Rerouting Curves on Surfaces

Timo Brand
Technische Universität München, Germany

Stefan Felsner
Technische Universität Berlin, Germany

Henry Förster
John Cabot University, Rome, Italy
Technische Universität München, Germany
Universität Tübingen, Germany

Stephen Kobourov
University of Arizona, Tucson, AZ, USA
Technische Universität München, Germany

Anna Lubiw
University of Waterloo, Canada

Yoshio Okamoto
The University of Electro-Communications, Chofu, Japan

János Pach
Rényi Institute of Mathematics, Budapest, Hungary
EPFL, Lausanne, Switzerland

Csaba D. Tóth
Cal State Northridge, Los Angeles, CA, USA
Tufts University, Medford, MA, US

Géza Tóth
Rényi Institute of Mathematics, Budapest, Hungary
Budapest Univ. of Technology, Hungary

Torsten Ueckerdt
Karlsruhe Institute of Technology, Germany

Pavel Valtr
Charles University Prague, Czech Republic

Abstract

We study the problem of reconfiguring a crossing-free embedding of a graph on a surface, with edges represented as curves, into another crossing-free embedding of the same graph on the same surface with the same fixed vertex positions. In this process, we reroute one edge at a time while maintaining crossing-free intermediate embeddings. This problem was introduced by Ito et al. [TALG 2025], who showed that even if the graph is a matching of two edges, reconfiguration is not always possible in the plane, but is always possible on the torus. For matchings of two or more edges, they gave a necessary and sufficient condition for reconfigurable embeddings in the plane, but not on the torus. Our main result is that for matchings, trees and forests, reconfiguration is always possible on the torus, and consequently, on any orientable surface of genus at least one. In addition, we provide sufficient conditions for reconfiguration on orientable surfaces of genus at least one and in the projective plane. For more general graphs, we show that reconfiguration is not always possible.

2012 ACM Subject Classification Mathematics of computing → Geometric topology; Mathematics of computing → Graph algorithms; Mathematics of computing → Trees; Mathematics of computing → Combinatorial algorithms

Keywords and phrases Combinatorial reconfiguration, orientable surface, non-orientable surface, rerouting

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.28

Related Version *Full Version*: `https://arxiv.org/abs/2607.05362` [8]

Funding *Yoshio Okamoto*: Partially supported by JSPS KAKENHI Grant Numbers JP23K10982, JP26K23806, and JST ERATO Grant Number JPMJER2301.
János Pach: Supported by ERC Advanced Grant "GeoScape" No. 882971.
Csaba D. Tóth: Research supported in part by the NSF award DMS-2154347.
Géza Tóth: Supported by ERC Advanced Grant "GeoScape" No. 882971 and by the National Research, Development and Innovation Office, NKFIH, K-131529 and ADVANCED 152590.

Torsten Ueckerdt: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 520708409
Pavel Valtr: Supported by grant no. 23-04949X of the Czech Science Foundation (GAČR).

Acknowledgements This work started at Dagstuhl Seminar 24062 "Beyond-Planar Graphs: Models, Structures and Geometric Representations".

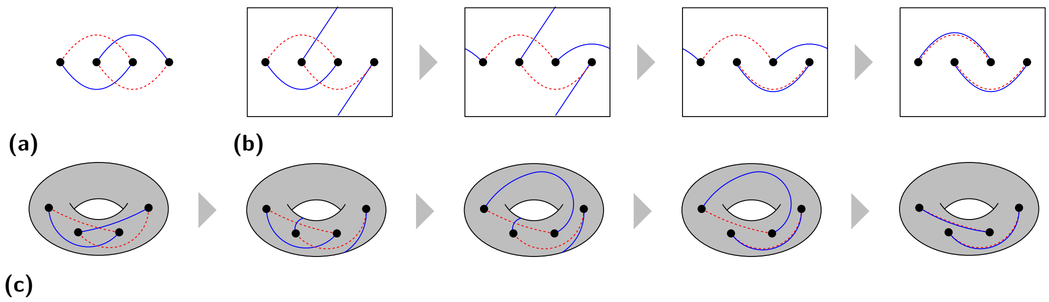

Figure 1 (a) Two embeddings (red & blue) that cannot be reconfigured on the plane. (b) Reconfiguration on the fundamental square. (c) Corresponding illustration on the torus.

1 Introduction

We study the problem of reconfiguring graph embeddings on a surface, where the vertices are fixed and each reconfiguration step redraws one edge (represented as a curve). Consider a set P of points on a surface Σ, and two embeddings $\mathcal{B}$ and $\mathcal{R}$ of the same simple graph G that share the same mapping of vertices to P. Here, an ***embedding*** means that each edge is drawn as a simple curve, which we call an ***edge curve***, on the surface, and no two edge curves intersect except at a common endpoint. The edge curves of $\mathcal{B}$ may cross the edge curves of $\mathcal{R}$ and the correspondence between edge curves of $\mathcal{B}$ and $\mathcal{R}$ is implied by the identical mapping of vertices. A ***reconfiguration step*** or ***move*** replaces one edge curve γ of an embedded graph G by a new curve γ' to obtain a new embedding of G; i.e., γ' may not cross any of the other edge curves in the embedding, though we allow γ and γ' to intersect. We address the question whether $\mathcal{B}$ can be reconfigured to $\mathcal{R}$ via a sequence of moves.

Ito et al. [22] showed that reconfiguration in the plane is not always possible, even if the graph is a matching of two edges; see Figure 1a. On the other hand, they proved that reconfiguration of a matching of two edges is always possible on a surface Σ of genus $g \geq 1$; see Figures 1b and 1c for an example on the torus. In this paper, we investigate the natural open problem of determining which classes of graphs can be reconfigured on surfaces of higher genus. In particular, we investigate general forests (including larger matchings).

Related Work. Rerouting curves on surfaces was introduced by Ito et al. [22]. They studied the case where the graph is a matching (a set of independent edges) and the surface is the plane (equivalently the sphere), for which they gave an algebraic condition characterizing the reconfigurable embeddings. This was a main step for studying the reconfigurability of vertex-disjoint paths in planar graphs. When the graph is a matching of two edges on a torus, they showed that reconfiguration is always possible, and left open the case of matchings of three or more edges. The problem of reconfiguring graph embeddings on a surface using elementary moves (studied in this paper) has obvious similarities to other problems in graph drawing and computational topology. We highlight key differences that may explain why the

Table 1 Our reconfiguration results and our open problems (numbered as in the list in Section 7).

GRAPH CLASS		ORIENTABLE	NON-ORIENTABLE
Forests	– *General Case*		Open Problem 1
	– *Perfect Matchings (arbitrary)*	Theorem 1	
	– *Perfect Matchings (inside disk)*		Theorem 8
Planar graphs	– *General Case*	Open Problems 2 & 3	
	– *with Fixed Rotation System*	Theorem 6	Open Problem 3
	– *Series-parallel graphs*	Theorem 7	
Negative examples	– *Same rotation system*		
	– *Different rotation systems, all implementing embeddings*	Theorem 11	

techniques developed for other problems do not seem to be helpful for our problem. The problem of ***morphing graph drawings on a surface*** [1, 9, 19] is different in that the vertices are allowed to move, and usually the edges must remain straight segments on the surface. The problem of ***tightening*** or ***untangling curves on a surface*** [10, 11, 12, 15] is also different in that they consider drawings with possible crossings (i.e., immersions rather than embeddings), and deform the edge curves continuously via so-called homotopy moves (local moves that modify the topology of the immersion). We also point the interested reader to Colin de Verdière's survey [14] on graphs on surfaces. Finally, our problem is a type of ***combinatorial reconfiguration*** problem; see [27, 30] for introductions to this field. Note that a reconfiguration of circle arrangements on orientable surfaces with moves that replace one circle with a transversely intersecting circle has been considered in [20, 23].

An overview of our results and open problems is presented in Table 1. Due to space constraints, some proof details are omitted and can be found in the full version [8].

2 Preliminaries

We study the reconfiguration of graphs embedded on a surface Σ where edges are represented as arcs. A curve on a surface Σ is a continuous function $\gamma : [0,1] \to \Sigma$. It is ***closed*** if $\gamma(0) = \gamma(1)$ and ***simple*** if γ is injective on $[0,1)$. An ***arc*** on Σ is the image of a simple curve $\gamma : [0,1] \to \Sigma$ such that $\gamma(0) \neq \gamma(1)$. The points $\gamma(0)$ and $\gamma(1)$ are called the endpoints of the arc. We will always consider simple curves unless mentioned otherwise.

Let $G = (V, E)$ be a simple graph with n vertices $V = \{v_1, \dots, v_n\}$ and with m edges $E \subseteq \binom{V}{2}$. Let Σ be a surface and let $P = \{p_1, \dots, p_n\} \subset \Sigma$ be a set of n distinct points on Σ. A mapping $\mathcal{E}$ defined on $V \cup E$ is a ***Σ-embedding*** of G on P if **1.** $\mathcal{E}(v_i) = p_i$ for $i \in \{1, \dots, n\}$, **2.** $\mathcal{E}(e)$ is a simple open arc on Σ with endpoints $\mathcal{E}(u)$ and $\mathcal{E}(v)$, called an ***edge curve***, for $e = uv \in E$, **3.** $\mathcal{E}(e) \cap \mathcal{E}(f) \subseteq \mathcal{E}(V)$ for $e, f \in E$, $e \neq f$. For simplicity, we use v_i and p_i interchangeably and denote by $\mathcal{E}$ both the mapping and its image on Σ. Let G' be a proper subgraph of a graph G. A Σ-embedding of G' is called a ***partial*** Σ-embedding of G; that is, it describes an embedding of only a subset of the edges of G. Given a Σ-embedding $\mathcal{E}$ of G, we denote by $\mathcal{E}[G']$ the restriction $\mathcal{E}$ to the subgraph G', i.e., $\mathcal{E}[G']$ contains exactly the edge curves of $\mathcal{E}$ corresponding to edges in G'. The ***rotation system*** of an embedding $\mathcal{E}$ specifies the clockwise cyclic order of edges incident to each vertex.

We investigate embeddings of a graph G where the mapping of vertices to points is fixed, that is, we consider Σ-embeddings $\mathcal{E}$ and $\mathcal{F}$, where $\mathcal{E}(v_i) = \mathcal{F}(v_i) = p_i$. Two such embeddings are ***adjacent*** if there is exactly one edge $e^* \in E$ such that $\mathcal{E}(e^*) \neq \mathcal{F}(e^*)$. Embedding $\mathcal{E}$ is ***reconfigurable*** to $\mathcal{F}$ if there is a ***reconfiguration sequence*** $(\mathcal{E}_0 = \mathcal{E}, \mathcal{E}_1, \dots, \mathcal{E}_k = \mathcal{F})$ from $\mathcal{E}$ to $\mathcal{F}$ where each $\mathcal{E}_i$ is a Σ-embedding of G on P for $i \in \{0, \dots, k\}$, and $\mathcal{E}_i$ is adjacent to $\mathcal{E}_{i+1}$ for $i \in \{0, \dots, k-1\}$. We study the following Σ-`Embedding Reconfiguration` problem:

> Σ-`Embedding Reconfiguration`
> **Input.** A graph $G = (V, E)$ and two embeddings $\mathcal{B}$ and $\mathcal{R}$ of G on surface Σ.
> **Question.** Is $\mathcal{B}$ reconfigurable to $\mathcal{R}$?

Only edges are reconfigured and relevant for crossings; hence assume that G has no isolated vertices. Ito et al. [22] solved the problem when G is a matching and Σ is the plane by giving necessary and sufficient conditions for reconfiguration. Their characterization yields a polynomial-time algorithm in special cases. Their example of two disjoint edges (Figure 1) is the smallest example where reconfiguration is not possible in the plane. However, they show how to reconfigure embeddings of this 2-edge graph on any surface of higher genus.

We denote by Σ a compact surface without boundary. Such a surface Σ is an ***orientable surface of genus* 0** if it is homeomorphic to the sphere. An ***orientable surface of genus g*** is obtained from a surface Σ' of genus $g-1$ by removing two disjoint open disks from Σ', and identifying their boundaries with opposite orientations. The orientable surface of genus 1 is the ***torus***. Embeddings on the sphere and torus are called ***plane*** and ***toroidal***, respectively. There are also ***non-orientable surfaces***. The ***non-orientable surface of genus* 1** is the projective plane $\mathbb{R}P^2$. It is obtained from a sphere $\mathbb{S}^2$ by removing the interior of a disk D and identifying antipodal points on the boundary ∂D; this operation is called ***attaching a crosscap***. We refer to the image of ∂D in the quotient as the ***crosscap***. We say that an $\mathbb{R}P^2$-embedding of a graph G has a ***clean crosscap*** if the embedding of every edge of G is disjoint from the crosscap. For $g \geq 2$, the ***non-orientable surface of genus g*** is obtained from the non-orientable surface of genus $g-1$ by attaching a crosscap.

A closed curve γ on a surface Σ is ***separating*** if $\Sigma \setminus \gamma$ is disconnected; otherwise γ is ***non-separating***. The curve γ is ***contractible*** if it is homotopic to a point. A ***disk*** on a surface is the region bounded by a contractible closed curve γ. On the sphere, each closed curve is contractible and separating; on the torus, all separating closed curves are contractible [31].

The ***fundamental square*** $\bar{\Sigma}$ is the following quotient-space representation of the torus $\Sigma = \mathbb{S}^1 \times \mathbb{S}^1$: it is the quotient of a unit square $[0,1]^2$, where **1.** the left and right sides are identified such that $(0, y)$ is identified with $(1, y)$ for all $y \in [0,1]$, and **2.** the top and bottom sides are identified such that $(x, 0)$ is identified with $(x, 1)$ for all $x \in [0,1]$. We also call the boundary (interior) of the unit square the ***boundary*** (***interior***) of the fundamental square, and denote it $\partial\bar{\Sigma} = \partial[0,1]^2$ (resp., $\operatorname{int}\bar{\Sigma} = \operatorname{int}[0,1]^2$). Note that the boundary (interior) of the fundamental square is not the boundary (interior) in a topological sense; instead it is associated with the particular representation of the torus as a fundamental square. If $\mathcal{B}$ is an embedding on the torus Σ, then its image under this representation is called a ***$\bar{\Sigma}$-embedding***. More generally, if Σ is an orientable surface of genus $g \geq 1$, then Σ is homeomorphic to the quotient of a $4g$-gon, called a ***fundamental polygon***, whose sides are identified according to the standard scheme $a_1 b_1 a_1^{-1} b_1^{-1} \cdots a_g b_g a_g^{-1} b_g^{-1}$. For $g = 1$, this is exactly the square model of the torus. If no edge curve of $\mathcal{B}$ intersects $\partial\bar{\Sigma}$, equivalently if $\mathcal{B}$ is contained in $\operatorname{int}\bar{\Sigma}$, then we call $\mathcal{B}$ an ***interior $\bar{\Sigma}$-embedding***. In this case, $\mathcal{B}$ is contained in a topological disk and can be viewed as a planar embedding inside the chosen fundamental polygon.

3 Reconfiguring Embeddings of a Forest on the Torus

We describe an algorithm that constructs a reconfiguration sequence from a given source Σ-embedding $\mathcal{B}$ of a forest $F = (V, E)$ to a given target Σ-embedding $\mathcal{R}$ of F where Σ is an orientable surface of genus $g \geq 1$. For now, let Σ be a torus; later, we generalize to $g > 1$.

To formalize our result algorithmically and analyze the performance of our algorithm, we consider a ***geometric setting*** with the following input specifications:

1. $\mathcal{B}$ and $\mathcal{R}$ are given as $\bar{\Sigma}$-embeddings of Σ,
2. no vertex is on the boundary $\partial\bar{\Sigma}$,
3. each edge curve is a polyline of straight-line segments, and
4. each segment crosses $\partial\bar{\Sigma}$ at most once.

For an embedding $\mathcal{E}$, let $s(\mathcal{E})$ be its number of segments, and let $k(\mathcal{E})$ be the number of these segments that cross $\partial\bar{\Sigma}$. The length of the reconfiguration sequence and the runtime will depend on the parameters s and k of $\mathcal{B}$ and $\mathcal{R}$. Note that although $\mathcal{B}$ and $\mathcal{R}$ (blue and red in all figures) are each crossing-free $\bar{\Sigma}$-embeddings on Σ, $\mathcal{B}$ and $\mathcal{R}$ may cross each other. Our main result is:

▶ **Theorem 1.** *Let $F = (V, E)$ be a forest having $c = c(F)$ connected components. Let $\mathcal{B}$ and $\mathcal{R}$ be two $\bar{\Sigma}$-embeddings of F on the fundamental square $\bar{\Sigma}$. Suppose that the edges of F are represented by polylines in $\mathcal{B}$ and $\mathcal{R}$ and let $s = s(\mathcal{B}) + s(\mathcal{R})$ be the total number of segments in $\mathcal{B}$ and $\mathcal{R}$ and let $k = k(\mathcal{B}) + k(\mathcal{R})$ be the total number of segments in $\mathcal{B}$ and $\mathcal{R}$ that intersect the boundary $\partial\bar{\Sigma}$ of $\bar{\Sigma}$. Then there is a reconfiguration sequence from $\mathcal{B}$ to $\mathcal{R}$ of length $O(c3^k s^2)$. Each embedded graph in the sequence has $O(c^2 3^{2k} s^3)$ segments. There is an algorithm to compute the reconfiguration sequence in time proportional to the output size, where the output size is the total number of segments in an explicit list of all embeddings in the sequence.*

Our algorithm proving Theorem 1 consists of the following two phases.

- **Phase 1: Reconfiguring $\mathcal{B}$ and $\mathcal{R}$ to $\mathcal{B}^*$ and $\mathcal{R}^*$:** Separately reconfigure $\mathcal{B}$ and $\mathcal{R}$ to embeddings $\mathcal{B}^*$ and $\mathcal{R}^*$, respectively, such that no edge curve in $\mathcal{B}^*$ or $\mathcal{R}^*$ crosses the boundary $\partial\bar{\Sigma}$. (We completely ignore $\mathcal{R}$ when reconfiguring $\mathcal{B}$, and vice-versa.)
- **Phase 2: Reconfiguring $\mathcal{B}^*$ to $\mathcal{R}^*$:** Reconfigure $\mathcal{B}^*$ to $\mathcal{R}^*$ using the freedom to temporarily reconfigure edges to wrap around the boundary of $\bar{\Sigma}$.

We obtain the final sequence by first reconfiguring $\mathcal{B}$ to $\mathcal{B}^*$ (**Phase 1**), then $\mathcal{B}^*$ to $\mathcal{R}^*$ (**Phase 2**), and finally $\mathcal{R}^*$ to $\mathcal{R}$ (**Phase 1**; reconfiguration sequences are reversible). For **Phase 1**, it suffices to show how to reconfigure $\mathcal{B}$ to $\mathcal{B}^*$, ignoring $\mathcal{R}$ ($\mathcal{R}$ to $\mathcal{R}^*$ is symmetric).

Phase 1. Reconfigure $\mathcal{B}$ to a $\bar{\Sigma}$-embedding $\mathcal{B}^*$ that does not intersect the boundary

▶ **Lemma 2.** *Let $\mathcal{B}$ be a $\bar{\Sigma}$-embedding of a forest $F = (V, E)$ on Σ. Let $s = s(\mathcal{B})$ and $k = k(\mathcal{B})$. Then there is a reconfiguration sequence of length $O(3^k)$ from $\mathcal{B}$ to a $\bar{\Sigma}$-embedding $\mathcal{B}^*$ of F that does not cross $\partial\bar{\Sigma}$. Each embedded graph in the sequence has $O(3^k s)$ segments. There is an algorithm that computes the reconfiguration sequence in time $O(3^{2k} s)$.*

Proof. We first prove the existence of $\mathcal{B}^*$ by induction and then prove the bounds and show how the proof yields an algorithm with the claimed runtime. Inspired by the machinery of weak embeddings by Chang, Erickson, and Xu [13], we use a sequence of embedded trees, $\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_k$, that we call the ***frame trees***. Each frame tree $\mathcal{F}_j$ is a Steiner tree: its vertices include some endpoints of segments of $\mathcal{B}$ and $2k$ additional points in $\bar{\Sigma} \setminus \partial\bar{\Sigma}$, at a fixed (small) distance $\delta > 0$ from $\partial\bar{\Sigma}$. Throughout the reconfiguration sequence, the rerouted

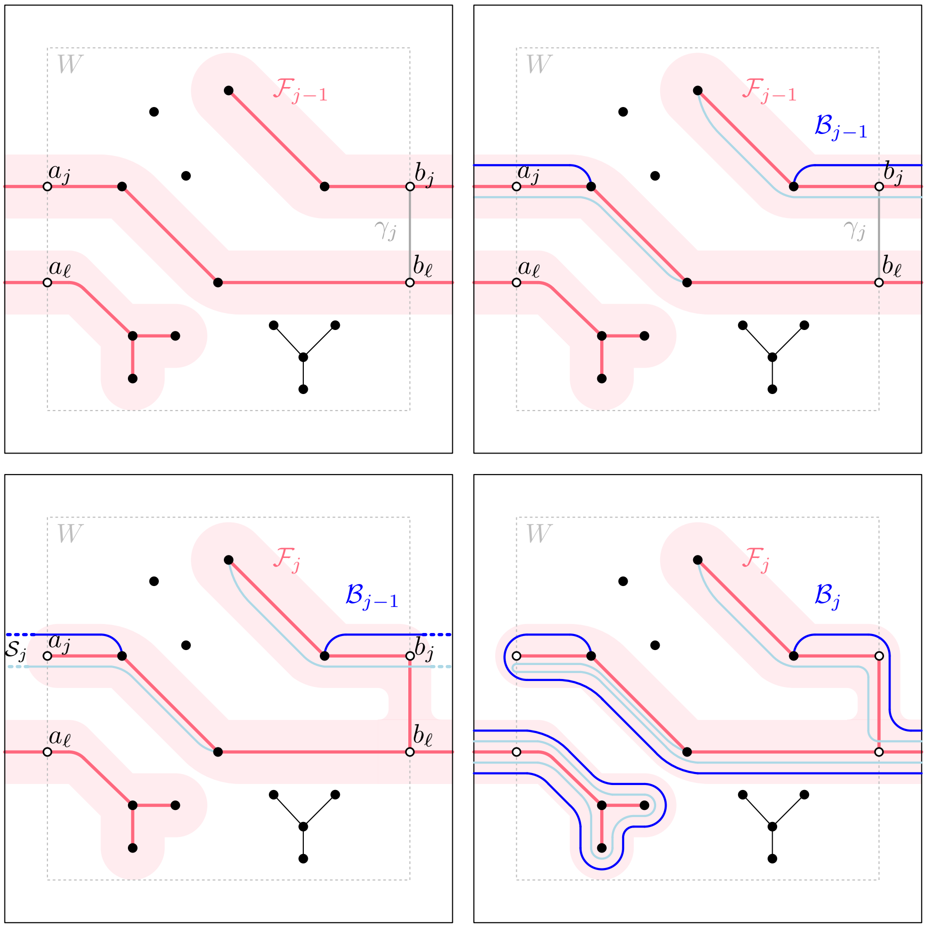

Figure 2 Phase 1: *Top-Left:* The smaller square W intersected by the segments crossing $\partial\bar{\Sigma}$ is shown dashed in gray, and frame trees $\mathcal{F}_{j-1}$ and $\mathcal{F}_j$ in pink, with their ε-neighborhoods in a lighter color. The frame tree $\mathcal{F}_{j-1}$ is cut by deleting the segment $f_j = a_j b_j$ (eliminating one crossing with $\partial\bar{\Sigma}$) and then reconnected by adding the candidate curve γ_j shown in gray. *Top-Right:* In the embedding $\mathcal{B}_{j-1}$, edges may be routed in the ε-neighborhood of the segment f_j (see the cyan and blue edge for two examples). *Bottom-Left:* After updating the frame tree to F_j, the segments of such edges in the ε-neighborhood of f_j (drawn dashed) must be reconfigured. *Bottom-Right:* The two edges that originally crossed W in the embedding $\mathcal{B}_{j-1}$ (cyan and blue) are reconfigured, first cyan, then blue, to obtain the embedding $\mathcal{B}_j$.
Note that the polylines representing the edges in the embeddings cannot cross themselves and each other, but they can cross the frame tree; in the figure, insertion of the candidate curve γ_j to the frame tree is responsible for the intersection of the frame tree with the cyan polyline near b_j. Moreover, observe that the parts of $\mathcal{F}$ not intersecting W are not in the neighborhood of $\mathcal{F}_{j-1}$ as they are not part of $\mathcal{B}_0$.

edge curves of $\mathcal{B}$ stay in a small neighborhood of the edges of one of $\mathcal{F}_0, \ldots, \mathcal{F}_k$. At the end of the reconfiguration process, they lie in a small neighborhood of the edges of the last frame tree $\mathcal{F}_k$, which the construction below shows does not cross $\partial\bar{\Sigma}$. Consequently, the rerouted edge curves in their final position also do not cross $\partial\bar{\Sigma}$. An edge curve of $\mathcal{B}$ may spiral around $\mathcal{F}_j$ multiple times.

If $k(\mathcal{B}) = 0$, then the lemma holds with $\mathcal{B}^*$ set to $\mathcal{B}$. In the following, we assume that $k > 0$. We fix $\delta > 0$ small enough such that all vertices of $\mathcal{B}$ (including the isolated vertices) lie at a distance larger than δ from $\partial\bar{\Sigma}$. Thus, all vertices of $\mathcal{B}$ lie inside the square $W \coloneqq [\delta, 1-\delta]^2$. Let $\mathcal{B}_0 \subseteq \mathcal{B}$ be the embedded forest obtained from $\mathcal{B}$ by deleting all connected components that do not cross $\partial\bar{\Sigma}$. Observe that the region $\bar{\Sigma} \setminus W$ is intersected only by those $k = k(\mathcal{B}) = k(\mathcal{B}_0)$ segments of $\mathcal{B}_0$ that cross $\partial\bar{\Sigma}$. Let $s_1, \ldots, s_k$ denote these k segments, in arbitrary order. Recall that we are in the geometric setting, where, in particular, we assume that in the input embedding each segment crosses the boundary at most once. Our goal is to reroute the segments $s_1, \ldots, s_k$ to new polylines (one per segment), called their ***polyline representations***, which will all lie in the union of small ε-neighborhoods[1] of the other segments of $\mathcal{B}_0$ and of ∂W. Each s_i intersects ∂W in two points, a_i and b_i. We set $A := \{a_1, \ldots, a_k, b_1, \ldots, b_k\}$. The $2k$ points of A partition ∂W into $2k$ polylines (each consisting of at most three axis-parallel segments), which we further call ***candidate curves***.

We obtain the initial (embedded) frame tree $\mathcal{F}_0$ from the forest $\mathcal{B}_0$ by subdividing each segment s_i by the two vertices a_i and b_i into three segments, and then adding a suitable set of candidate curves to ensure that $\mathcal{F}_0$ is connected. This set of candidate curves can be chosen, for example, greedily by considering them one by one and inserting each if and only if its insertion does not create a cycle. We further inductively construct an auxiliary sequence $\mathcal{F}_1, \ldots, \mathcal{F}_k$ of k frame trees having a gradually decreasing number of crossings with $\partial\bar{\Sigma}$, until the last of them $\mathcal{F}_k$ does not cross $\partial\bar{\Sigma}$ at all. Later, we will show how to use these trees to inductively redraw the segments s_i so that their final polyline representations lie in the union of the small neighborhoods of the edges of $\mathcal{F}_k$, and thus not cross $\partial\bar{\Sigma}$. For $j \in \{1, \ldots, k\}$, we obtain $\mathcal{F}_j$ from $\mathcal{F}_{j-1}$ by deleting the segment $f_j := a_j b_j$ and adding any candidate curve γ_j connecting the two different components of $\mathcal{F}_{j-1} - f_j$. Such a candidate curve must exist since the candidate curves form a cycle which intersects each of the two components of $\mathcal{F}_{j-1} - f_j$ (for example, in a_j and in b_j, respectively). The last frame tree, $\mathcal{F}_k$, indeed does not cross $\partial\bar{\Sigma}$, as it contains none of the segments $f_j = a_j b_j$.

It remains to describe how the frame trees $\mathcal{F}_0, \ldots, \mathcal{F}_k$ are used to change the polyline representations for the segments s_i in k steps so that after the last step none of them crosses $\partial\bar{\Sigma}$. We remark that the other polyline representations are not changed during the process, thus none of them crosses $\partial\bar{\Sigma}$ either. During the process, we keep the invariant that after the j-th step ($j \in \{1, \ldots, k\}$), the polyline representation of every segment s_i, $i \in \{1, \ldots, k\}$, lies in the union of the small neighborhoods of the edges of the frame tree $\mathcal{F}_j$.

We now describe the j-th step. For every $i \in \{1, \ldots, k\}$, the polyline representation of s_i lies in the union of small neighborhoods of the edges of $\mathcal{F}_{j-1}$. Let ρ_j be the boundary of a sufficiently small neighborhood of the frame tree $\mathcal{F}_j$. Since $\mathcal{F}_j$ is a tree, ρ_j is a contractible closed curve. We define $\mathcal{S}_j$ to be the set of all connected components of the intersections of the current polyline representations of the segments s_i (after the first $j-1$ steps) with the exterior of ρ_j. In other words, each element of $\mathcal{S}_j$ is a maximal portion of some current polyline representation that lies outside ρ_j (see dashed blue and cyan segments in Figure 2). These curve portions cross $\partial\bar{\Sigma}$ near f_j (the edge of $\mathcal{F}_{j-1} \setminus \mathcal{F}_j$), and connect points (their "raw ends") on ρ_j near a_j with points (their "raw ends") on ρ_j near b_j. If f_j crosses a vertical part of $\partial\bar{\Sigma}$, then these raw ends appear in the same bottom-to-top order near a_j and near b_j. Otherwise, they appear in the same left-to-right order near a_j and near b_j. We choose arbitrarily one of the two possible directions along ρ_j from the vicinity of a_j to the vicinity of b_j. We reroute the curve portions in $\mathcal{S}_j$ so that they are newly drawn as polylines next to each other along this direction in an ε-neighborhood of ρ_j, and each of them connects

[1] In a slight abuse of notation, we will denote by ε a suitably small distance without explicitly specifying its value. Observe that ε is not constant: Most occurrences of ε could be numbered consecutively $\varepsilon_1, \varepsilon_2, \ldots$ so that for each i, we have $\varepsilon_{i+1} \geq \varepsilon_i$, and the last (biggest) one of them is still very small.

the corresponding pair of raw ends (one near a_j and the other near b_j); see Figure 2. Due to the above-mentioned consistency in the bottom-to-top or left-to-right order, and since ρ_j is a simple closed curve on an orientable surface, a sufficiently small neighborhood of ρ_j is homeomorphic to an annulus. Hence, the rerouted curve portions can be drawn there as pairwise disjoint parallel polylines, without crossing each other or any polyline representation in the current drawing. Thus, we obtain $\mathcal{B}_j$ from $\mathcal{B}_{j-1}$ by the rerouting of the segments $\mathcal{S}_j$; see Figure 2. This completes the inductive description of how to reconfigure $\mathcal{B}$ to $\mathcal{B}_k = \mathcal{B}^*$. We discuss the number of moves and the runtime in the full version [8, Appendix A]. ◀

Phase 2. Use the boundary of the fundamental square to solve the problem

After Phase 1, no edge of $\mathcal{B}^*$ or $\mathcal{R}^*$ crosses the boundary of $\bar{\Sigma}$. In Phase 2 we reconfigure $\mathcal{B}^*$ to $\mathcal{R}^*$. We first consider the case where F is a tree and then generalize to forests.

▶ **Lemma 3.** *Let $\mathcal{B}^*$ be a partial $\bar{\Sigma}$-embedding of subforest F of a tree $T = (V, E)$ and $\mathcal{R}^*$ be a $\bar{\Sigma}$-embeddings of T such that no edge curve of $\mathcal{B}^*$ or $\mathcal{R}^*$ crosses the boundary of the fundamental square $\bar{\Sigma}$. Let χ be the number of crossings between $\mathcal{B}^*$ and $\mathcal{R}^*$ and $s^* = s(\mathcal{B}^*)+s(\mathcal{R}^*)$. There is a reconfiguration sequence from $\mathcal{B}^*$ to the restriction $\mathcal{R}^*[F]$ of $\mathcal{R}^*$ to F of length $O(\chi+s^*)$. Each embedded graph in the sequence has at most $O(s^*(\chi+s^*))$ segments. There is an algorithm to compute the reconfiguration sequence in time $O(s^*(\chi+s^*)^2)$.*

Proof. For the reconfiguration algorithm of Phase 2, we iterate over the edges e of T in leaf-to-root order after choosing an arbitrary root for T (i.e., each edge is processed only after all edges in the subtree below have been processed, for example, by processing the tree in a post-order traversal). In each stage, we ***fix*** e, meaning that we reconfigure the edge curve $b = \mathcal{B}^*(e)$ to match $r = \mathcal{R}^*(e)$, or – if b does not exist (which may happen if $\mathcal{B}^*$ is a partial embedding which we make use of in Lemma 4) – just ensure that no edge curve of $\mathcal{B}^*$ crosses r. After each stage, we ensure that no edge curve of $\mathcal{B}^*$ crosses the top/bottom boundary of $\bar{\Sigma}$, but allow right/left crossings. Embedding $\mathcal{R}^*$ will not change.

The general subproblem is as follows. We have an edge $e = xy$ directed toward the root of T, with corresponding edge curves $r = \mathcal{R}^*(e)$ and $b = \mathcal{B}^*(e)$ (if it exists). Every edge f in the subtree rooted at x is a ***fixed*** edge (that is, no edge of $\mathcal{B}^*$ crosses $\mathcal{R}^*(f)$ and, if $\mathcal{B}^*(f)$ exists, then $\mathcal{B}^*(f) = \mathcal{R}^*(f)$). No edge curve of $\mathcal{R}^*$ crosses $\partial\bar{\Sigma}$. No edge curve of $\mathcal{B}^*$ crosses the top/bottom of $\bar{\Sigma}$. The goal is to fix edge e while reconfiguring only non-fixed edges of $\mathcal{B}^*$.

In order to fix e, we must eliminate the crossings of r with edge curves of $\mathcal{B}^*$. The basic idea is to reroute the edge curves at crossings to detour around $\mathcal{R}^*_x$, the $\mathcal{R}^*$ embedding of the subtree rooted at x. This is not immediately possible if b exists, since such a detour would cross b where it is incident to x. For example, in the first pane of Figure 3, $\mathcal{B}^*(yz)$ crosses r, but rerouting it around $\mathcal{R}^*_x$, as shown in the last pane of the figure, would introduce a crossing with b. Furthermore, we know that we must make use of the torus. We therefore begin by rerouting b (if it exists) around the torus. There are three steps (see Figure 3).

1. Reconfigure b to a ***desire path***, an edge curve d from x to y that initially goes alongside r, never crosses r, and crosses the boundary of $\bar{\Sigma}$ once along the top/bottom boundary.
2. Eliminate any crossing between r and an edge curve of $\mathcal{B}^*$ by reconfiguring the $\mathcal{B}^*$ piece of the crossing to walk around $\mathcal{R}^*_x$.
3. Reconfigure d to r. (This step is trivial and not further discussed.)

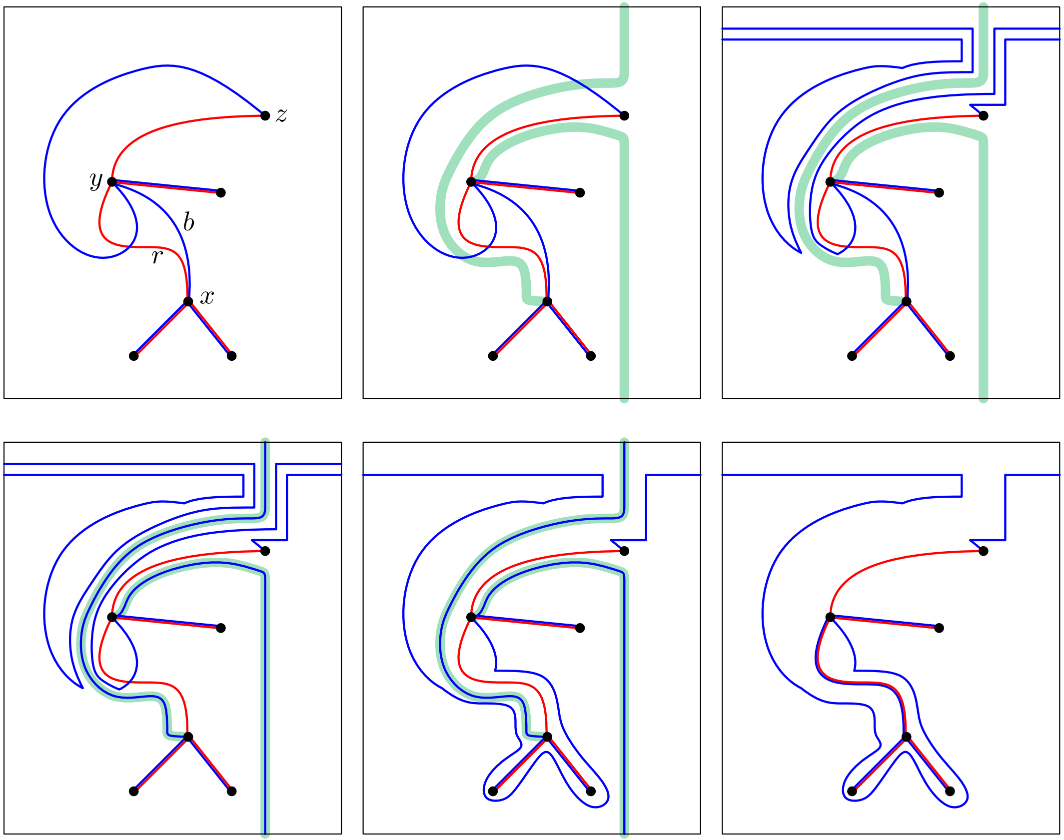

Figure 3 Phase 2 applied to edge $r = \mathcal{R}^*(xy)$ after fixing the subtree rooted at x. Top: input; Step 1a, choosing desire path d (green); Step 1b, eliminating the two crossings along d_x. Bottom: Step 1c, rerouting b to d; Step 2, eliminating the crossing along r; Step 3, reconfiguring d to r.

Step 1: Reconfigure b to a desire path d. d must not cross any blue edge curve, but in planning d, we initially allow it to cross non-fixed blue edge curves and then reroute those to eliminate the crossings. Although not strictly necessary, we impose the stronger condition that the desire path d does not cross any edge curve of $\mathcal{R}^*$. We use three substeps.

1a. Construct d, allowing it to cross edge curves of $\mathcal{B}^*$ but not $\mathcal{R}^*$.
1b. Eliminate any crossing between d and an edge curve of $\mathcal{B}^*$ by reconfiguring the $\mathcal{B}^*$ piece of the crossing to a curve that crosses the left/right boundary of $\bar{\Sigma}$.
1c. Reconfigure b to d.

Step 1a: Construct d. Let ρ_0 be the boundary of an ε-neighborhood of $\mathcal{R}^*$, where $\varepsilon > 0$ is chosen small enough that ρ_0 is simple (i.e., non-crossing). Near the root vertex z of the tree $\mathcal{R}^*$, remove a tiny arc of ρ_0 between two points z_1 and z_2 (in clockwise order around ρ_0) located just before and just after the closest approach of ρ_0 to z; by construction of ρ_0 as the boundary of a neighborhood, this arc does not cross $\mathcal{R}^*$. Augment the resulting path $\rho_0 \setminus z_1z_2$ to a closed curve ρ by finding a path starting at z_1, traveling in the interior of $\bar{\Sigma} \setminus \mathcal{R}^*$ to a point on the top boundary of $\bar{\Sigma}$, crossing to the bottom boundary, and then traveling in the interior of $\bar{\Sigma} \setminus \mathcal{R}^*$ to z_2. See Figure 4 (left). Note that ρ is a non-separating

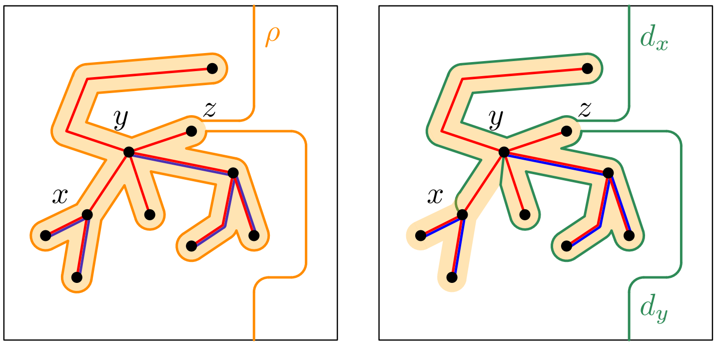

Figure 4 Step 1a: constructing d for tree $\mathcal{R}^*$ rooted at z. (left) The neighborhood of $\mathcal{R}^*$ (shaded yellow) and the curve ρ (yellow). (right) For edge xy, the two parts d_x and d_y of the desire path d.

curve (w.r.t. Σ) and has $O(s(\mathcal{R}^*))$ segments – we will use this in our analysis at the end of the proof. In fact, we can choose ρ so that all but $O(1)$ of its segments travel alongside edge curves of $\mathcal{R}^*$, with each segment of $\mathcal{R}^*$ traversed $O(1)$ times.

Desire path d follows ρ in the direction that keeps $\mathcal{R}^*$ on the right (from z_2 to z_1 along ρ_0, then across the top/bottom of $\bar{\Sigma}$), from x to y. Since ρ visits a neighborhood of each vertex v $\deg(v)$ times, we must specify this more precisely. If we follow ρ_0 in this direction from z_2 to z_1, the first visit to any vertex follows the first (downward) traversal of its parent edge, and the last visit to any vertex precedes the second (upward) traversal of its parent edge. To construct d, we join x to ρ_0 at the last visit of the edge $\{x, y\}$ (counterclockwise of r), then follow ρ past z_1, across the top/bottom of $\bar{\Sigma}$, and past z_2, joining to y at the last visit of the edge $\{x, y\}$; see Figure 4 (right). Observe that d starts along r, does not cross $\mathcal{R}^*$, and crosses the boundary of $\bar{\Sigma}$ once on the top/bottom; i.e., d is valid.

Step 1b. Eliminate crossings between d and $\mathcal{B}^*$. First, we resolve crossings on the directed portion d_x from x to the top of $\bar{\Sigma}$ in order, starting with the last one. Suppose d_x crosses an edge curve $\mathcal{B}^*(f)$ at point p; see Figure 5. Add new vertices p_L and p_R to $\mathcal{B}^*(f)$ just to the left and right of d_x. Replace the part of $\mathcal{B}^*(f)$ from p_L to p_R by a curve traveling forward left of d_x to just before the top of $\bar{\Sigma}$, then going left to the left boundary of $\bar{\Sigma}$, crossing to the other side, and continuing to just before d_x, then traveling backward left of d_x until reaching p_R. This curve does not cross d or any curve of $\mathcal{B}^*$ and each successively rerouted curve will "nest inside" the previous ones, i.e., lie closer to d and closer to the top boundary of $\bar{\Sigma}$. After eliminating all crossings along d_x, eliminate crossings on the portion of d_y from y to the bottom of $\bar{\Sigma}$ by rerouting them to cross the left/right boundary of $\bar{\Sigma}$ near the bottom.

Step 1c. Reconfigure b to d. As d is not crossed anymore, we can now reconfigure b to d.

Step 2. Eliminate any crossing between r and an edge curve of $\mathcal{B}^*$. Direct r from x to y and eliminate crossings in order, starting with the first one. Suppose that r crosses an edge curve $\mathcal{B}^*(f)$ at point p. See Figure 5. If b does not exist, then let p_L and p_R be points on $\mathcal{B}^*(f)$ just to the left and right of p. Otherwise (if b exists), we chose d in Step 1a to start at x and hug the edge curve r, so we know that $\mathcal{B}^*(f)$ used to cross d, and was rerouted in Step 1b to eliminate that crossing. Let p_L be the point on $\mathcal{B}^*(f)$ where we diverted it to eliminate the crossing with d. Let p_R be a point on $\mathcal{B}^*(f)$ just to the right of r. Having specified p_L and p_R, we replace the part of $\mathcal{B}^*(f)$ from p_L to p_R (which may detour along d and loop

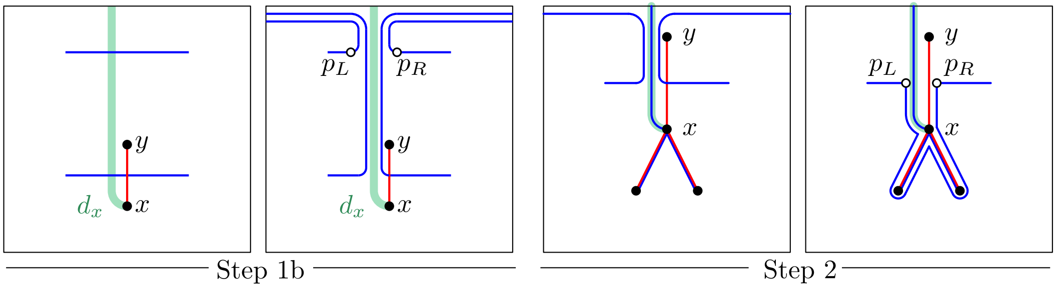

Figure 5 Steps 1b and 2: Eliminating crossings between $\mathcal{B}^*$ and d_x and r, respectively; shown schematically. Step 1: (left) d_x is crossed by two segments of $\mathcal{B}^*$. (right) The crossings are rerouted to go around the left/right boundary of $\bar{\Sigma}$ near the top. Step 2: (left) r is crossed by a segment of $\mathcal{B}^*$. (right) The crossed segment is rerouted to go around the fixed subtree rooted at x.

around the left/right boundary of $\bar{\Sigma}$ near the top) by a curve that goes counterclockwise around $\mathcal{R}^*_x$, the fixed subtree rooted at x. This curve does not cross any curve of the current $\mathcal{B}^*$ and each successive rerouted curve will nest inside the previous ones, i.e., lie closer to $\mathcal{R}^*_x$.

This completes the description of the reconfiguration algorithm. We describe how to reuse the same ρ in each step and discuss the resulting complexity in the full version [8, Appendix A]. ◀

▶ **Lemma 4.** *Let $\mathcal{B}^*$ and $\mathcal{R}^*$ be two $\bar{\Sigma}$-embeddings of a forest $F = (V, E)$ such that no edge curve of $\mathcal{B}^*$ or $\mathcal{R}^*$ crosses the boundary of the fundamental square $\bar{\Sigma}$. Let c be the number of connected components of F, let χ be the number of crossings between $\mathcal{B}^*$ and $\mathcal{R}^*$, and let $s^* = s(\mathcal{B}^*) + s(\mathcal{R}^*)$. Then there is a reconfiguration sequence from $\mathcal{B}^*$ to $\mathcal{R}^*$ of length $O(c(\chi + s^*))$. Each embedded graph in the sequence has at most $O(c^2 s^*(\chi + s^*))$ segments. There is an algorithm to compute the reconfiguration sequence in time $O(c^3 s^*(\chi + s^*)^2)$.*

Proof Sketch. We recursively augment F to a tree T and simultaneously augment $\mathcal{R}^*$ to an embedding $\mathcal{R}^A$ of T. In each step, we reduce the number of connected components by adding an augmenting edge e_a that closely follows previously existing edge curves and one of the noncrossing triangulating edges chosen to connect the components of the planar straight-line graph induced by $\mathcal{R}^*$. Then, we apply Lemma 3. In the process, we ignore the reconfiguration of augmenting edges, which have no corresponding curve in $\mathcal{B}^*$; i.e., $\mathcal{B}^*$ is a partial $\bar{\Sigma}$-embedding of T. See the full version [8, Appendix A] for details. ◀

Proof of Theorem 1. From Phase 1 (Lemma 2) $\mathcal{B}^*$ has $O\left(3^{k(\mathcal{B})} s(\mathcal{B})\right)$ segments and similarly for $\mathcal{R}^*$. In the worst case, every segment of $\mathcal{B}^*$ crosses every segment of $\mathcal{R}^*$, so χ is in $O(3^k s^2)$, where $k = k(\mathcal{B}) + k(\mathcal{R})$ and $s = s(\mathcal{B}) + s(\mathcal{R})$. Also s^* is in $O(3^k s)$. In Phase 2 (Lemma 4) the number of steps is $O(c(\chi + s^*))$ which is $O(c3^k s^2)$. Each embedded graph in the sequence has $O(c^2 s^*(\chi + s^*)) = O(c^2 3^{2k} s^3)$ segments. If we output an explicit list of each embedding in the sequence, the output size is $O(c^3 3^{3k} s^5)$. If we only output the reconfigured portion of each edge curve after each step, the bound is $O(c^2 3^{2k} s^3)$. The exponential dependence on k arises only in Phase 1. In positive terms, our algorithm is fixed parameter tractable in k. ◀

We discuss the following generalization to orientable surfaces of higher genus in the full version [8, Appendix A].

▶ **Corollary 5.** *Let $F = (V, E)$ be a forest and let $\mathcal{B}$ and $\mathcal{R}$ be two $\bar{\Sigma}$-embeddings of F, where $\bar{\Sigma}$ is a fundamental polygon of genus $g \geq 1$. Let c be the number of connected components of F, $s = s(\mathcal{B}) + s(\mathcal{R})$ denote the total number of segments in $\mathcal{B}$ and $\mathcal{R}$ and* *let $k = k(\mathcal{B}) + k(\mathcal{R})$ denote the total number of segments in $\mathcal{B}$ and $\mathcal{R}$ that intersect $\partial\bar{\Sigma}$. Then there is a reconfiguration sequence from $\mathcal{B}$ to $\mathcal{R}$ of length $O(c3^k g^2 s^2)$. Each embedded graph in the sequence has $O(c^2 3^{2k} g^3 s^3)$ segments. There is an algorithm to compute the reconfiguration sequence in time proportional to the output size $O(c^3 3^{3k} g^5 s^5)$.*

4 Reconfiguration of Planar Graphs on Orientable Surfaces

We study generalizations to planar graphs embedded on an orientable surface of genus $g \geq 1$.

4.1 Reconfiguration with Fixed Rotation System

▶ **Theorem 6.** *Let $\mathcal{B}$ and $\mathcal{R}$ be two Σ-embeddings of a planar graph G on an orientable surface Σ of genus $g \geq 1$. Moreover, let $\mathcal{B}$ and $\mathcal{R}$ have the same rotation system and suppose that there is a planar embedding $\mathcal{E}$ of G with the same rotation system as $\mathcal{B}$ and $\mathcal{R}$. Then $\mathcal{B}$ can be reconfigured to $\mathcal{R}$. Furthermore, the rotation system of the Σ-embedding of G remains the same throughout the reconfiguration sequence.*

Proof Sketch. We give a brief overview of the proof of Theorem 6; the details and the supporting lemmas for hugging a spanning tree, fixing vertex neighborhoods, and augmenting disconnected graphs can be found in the full version [8, Appendix B]. Assume first that G is connected. We choose a spanning tree of G and reroute the edge curves of $\mathcal{B}$ and $\mathcal{R}$ to hug the spanning tree. If G itself is a tree, we can reduce the problem to Corollary 5 by first reconfiguring $\mathcal{B}$ to be identical to $\mathcal{R}$ in some small neighborhoods of the vertices, and then applying the algorithm in Section 3, which does not change these neighborhoods. The rotation system remains unchanged (it is determined by the neighborhoods of the vertices).

G is not necessarily connected. We can augment its spanning forest to a spanning tree if the components each lie in small neighborhoods of their spanning trees, and they have a consistent "outer face". Finally, if G is connected and $\mathcal{B}$ and $\mathcal{R}$ lie in the neighborhood of the embedding of a spanning tree T, we can follow the steps of the reconfiguration algorithm in Section 3 for T: Each time the algorithm reroutes an edge e of T, we reroute a "bundle" of edges in the neighborhood of the edge curve of e. ◀

4.2 Reconfiguration of Series-Parallel Graphs

The family of ***series-parallel graphs*** [17] can be defined recursively as follows.

1. The graph consisting of a single edge st is a series-parallel graph with poles s and t.
2. Given two series-parallel graphs G_1 with poles s_1 and t_1 and G_2 with poles s_2 and t_2, the following are series-parallel: (i) The ***series composition*** obtained by identifying t_1 and s_2 (with poles s_1 and t_2). (ii) The ***parallel composition*** obtained by identifying s_1 and s_2 as well as identifying t_1 and t_2 (with poles $s_1 = s_2$ and $t_1 = t_2$).

Series-parallel graphs were studied as early as 1892 [24] as they naturally occur in electrical networks [17, 24, 29]. Recently, series-parallel graphs have received attention both in graph algorithms [5, 7, 25, 28] and in graph drawing [2, 3, 16, 18]. The families of edge-maximal series-parallel graphs and 2-trees coincide [21] and the family of series-parallel graphs includes all 2-connected graphs without 3-connected minors [6]. Results on series-parallel graphs often cover all graphs of treewidth two and can be intermediate steps towards results on all 2-connected graphs – embeddings of 2-connected graphs can be enumerated using SPQR-trees that describe their construction using three operations (parallel, series and rigid) [4].

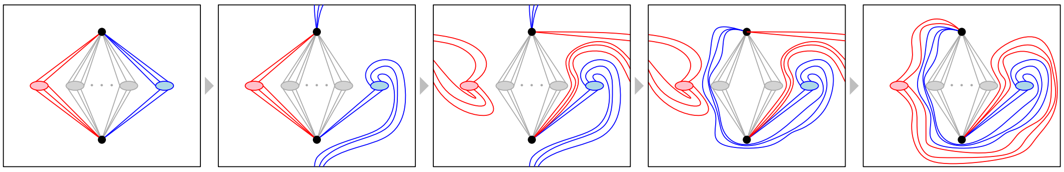

Figure 6 Reordering of two parallel components (red & blue) based on an interior $\bar{\Sigma}$-embedding.

Notably, all plane embeddings of series-parallel graphs differ only in the order in which parallel subgraphs are sorted at their common poles; see e.g. [6]. We will assume without loss of generality that the final step in the recursive construction is a parallel composition (otherwise, we add a parallel composition with an edge connecting both poles at the end).

▶ **Theorem 7.** *Let G be a series-parallel graph and $\mathcal{B}$ and $\mathcal{R}$ two interior $\bar{\Sigma}$-embeddings of G on a fundamental polygon $\bar{\Sigma}$ with $g \geq 1$. Then $\mathcal{B}$ can be reconfigured to $\mathcal{R}$.*

Proof Sketch. For a pair of poles s and t, we describe a procedure to exchange two parallel components C_1 and C_2 occurring consecutively in clockwise orientation around s (and hence in counterclockwise orientation around t). We first make the face between C_1 and C_2 the outer face of the embedding using Theorem 6, yielding the starting configuration shown in Figure 6. Then, we first reconfigure the edges of C_1 incident to s so that they intersect the vertical part of $\partial\bar{\Sigma}$, then we let the edges of C_2 incident to t intersect the horizontal part of $\partial\bar{\Sigma}$ so that C_1 and C_2 have successfully exchanged their positions; see the second and third subfigures in Figure 6. Then, we transform back to an embedding contained within the torus boundary while maintaining the rotation system. See the full version [8, Appendix C] for more details. ◀

5 Reconfiguration in the Projective Plane

▶ **Theorem 8.** *Let G be a perfect matching on $2n$ vertices, and let $\mathcal{B}$ and $\mathcal{R}$ be two $\mathbb{R}P^2$-embeddings of G such that there is a single region homeomorphic to a disk that contains both embeddings. Then, $\mathcal{B}$ and $\mathcal{R}$ are reconfigurable.*

Proof. By applying a homeomorphism, we may assume that $\mathcal{B}$ and $\mathcal{R}$ are given in the above described representation of $\mathbb{R}P^2$, with a sphere and a crosscap ∂D, in which the antipodal points of the circle ∂D are identified. The homeomorphism should map the region containing both embeddings to a disk disjoint from the crosscap. Let $G = (V, E)$ be a perfect matching on $2n$ vertices $V = \{s_1, \dots, s_n, t_1, \dots, t_n\}$ and n edges $E = \{e_i = s_i t_i : 1 \leq i \leq n\}$. Let $\mathcal{B}$ and $\mathcal{R}$ be two $\mathbb{R}P^2$-embeddings of G such that $\mathcal{B}(e_i) = P_i$ and $\mathcal{R}(e_i) = Q_i$ for all i, $1 \leq i \leq n$.

We construct a ***canonical embedding*** $\mathcal{M}$, where every curve passes through the crosscap exactly once, and then show how to transform $\mathcal{B}$ (resp., $\mathcal{R}$) to $\mathcal{M}$. We construct the canonical matching $\mathcal{M}$ as follows: Choose $2n$ equally spaced points along the circle ∂D, denoted $(a_1, \dots a_n, b_1, \dots, b_n)$ in counterclockwise order. Note that a_i and b_i are antipodal points in ∂D, hence they are identified in $\mathbb{R}P^2$. Since both embeddings lie in a disk disjoint from the crosscap, we may first choose pairwise disjoint arcs from the points $a_1, \dots, a_n, b_1, \dots, b_n$ to the boundary of that disk in the annulus between the two boundaries. We then extend these arcs inside the disk one by one to the vertices $s_1, \dots, s_n, t_1, \dots, t_n$ while avoiding the previously constructed arcs. Thus, for $i = 1, \dots, n$, we obtain pairwise disjoint paths $a_i s_i$ and $b_i t_i$ that cross the curves P_j and Q_j, $1 \leq j \leq n$, only finitely many times; moreover, each path $a_i s_i$ avoids all paths $a_j s_j$, $j < i$, and each path $b_i t_i$ avoids all paths $a_j s_j$, $1 \leq j \leq n$, and $b_j t_j$, $j < i$. The canonical matching is now $\mathcal{M} = \{M_i = s_i a_i \cup b_i t_i : i \in \{1 \dots, n\}\}$.

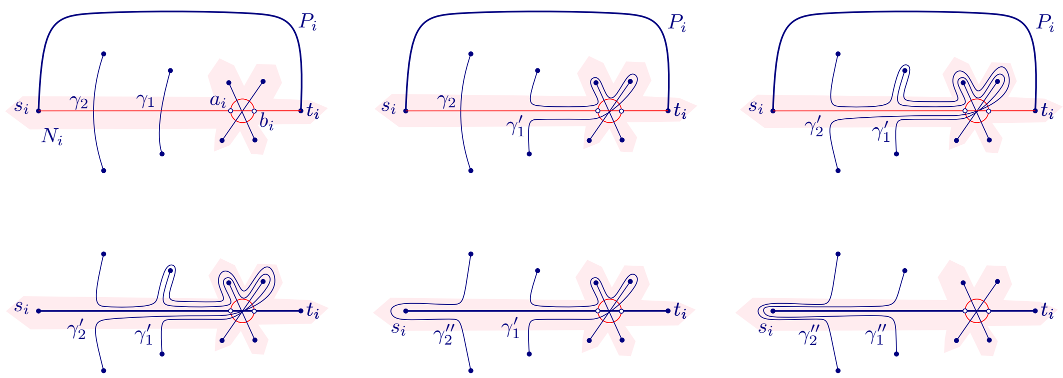

Figure 7 Replacing curve P_i (blue) with M_i (red). The thickening N_i of $\partial D \cup M_i \cup \left(\bigcup_{j<i} M_j\right)$ is highlighted in pink. Initially, γ_1 and γ_2 cross N_i. Top row: Stage 2. Bottom row: Stage 4.

Next, construct a reconfiguration sequence from $\mathcal{B}$ to $\mathcal{M}$. For $i = 1, \ldots, n$, replace the curve P_i with M_i (in several moves) as follows. Suppose all curves P_j, $j < i$, have already been replaced by M_j (curves $P_i, \ldots, P_n$ may have been modified in the process). We seek to replace P_i with M_i without modifying any curve M_j, $j < i$. To do so, successively modify the curves $P_i, \ldots, P_n$ and eliminate all their crossings with M_i. Then, replace P_i with M_i. We implement this strategy in two phases: First, eliminate all crossings of $P_i, \ldots, P_n$ with $a_i s_i$ (without introducing new crossings with $b_i t_i$); then eliminate all crossings with $b_i t_i$.

Consider a ***thickening*** N_i of $\partial D \cup M_i \cup \left(\bigcup_{j<i} M_j\right)$, that is, a δ-neighborhood for a small $\delta > 0$; see Figure 7 (upper left). Assume $\delta > 0$ is sufficiently small so that for all $\ell > i$, every connected component of $N_i \cap P_\ell$ intersects M_i. Denote these connected components (arcs) by $\gamma_1, \ldots, \gamma_t$ in the order in which they cross $a_i s_i$ (from a_i to s_i). Let $P(\gamma_j)$ be the curve that contains the arc γ_j for $j = 1, \ldots, t$. We modify $P_i, \ldots, P_n$ in four stages.

- **Stage 1:** A simplification step (described below) that modifies $P_i, \ldots, P_n$ such that (i) P_i does not cross M_i, and (ii) for every j, $i < j \le n$, the curve P_j crosses M_i at most once.
- **Stage 2:** For $\ell = 1, 2, \ldots, t$, replace γ_ℓ with an arc γ_ℓ' between the same endpoints that closely follows M_i (without crossing it) to ∂D, then crosses the crosscap, and finally closely follows $M_i \cup (\bigcup_{j<i} M_j) \cup (\bigcup_{j<\ell} P(\gamma_j))$; see Figure 7 (top row).
- **Stage 3:** Replace P_i with M_i.
- **Stage 4:** For each $\ell = t, t-1, \ldots, 1$, replace γ_ℓ' with an arc γ_ℓ'' that closely follows the boundary of the neighborhood N_i, and goes around s_i; see Figure 7 (bottom row).

It remains to describe Stage 1 (the simplification stage) in detail. In Stage 1, we apply the following claim while there is a curve in $\{P_i, \ldots, P_n\}$ that intersects $a_i s_i$ more than once.

▷ Claim 9. While some curve in $\mathcal{P} = \{P_i, \ldots, P_n\}$ intersects $M_i \setminus \{s_i\}$ at least twice, one can modify $\mathcal{P}$ to reduce the number of intersections between $M_i \setminus \{s_i\}$ and curves in $\mathcal{P}$.

We prove Claim 9 in the full version [8, Appendix D]. Consequently, $\mathcal{B}$ and $\mathcal{R}$ can each be reconfigured to $\mathcal{M}$ and combining the two reconfiguration sequences shows that $\mathcal{B}$ is reconfigurable to $\mathcal{R}$. ◀

▶ Remark 10. In Theorem 8, none of the edges pass through the crosscap; and we constructed an intermediate embedding in which every edge passes through the crosscap precisely once. It is not difficult to generalize Theorem 8 to input embeddings in which one edge may pass through the crosscap arbitrarily many times, and all other edges pass through the crosscap at most once each (in this case, we can create a clean crosscap). However, it is unclear how to reconfigure $\mathbb{R}P^2$-embeddings where multiple edges pass through the crosscap multiple times.

6 Non-Reconfigurable Graph Embeddings on Surfaces

▶ **Theorem 11.** *Let Σ be a surface.*

1. *There exists a graph H_1 and two Σ-embeddings $\mathcal{B}_1$ and $\mathcal{R}_1$ of H_1 that are not reconfigurable into each other. This holds even if we require that $\mathcal{B}_1$ and $\mathcal{R}_1$ have the same rotation system.*
2. *There exist a graph H_2 and two rotation systems B_2 and R_2 such that H_2 admits Σ-embeddings implementing B_2 and R_2, but any pair of Σ-embeddings $\mathcal{B}_2$ and $\mathcal{R}_2$ implementing B_2 and R_2 that have the vertices of H_2 embedded on the same points in Σ are not reconfigurable into each other.*

Proof Sketch. For the plane (and the sphere), Ito et al. [22] showed that a perfect matching on four vertices G_0 has two embeddings, $\mathcal{B}_0$ and $\mathcal{R}_0$, that are not reconfigurable; see also Figure 1a. On any surface Σ, we can embed G_0 within a triangle $\Delta(abc)$ as shown in Figure 8 (ignoring the dashed edges), yielding the gadget graph G_1. For statement 1, we use a uniquely embeddable triangulation $H = H(\Sigma)$ in which every 3-cycle is facial in every embedding; its construction is given in the full version [8, Appendix E] using a theorem of Negami [26, Theorem 3.3]. We identify one of its faces with $\Delta(abc)$ and insert the two embeddings of G_1 shown in Figure 8, respectively.

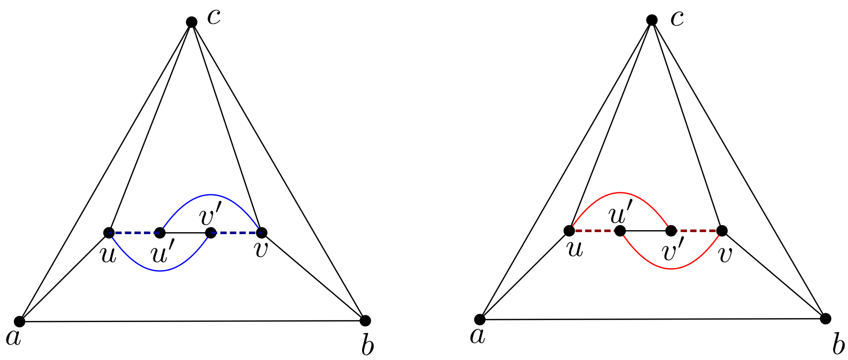

Figure 8 Illustration for the proof of Theorem 11.

As a result, reconfiguration is not possible due to the result by Ito et al. [22]. For statement 2, we augment the embeddings of G_1 with the two dashed edges in Figure 8, obtaining graph G_2 (part of the augmented graph H_2). We then argue that in any embedding with one of the two fixed rotation systems, we must have the same topology for G_2. This again contradicts reconfigurability. ◀

7 Open Problems

1. We do not know if our main result for forests extends to non-orientable surfaces. We conjecture that a forest can always be reconfigured on any non-orientable surface. Phase 1 of our algorithm from Section 3 (see Lemma 2) likely extends to non-orientable surfaces, since its shortcuts are constructed within the chosen flat representation. In contrast, this extension does not seem readily available for Phase 2 (see Lemma 3), where the rerouting argument uses that crossing an identified side of the flat representation preserves the left-to-right order of parallel arcs. For orientation-reversing side identifications on non-orientable surfaces, this order is reversed, so the same detour argument breaks down.

2. Our result on the reconfigurability of two different interior $\bar{\Sigma}$-embeddings of series-parallel graphs may generalize to general planar graphs. The missing link is triconnected planar graphs, where reconfiguration into a mirrored embedding is required.
3. Given a positive answer to Open Problem 1, one may wonder if we can obtain positive results for other (subclasses of) planar graphs on non-orientable surfaces.
4. The runtime of our algorithm in Section 3 is exponential in the complexity of the input embeddings. We conjecture that there is an exponential lower bound. We also ask what is the computational complexity of the Σ-`Embedding Reconfiguration` problem.

References

1 Soroush Alamdari, Patrizio Angelini, Fidel Barrera-Cruz, Timothy M. Chan, Giordano Da Lozzo, Giuseppe Di Battista, Fabrizio Frati, Penny Haxell, Anna Lubiw, Maurizio Patrignani, Vincenzo Roselli, Sahil Singla, and Bryan T. Wilkinson. How to morph planar graph drawings. *SIAM Journal on Computing*, 46(2):824–852, 2017. `doi:10.1137/16M1069171`.

2 Patrizio Angelini, Michael A. Bekos, Philipp Kindermann, and Tamara Mchedlidze. On mixed linear layouts of series-parallel graphs. *Theor. Comput. Sci.*, 936:129–138, 2022. `doi:10.1016/J.TCS.2022.09.019`.

3 Sk Ruhul Azgor and Md. Saidur Rahman. On the rique number of series-parallel graphs and planar bipartite graphs. In *Proc. 2nd International Conference on Applied Algorithms (ICAA)*, volume 15505 of *LNCS*, pages 3–14. Springer, 2025. `doi:10.1007/978-3-031-84543-7_1`.

4 Giuseppe Di Battista and Roberto Tamassia. On-line planarity testing. *SIAM J. Comput.*, 25(5):956–997, 1996. `doi:10.1137/S0097539794280736`.

5 Ferenc Bencs, Jeroen Huijben, and Guus Regts. On the location of chromatic zeros of series-parallel graphs. *Electron. J. Comb.*, 30(3), 2023. `doi:10.37236/11204`.

6 Daniel Bienstock and Clyde L. Monma. On the complexity of embedding planar graphs to minimize certain distance measures. *Algorithmica*, 5(1):93–109, 1990. `doi:10.1007/BF01840379`.

7 Andrew Bloch-Hansen and Roberto Solis-Oba. The thief orienteering problem on 2-terminal series-parallel graphs. *Acta Informatica*, 62(2):18, 2025. `doi:10.1007/S00236-025-00486-Y`.

8 Timo Brand, Stefan Felsner, Henry Förster, Stephen Kobourov, Anna Lubiw, Yoshio Okamoto, János Pach, Csaba D. Tóth, Géza Tóth, Torsten Ueckerdt, and Pavel Valtr. Rerouting curves on surfaces, 2026. `arXiv:2607.05362`.

9 Erin Wolf Chambers, Jeff Erickson, Patrick Lin, and Salman Parsa. How to morph graphs on the torus. In *Proc. 32nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2759–2778, 2021. `doi:10.1137/1.9781611976465.164`.

10 Hsien-Chih Chang and Arnaud de Mesmay. Tightening curves on surfaces monotonically with applications. *ACM Trans. Algorithms*, 18(4):36:1–36:32, 2022. `doi:10.1145/3558097`.

11 Hsien-Chih Chang and Jeff Erickson. Untangling planar curves. *Discret. Comput. Geom.*, 58(4):889–920, 2017. `doi:10.1007/S00454-017-9907-6`.

12 Hsien-Chih Chang, Jeff Erickson, David Letscher, Arnaud de Mesmay, Saul Schleimer, Eric Sedgwick, Dylan Thurston, and Stephan Tillmann. Tightening curves on surfaces via local moves. In *Proc. 29th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 121–135, 2018. `doi:10.1137/1.9781611975031.8`.

13 Hsien-Chih Chang, Jeff Erickson, and Chao Xu. Detecting weakly simple polygons. In Piotr Indyk, editor, *Proc. 26th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1655–1670, 2015. `doi:10.1137/1.9781611973730.110`.

14 Éric Colin de Verdière. Computational topology of graphs on surfaces. In Csaba D. Tóth, Joseph O'Rourke, and Jacob E. Goodman, editors, *Handbook of Discrete and Computational Geometry*, chapter 23, pages 605–636. Chapman and Hall/CRC, 2017. `doi:10.1201/9781315119601`.

15 Éric Colin de Verdière, Vincent Despré, and Loïc Dubois. Untangling graphs on surfaces. In *Proc. 35th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4909–4941, 2024. doi:10.1137/1.9781611977912.175.

16 Walter Didimo, Michael Kaufmann, Giuseppe Liotta, and Giacomo Ortali. Computing bend-minimum orthogonal drawings of plane series-parallel graphs in linear time. *Algorithmica*, 85(9):2605–2666, 2023. doi:10.1007/S00453-023-01110-6.

17 Richard J. Duffin. Topology of series-parallel networks. *Journal of Mathematical Analysis and Applications*, 10(2):303–318, 1965. doi:10.1016/0022-247X(65)90125-3.

18 David Eppstein. Bipartite and series-parallel graphs without planar lombardi drawings. *J. Graph Algorithms Appl.*, 25(1):549–562, 2021. doi:10.7155/JGAA.00571.

19 Jeff Erickson and Patrick Lin. Planar and toroidal morphs made easier. *Journal of Graph Algorithms and Applications*, 27:95–118, 2023. doi:10.7155/jgaa.00616.

20 Allen Hatcher and William Thurston. A presentation for the mapping class group of a closed orientable surface. *Topology*, 19(3):221–237, 1980. doi:10.1016/0040-9383(80)90009-9.

21 Frank K. Hwang, Dana S. Richards, and Pawel Winter. Chapter 5 polynomially solvable cases. In *The Steiner Tree Problem*, volume 53 of *Annals of Discrete Mathematics*, pages 177–188. Elsevier, 1992. doi:10.1016/S0167-5060(08)70202-4.

22 Takehiro Ito, Yuni Iwamasa, Naonori Kakimura, Yusuke Kobayashi, Shun-Ichi Maezawa, Yuta Nozaki, Yoshio Okamoto, and Kenta Ozeki. Rerouting planar curves and disjoint paths. *ACM Trans. Algorithms*, 21(2), 2025. doi:10.1145/3715694.

23 Marc Lackenby and Mehdi Yazdi. Bounds for the number of moves between pants decompositions, and between triangulations. *Journal of Topology and Analysis*, pages 1–68, 2026. doi:10.1142/S1793525326500056.

24 Percy A. Macmahon. The combinations of resistances. *Discrete Applied Mathematics*, 54(2):225–228, 1994. doi:10.1016/0166-218X(94)90024-8.

25 Marco A. López Medina, J. Leonardo González-Ruiz, José Raymundo Marcial-Romero, and José Antonio Hernández Servín. Computing the clique-width on series-parallel graphs. *Computación y Sistemas*, 26(2), 2022. doi:10.13053/CYS-26-2-4250.

26 Seiya Negami. Construction of graphs which are not uniquely and not faithfully embeddable in surfaces. *Yokohama Mathematical Journal*, 33(1–2):67–91, 1985. URL: https://ynu.repo.nii.ac.jp/records/6751.

27 Naomi Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4):52, 2018. doi:10.3390/A11040052.

28 Zhishi Pan and Xuding Zhu. The circular chromatic numbers of signed series-parallel graphs. *Discret. Math.*, 345(3):112733, 2022. doi:10.1016/J.DISC.2021.112733.

29 John Riordan and Claude E. Shannon. The number of two-terminal series-parallel networks. *Journal of Mathematics and Physics*, 21(1-4):83–93, 1942. doi:10.1002/sapm194221183.

30 Jan van den Heuvel. The complexity of change. In Simon R. Blackburn, Stefanie Gerke, and Mark Wildon, editors, *Surveys in Combinatorics 2013*, volume 409 of *London Mathematical Society Lecture Note Series*, pages 127–160. Cambridge University Press, 2013. doi:10.1017/CBO9781139506748.005.

31 Oleg Yanovich Viro, Oleg Aleksandrovich Ivanov, Nikita Yurévich Netsvetaev, and Viatcheslav Mikhaĭlovich Kharlamov. *Elementary Topology: Problem Textbook*, volume 54. American Mathematical Society, 2024. doi:10.1090/mbk/054.