycles with a constrained outer-1-string representation.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases String representation, outer-string, outer-1-string, chordal graphs, trees, polynomial-time algorithms, computational complexity

Digital Object Identifier 10.4230/LIPIcs.GD.2024.10

Funding *Therese Biedl*: Research supported by NSERC FRN RGPIN-2020-03958.
Jan Kratochvíl: Research supported by Czech Science Foundation grant No. GAČR 23-04949X.

Acknowledgements This work was initiated at the Dagstuhl Seminar 24062 on Beyond-Planar Graphs: Models, Structures and Geometric Representations, Schloss Dagstuhl, Germany, February 2024. The authors would like to thank the other participants (and especially Stefan Felsner) for stimulating discussions.

1 Introduction

In a *string representation* [6, 15] of a graph $G = (V, E)$, each vertex v is drawn as a simple curve $\partial(v)$ such that the curves of two vertices intersect if and only if the two vertices are adjacent. We study here only *outer-string representations* where all curves reside within a disk or simple closed region D, and the curve of every vertex has at least one endpoint

© Therese Biedl, Sabine Cornelsen, Jan Kratochvíl, and Ignaz Rutter;
licensed under Creative Commons License CC-BY 4.0
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).
Editors: Stefan Felsner and Karsten Klein; Article No. 10; pp. 10:1–10:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

on the boundary of D, the so-called *anchor* of the vertex. See Figure 1a. Outer-string representations were named as such in 1982 [10], though they were implicitly defined and first results obtained already in 1966 [15]. It follows from a construction of Middendorf and Pfeiffer [12] that testing whether a graph admits an outer-string representation is NP-hard; see [14] for a sketch of the proof. One key result about outer-string graphs is that they are *χ-bounded*, i.e., their chromatic number is bounded by a function of the clique number [14]. By contrast, the chromatic number even of triangle-free segment graphs, which are a subclass of string graphs, can be $\Omega(\log n)$ [13]. A graph is a *chordal graph*, if it does not contain an induced cycle of length greater than three. By its tree representation [7, 16], every chordal graph admits an outer-string representation, and so do the complements of chordal graphs. Unfortunately, outer-string representations sometimes need exponentially many crossings [1]. So it is interesting to investigate which graphs allow an outer-string representation with a restricted number of crossings. In an *outer-1-string representation*, it is additionally required that the curves of two vertices intersect at most once. This is similar to the intersection graph of pseudosegments [6], however, with the additional constraint that the anchors still have to be on the boundary of a simple closed region containing all pseudosegments. Representing chordal graphs as intersections of pseudosegments was considered in [3].

Biedl and Derka [2] considered outer-string representation where the order of crossings along a curve was constrained. In this paper, we study outer-string representations that are *constrained* in the sense that the cyclic order of the anchors is fixed, i.e., we consider as input *cyclically ordered graphs*[1] (that is, graphs together with a cyclic order of the vertices) and we ask whether there is an outer-string, or an outer-1-string representation within a disk D in which the anchors occur on the boundary of D in the given cyclic order. Constrained outer-string representations were called the *constrained case* in [15]. Sinden [15] showed that the constrained case with n vertices can be reduced to the unconstrained case with $2n$ additional vertices and $4n$ additional edges.

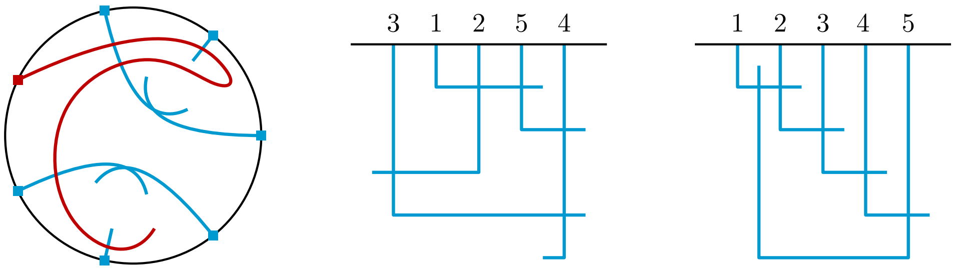
One can restrict the shapes of the curves further. In particular, we also consider *L-shaped* [11, 9, 4] and *U-shaped representations* in which the anchors are on a horizontal line ℓ and the vertices are 1- or 2-bend orthogonal polylines below that line; see Figures 1b and 1c. More precisely, in the case of L-shaped representations, the curves are required to consist of a vertical segment going downward from its anchor on ℓ followed (optionally) by a single horizontal segment. I.e., in particular, we also allow ⅃s. In the case of U-shaped representations, there may be an additional final vertical segment pointing upward. In the *constrained* version the input are *ordered graphs*, i.e., graphs with a linear order of the vertices and we require that the anchors on ℓ appear in this specific order.

Besides some sufficient conditions for constrained outer-string representations, Sinden [15] also observed the following necessary condition: The *complement of an anchor-ordered cycle* with at least four vertices does not have a constrained outer-string representation, i.e., if the cyclic order is $v_1, \ldots, v_n$ then the graph with edge set $E = \{\{v_i, v_j\};\ |i-j| \notin \{1, n-1\}\}$ does not have a constrained outer-string representation; see Figure 2.

Our Results. We show that a graph admits a constrained outer-string representation for every circular order of the vertices if and only if its complement is chordal (Theorem 2 in Section 2). In Section 3 we show that a cyclically ordered chordal graph admits a constrained outer-string representation if and only if it does not contain the complement of an anchor-ordered 4-cycle. The proof is constructive and yields a construction algorithm as

---

[1] Sinden [15] used the term *constrained graphs*.



(a) Outer-string. **(b)** L-shaped. **(c)** U-shaped.

Figure 1 (a) An outer-string representation of a tree that is not an outer-1-string representation (b+c) two outer-1-string representations of a 5-cycle with special shapes.

(a) Complement of a 4-cycle. **(b)** Complement of a 5-cycle.

Figure 2 Complements of anchor-ordered cycles; vertices at anchor positions.

well as a polynomial time testing algorithm. In order to characterize the cyclically ordered trees that admit a constrained outer-1-string representation, we need two more forbidden substructures, which we define in Section 4. We then provide a linear-time algorithm that either constructs a constrained outer-1-string representation of a cyclically ordered tree, or returns a forbidden substructure. In Section 5, we show how to test in time quadratic in the number of vertices whether any ordered graph admits a constrained L-shaped outer-1-string representation. In Section 6, we characterize cyclically ordered simple cycles that admit a constrained outer-1-string representation. The characterization yields a linear time testing algorithm. We further show that every cyclically ordered tree (Corollary 18) or simple cycle (Corollary 23) that admits a constrained outer-1-string representation already admits one with U-shapes for every induced linear order.[2] Full proofs of statements marked with ($\star$) will appear in the forthcoming full version of the paper.

2 Preliminaries

Let $G = (V, E)$ be a simple graph. For $e \in E$, let $G - e = (V, E \setminus \{e\})$. For $V' \subseteq V$, let $G - V'$ be the graph obtained from G by removing V' and all edges incident to a vertex in V'; we write $G - v$ for $G - \{v\}$. A set $A \subseteq V$ is *connected* if A induces a connected subgraph in G. The degree $\deg(v)$ of a vertex v is the number of edges that are incident to v. A *bridge* of a graph G is an edge e of G such that $G - e$ has more connected components than G. If G is connected, then the *bridge components* of a bridge $e = \{x, y\}$ are the vertex sets X and Y of the two connected components of $G - e$, named such that $x \in X$ and $y \in Y$.

[2] A cyclic order $\langle v_1, \ldots, v_n \rangle$ induces n linear orders $\langle v_{k+1}, \ldots, v_n, v_1, \ldots, v_k \rangle$, $1 \le k \le n$.

2.1 Input and Output

An instance $(G, \circlearrowright)$ of the problem of testing for a constrained outer-string or outer-1-string representation consists of a graph G and a cyclic order $\circlearrowright$ of the anchors around the disk D. During one of our algorithms, for some curves we need to fix both endpoints to the boundary of the disk D at specific positions. We call such vertices *doubly-anchored*, and they occur twice in $\circlearrowright$. For our algorithms we assume that the graph G is given as an adjacency list and $\circlearrowright$ is given as a doubly-linked circular list of vertex-references. Moreover, each vertex is equipped with pointers to its one or two entries in $\circlearrowright$.

A representation is stored as a plane graph H. Every anchor corresponds to an *anchor-vertex* in H, and these are connected in an *anchor-cycle* according to $\circlearrowright$ with doubly-anchored vertices appearing twice in the anchor-cycle. Every crossing of two curves corresponds to a *crossing-vertex*. In an outer-1-string representation this means that each edge of G corresponds to a crossing-vertex. Every vertex-curve $\partial(v)$ gives rise to edges in H that correspond to maximal sub-curves of $\partial(v)$ between its anchor(s) and crossings or between two crossings, connecting the corresponding vertices. Finally, H comes with a fixed circular order of the edges around each vertex that corresponds to the representation and in which the anchor-cycle bounds the outer face. Any embedding-preserving planar drawing of H yields then the desired representation of the instance.

2.2 A Necessary Condition for Constrained Outer-String Representations

Two sets V_1 and V_2 of vertices are *independent* if they have no vertex in common, and there is no edge with one endvertex in V_1 and the other in V_2. In an instance of constrained outer-string representation, we call two disjoint sets A_1 and A_2 of anchors *interleaved* if the cyclic order $\circlearrowright$ of anchors contains a subsequence a_1, a_2, a'_1, a'_2 where $a_i, a'_i \in A_i$ for $i = 1, 2$. Note that a_i and a'_i can be different anchors of the same doubly-anchored vertex. Two sets V_1 and V_2 of vertices are *interleaved* if their anchors in $\circlearrowright$ are interleaved. Observe that the complement of an anchor-ordered 4-cycle is a pair of interleaved independent edges.

▶ **Lemma 1** (interleaved independent pairs ⋆). *If $(G, \circlearrowright)$ has a constrained outer-string representation, then there are no two independent connected vertex sets that are interleaved.*

2.3 Complements of Chordal Graphs

The necessary condition of Sinden [15] implies that, if the complement of an input graph contains an induced cycle of length at least 4, then there exists a cyclic order for which it does not admit a constrained outer-string representation. This yields the necessity of the following characterization.

▶ **Theorem 2.** *A graph admits a constrained outer-string representation for any cyclic ordering of its vertices if and only if its complement is chordal.*

Proof. If the complement of a graph G is not chordal, then it contains an induced cycle C_k of length $k > 3$. Let $u_1, \ldots, u_k$ be the vertices of this cycle in the natural order. Then no circular order of the vertices of G which extends this order allows an outer-string representation of G, because it contains the complement of C_k with the natural order of its vertices as an induced subgraph.

We prove the opposite implication by induction on the number of vertices of G. Clearly, the one-vertex graph allows an outer-string representation for any (i.e., just one) circular ordering of its vertices (i.e., vertex). Suppose $G = (V, E)$ has more than one vertex and that

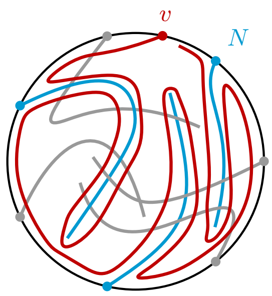


Figure 3 How to construct a constrained outer-string representation for a complement of a chordal graph.

the complement $\overline{G}$ of G is a chordal graph. Let a circular order $\circlearrowright$ of V be given. Let v be a *simplicial vertex* of $\overline{G}$, i.e., a vertex v whose neighborhood $N = N_{\overline{G}}(v)$ in $\overline{G}$ induces a clique in $\overline{G}$, and hence an independent set in G. Observe that every chordal graph contains a simplicial vertex [5]. Consider $G' = G - v$ and its complement $\overline{G'} = \overline{G - v} = \overline{G} - v$. By the induction hypothesis, $\overline{G'}$ has an outer-string representation that respects the circular order $\circlearrowright'$ of $V \setminus \{v\}$ induced by $\circlearrowright$. In this representation, the neighbors of v (in $\overline{G}$) are represented by disjoint curves; see Figure 3. We add a curve $\partial(v)$ starting at the anchor of v and contouring the boundary of the region $D \setminus \bigcup_{x \in N} \partial(x)$. In this way $\partial(v)$ intersects all curves $\partial(y)$ for $y \in V \setminus (N \cup \{v\})$ and avoids crossing all curves $\partial(x)$ for $x \in N$. Thus we constructed a constrained outer-string representation of G that respects $\circlearrowright$. ◀

3 Chordal Graphs and Constrained Outer-String Representations

We characterize chordal graphs with a cyclic order of the vertices that admit a constrained outer-string representation. The proof is by induction on the number of pairs of independent edges. For example, a path of length five has three pairs of independent edges; see Figure 4.

▶ **Theorem 3.** *A chordal graph $G = (V, E)$ with a cyclic order $\circlearrowright$ of V has a constrained outer-string representation if and only if no two independent edges are interleaved.*

Proof. By Lemma 1, $(G, \circlearrowright)$ has no constrained outer-string representation if there are two independent edges that are interleaved. So assume that there is no pair of interleaved independent edges. We show by induction on the number of pairs of independent edges that $(G, \circlearrowright)$ has a constrained outer-string representation within a simple connected region D. We may assume that there are no isolated vertices.

In the base case, G has no pair of independent edges. Thus [8, Theorem 6.3], G is a *split graph*, i.e., it consists of a clique $K = \{k_1, \ldots, k_r\}$ and a set S of independent vertices, with an arbitrary set of edges between K and S. To obtain an outer-string representation of G, add $|K|$ concentric circles inside D, and assign them to $k_1, \ldots, k_r$. For every clique-vertex k_i, go perpendicular from the anchor to the circle assigned to k_i, then along this circle until we almost touch the curve $\partial(k_i)$. This creates an intersection for each edge $\{k_i, k_j\}$: Assume

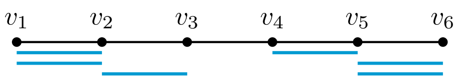
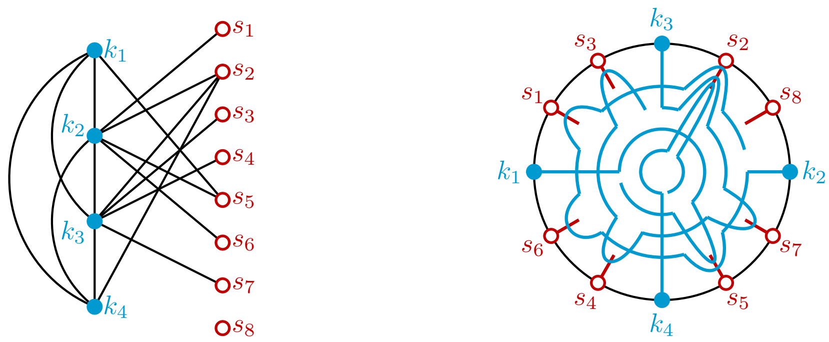


Figure 4 A path of length five contains three pairs of independent edges.



(a) **(b)**

Figure 5 Illustration of the base case in the proof of Theorem 3 on chordal graphs.

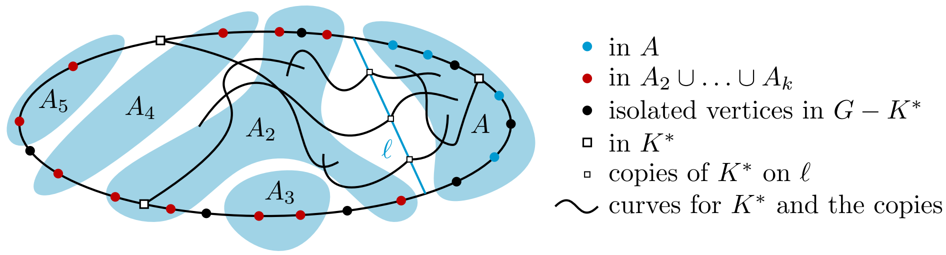


Figure 6 Splitting a chordal graph into smaller instances. If G is chordal but not a split graph, we find a minimal set K^* of vertices such that $G - K^*$ contains at least two non-trivial components. Let A be such a component for which the anchors are consecutive up to isolated vertices of $G - K^*$. Split D by a curve ℓ separating the anchors of A from the anchors of the other non-trivial components $A_2, \ldots, A_k$ of $G - K^*$. Insert copies of K^* on ℓ.

that the circle for k_i has greater radius. Then $\partial(k_j)$ intersects this circle when connecting from the anchor of k_j to its circle. This represents the clique K. Now for every vertex $s \in S$, add a short segment $\partial(s)$ from the anchor of s perpendicular to the boundary of D, and for all $k \in K$ with $\{k, s\} \in E$, add a detour to $\partial(k)$ to intersect $\partial(s)$; see Figure 5.

Now assume that G contains at least one pair of independent edges. Let K^* be a minimal set of vertices such that $G - K^*$ contains at least two *non-trivial components*, i.e., connected components that contain an edge. The following claim is an implication of [8, Theorem 4.1].

▷ Claim 4 (⋆). K^* exists and is a clique.

Let $A_1, \ldots, A_k$ be the non-trivial components of $G - K^*$. Since there is no pair of interleaved independent edges, it follows that the anchors of $A_1, \ldots, A_k$ on the boundary of D are nested; see Figure 6. In particular, there must be a component, say $A = A_1$, whose anchors are consecutive, except for perhaps some isolated vertices of $G - K^*$. Split D along a line ℓ that separates the anchors of A from the anchors of $A_2, \ldots, A_k$. Place $|K^*|$ anchor points along ℓ, one per vertex of K^* in arbitrary order.

Now we get two instances, an instance I_A and an instance $I_{\overline{A}}$, by cutting along ℓ. The instance I_A contains (1) all vertices of A, (2) all vertices whose anchor were on the same side of ℓ as the anchors of A; these might be anchors of isolated vertices of $G - K^*$ or anchors of vertices in K^*, and (3) copies of vertices in K^* with an anchor on ℓ. Here the point assigned to $k \in K^*$ is taken as the endpoint for k if the actual endpoint of k is not in this part of D, and it gets taken as endpoint for a new vertex k' otherwise, where k' is adjacent only to k. The instance $I_{\overline{A}}$ is defined analogously.

▷ Claim 5. I_A and $I_{\overline{A}}$ are both chordal.

Proof. I_A and $I_{\overline{A}}$ are obtained from chordal graphs by removing vertices and adding leaves. This neither creates new cycles, nor does it remove chords from remaining cycles. ◁

▷ Claim 6 (⋆). If I_A and $I_{\overline{A}}$ have constrained outer-string representations then so does $(G, \circlearrowleft)$.

Sketch of Proof. A curve for a vertex k in K^* can be obtained by starting at the original anchor of k, following $\partial(k)$ until its end and back to an intersection point with a curve ∂'_1 anchored at a copy k' of k on ℓ, following ∂'_1 up to ℓ and finally along the curve ∂'_2 of k' in the other subinstance until its end. ◁

▷ Claim 7. If I_A or $I_{\overline{A}}$ contains a pair of interleaved independent edges then so does $(G, \circlearrowleft)$.

Proof. Assume I_A contains a pair $\{v, w\}$, $\{a, b\}$ of interleaved independent edges, the case for $I_{\overline{A}}$ is symmetric. Unless this involves vertices with an anchor on the curve ℓ, the same pair is already contained in $(G, \circlearrowleft)$. If both $\{v, w\}$ and $\{a, b\}$ contain a vertex of K^*, then the pair is not independent. Since I_A contains only vertices of A, K^*, and degree-one neighbours of K^*, we may assume that $\{a, b\} \subseteq A$. If the anchors of both v and w are on ℓ, then $\{v, w\}$ and $\{a, b\}$ are not interleaved. So, we may assume that the anchor of v is on ℓ and the anchor of w is not. We distinguish two cases based on whether $w \in K^*$ or not.

If $w \in K^*$ then, by the minimality of K^*, vertex w had a neighbour v' in component A_2. It follows that $\{a, b\}$ and $\{w, v'\}$ are interleaved and independent. If $w \in A$ or w is an isolated vertex in $G - K^*$, then v is the copy of a vertex $v' \in K^*$ whose anchor is on the other side of ℓ than A. Thus, $\{a, b\}$ and $\{w, v'\}$ are interleaved and independent. ◁

▷ Claim 8 (⋆). The number of pairs of independent edges in I_A and $I_{\overline{A}}$, respectively, is smaller than in G.

Sketch of Proof. By construction, there is a pair of independent edges $e_1 \subseteq A_1 = A$, $e_2 \subseteq A_2$ in G which is neither contained in I_A nor $I_{\overline{A}}$. On the other hand the pairs of independent edges of I_A ($I_{\overline{A}}$, respectively), can be mapped into the pairs of independent edges of G with at least one endvertex in K^* or A ($A_2 \cup \cdots \cup A_k$, respectively). ◁

This concludes the proof: If $(G, \circlearrowleft)$ contains no pair of interleaved independent edges, then, by Claim 7, none of the sub-instances has one. By Claim 8, they have fewer independent edge pairs than $(G, \circlearrowleft)$ and they hence have a constrained outer-string representation by the inductive hypothesis. By Claim 6 these representations can be combined to a constrained outer-string representation for the original instance $(G, \circlearrowleft)$. ◀

▶ **Corollary 9.** *It can be tested in polynomial time whether a chordal graph with a given cyclic order of the vertices admits a constrained outer-string representation.*

4 Constrained Outer-1-String Representations for Trees

In this section, we show how to test for a constrained outer-1-string representation if the graph is a tree. We first give an outline of our approach. See Figure 7. Let $(G, \circlearrowleft)$ be the given instance where G is a tree. If G is a single vertex, then the answer is always true. Otherwise, we root G, preferably at a vertex that has at least two neighbors, and process the vertices in post-order, i.e., children are processed before their parents. Whenever we encounter a vertex x that is not a leaf, we either find an obstruction, i.e., a sub-instance that makes a constrained outer-1-string representation impossible, or we remove the children of

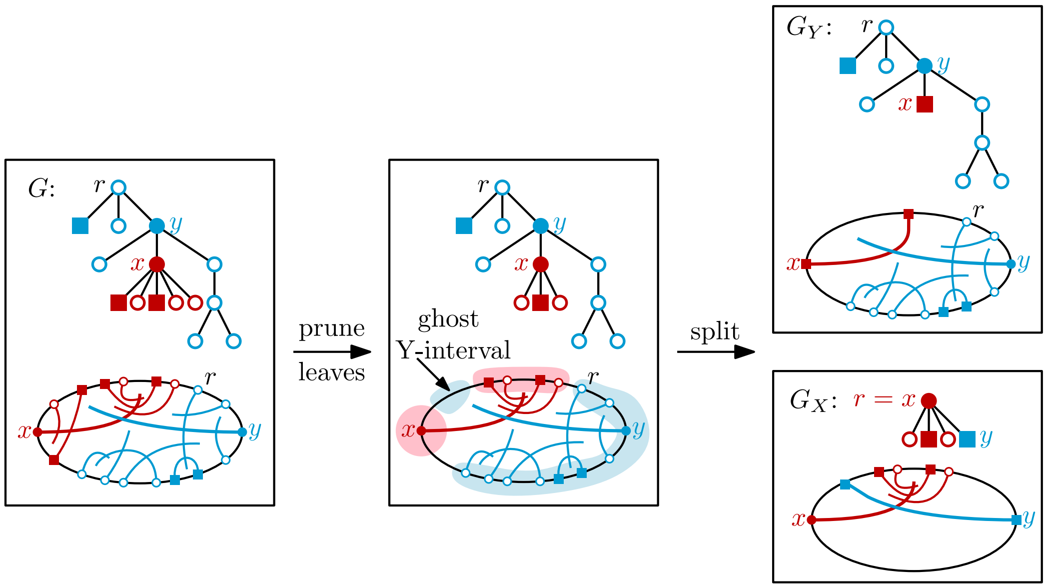


Figure 7 We solve constrained outer-1-string on trees in a postorder traversal. Some leaves might be doubly-anchored (squared vertices). When processing a vertex x then all its children are leaves. We first prune some of x's children (Rule 1). We then make sure that the sequence of anchors contains exactly two X-intervals (pink regions), one of which contains only x. If this is impossible, we reject the instance (Rule 2, Rule 4, and Rule 5). Finally, we split the instance along the edge between x and its parent y, keeping y and x as a doubly-anchored vertex in the opposite component (Rule 6). The base case is reached after pruning the leaves of a star (Rule 2 and Rule 3).

x until x is a leaf. Thus at the end only a single vertex remains and we are done. For the recursions, we will sometimes have doubly-anchored vertices, but we maintain as invariant that only leaves of the rooted tree can be doubly-anchored.

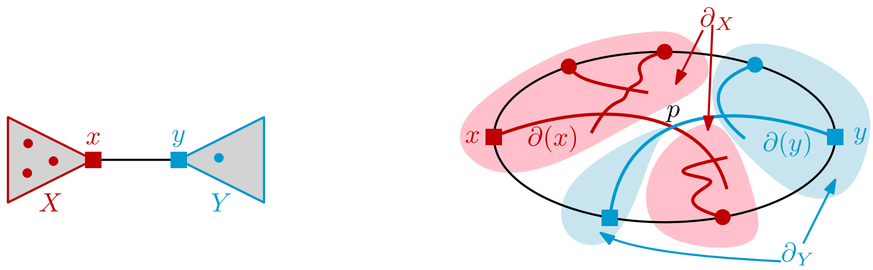
4.1 Obstructions

By Lemma 1, there cannot be a constrained outer-string representation if there is a pair of independent connected vertex-sets that are interleaved. We call such an interleaved independent pair a *pair-obstruction* if each of the two vertex-sets contains at most two vertices, i.e., it is an edge or a set containing a doubly-anchored vertex. We will have two other obstructions for constrained outer-1-string representations. Recall that a bridge $e = \{x, y\}$ defines the bridge-components X and Y of $G - e$ with $x \in X$ and $y \in Y$. An X*-interval* is a maximal sub-sequence of $\circlearrowright$ that only contains anchors of X. We define Y*-interval* analogously. See Figure 8 for the following lemma.

▶ **Lemma 10** (bridge-obstruction ⋆). *If $(G, \circlearrowright)$ has a constrained outer-1-string representation, then no bridge $\{x, y\}$ has three or more X-intervals.*

We use the term *bridge-obstruction* for a bridge that has three or more X-intervals and hence prevents a constrained outer-1-string representation. See Figure 9.

Finally, observe that two adjacent doubly-anchored vertices must be interleaved. The third kind of obstruction generalizes this observation and is based on a *central path* Π with $\ell \geq 0$ edges and hence will be called Π_ℓ*-obstruction*, or simply *path-obstruction*. See Figure 10. Let $\Pi = \langle v_0, v_1, \ldots, v_{\ell-1}, v_\ell \rangle$ be a path, and note that $\ell = 0$is specifically allowed. For the ends of the central path, there are three variants. In the first variant, there are additionally



(a) **(b)**

Figure 8 Illustration of the proof of Lemma 10. Red curves represent vertices in X, blue curves represent vertices in Y. The union ∂_X and ∂_Y of all red and all blue curves, respectively, is connected and the two sets intersect in $p = \partial(x) \cap \partial(y)$. Removing p from $\partial_X \cup \partial_Y$ yields four connected components (pink and light-blue areas) the anchor of which form X- or Y-intervals in $\circlearrowright$.

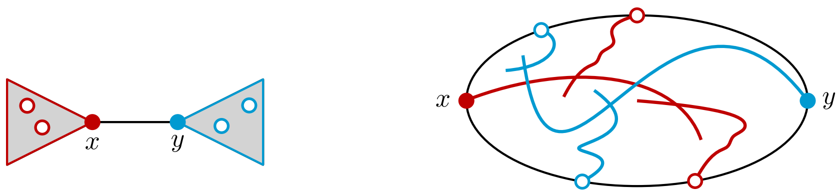


Figure 9 An outer-string representation of a bridge-obstruction.

two *bounding paths* $Q = \langle c, v_0, b, a\rangle$ and $Q' = \langle c', v_\ell, b', a'\rangle$ that are disjoint from Π and each other except at v_0 and v_ℓ. The anchor-order $\circlearrowright$ is such that in the order induced by the vertices in Π, Q, and Q' satisfies two things: (i) the anchors of the sets $\{a, b, c\}$, $\{a, b, c, v_0\}$,..., and $\{a, b, c, v_0, \dots, v_\ell\}$, respectively, appear consecutive and (ii) the pair $\{a, b\}$ and $\{v_0, c\}$ as well as the pair $\{a', b'\}$ and $\{v_\ell, c'\}$ are interleaved. In the second variant, one of the bounding paths, say Q, is replaced by the condition that v_0 is doubly-anchored and that the anchors of v_0 are consecutive in the induced anchor-order. The other conditions on the anchor order remain. The third variant is defined only for $\ell \geq 1$ and is obtained from the second variant by similarly replacing Q' with the requirement that v_ℓ be doubly-anchored with consecutive anchors.

▶ **Lemma 11** (path-obstruction ⋆). *If $(G, \circlearrowright)$ has a constrained outer-1-string representation, then there is no path-obstruction.*

Clearly, no instance with a constrained outer-1-string representation can contain any of the three obstructions. As our main result for trees, we show that this necessary condition is also sufficient, and furthermore an efficient constructive testing algorithm exists. We prove the following theorem in the next section.

▶ **Theorem 12.** *An instance $(G, \circlearrowright)$ where G is a tree admits a constrained outer-1-string representation if and only if it contains no pair-obstruction, no bridge-obstruction, and no path-obstruction. Furthermore, there is a linear-time algorithm that either finds such an obstruction or returns a constrained outer-1-string representation for $(G, \circlearrowright)$.*

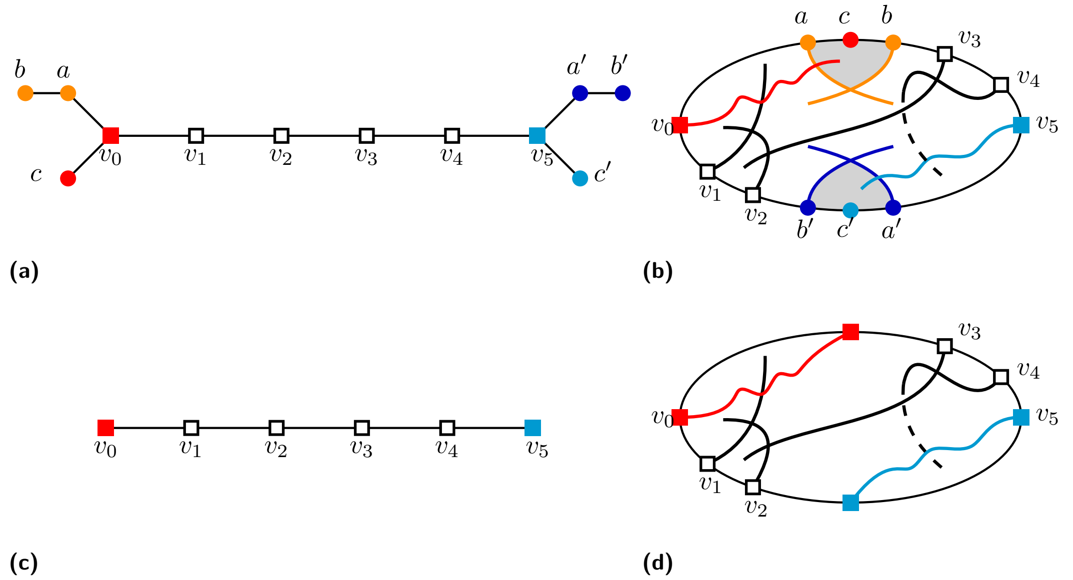


Figure 10 (a) The graph of a path-obstruction (b) and a forbidden order of the anchors along with an outer-string representation. (c,d) A doubly-anchored version of a path-obstruction.

4.2 Reduction-Rules

As outlined, we root the tree, preferably at a vertex of degree at least two, and process the vertices in post-order. Furthermore, we maintain that all vertices that have been processed are leaves in the tree or have been deleted altogether. Finally only leaves that are not the root may be doubly-anchored. Let x be the currently processed vertex. If x is already a leaf, then we proceed to the next vertex. So assume that x has children. These children have been processed already, so they are leaves. Let X be the set consisting of x and all its children. When processing x, we will apply a number of reduction rules, each of which yielding one or two smaller instances. In particular, all children of x are deleted eventually.

For the reduction rules we have to argue that they are *correct*, which means two things. First, if the smaller instances have solutions, then so does $(G, \circlearrowleft)$. Second, if one of the smaller instances contains an obstruction, then so does $(G, \circlearrowleft)$. The second one implies that if $(G, \circlearrowleft)$ has a solution, then so do the smaller instances: If $(G, \circlearrowleft)$ has a solution, then it has no obstruction, so the smaller instances have no obstructions; by the inductive hypothesis, this implies that the smaller instances have a constrained outer-1-string representation. Our arguments for this will be constructive, which means that there will be an easy algorithm to retrieve the solution or the obstruction from the ones for the smaller instances. A special type of these rules are *obstruction-rules*, where the returned instance is an obstruction that is contained in the instance. In that case the instance is a no-instance and we show how to exhibit the obstruction in the proof of correctness.

▶ **Rule 1** (leaves)**.** *If x is adjacent to a leaf v that either (i) is singly-anchored and x, v are consecutive in $\circlearrowleft$ or (ii) is doubly-anchored and v, x, v are consecutive in $\circlearrowleft$, then remove v and its anchors from $(G, \circlearrowleft)$.*

▶ **Lemma 13.** *Rule 1 is correct.*

Proof. Given a constrained outer-1-string representation of the smaller instance $(G', \circlearrowleft')$, we can add a curve for v that is anchored on the correct side of x in case (i) or anchored on both sides of x in case (ii) to obtain a constrained outer-1-string representation of $(G, \circlearrowleft)$.

If $(G', \circlearrowright')$ contains an obstruction, then the exact same subgraph is also an obstruction for $(G, \circlearrowright)$ because $(G', \circlearrowright')$ is an induced sub-instance of $(G, \circlearrowright)$, and adding more vertices and anchors does not destroy an obstruction. ◀

▶ **Rule 2** (anchor of x surrounded by X)**.** *If Rule 1 cannot be applied, but there are at least three anchors in $\circlearrowright$, and both anchors immediately before and after the anchor of x in $\circlearrowright$ belong to vertices of X, then report that the instance is a no-instance.*

▶ **Lemma 14.** *Rule 2 is correct.*

Proof. Let u and v be the vertices in X whose anchors are next to the anchor of x in $\circlearrowright$. Since Rule 1 cannot be applied, both u and v are doubly-anchored and $u \neq v$. We have $u \neq x \neq v$ since otherwise x would be doubly-anchored, but x is not a leaf and so cannot be doubly-anchored. We distinguish two cases: If the anchors of u, v, and x are in the cyclic order u, u, x, v, v then this is a Π_2-obstruction for path $\langle u, x, v\rangle$. If the cyclic order is v, u, x, v, u then u and v are two independent doubly-anchored vertices that are interleaved, so this is a pair-obstruction. ◀

If x is the root and neither Rule 1 nor Rule 2 applies then x is the only vertex that is left. Hence, we get the following rule.

▶ **Rule 3** (base case)**.** *If x is the root and neither Rule 1 nor Rule 2 applies, then the current (sub-)instance is a yes-instance.*

So for the following rules, we assume that x has a parent y. Vertex y in turn either has a parent, or it is the root and, by the choice of the root, has at least one other child, so y has neighbors other than x. Consider the bridge $e = \{x, y\}$. Let X and Y be the respective bridge components of $G - e$ with $x \in X$ and $y \in Y$. This is consistent with our earlier definition of X. The following rule is clearly correct since it directly exhibits a bridge-obstruction.

▶ **Rule 4** (three X-intervals)**.** *If the bridge $\{x, y\}$ has three or more X-intervals, then report that the instance is a no-instance.*

So from now on we assume that there are one or two X-intervals. Actually both these cases can be handled at once. We first identify another obvious no-instance.

▶ **Rule 5** (two X-intervals, x not alone)**.** *If there are two X-intervals, Rule 1 and Rule 2 cannot be applied, and the X-interval containing the anchor of x contains at least two anchors, then report that the instance is a no-instance.*

▶ **Lemma 15.** *Rule 5 is correct.*

Proof. Assume that the X-interval containing the anchor of x contains at least two anchors. The anchor of x is the first or last anchor in the X-interval, otherwise Rule 2 would apply. Up to symmetry assume that it is the first. Let x' be the vertex of X whose anchor follows x in this X-interval. Since Rule 1 cannot be applied, x' is doubly-anchored. Note that $x' \neq x$ since otherwise x would be doubly-anchored, but x is not a leaf and so not doubly-anchored.

Consider one anchor in each of the two Y-intervals such that the respective vertices $y', y'' \in Y$ are either adjacent or identical. If the two anchors of x' are in the two X-intervals, then x' and $\{y', y''\}$ are independent and interleaved, so a pair-obstruction. If the two anchors of x' are in the same X-interval, then let x'' be any vertex of X whose anchor is in the other X-interval. This is a child of x. If one of y', y'', say y'', is y, then we obtain a Π_1-obstruction with central path $\langle x', x\rangle$, the doubly-anchored vertex x' and the bounding path $\langle x'', x, y, y'\rangle$. Observe that the anchors appear in a suitable order. If neither of y', y'' is y, then $\{y', y''\}$ forms an independent interleaved pair with edge $\{x, x''\}$, so we have a pair-obstruction. ◀

If none of the above rules apply, we are in one of two possible situations. Either (a) there is only one X-interval, and x is the first or last vertex in it, or (b) there are two X-intervals, but one of them contains only x. Both situations can be handled as one if in situation (a) we view the range between x and the rest of the X-interval as a “ghost Y-interval” that has no anchors of Y in it. So we always have two X-intervals (one of them contains only the anchor of x) and two Y-intervals (one of which may be a ghost-interval).

▶ **Rule 6** (split the instance)**.** *If none of the previous rules can be applied, then we split the instance into the graphs G_X and G_Y induced by $X \cup \{y\}$ and $Y \cup \{x\}$, respectively. In G_X, we doubly-anchor y in place of the two Y-intervals. In G_X, we doubly-anchor x in place of the two X-intervals. All other vertices use the same anchors as in $\circlearrowright$.*

▶ **Lemma 16** ($\star$)**.** *Rule 6 is correct.*

Sketch of Proof. A constrained outer-1-string representation for G can be constructed by combining constrained outer-1-string representation for G_X and G_Y at the intersection point of $\partial(x)$ and $\partial(y)$ such that the anchors are in the correct order.

It remains to show how to reconstruct obstructions of $(G, \circlearrowright)$ from obstructions of the reduced instances. To this end we have to show that if the new anchor of the vertex x or y is contained in an obstruction of a reduced instance, then we can use omitted vertices to find an obstruction in $(G, \circlearrowright)$. Since G_X is a star centered at x and yxy is a subsequence of the anchor order, it follows that G_X cannot contain an obstruction that uses y.

So, let $\mathcal{O}$ be an obstruction of G_Y that contains x. Let $x' \in X \setminus \{x\}$ be some child of x, preferably a doubly-anchored one. Any independent interleaved pair of G_Y that uses x can be expanded into one of G by adding x' to the set that contains x. Any bridge-obstruction at some bridge e of G_Y is also one in G. In both cases, an anchor of x' can take the place of the second anchor of x in G_Y. It remains to consider the case that $\mathcal{O}$ is a path-obstruction.

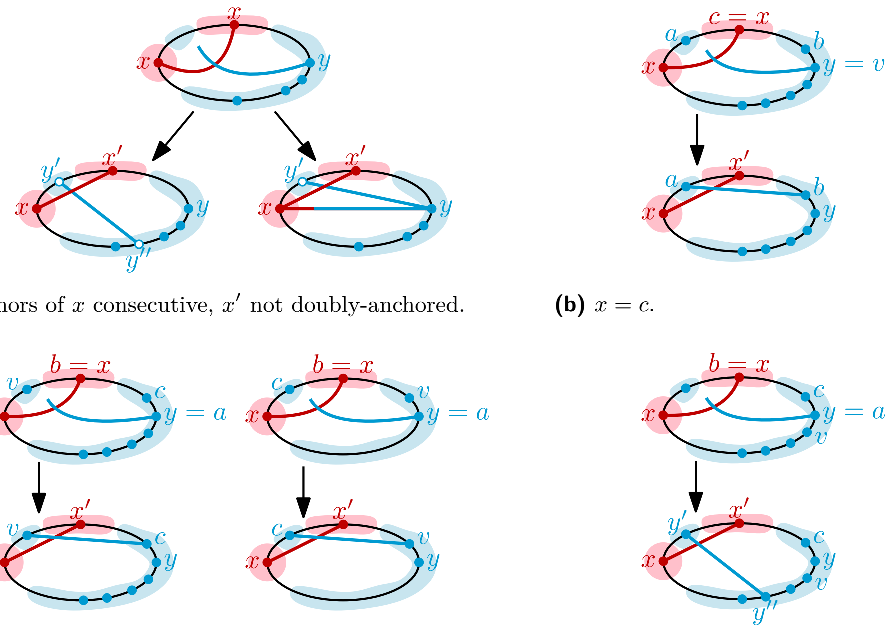
If the two anchors of x are consecutive among the anchors of $\mathcal{O}$, we may assume that x is the endvertex of the central path. If x has a doubly-anchored child, then we obtain a path-obstruction of G by appending x' to x. Otherwise there is no ghost Y-interval. Let $\{y', y''\}$ be an edge with anchors in different Y-intervals. Depending on whether $y' \neq y \neq y''$ or not either $\{x, x'\}$ and $\{y', y''\}$ is a pair-obstruction or G contains a path-obstruction with bounding path $\langle x', x, y, y' \rangle$. See Figure 11a.

If the two anchors of x are not consecutive among the anchors of $\mathcal{O}$, then x is the endvertex of a bounding path and G contains an interleaved independent pair. See Figure 11. ◀

Note that Rule 6 can *always* be applied if none of the previous rules apply. Observe that G_X is a star and so directly brings us to the base case after rooting G_X at x and applying Rule 1, as well as Rule 2 or Rule 3. Observe further that G_Y is obtained from G by removing the children of x and by doubly-anchoring x. Hence, in G_Y, vertex x has become a leaf as desired and we continue processing the rest of G_Y in post-order. This proves the characterization stated in Theorem 12. For the linear run time, we refer to the full version of the paper.

▶ **Corollary 17.** *A cyclically ordered path has a constrained outer-1-string representation if and only if there are no two independent edges that are interleaved.*

Proof. A path cannot have a path-obstruction and if a path has a bridge-obstruction then this already implies two independent edges that are interleaved. ◀



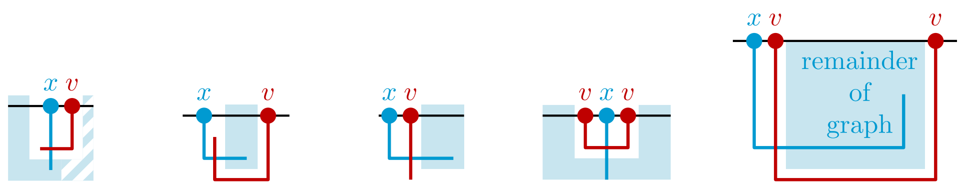
(a) Anchors of x consecutive, x' not doubly-anchored. **(b)** $x = c$.

(c) $x = b$, v and c in distinct intervals. **(d)** $x = b$, v and c in the same interval.

Figure 11 How to reconstruct an obstruction of G from a path-obstruction $\mathcal{O}$ of G_Y. In each case the topmost drawing is a string-representation while in the bottommost drawing the edges are represented as segments. (a) If x is the endvertex of the central path and no child x' of x is doubly-anchored then G contains an interleaved independent pair or $\{x, y\}$ is replaced by the bounding path $\langle x', x, y, y' \rangle$ in $\mathcal{O}$. (b-d) If x is the endvertex of a bounding path $\langle c, v, a, b \rangle$ and the two anchors of x are not consecutive then G contains an interleaved independent pair.

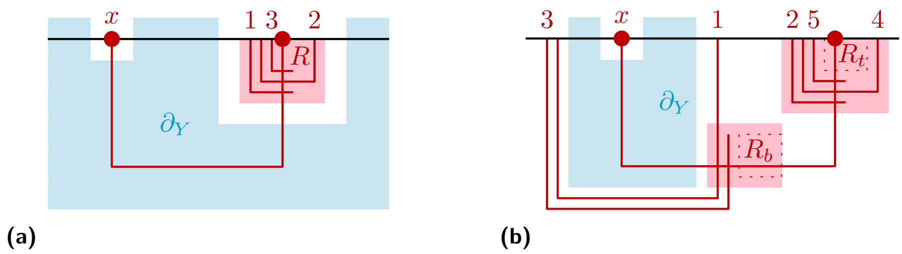
▶ **Corollary 18** (⋆). *A tree with a given cyclic order ⟳ of the vertices that admits a constrained outer-1-string representation also has a constrained U-shaped outer-1-string representation with respect to any linear order induced by ⟳.*

Sketch of Proof. We follow the construction for constrained outer-1-string representations. Whenever Rule 1 or Rule 6 yields yes-instances, we show how to obtain a constrained U-shaped outer-1-string representation for the original instance. See Figures 12 and 13. We maintain the property that nothing is drawn to the left of the left-most or to the right of the right-most anchor. If the linear order is such that an X-interval is split into a right-most and a left-most part and one of the two parts contains both anchors of a doubly-anchored vertex then the second anchor of x in G_Y is put in this sub-interval. ◀

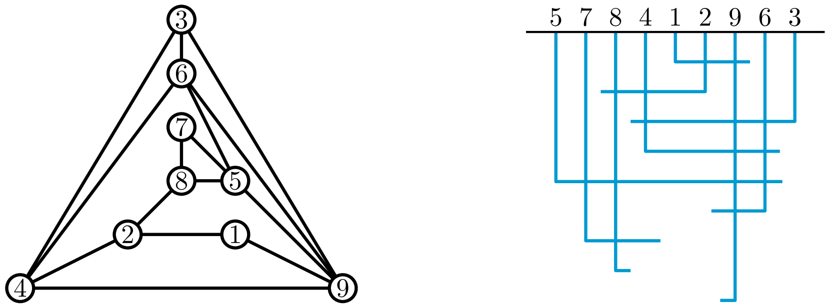


(a) **(b)** **(c)** **(d)** **(e)**

Figure 12 Constructing a U-shaped representation of a tree after the application of Rule 1.



■ **Figure 13** Constructing a U-shaped representation of a tree after the application of Rule 6, by inserting $X \setminus \{x\}$ into a representation of G_Y in the order in which we would apply Rule 1 in G_X.



(a) Graph with vertex order. **(b)** L-shaped representation.

■ **Figure 14** How to construct a constrained L-shaped outer-1-string representation. First order the vertices such that the anchores of the future neighbors of any vertex v are consecutive and next to v. Then draw the L-shaped curves in this order with decreasing y-coordinates of the horizontal part.

5 Constrained L-Shaped Outer-1-String Representations

We now show how to test in quadratic time whether an ordered graph admits a constrained L-shaped outer-1-string representation. See Figure 14.

▶ **Lemma 19.** *Let $G = (V, E)$ be a graph and let $\prec$ be an order of the vertices. Then $(G, \prec)$ admits a constrained L-shaped outer-1-string representation if and only if the vertices of G can be ordered $v_1, \dots, v_n$ such that for $i = 1, \dots, n$ the set of* future neighbors *$V_i = \{v_j;\ j > i \text{ and } \{v_i, v_j\} \in E\}$ of v_i as well as $V_i \cup \{v_i\}$ are consecutive in $\prec$.*

Proof. Assume that $v_1, \dots, v_n$ is such an order. Let the horizontal line from which the vertices hang have y-coordinate 0. For $i = 1, \dots, n$ we draw the vertical part of $\partial(v_i)$ from 0 to $-i$. The future neighbors V_i are all directly to the left or all directly to the right of v_i. Draw the horizontal part of $\partial(v_i)$ in that direction until the last future vertex is met.

Assume now that there is a constrained L-shaped outer-1-string representation of $(G, \prec)$. Order the vertices $v_1, \dots, v_n$ according to the y-coordinate of the horizontal part of their curve from top to bottom. Then the curve of all future neighbors of v_i must intersect the horizontal part of $\partial(v_i)$ and all vertical segments of all vertices v_j, $j > i$ must be at least as long as the one of v_i. It follows that V_i must be consecutive and directly next to v_i. ◀

An example of an ordered graph without a constrained L-shaped outer-1-string representation is the path $\langle 1234 \rangle$ with vertex ordering $\langle 2413 \rangle$.

▶ **Corollary 20.** *It can be tested in $\mathcal{O}(n^2)$ time whether an ordered graph with n vertices admits a constrained L-shaped outer-1-string representation.*

Proof. Let $G = (V, E)$ be a graph with a linear order $\prec$ of its n vertices. For $i = 1, \ldots, n$, iteratively check whether there is a vertex, such that the set of its neighbors is directly to the left or to the right of v_i. If so remove v_i and continue. Otherwise report that there is no constrained L-shaped outer-1-string representation.

This can be tested in $\mathcal{O}(n^2)$ time. For a vertex v, let $N(v)$ be the set of neighbors of v. An $N(v)$-interval is a maximal subsequence of $\prec$ that contains only anchors of vertices of $N(v)$. Let $k(v)$ be the number of $N(v)$-intervals. We first compute $k(v)$ for each vertex v. This can be done in linear time for each vertex. Observe that we can choose v as a next vertex if and only if $k(v) = 0$ or $k(v) = 1$ and at least one of the neighbors of v in $\prec$ is in $N(v)$. Each time we remove a vertex w, we update $k(v)$ as follows. Decrease $k(v)$ by one if and only if either $w \in N(v)$ and both neighbors of w in $\prec$ are not in $N(v)$ or $w \notin N(v)$ and both neighbors of w in $\prec$ are in $N(v)$. Otherwise do not change $k(v)$. This update can be done in constant time per removed vertex w and remaining vertex v. ◀

6 Constrained Outer-1-String Representations for Simple Cycles

An *extended complement of a 5-cycle* (Figure 15a) is the complement of an anchor-ordered 5-cycle or a subpath $w_1v'vuu'w_2$ of a cycle whose anchors are in the order $w_1uv'u'vw_2$. A cyclically ordered cycle admits a constrained outer-1-string representation if and only if it neither contains a pair of interleaved independent edges nor an extended complement of a 5-cycle:

▶ **Theorem 21.** *Let $G = (V, E)$ be a simple cycle and let $\circlearrowright$ be a cyclic order of V. Then the following are equivalent.*

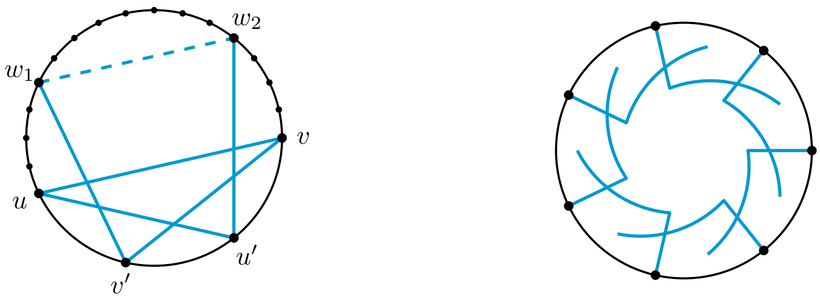
1. *$(G, \circlearrowright)$ has a constrained outer-1-string representation.*
2. *For every path $\langle u'uvv' \rangle$ of G, at least one among the sequences uv, $uu'v'v$, $uu'v$, or $uv'v$, or their reverse is a subsequence of $\circlearrowright$.*
3. *$(G, \circlearrowright)$ does not contain two interleaved independent edges nor an extended complement of a 5-cycle.*

Proof. We show equivalence of (1) and (3) to (2).

1 ⇒ 2: Let P be the path obtained from G after removing u, v, u', and v'. Let $\circlearrowright_1$ and $\circlearrowright_2$ be the subsequences obtained by splitting $\circlearrowright$ at the anchors of u and v. Then the anchors of P are either all in $\circlearrowright_1$ or all in $\circlearrowright_2$: If there were two adjacent vertices v_1 and v_2 in P such that the anchor of v_i is in $\circlearrowright_i$, $i = 1, 2$ Then $\{u, v\}$ and $\{v_1, v_2\}$ would be two interleaved independent edges.
So assume that the anchors of P are in $\circlearrowright_1$, i.e., $\circlearrowright_2$ contains either nothing, or the anchor of u', of v', or of both. If $\circlearrowright_1$ is not empty and $\circlearrowright_2$ contains the anchor of u' and of v', then u' must be next to u: The curve $\partial(u')$ must intersect $\partial(u)$ and reach the curve of the neighbor of u' in P in $\circlearrowright_1$. Similarely, the curve $\partial(v')$ must intersect $\partial(v)$ and reach the curve of the neighbor of v' in P in $\circlearrowright_1$. This is impossible if the order is $uv'u'v$.

3 ⇒ 2: Let again P, $\circlearrowright_1$, and $\circlearrowright_2$ be defined as above. As above the anchors of P are either all in $\circlearrowright_1$ or all in $\circlearrowright_2$. The fact that there is no extended complement of a 5-cycle forbids the sequence $wuv'u'v$ for any neighbor w of v' or u' other than v or u.

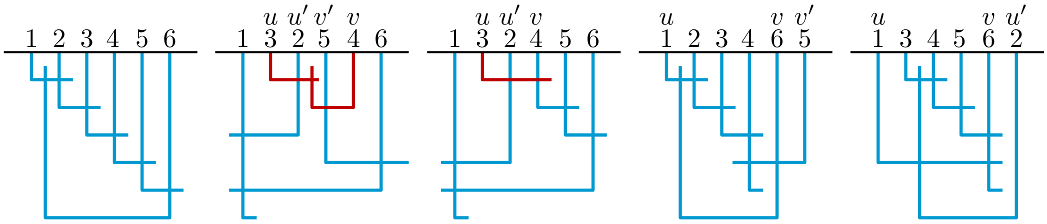
2 ⇒ 3: Assume there were two independent interleaved edges $\{u, v\}$ and $\{x, y\}$. Then one subsequence of $\circlearrowright$ between u and v would contain x and the other y. But neither x nor y is a neighbor of u or v. Assume now that there is an extended complement of a 5-cycle. Then there is the sequence $wuv'u'v$ for some neighbor $w \neq v$ of v'.



(a) Extended complement of a 5-cycle.

(b) Representation of a simple cycle.

Figure 15 a) An obstruction for simple cycles. $w_1 = w_2$ is also possible. b) How to draw a simple cycle if for any edge the anchors of the two endvertices are next to each other.



(a) Case 1. **(b)** Case 2a. **(c)** Case 2a. **(d)** Case 2b(i). **(e)** Case 2b(ii).

Figure 16 How to draw a simple cycle as a constrained U-shaped outer-1-string representation. If not all vertices of the cycle are consecutive in the cyclic order of the anchors, we remove one or two vertices (indicated in red), draw the resulting path and reinsert the vertices.

2 ⇒ 1: If the anchors for any pair of adjacent vertices u, v of G form a subsequence of $\circlearrowright$ then there is a constrained outer-1-string representation. See Figure 15b.
Assume now that G contains a path $u'uvv'$ such that $\circlearrowright$ contains $uu'v$ as a subsequence (the case $uv'v$ or their reverse being symmetric). Let P be the path obtained from G by removing u. Since Item 2 implies Item 3, there are no two interleaved independent edges. Thus, P has a constrained outer-1-string representation by Corollary 17. Route $\partial(u)$ closely along the border of the disk until it intersects first $\partial(u')$ and then $\partial(v)$.
Assume now that G contains a path $u'uvv'$ such that the anchors are in the order $uu'v'v$. Let P be the path obtained from G by removing u and v. Again by Corollary 17, it follows that P has a constrained outer-1-string representation. Route $\partial(u)$ and $\partial(v)$ closely along the border of the disk until they intersect between the anchor of u' and v'. ◀

The second condition of Theorem 21 can be tested in constant time per edge.

▶ **Corollary 22.** *It can be tested in linear time, whether a simple cycle with a given cyclic order of the vertices admits a constrained outer-1-string representation.*

Simple cycles do not necessarily have a constrained L-shaped outer-1-string representation even if the respective cyclic order of the anchors admits a constrained outer-1-string representation, consider for example 12345, 13452, or 34127856. The existence of a constrained outer-1-string representation follows from Theorem 21 and the non-existence of an L-shaped outer-1-string representation follows from Lemma 19. However, U-shapes suffice.

▶ **Corollary 23** ($\star$). *Each simple cycle with a given cyclic order $\circlearrowright$ of the vertices that admits a constrained outer-1-string representation also has a constrained U-shaped outer-1-string representation with respect to any linear order induced by $\circlearrowright$.*

Sketch of Proof. We follow "2 ⇒ 1" in the proof of Theorem 21. We distinguish whether any two adjacent vertices are consecutive in ⟳ (Figure 16a), some adjacent vertices contain some neighbors between them (Figures 16b and 16c), or an adjacent pair contains everything but a neighbor between them (Figures 16d and 16e). ◀

7 Conclusion

We considered outer-string and outer-1-string representations of graphs in which the sequence of the anchors of the vertices was fixed. In particular, we considered outer-string representations of chordal graphs, outer-1-string representations of trees and cycles, as well as L-shaped representations of general graphs. We leave some interesting open problems.

What is the complexity of testing whether a graph has an outer-1-string, a constrained outer-1-string, or a constrained outer-string representation? Can these problems be efficiently solved for cacti or graphs with constant treewidth? Can it be tested efficientlywhether an ordered graph admits a constrained U-shaped outer-1-string representation?

A variant of the problem would be to specify for each vertex a set of anchors and to require that these points are within its curve. What can be said about this variant?

References

1 Therese Biedl, Ahmad Biniaz, and Martin Derka. On the size of outer-string representations. In *16th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT*, volume 101 of *LIPIcs*, pages 10:1–10:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. `doi:10.4230/LIPICS.SWAT.2018.10`.

2 Therese Biedl and Martin Derka. Order-preserving 1-string representations of planar graphs. In *SOFSEM 2017: Theory and Practice of Computer Science*, volume 10139 of *Lecture Notes in Computer Science*, pages 283–294. Springer, 2017. `doi:10.1007/978-3-319-51963-0_22`.

3 Cornelia Dangelmayr, Stefan Felsner, and William T. Trotter. Intersection graphs of pseudosegments: Chordal graphs. *J. Graph Algorithms Appl.*, 14(2):199–220, 2010. `doi:10.7155/JGAA.00204`.

4 James Davies, Tomasz Krawczyk, Rose McCarty, and Bartosz Walczak. Grounded L-graphs are polynomially χ-bounded. *Discret. Comput. Geom.*, 70(4):1523–1550, 2023. `doi:10.1007/S00454-023-00592-Z`.

5 G.A. Dirac. On rigid circuit graphs. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 25:71–76, 1961. `doi:doi.org/10.1007/BF02992776`.

6 G. Ehrlich, S. Even, and R.E. Tarjan. Intersection graphs of curves in the plane. *Journal of Combinatorial Theory, Series B*, 21(1):8–20, 1976. `doi:10.1016/0095-8956(76)90022-8`.

7 F. Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *Journal of Combinatorial Theory Ser. B*, 16:47–56, 1974. `doi:10.1016/0095-8956(74)90094-X`.

8 Martin Charles Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Elsevier, 1980.

9 Vít Jelínek and Martin Töpfer. On grounded L-graphs and their relatives. *Electron. J. Comb.*, 26(3):P3.17, 2019. `doi:10.37236/8096`.

10 Jan Kratochvíl. String graphs. In *Graphs and Other Combinatorial Topics, Proceedings Third Czechoslovak Symposium on Graph Theory, Prague*, volume 59 of *Teubner Texte zur Mathematik*, pages 168–172. Teubner, Berlin, 1982.

11 Sean McGuinness. On bounding the chromatic number of L-graphs. *Discret. Math.*, 154(1–3):179–187, 1996. `doi:10.1016/0012-365X(95)00316-O`.

12 M. Middendorf and F. Pfeiffer. The max clique problem in classes of string graphs. *Discret. Math.*, 108(1–3):365–372, 1992. `doi:10.1016/0012-365X(92)90688-C`.

13 Arkadiusz Pawlik, Jakub Kozik, Tomasz Krawczyk, Michal Lason, Piotr Micek, William T. Trotter, and Bartosz Walczak. Triangle-free intersection graphs of line segments with large chromatic number. *J. Comb. Theory, Ser. B*, 105:6–10, 2014. doi:10.1016/J.JCTB.2013.11.001.

14 Alexandre Rok and Bartosz Walczak. Outerstring graphs are χ-bounded. *SIAM J. Discret. Math.*, 33(4):2181–2199, 2019. doi:10.1137/17M1157374.

15 F. W. Sinden. Topology of thin film RC circuits. *Bell System Tech. J.*, 45:1639–1662, 1966. doi:10.1002/j.1538-7305.1966.tb01713.x.

16 James R. Walter. Representations of chordal graphs as subtrees of a tree. *Journal of Graph Theory*, 2(3):265–267, 1978.