Rerouting Planar Curves and Disjoint Paths

Takehiro Ito
Graduate School of Information Sciences,
Tohoku University, Sendai, Japan

Yuni Iwamasa
Graduate School of Informatics,
Kyoto University, Japan

Naonori Kakimura
Faculty of Science and Technology,
Keio University, Yokohama, Japan

Yusuke Kobayashi
Research Institute for Mathematical Sciences,
Kyoto University, Japan

Shun-ichi Maezawa
Department of Mathematics,
Tokyo University of Science, Japan

Yuta Nozaki
Faculty of Environment and Information Sciences,
Yokohama National University, Japan
SKCM^2, Hiroshima University, Japan

Yoshio Okamoto
Graduate School of Informatics and Engineering, The University of Electro-Communications,
Tokyo, Japan

Kenta Ozeki
Faculty of Environment and Information Sciences,
Yokohama National University, Japan

Abstract

In this paper, we consider a transformation of k disjoint paths in a graph. For a graph and a pair of k disjoint paths $\mathcal{P}$ and $\mathcal{Q}$ connecting the same set of terminal pairs, we aim to determine whether $\mathcal{P}$ can be transformed to $\mathcal{Q}$ by repeatedly replacing one path with another path so that the intermediates are also k disjoint paths. The problem is called Disjoint Paths Reconfiguration. We first show that Disjoint Paths Reconfiguration is PSPACE-complete even when $k = 2$. On the other hand, we prove that, when the graph is embedded on a plane and all paths in $\mathcal{P}$ and $\mathcal{Q}$ connect the boundaries of two faces, Disjoint Paths Reconfiguration can be solved in polynomial time. The algorithm is based on a topological characterization for rerouting curves on a plane using the algebraic intersection number. We also consider a transformation of disjoint s-t paths as a variant. We show that the disjoint s-t paths reconfiguration problem in planar graphs can be determined in polynomial time, while the problem is PSPACE-complete in general.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Disjoint paths, combinatorial reconfiguration, planar graphs

Digital Object Identifier 10.4230/LIPIcs.ICALP.2023.81

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2210.11778

Funding *Takehiro Ito*: JSPS KAKENHI Grant Numbers JP18H04091, JP19K11814, JP20H05793.
Yuni Iwamasa: JSPS KAKENHI Grant Numbers JP20K23323, JP20H05795, JP22K17854.
Naonori Kakimura: JSPS KAKENHI Grant Numbers JP20H05795, JP21H03397, JP22H05001.
Yusuke Kobayashi: JSPS KAKENHI Grant Numbers JP20K11692, JP20H05795, JP22H05001.
Shun-ichi Maezawa: JSPS KAKENHI Grant Numbers JP20H05795, JP22K13956.
Yuta Nozaki: JSPS KAKENHI Grant Numbers JP20H05795, JP20K14317, JP23K12974.
Yoshio Okamoto: JSPS KAKENHI Grant Numbers JP20H05795, JP20K11670, JP23K10982.
Kenta Ozeki: JSPS KAKENHI Grant Numbers JP18K03391, JP19H01803, JP20H05795, 23K03195.

Acknowledgements We thank Naoyuki Kamiyama for the discussion and the anonymous referees for their helpful comments.

1 Introduction

1.1 Disjoint Paths and Reconfiguration

The *disjoint paths problem* is a classical and important problem in algorithmic graph theory and combinatorial optimization. In the problem, the input consists of a graph $G = (V, E)$ and $2k$ distinct vertices $s_1, \dots, s_k, t_1, \dots, t_k$, called *terminals*, and the task is to find k vertex-disjoint paths $P_1, \dots, P_k$ such that P_i connects s_i and t_i for $i = 1, \dots, k$ if they exist. A tuple $\mathcal{P} = (P_1, \dots, P_k)$ of paths satisfying this condition is called a *linkage*. The disjoint paths problem has attracted attention since the 1970s because of its practical applications to transportation networks, network routing [46], and VLSI-layout [16, 29]. When the number k of terminal pairs is part of the input, the disjoint paths problem was shown to be NP-hard by Karp [25], and it remains NP-hard even for planar graphs [30]. For the case when the graph is undirected and k is a fixed constant, Robertson and Seymour [42] gave a polynomial-time algorithm based on the graph minor theory, which is one of the biggest achievements in this area. Although the setting of the disjoint paths problem is quite simple and easy to understand, a deep theory in discrete mathematics is required to solve the problem, which is a reason why this problem has attracted attention in the theoretical study of algorithms.

In this paper, we consider a transformation of linkages in a graph. Roughly, in a transformation, we pick up one path among the k paths in a linkage and replace it with another path to obtain a new linkage. To give a formal definition, suppose that G is a graph and $s_1, \dots, s_k, t_1, \dots, t_k$ are distinct terminals. For two linkages $\mathcal{P} = (P_1, \dots, P_k)$ and $\mathcal{Q} = (Q_1, \dots, Q_k)$, we say that $\mathcal{P}$ is *adjacent* to $\mathcal{Q}$ if there exists $i \in \{1, \dots, k\}$ such that $P_j = Q_j$ for $j \in \{1, \dots, k\} \setminus \{i\}$ and $P_i \neq Q_i$. We say that a sequence $\langle \mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_\ell \rangle$ of linkages is a *reconfiguration sequence* from $\mathcal{P}_1$ to $\mathcal{P}_\ell$ if $\mathcal{P}_i$ and $\mathcal{P}_{i+1}$ are adjacent for $i = 1, \dots, \ell - 1$. If such a sequence exists, we say that $\mathcal{P}_1$ is *reconfigurable* to $\mathcal{P}_\ell$. In this paper, we focus on the following reconfiguration problem, which we call Disjoint Paths Reconfiguration.

Disjoint Paths Reconfiguration
Input. A graph $G = (V, E)$, distinct terminals $s_1, \dots, s_k, t_1, \dots, t_k$, and two linkages $\mathcal{P}$ and $\mathcal{Q}$.
Question. Is $\mathcal{P}$ reconfigurable to $\mathcal{Q}$?

The problem can be regarded as the problem of deciding the reachability between linkages via rerouting paths. Such a problem falls in the area of *combinatorial reconfiguration*; see Section 1.3 for prior work on combinatorial reconfiguration. Note that Disjoint Paths Reconfiguration is a decision problem that just returns "YES" or "NO" and does not necessarily find a reconfiguration sequence when the answer is YES[1].

Although our study is motivated by a theoretical interest in the literature on combinatorial reconfiguration, the problem can model a rerouting problem in a telecommunication network as follows. Suppose that a linkage represents routing in a telecommunication network, and we want to modify linkage $\mathcal{P}$ to another linkage $\mathcal{Q}$ which is better than $\mathcal{P}$ in some sense. If we can change only one path in a step in the network for some technical reasons, and we have to keep a linkage in the modification process, then this situation is modeled as Disjoint Paths Reconfiguration.

[1] Our positive results in this paper hold also for the problem of finding a reconfiguration sequence.

We also study internally vertex-disjoint s-t paths instead of disjoint paths. In the *disjoint s-t paths problem*, for a graph and two terminals s and t, we seek for k internally vertex-disjoint paths connecting s and t. It is well-known that the disjoint s-t paths problem can be solved in polynomial time. The study of disjoint s-t paths originated from Menger's min-max theorem [33] and the max-flow algorithm by Ford and Fulkerson [14]. Faster algorithms for finding maximum disjoint s-t paths or a maximum s-t flow have been actively studied in particular for planar graphs; see e.g. [12, 24, 26, 51].

In the same way as Disjoint Paths Reconfiguration, we consider a reconfiguration of internally vertex-disjoint s-t paths. Let $G = (V, E)$ be a graph with two distinct terminals s and t. We say that a set $\mathcal{P} = \{P_1, \dots, P_k\}$ of k paths in G is an s-t *linkage* if $P_1, \dots, P_k$ are internally vertex-disjoint s-t paths. Note that $\mathcal{P}$ is not a tuple but a set, that is, we ignore the ordering of the paths in $\mathcal{P}$. We say that s-t linkages $\mathcal{P}$ and $\mathcal{Q}$ are *adjacent* if $\mathcal{Q} = (\mathcal{P} \setminus P) \cup \{Q\}$ for some s-t paths P and Q with $P \neq Q$. We define the reconfigurability of s-t linkages in the same way as linkages. We consider the following problem.

Disjoint s-t Paths Reconfiguration
Input. A graph $G = (V, E)$, distinct terminals s and t, and two s-t linkages $\mathcal{P}$ and $\mathcal{Q}$.
Question. Is $\mathcal{P}$ reconfigurable to $\mathcal{Q}$?

1.2 Our Contributions

Since finding disjoint s-t paths is an easy combinatorial optimization problem, we may wonder whether Disjoint s-t Paths Reconfiguration is also tractable. In this paper, we show that Disjoint s-t Paths Reconfiguration is PSPACE-hard even when $k = 2$.

▶ **Theorem 1.** *The Disjoint s-t Paths Reconfiguration is* PSPACE*-complete even when $k = 2$ and the maximum degree of G is four.*

Note that Disjoint s-t Paths Reconfiguration can be easily reduced to Disjoint Paths Reconfiguration by splitting each of s and t into k terminals. Thus, this theorem implies the PSPACE-hardness of Disjoint Paths Reconfiguration with $k = 2$.

In this paper, we mainly focus on the problems in planar graphs. To better understand Disjoint Paths Reconfiguration in planar graphs, we show a topological necessary condition.

Topological conditions play important roles in the disjoint paths problem. If there exist disjoint paths connecting terminal pairs in a graph embedded on a surface Σ, then there must exist disjoint curves on Σ connecting them. For example, when terminals s_1, s_2, t_1 and t_2 lie on the outer face F in a plane graph G in this order, there exist no disjoint curves connecting the terminal pairs in the disk $\Sigma = \mathbb{R}^2 \setminus F$, and hence we can conclude that G contains no disjoint paths. Such a topological condition is used to design polynomial-time algorithms for the disjoint paths problem with $k = 2$ [44, 45, 49], and to deal with the problem on a disk or a cylinder [40]. When Σ is a plane (or a sphere), we can always connect terminal pairs by disjoint curves on Σ, and hence nothing is derived from the above argument. Indeed, Robertson and Seymour [41] showed that if the input graph is embedded on a surface and the terminals are mutually "far apart," then desired disjoint paths always exist.

In contrast, as we will show below in Theorem 2, there exists a topological necessary condition for the reconfigurability of disjoint paths. Thus, even when the terminals are mutually far apart, the reconfiguration of disjoint paths is not always possible. This shows a difference between the disjoint paths problem and Disjoint Paths Reconfiguration.

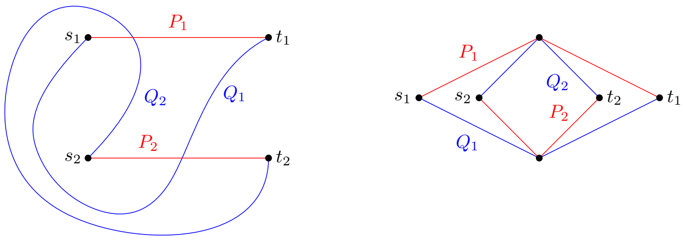

Figure 1 (Left) An example on the plane where (P_1, P_2) is not reconfigurable to (Q_1, Q_2). (Right) An example in a graph where the condition in Theorem 2 holds but (P_1, P_2) is not reconfigurable to (Q_1, Q_2).

To formally discuss the topological necessary condition, we consider the reconfiguration of curves on a surface. Suppose that Σ is a surface and let $s_1, \ldots, s_k, t_1, \ldots, t_k$ be distinct points on Σ. By abuse of notation, we say that $\mathcal{P} = (P_1, \ldots, P_k)$ is a *linkage* if it is a collection of disjoint simple curves on Σ such that P_i connects s_i and t_i. We also define the adjacency and reconfiguration sequences for linkages on Σ in the same way as linkages in a graph. Then, the reconfigurability between two linkages on a plane can be characterized with a word w_j associated to Q_j which is an element of the free group[2] F_k generated by $x_1, \ldots, x_k$ as follows; see Section 3 for the definition of w_j.

▶ **Theorem 2.** *Let $\mathcal{P} = (P_1, \ldots, P_k)$ and $\mathcal{Q} = (Q_1, \ldots, Q_k)$ be linkages on a plane (or a sphere). Then, $\mathcal{P}$ is reconfigurable to $\mathcal{Q}$ if and only if $w_j \in \langle x_j \rangle$ for any $j \in \{1, \ldots, k\}$, where $\langle x_j \rangle$ denotes the subgroup generated by x_j.*

See Figure 1 (left) for an example. It is worth noting that, if $k = 2$ and Σ is a connected orientable closed surface of genus $g \geq 1$, then such a topological necessary condition does not exist, i.e., the reconfiguration is always possible; see the full version [20].

For a graph embedded on a plane, we can identify paths and curves. Then, Theorem 2 gives a topological necessary condition for Disjoint Paths Reconfiguration in planar graphs. However, the converse does not necessarily hold: even when the condition in Theorem 2 holds, an instance of Disjoint Paths Reconfiguration may have no reconfiguration sequence. See Figure 1 (right) for a simple example. The polynomial solvability of Disjoint Paths Reconfiguration in planar graphs is open even for the case of $k = 2$.

With the aid of the topological necessary condition, we design polynomial-time algorithms for special cases, in which all the terminals are on a single face (called *one-face instances*), or $s_1, \ldots, s_k$ are on some face and $t_1, \ldots, t_k$ are on another face (called *two-face instances*). Note that one/two-face instances have attracted attention in the disjoint paths problem [40, 47, 48], in the multicommodity flow problem [18, 35, 36], and in the shortest disjoint paths problem [8, 9, 11, 28]. We show that any one-face instance of Disjoint Paths Reconfiguration has a reconfiguration sequence (Proposition 13). Moreover, we prove a topological characterization for two-face instances of Disjoint Paths Reconfiguration with a certain condition (Theorem 14), which leads to a polynomial-time algorithm in this case.

▶ **Theorem 3.** *When the instances are restricted to two-face instances, Disjoint Paths Reconfiguration can be solved in polynomial time.*

[2] Each element of the free group can be expressed as a word consisting of $x_1, x_1^{-1}, \ldots, x_k, x_k^{-1}$ in which x_i and x_i^{-1} are not adjacent.

Based on this theorem, we give a polynomial-time algorithm for Disjoint s-t Paths Reconfiguration in planar graphs.

▶ **Theorem 4.** *There is a polynomial-time algorithm for Disjoint s-t Paths Reconfiguration in planar graphs.*

Note that the number k of paths in Theorems 3 and 4 can be part of the input.

It is well known that G has an s-t linkage of size k if and only if G has no s-t separator of size $k-1$ (Menger's theorem). The characterization for two-face instances (Theorem 14) implies the following theorem, which is interesting in the sense that one extra s-t connectivity is sufficient to guarantee the existence of a reconfiguration sequence.

▶ **Theorem 5.** *Let $G = (V, E)$ be a planar graph with distinct vertices s and t, and let $\mathcal{P}$ and $\mathcal{Q}$ be s-t linkages of size k. If there is no s-t separator of size k, then $\mathcal{P}$ is reconfigurable to $\mathcal{Q}$.*

As mentioned above, the polynomial solvability of Disjoint Paths Reconfiguration in planar graphs is open even for the case of $k = 2$. On the other hand, when k is not bounded, Disjoint Paths Reconfiguration is PSPACE-complete as the next theorem shows.

▶ **Theorem 6.** *The Disjoint Paths Reconfiguration is PSPACE-complete when the graph G is planar and of bounded bandwidth.*

Here, we recall the definition of the bandwidth of a graph. Let $G = (V, E)$ be an undirected graph. Consider an injective map $\pi \colon V \to \mathbb{Z}$. Then, the *bandwidth* of π is defined as $\max\{|\pi(u) - \pi(v)| \mid \{u, v\} \in E\}$. The *bandwidth* of G is defined as the minimum bandwidth of all injective maps $\pi \colon V \to \mathbb{Z}$.

1.3 Related Work

There are a lot of studies on the disjoint paths problem and its variant. For the case of $k = 2$, polynomial-time algorithms were presented in [44, 45, 49], while the directed variant was shown to be NP-hard [15]. In the early stages of the study of the disjoint paths problem, for the case when G is embedded on a plane and all the terminals are on one face or two faces, polynomial-time algorithms were given in [40, 47, 48]. For fixed k, Robertson and Seymour [41] gave a polynomial time algorithm for the disjoint paths problem on a plane or a fixed surface. By extending this result, for the case when the graph is undirected and k is a fixed constant, Robertson and Seymour [42] gave a polynomial-time algorithm based on the graph minor theory, which is one of the biggest achievements in this area. For the planar case, faster algorithms were presented in [1, 38, 39]. The directed variant of the problem can be solved in polynomial time if the input digraph is planar and k is a fixed constant; an XP algorithm was given by Schrijver [43] and an FPT algorithm was given by Cygan et al. [10] for the parameter k.

Combinatorial reconfiguration is an emerging field in discrete mathematics and theoretical computer science. In typical problems of combinatorial reconfiguration, we consider two discrete structures and ask whether one can be transformed to the other by a sequence of local changes. See surveys of Nishimura [34] and van den Heuvel [50]. Refer to [22] for a general solver.

Path reconfiguration problems have been studied in this framework. The first problem is the shortest path reconfiguration, introduced by Kaminski et al. [23]. In this problem, we are given an undirected graph with two designated vertices s, t and two s-t shortest paths P and

Q. Then, we want to decide whether P can be transformed to Q by a sequence of one-vertex changes in such a way that all the intermediate s-t paths remain the shortest. Bonsma [6] proved that the shortest path reconfiguration is PSPACE-complete, but polynomial-time solvable when the input graph is chordal or claw-free. Bonsma [7] further proved that the problem is polynomial-time solvable for planar graphs. Wrochna [52] proved that the problem is PSPACE-complete even for graphs of bounded bandwidth. Gajjar et al. [17] proved that the problem is polynomial-time solvable for circle graphs, circular-arc graphs, permutation graphs, and hypercubes. They also considered a variant where a change can involve k successive vertices; in this variant, they proved that the problem is PSPACE-complete even for line graphs. Properties of the adjacency relation in the shortest path reconfiguration have also been studied [4, 5].

Another path reconfiguration problem has been introduced by Amiri et al. [3] who were motivated by a problem in software-defined networks. In their setup, we are given a directed graph with edge capacity and two designated vertices s, t. We are also given k pairs of s-t paths (P_i, Q_i), $i = 1, 2, \dots, k$, where the number of paths among $P_1, P_2, \dots, P_k$ (and among $Q_1, Q_2, \dots, Q_k$ respectively) traversing an edge is at most the capacity of the edge. The problem is to determine whether one set of paths can be transformed into the other set of paths by a sequence of the following type of changes: specify one vertex v and then switch the usable outgoing edges at v from those in the P_i to those in the Q_i. In each of the intermediate situations, there must be a unique path through usable edges in $P_i \cup Q_i$ for each i. See [3] for the precise problem specification. Amiri et al. [3] proved that the problem is NP-hard even when $k = 2$. For directed acyclic graphs, they also proved that the problem is NP-hard (for unbounded k) but fixed-parameter tractable with respect to k. A subsequent work [2] studied an optimization variant in which the number of steps is to be minimized when a set of "disjoint" changes can be performed simultaneously.

Matching reconfiguration in bipartite graphs can be seen as a certain type of disjoint paths reconfiguration problems. In matching reconfiguration, we are given two matchings (with extra properties) and want to determine whether one matching can be transformed to the other matching by a sequence of local changes. There are several choices for local changes. One of the most studied local change rules is the token jumping rule, where we remove one edge and add one edge at the same time. Ito et al. [19] proved that the matching reconfiguration (under the token jumping rule) can be solved in polynomial time.[3]

To see a connection of matching reconfiguration with disjoint paths reconfiguration, consider the matching reconfiguration problem in bipartite graphs G under the token jumping rule, where we are given two matchings M, M' of G. Then, we add two extra vertices s, t to G, and for each edge $e \in M$ (and M') we construct a unique s-t path of length three that passes through e. This way, we obtain two s-t linkages $\mathcal{P}$ and $\mathcal{P}'$ from M and M', respectively. It is easy to observe that $\mathcal{P}$ can be reconfigured to $\mathcal{P}'$ in Disjoint s-t Paths Reconfiguration if and only if M can be reconfigured to M' in the matching reconfiguration problem in G.

1.4 Organization

In Section 2, we introduce some notation and basic concepts in topology. Section 3 deals with rerouting disjoint curves, giving the proof of Theorem 2. In Sections 4 and 5, we prove Theorems 3, 4, and 5. Hardness results (Theorems 1 and 6) are proven in the full version [20].

[3] The theorem by Ito et al. [19] only gave a polynomial-time algorithm for a different local change, the so-called token addition and removal rule. However, their result can easily be adapted to the token jumping rule, too. See [21].

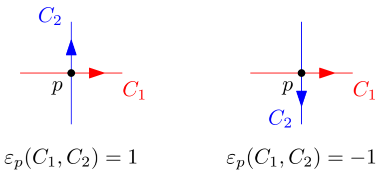

Figure 2 Local intersection numbers of curves C_1 and C_2 at p.

2 Preliminaries

For a positive integer k, let $[k] = \{1, 2, \ldots, k\}$.

Let $G = (V, E)$ be a graph. For a subgraph H of G, the vertex set of H is denoted by $V(H)$. Similarly, for a path P, let $V(P)$ denote the set of vertices in P. For $X \subseteq V$, let $N(X) = \{v \in V \setminus X \mid \{u, v\} \in E \text{ for some } u \in X\}$. For a vertex set $U \subseteq V$, let $G \setminus U$ denote the graph obtained from G by removing all the vertices in U and the incident edges. For a path P in G, we denote $G \setminus V(P)$ by $G \setminus P$ to simplify the notation. For disjoint vertex sets $X, Y \subseteq V$, we say that a vertex subset $U \subseteq V \setminus (X \cup Y)$ *separates* X and Y if $G \setminus U$ contains no path between X and Y. For distinct vertices $s, t \in V$, $U \subseteq V \setminus \{s, t\}$ is called an *s-t separator* if U separates $\{s\}$ and $\{t\}$.

For Disjoint Paths Reconfiguration (resp. Disjoint s-t Paths Reconfiguration), an instance is denoted by a triplet $(G, \mathcal{P}, \mathcal{Q})$, where G is a graph and $\mathcal{P}$ and $\mathcal{Q}$ are linkages (resp. s-t linkages). Note that we omit the terminals because they are determined by $\mathcal{P}$ and $\mathcal{Q}$. Since any instance has a trivial reconfiguration sequence when $k = 1$, we may assume that $k \geq 2$. For linkages (resp. s-t linkages) $\mathcal{P}$ and $\mathcal{Q}$, we denote $\mathcal{P} \leftrightarrow \mathcal{Q}$ if $\mathcal{P}$ and $\mathcal{Q}$ are adjacent. Recall that $\mathcal{P} = (P_1, \ldots, P_k)$ is *adjacent* to $\mathcal{Q} = (Q_1, \ldots, Q_k)$ if there exists $i \in [k]$ such that $P_j = Q_j$ for $j \in [k] \setminus \{i\}$ and $P_i \neq Q_i$.

For a graph G embedded on a surface Σ, each connected region of $\Sigma \setminus G$ is called a *face* of G. For a face F, its boundary is denoted by ∂F. When a graph G is embedded on a surface Σ, a path in G is sometimes identified with the corresponding curve in Σ. A graph embedded on a plane is called a *plane graph*. A graph is said to be *planar* if it has a planar embedding.

The following notion is well-known in topology. See [13, Section 1.2.3] for instance.

▶ **Definition 7.** *Let C_1 and C_2 be piecewise smooth oriented curves on an oriented surface and let $p \in C_1 \cap C_2$ be a transverse double point*[4]*. The* local intersection number *$\varepsilon_p(C_1, C_2)$ of C_1 and C_2 at p is defined by $\varepsilon_p(C_1, C_2) = 1$ if C_1 crosses C_2 from left to right and $\varepsilon_p(C_1, C_2) = -1$ if C_1 crosses C_2 from right to left (see Figure 2). When $\partial C_1 \cap C_2 = C_1 \cap \partial C_2 = \emptyset$, the* algebraic intersection number *$\mu(C_1, C_2) \in \mathbb{Z}$ is defined to be the sum of $\varepsilon_p(C_1, C_2)$ over all $p \in C_1 \cap C_2$ (after a small perturbation if necessary). Note that ∂C_i denotes the set of endpoints of C_i.*

When a graph is embedded on an oriented surface, paths in the graph are piecewise smooth curves, and hence we can define the algebraic intersection number for a pair of paths (see Figure 3).

[4] Intuitively, a "transverse double point" means that at the intersection two curves are not tangent with each other and no three segments of curves do not intersect simultaneously.

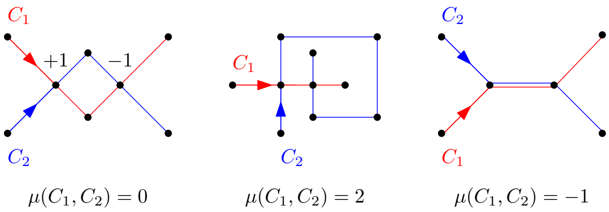

Figure 3 Algebraic intersection numbers of paths C_1 and C_2 on a graph.

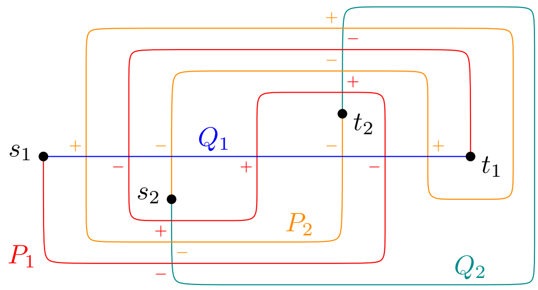

Figure 4 An example of linkages with $w_1 = x_2x_1^{-1}x_2^{-1}x_1x_2^{-1}x_1^{-1}x_2$ and $w_2 = x_1x_2^{-1}x_1^{-1}x_2x_1^{-1}x_2^{-1}x_1$.

3 Curves on a Plane

In this section, we consider the reconfiguration of curves on a plane and prove Theorem 2. Suppose that we are given distinct points $s_1, \ldots, s_k, t_1, \ldots, t_k$ on a plane and linkages $\mathcal{P}$ and $\mathcal{Q}$ that consist of curves on the plane connecting s_i and t_i.

Throughout this section, all intersections of curves are assumed to be transverse double points. Fix $j \in [k]$ and let $\bigcup_{i \in [k]} P_i \cap Q_j = \{s_j, p_1, \ldots, p_n, t_j\}$, where the $n + 2$ points are aligned on Q_j in this order. We now define $w_j \in F_k$ by

$$w_j = \prod_{\ell \in [n]} x_{i_\ell}^{\varepsilon_{p_\ell}(P_{i_\ell}, Q_j)},$$

where $i_\ell \in [k]$ satisfies $p_\ell \in P_{i_\ell} \cap Q_j$. Recall that F_k denotes the free group generated by $x_1, \ldots, x_k$. We give an example in Figure 4.

▶ Remark 8. Let $\mathrm{ab}\colon F_k \to \mathbb{Z}^k$ denote the abelianization, that is, the ℓth entry of $\mathrm{ab}(w)$ is the sum of the exponents of x_ℓ's in w. For distinct $i, j \in [k]$, the ith entry of $\mathrm{ab}(w_j)$ is equal to the algebraic intersection number $\mu(P_i, Q_j) \in \mathbb{Z}$ of P_i and Q_j. Thus, $w_j \in \langle x_j \rangle$ implies that $\mu(P_i, Q_j) = 0$ for any $i \in [k] \setminus \{j\}$.

In the following two lemmas, we observe the behavior of w_j under certain moves of curves. For $j \in [k]$, let w'_j denote the word defined by a linkage $\mathcal{P}'$ and the curve Q_j.

▶ **Lemma 9.** *Let $i \in [k]$ and let $\mathcal{P}' = (P'_1, \ldots, P'_k)$ be a linkage such that $P'_\ell = P_\ell$ if $\ell \neq i$, and P'_i is isotopic[5] to P_i relative to $\{s_i, t_i\}$ in $\mathbb{R}^2 \setminus \bigcup_{\ell \neq i} P_\ell$. Then, $w'_j = w_j$ for $j \in [k] \setminus \{i\}$, and $w'_i = x_i^{e_1} w_i x_i^{e_2}$ for some $e_1, e_2 \in \mathbb{Z}$.*

[5] Intuitively, this means P'_i can be obtained from P_i by a continuous deformation in the plane that fixes the endpoints s_i and t_i and avoids passing any point in $\bigcup_{\ell \neq i} P_\ell$.

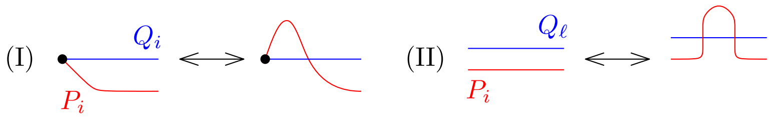

Figure 5 Local pictures of isotopies of P_i.

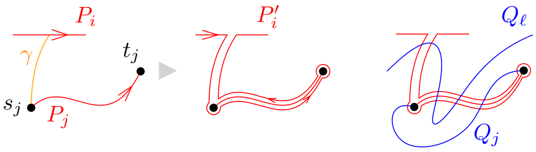

Figure 6 (Left) A move of P_i along γ. (Right) Intersections of P_i' and $\bigcup_\ell Q_\ell$.

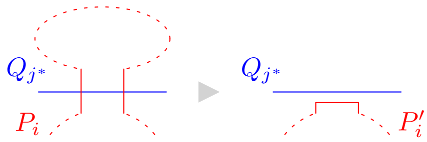

Figure 7 A reconfiguration of P_i to P_i'.

Proof. By the definition of an isotopy (see [13, Section 1.2.5]), P_i' is obtained from P_i by a finite sequence of the moves illustrated in Figure 5. By (I), one intersection of P_i and Q_i is created or eliminated, and thus (I) changes w_i to $w_i x_i^{\pm 1}$ or $x_i^{\pm 1} w_i$. In (II), two intersections of P_i and Q_ℓ are created or eliminated for some $\ell \in [k]$. Since $x_i^{\pm 1} x_i^{\mp 1} = 1$, w_j is unchanged under (II) for any $j \in [k]$. ◀

Recall here that $\langle x_\ell \rangle$ denotes the subgroup of F_k generated by x_ℓ.

▶ **Lemma 10.** *Let γ be a simple curve connecting P_i and s_j $(i \neq j)$ whose interior is disjoint from $\bigcup_{\ell \in [k]} P_\ell$, and define P_i' as illustrated in Figure 6. Let $\mathcal{P}'$ be the linkage obtained from $\mathcal{P}$ by replacing P_i with P_i'. For $\ell \in [k]$, if $w_\ell \in \langle x_\ell \rangle$, then $w_\ell' = w_\ell$.*

Proof. Define a group homomorphism $f_{ij} \colon F_k \to F_k$ by $f_{ij}(x_\ell) = x_\ell$ if $\ell \neq j$, and $f_{ij}(x_j) = x_i x_j x_i^{-1}$. Then, one can check that $w_\ell' = f_{ij}(w_\ell)$ if $\ell \neq j$, and $w_j' = x_i^{-1} f_{ij}(w_j) x_i$ (see Figure 6). Since $w_\ell = x_\ell^{e_\ell}$ for some $e_\ell \in \mathbb{Z}$ by the assumption, we have $w_\ell' = w_\ell$ if $\ell \neq j$. Also, one has

$$w_j' = x_i^{-1} f_{ij}(w_j) x_i = x_i^{-1} (x_i x_j x_i^{-1})^{e_j} x_i = w_j.$$

This completes the proof. ◀

As a consequence of Lemmas 9 and 10, we obtain the following key lemma.

▶ **Lemma 11.** *Suppose that $\mathcal{P}$ is reconfigurable to $\mathcal{P}'$. For $j \in [k]$, if $w_j \in \langle x_j \rangle$, then $w_j' \in \langle x_j \rangle$.*

Proof. It suffices to consider the case when there is $i \in [k]$ such that $P_\ell' = P_\ell$ if $\ell \neq i$, and $P_i' \neq P_i$. Since P_i is isotopic to P_i' (relative to $\{s_i, t_i\}$) in $\mathbb{R}^2$, the curve P_i' is obtained from P_i by the moves in Lemmas 9 and 10. Therefore, these lemmas imply that if $w_j \in \langle x_j \rangle$ then $w_j' \in \langle x_j \rangle$. ◀

With this key lemma, we can prove Theorem 2 stating that $\mathcal{P}$ is reconfigurable to $\mathcal{Q}$ if and only if $w_j \in \langle x_j \rangle$ for any $j \in [k]$.

Proof of Theorem 2. First suppose that $\mathcal{P}$ is reconfigurable to $\mathcal{Q}$, namely $\mathcal{P}$ is reconfigurable to $\mathcal{P}'$ such that $P_i' \cap Q_i = \{s_i, t_i\}$ and $P_i' \cap Q_j = \emptyset$ for $j \in [k] \setminus \{i\}$. Then, $w_j' = 1$ for any $j \in [k]$. Since $\mathcal{P}'$ is reconfigurable to $\mathcal{P}$, Lemma 11 implies that $w_j \in \langle x_j \rangle$ for any $j \in [k]$.

The converse is shown by induction on the number, say n, of intersections of $\mathcal{P}$ and $\mathcal{Q}$ except their endpoints. The case $n = 0$ is obvious. Let us consider the case $n \geq 1$. If $P_i \cap Q_j = \emptyset$ for any pair of distinct $i, j \in [k]$, then the reconfiguration is obviously possible. Otherwise, there exists $x_i x_i^{-1}$ or $x_i^{-1} x_i$ in the product of the definition of w_{j^*} for some $i, j^* \in [k]$ (possibly $i = j^*$). This means that P_i can be reconfigured to a curve P_i' as illustrated in Figure 7. This process eliminates at least two intersections and we have $w_j' \in \langle x_j \rangle$ for any $j \in [k]$ by Lemma 11. Thus, the induction hypothesis concludes that $\mathcal{P}'$ is reconfigurable to $\mathcal{Q}$. ◀

By Theorem 2 and Remark 8, we obtain the following corollary.

▶ **Corollary 12.** *Let $\mathcal{P} = (P_1, \dots, P_k)$ and $\mathcal{Q} = (Q_1, \dots, Q_k)$ be linkages on a plane (or a sphere). If $\mathcal{P}$ is reconfigurable to $\mathcal{Q}$, then $\mu(P_i, Q_j) = 0$ for any distinct $i, j \in [k]$.*

It is worth mentioning that the converse is not necessarily true as illustrated in Figure 4. This means that a "non-commutative" tool such as the free group F_k is essential to describe the complexity of the reconfiguration of curves on a plane.

4 Algorithms for Planar Graphs

In this section, we consider the reconfiguration in planar graphs and prove Theorems 3, 4, and 5. We deal with one-face instances and two-face instances of Disjoint Paths Reconfiguration in Section 4.1. Then, we discuss Disjoint s-t Paths Reconfiguration in Section 4.2. A proof of a key theorem (Theorem 14) is postponed to Section 5.

4.1 One-Face Instance and Two-Face Instance

We say that an instance $(G, \mathcal{P}, \mathcal{Q})$ of Disjoint Paths Reconfiguration is a *one-face instance* if G is a plane graph and all the terminals are on the boundary of some face. We show that $\mathcal{P}$ is always reconfigurable to $\mathcal{Q}$ in a one-face instance, whose proof is given in the full version [20].

▶ **Proposition 13.** *For any one-face instance $(G, \mathcal{P}, \mathcal{Q})$ of Disjoint Paths Reconfiguration, $\mathcal{P}$ is reconfigurable to $\mathcal{Q}$.*

Let $k \geq 2$. We say that an instance $(G, \mathcal{P}, \mathcal{Q})$ of Disjoint Paths Reconfiguration is a *two-face instance* if $G = (V, E)$ is a plane graph, $s_1, \dots, s_k$ are on the boundary of some face S, and $t_1, \dots, t_k$ are on the boundary of another face T. The objective of this subsection is to present a polynomial-time algorithm for two-face instances.

It suffices to consider the case when the graph is 2-connected since otherwise we can easily reduce to the 2-connected case. Hence, we may assume that the boundary of each face forms a cycle. For ease of explanation, without loss of generality, we assume that G is embedded on $\mathbb{R}^2$ so that S is an inner face and T is the outer face. Furthermore, we may assume that $s_1, \dots, s_k$ lie on the boundary of S clockwise in this order and $t_1, \dots, t_k$ lie on the boundary of T clockwise in this order, because there is a linkage.

A vertex set $U \subseteq V$ is called a *terminal separator* if U separates $\{s_1, \dots, s_k\}$ and $\{t_1, \dots, t_k\}$. For two curves (or paths) P and Q between ∂S and ∂T that share no endpoints, define $\mu(P, Q)$ as in Definition 7. That is, $\mu(P, Q)$ is the number of times P crosses Q from left to right minus the number of times P crosses Q from right to left, where we suppose that P and Q are oriented from ∂S to ∂T. Since $\mu(P_i, Q_j)$ takes the same value for distinct $i, j \in [k]$ (see the full version [20] for details), this value is denoted by $\mu(\mathcal{P}, \mathcal{Q})$. Roughly, $\mu(\mathcal{P}, \mathcal{Q})$ indicates the difference in the numbers of rotations around S of the linkages.

The existence of a linkage shows that the graph has no terminal separator of size less than k. If the graph has no terminal separator of size k, then we can characterize the reconfigurability by using $\mu(\mathcal{P}, \mathcal{Q})$. The following is a key theorem in our algorithm, whose proof is given in Section 5.

▶ **Theorem 14.** *Let $k \geq 2$. Suppose that a two-face instance $(G, \mathcal{P}, \mathcal{Q})$ of* Disjoint Paths Reconfiguration *has no terminal separator of size k. Then, $\mathcal{P}$ is reconfigurable to $\mathcal{Q}$ if and only if $\mu(\mathcal{P}, \mathcal{Q}) = 0$.*

By using this theorem, we can design a polynomial-time algorithm for two-face instances of Disjoint Paths Reconfiguration and prove Theorem 3.

Proof of Theorem 3. Suppose that we are given a two-face instance $I = (G, \mathcal{P}, \mathcal{Q})$ of Disjoint Paths Reconfiguration.

We first test whether I has a terminal separator of size k, which can be done in polynomial time by a standard minimum cut algorithm. If there is no terminal separator of size k, then Theorem 14 shows that we can easily solve Disjoint Paths Reconfiguration by checking whether $\mu(\mathcal{P}, \mathcal{Q}) = 0$ or not.

Suppose that we obtain a terminal separator U of size k. Then, we obtain subgraphs G_1 and G_2 of G such that $G = G_1 \cup G_2$, $V(G_1) \cap V(G_2) = U$, $\{s_1, \ldots, s_k\} \subseteq V(G_1)$, and $\{t_1, \ldots, t_k\} \subseteq V(G_2)$. We test whether $V(P_i) \cap U = V(Q_i) \cap U$ holds for any $i \in [k]$ or not, where we note that each of $V(P_i) \cap U$ and $V(Q_i) \cap U$ consists of a single vertex. If this does not hold, then we can immediately conclude that $\mathcal{P}$ is not reconfigurable to $\mathcal{Q}$, because $V(P_i) \cap U$ does not change in the reconfiguration. If $V(P_i) \cap U = V(Q_i) \cap U$ for $i \in [k]$, then we consider the instance $I_i = (G_i, \mathcal{P}_i, \mathcal{Q}_i)$ for $i = 1, 2$, where $\mathcal{P}_i$ and $\mathcal{Q}_i$ are the restrictions of $\mathcal{P}$ and $\mathcal{Q}$ to G_i. That is, I_i is the restriction of I to G_i. Then, we see that $\mathcal{P}$ is reconfigurable to $\mathcal{Q}$ if and only if $\mathcal{P}_i$ is reconfigurable to $\mathcal{Q}_i$ for $i = 1, 2$. Since I_1 and I_2 are one-face or two-face instances, by solving them recursively, we can solve the original instance I in polynomial time. ◀

4.2 Reconfiguration of s-t Paths

In this subsection, for Disjoint s-t Paths Reconfiguration in planar graphs, we show results that are analogous to Theorems 14 and 3, which have been already stated in Section 1.2.

▶ **Theorem 5.** *Let $G = (V, E)$ be a planar graph with distinct vertices s and t, and let $\mathcal{P}$ and $\mathcal{Q}$ be s-t linkages of size k. If there is no s-t separator of size k, then $\mathcal{P}$ is reconfigurable to $\mathcal{Q}$.*

Proof. Suppose that G, s, t, $\mathcal{P}$, and $\mathcal{Q}$ are as in the statement, and assume that there is no s-t separator of size k. We fix an embedding of G on the plane. If there is an edge connecting s and t, then s and t are on the boundary of some face, and hence $\mathcal{P}$ is reconfigurable to $\mathcal{Q}$ in the same way as Proposition 13. Thus, it suffices to consider the case when there is no edge connecting s and t.

We now construct an instance of Disjoint Paths Reconfiguration by replacing s and t with large "grids" as follows. Let $e_1, e_2, \ldots, e_\ell$ be the edges incident to s clockwise in this order. Note that $\ell \geq k + 1$ holds, because G has no s-t separator of size k. For $i \in [\ell]$, we subdivide e_i by introducing p new vertices $v_i^1, v_i^2, \ldots, v_i^p$ such that they are aligned in this order and v_i^1 is closest to s, where p is a sufficiently large integer (e.g., $p \geq |V|^2$). For $i \in [\ell]$ and for $j \in [p]$, we introduce a new edge connecting v_i^j and v_{i+1}^j, where $v_{\ell+1}^j = v_1^j$. Define $s_i = v_i^1$ for $i \in [k]$ and remove s. Then, the graph is embedded on the plane and $s_1, \ldots, s_k$ are on the boundary of some face clockwise in this order; see Figure 8. By applying a similar procedure to t, we modify the graph around t and define $t_1, \ldots, t_k$ that are on the boundary of some face counter-clockwise in this order. Let G' be the obtained graph. Observe that G' contains no terminal separator of size k, because G has no s-t separator of size k.

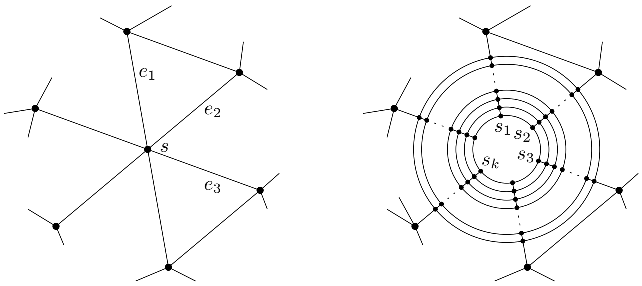

Figure 8 (Left) Original graph G. (Right) Modification around s.

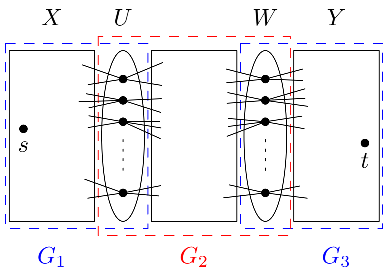

Figure 9 Construction of G_1, G_2, and G_3.

By rerouting the given s-t linkages $\mathcal{P}$ and $\mathcal{Q}$ around s and t, we obtain linkages $\mathcal{P}'$ and $\mathcal{Q}'$ from $\{s_1, \dots, s_k\}$ to $\{t_1, \dots, t_k\}$ in G'. Note that the restrictions of $\mathcal{P}$ and $\mathcal{Q}$ to $G \setminus \{s, t\}$ coincide with those of $\mathcal{P}'$ and $\mathcal{Q}'$, respectively. Then, we can take $\mathcal{P}'$ and $\mathcal{Q}'$ so that $|\mu(\mathcal{P}', \mathcal{Q}')| \leq |V|$. Furthermore, using at most $|V|$ concentric cycles around s and t, we can reroute the linkages so that the value $\mu(\mathcal{P}', \mathcal{Q}')$ decreases or increases by one. Therefore, using $p \geq |V|^2$ concentric cycles, we can reroute $\mathcal{P}'$ and $\mathcal{Q}'$ so that $\mu(\mathcal{P}', \mathcal{Q}')$ becomes zero.

By Theorem 14, $\mathcal{P}'$ is reconfigurable to $\mathcal{Q}'$ in G' (in terms of Disjoint Paths Reconfiguration). Then, the reconfiguration sequence from $\mathcal{P}'$ to $\mathcal{Q}'$ corresponds to that from $\mathcal{P}$ to $\mathcal{Q}$ in G (in terms of Disjoint s-t Paths Reconfiguration). Therefore, $\mathcal{P}$ is reconfigurable to $\mathcal{Q}$ in G. ◀

▶ **Theorem 4.** *There is a polynomial-time algorithm for Disjoint s-t Paths Reconfiguration in planar graphs.*

Proof. Suppose that we are given a planar graph $G = (V, E)$ with $s, t \in V$ and s-t linkages $\mathcal{P} = \{P_1, \dots, P_k\}$ and $\mathcal{Q} = \{Q_1, \dots, Q_k\}$ in G. We first test whether G has an s-t separator of size k. If there is no such a separator, then we can immediately conclude that $\mathcal{P}$ is reconfigurable to $\mathcal{Q}$ by Theorem 5.

Suppose that G has an s-t separator of size k. Let X be the inclusionwise minimal vertex set subject to $s \in X$ and $N(X)$ is an s-t separator of size k. Note that such X is uniquely determined by the submodularity of $|N(X)|$ and it can be computed in polynomial time by a standard minimum cut algorithm. Similarly, let Y be the unique inclusionwise minimal vertex set subject to $t \in Y$ and $N(Y)$ is an s-t separator of size k. Let $U = N(X)$, $W = N(Y)$, $G_1 = G[X \cup U]$, $G_2 = G \setminus (X \cup Y)$, and $G_3 = G[Y \cup W]$; see Figure 9. Since $V(P_i) \cap U$ and $V(P_i) \cap W$ do not change in the reconfiguration, we can consider the reconfiguration in G_1, G_2, and G_3, separately.

We first consider the reconfiguration in G_1. Observe that each path in $\mathcal{P}$ contains exactly one vertex in U, and the restriction of $\mathcal{P}$ to G_1 consists of k paths from s to U that are vertex-disjoint except at s. The same for $\mathcal{Q}$. By the minimality of X, G_1 contains no vertex set of size k that separates $\{s\}$ and U. Therefore, by the same argument as Theorem 5, the restriction of $\mathcal{P}$ to G_1 is reconfigurable to that of $\mathcal{Q}$.

If $U \cap W \neq \emptyset$, then $G \setminus X$ contains no vertex set of size k that separates U and $\{t\}$ by the minimality of Y. In such a case, by applying the same argument as above, the restriction of $\mathcal{P}$ to $G \setminus X$ is reconfigurable to that of $\mathcal{Q}$. By combining the reconfiguration in G_1 and that in $G \setminus X$, we obtain a reconfiguration sequence from $\mathcal{P}$ to $\mathcal{Q}$.

Therefore, it suffices to consider the case when $U \cap W = \emptyset$. In the same way as G_1, we see that the restriction of $\mathcal{P}$ to G_3 is reconfigurable to that of $\mathcal{Q}$. This shows that the reconfigurability from $\mathcal{P}$ to $\mathcal{Q}$ in G is equivalent to that in G_2. By changing the indices if necessary, we may assume that $P_i \cap U = Q_i \cap U$ for $i \in [k]$. If $P_i \cap W \neq Q_i \cap W$ for some $i \in [k]$, then we can conclude that $\mathcal{P}$ is not reconfigurable to $\mathcal{Q}$. Otherwise, let $\mathcal{P}'$ and $\mathcal{Q}'$ be the restrictions of $\mathcal{P}$ and $\mathcal{Q}$ to G_2, respectively. Since $(G_2, \mathcal{P}', \mathcal{Q}')$ is a one-face or two-face instance of Disjoint Paths Reconfiguration, we can solve it in polynomial time by Proposition 13 and Theorem 3. Therefore, we can test the reconfigurability from $\mathcal{P}$ to $\mathcal{Q}$ in polynomial time; ◀

5 Proof of Theorem 14

The necessity ("only if" part) in Theorem 14 is immediately derived from Corollary 12.

In what follows in this section, we show the sufficiency ("if" part) in Theorem 14, which is one of the main technical contributions of this paper. Assume that $\mu(\mathcal{P}, \mathcal{Q}) = 0$ and there is no terminal separator of size k. The objective is to show that $\mathcal{P}$ is reconfigurable to $\mathcal{Q}$. Our proof is constructive, and based on topological arguments. A similar technique is used in [27, 32, 31, 37].

5.1 Preliminaries for the Proof

Let C be a simple curve connecting the boundaries of S and T such that C contains no vertex in G, C intersects the boundaries of S and T only at its endpoints, and $\mu(P_i, C) = 0$ for $i \in [k]$. Note that such C always exists, because the last condition is satisfied if C is disjoint from $\mathcal{P}$. Note also that $\mu(Q_i, C) = 0$ holds for $i \in [k]$, because $\mu(\mathcal{P}, \mathcal{Q}) = 0$.

Since T is the outer face, $\mathbb{R}^2 \setminus (S \cup T)$ forms an annulus (or a cylinder).[6] Thus, by cutting it along C, we obtain a rectangle whose boundary consists of ∂S, ∂T, and two copies of C. We take infinite copies of this rectangle and glue them together to obtain an infinitely long strip R. That is, for $j \in \mathbb{Z}$, let C^j be a copy of C, let R^j be a copy of the rectangle whose boundary contains C^j and C^{j+1}, and define $R = \bigcup_{j \in \mathbb{Z}} R^j$; see Figure 10. By taking C appropriately, we may assume that the copies of $s_1, \dots, s_k$ lie on the boundary of R^j in this order so that s_1 is closest to C^j and s_k is closest to C^{j+1}. The same for $t_1, \dots, t_k$. Note that R is called the *universal cover* of $\mathbb{R}^2 \setminus (S \cup T)$ in the terminology of topology.

Since G is embedded on $\mathbb{R}^2 \setminus (S \cup T)$, this operation naturally defines an infinite periodic graph $\hat{G} = (\hat{V}, \hat{E})$ on R that consists of copies of G. A path in $\hat{G}$ is identified with the corresponding curve in R. For $v \in V$ and $j \in \mathbb{Z}$, let $v^j \in \hat{V}$ denote the unique vertex in R^j

[6] More precisely, the annulus is degenerated when $\partial S \cap \partial T \neq \emptyset$, but the same argument works even for this case.

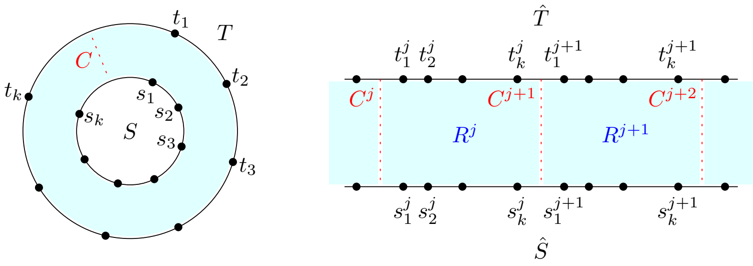

Figure 10 (Left) Curve C in $\mathbb{R}^2 \setminus (S \cup T)$. (Right) Construction of R.

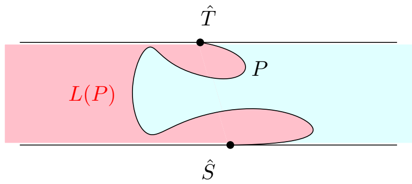

Figure 11 Definition of $L(P)$.

that corresponds to v. Since $\mu(P_i, C) = 0$ for $i \in [k]$, each path in $\hat{G}$ corresponding to P_i is from s_i^j to t_i^j for some $j \in \mathbb{Z}$, and we denote such a path by P_i^j. We define Q_i^j in the same way. Since $\mathcal{P}$ and $\mathcal{Q}$ are linkages in G, $\{P_i^j \mid i \in [k],\ j \in \mathbb{Z}\}$ and $\{Q_i^j \mid i \in [k],\ j \in \mathbb{Z}\}$ are sets of vertex-disjoint paths in $\hat{G}$.

A path in $\hat{G}$ connecting the boundary of R corresponding to ∂S and that corresponding to ∂T is called an *$\hat{S}$-$\hat{T}$ path*. For an $\hat{S}$-$\hat{T}$ path P, let $L(P)$ be the region of $R \setminus P$ that is on the "left-hand side" of P. Formally, let r be a point in R^j for sufficiently small j, and define $L(P)$ as the set of points $x \in R \setminus P$ such that any curve in R between r and x crosses P an even number of times; see Figure 11. For two $\hat{S}$-$\hat{T}$ paths P and Q, we denote $P \preceq Q$ if $L(P) \subseteq L(Q)$, and denote $P \prec Q$ if $L(P) \subsetneq L(Q)$. For two linkages $\mathcal{P} = (P_1, \dots, P_k)$ and $\mathcal{Q} = (Q_1, \dots, Q_k)$ in G with $\mu(P_i, C) = \mu(Q_i, C) = 0$ for $i \in [k]$, we denote $\mathcal{P} \preceq \mathcal{Q}$ if $P_i^j \preceq Q_i^j$ for any $i \in [k]$ and $j \in \mathbb{Z}$, and denote $\mathcal{P} \prec \mathcal{Q}$ if $\mathcal{P} \preceq \mathcal{Q}$ and $\mathcal{P} \neq \mathcal{Q}$.

5.2 Case When $\mathcal{P} \preceq \mathcal{Q}$

In this subsection, we consider the case when $\mathcal{P} \preceq \mathcal{Q}$, and the general case will be dealt with in Section 5.3. To show that $\mathcal{P}$ is reconfigurable to $\mathcal{Q}$, we show the following lemma.

▶ **Lemma 15.** *If $\mathcal{P} \prec \mathcal{Q}$, then there exists a linkage $\mathcal{P}'$ such that $\mathcal{P} \leftrightarrow \mathcal{P}'$ and $\mathcal{P} \prec \mathcal{P}' \preceq \mathcal{Q}$.*

Proof. Let $\hat{W} := \{\hat{v} \in \hat{V} \mid \hat{v} \in P_i^j \setminus Q_i^j$ for some $i \in [k]$ and $j \in \mathbb{Z}\}$ and let W be the subset of V corresponding to $\hat{W}$. If $W = \emptyset$, then take an index $i \in [k]$ such that $P_i \neq Q_i$ and let $\mathcal{P}' = (P'_1, \dots, P'_k)$ be the set of paths obtained from $\mathcal{P}$ by replacing P_i with Q_i. Since $\mathcal{Q}$ is a linkage and all the vertices in P'_h are contained in Q_h for $h \in [k]$, $\mathcal{P}'$ is a desired linkage. Thus, it suffices to consider the case when $W \neq \emptyset$.

Let $u \in W$. Let $\hat{u} \in \hat{W}$ be a vertex corresponding to u and let $i \in [k]$ and $j \in \mathbb{Z}$ be the indices such that $\hat{u} \in P_i^j \setminus Q_i^j$. Since $\hat{u} \in P_i^j \setminus Q_i^j$ implies $\hat{u} \in L(Q_i^j) \setminus L(P_i^j)$, there exists a face $\hat{F}$ of $\hat{G}$ such that $\partial \hat{F}$ contains an edge of P_i^j incident to $\hat{u}$ and $\hat{F} \subseteq L(Q_i^j) \setminus L(P_i^j)$. Define $(P_i^j)'$ as the s_i^j-t_i^j path in $\hat{G}$ with maximal $L((P_i^j)')$ subject to $(P_i^j)' \subseteq P_i^j \cup \partial \hat{F}$; see Figure 12. Note that such a path is uniquely determined, it satisfies $P_i^j \prec (P_i^j)' \preceq Q_i^j$, and it can be found in polynomial time.

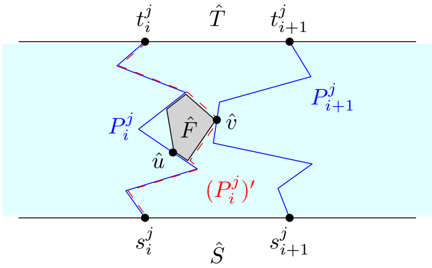

Figure 12 The blue thick paths are P_i^j and P_{i+1}^j, and the red dashed path is $(P_i^j)'$. There exists a vertex $\hat{v} \in \partial\hat{F} \cap P_{i+1}^j$.

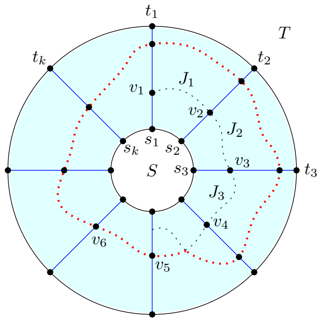

Figure 13 Each blue path represents P_i. The dotted curve is part of J and the red dotted thick curve is C^*.

Let P_i' be the s_i-t_i path in G that corresponds to $(P_i^j)'$. If P_i' is disjoint from P_h for any $h \in [k] \setminus \{i\}$, then we can obtain a desired linkage $\mathcal{P}'$ from $\mathcal{P}$ by replacing P_i with P_i'. Otherwise, P_i' intersects P_h for some $h \in [k] \setminus \{i\}$. This together with $P_i^j \prec (P_i^j)'$ shows that $(P_i^j)'$ intersects P_{i+1}^j, where P_{k+1}^j means P_1^{j+1}. Since P_i^j and P_{i+1}^j are vertex-disjoint, the intersection of $(P_i^j)'$ and P_{i+1}^j is contained in $\partial\hat{F}$, which implies that $\partial\hat{F} \cap P_{i+1}^j$ contains a vertex $\hat{v} \in \hat{V}$; see Figure 12 again. Since $\hat{F} \subseteq L(Q_i^j)$, we obtain $\hat{v} \in L(Q_i^j) \cup Q_i^j \subseteq L(Q_{i+1}^j)$, and hence $\hat{v} \notin Q_{i+1}^j$. Let v and F be the vertex and the face of G that correspond to $\hat{v}$ and $\hat{F}$, respectively. Then, $\hat{v} \in P_{i+1}^j \setminus Q_{i+1}^j$ implies that $\hat{v} \in \hat{W}$ and $v \in W$. Note that there exists a curve in F from u to v.

By the above argument, for any $u \in W$, we can obtain

(i) a desired linkage $\mathcal{P}'$, or

(ii) a vertex $v \in W$ on P_{i+1} and a curve from u to v contained in some face of G, where i is the index with $u \in V(P_i)$.

Therefore, it suffices to show that we obtain the outcome (i) for some $u \in W$. To derive a contradiction, assume to the contrary that we obtain the outcome (ii) for any $u \in W$.

By using the outcome (ii) repeatedly and by shifting the indices of P_i if necessary, we obtain v_i and J_i for $i = 1, 2, \dots$ such that $v_i \in W$ is on P_i (where the index is modulo k) and J_i is a curve from v_i to v_{i+1} contained in some face. We consider the curve J obtained by concatenating $J_1, J_2, \dots$ in this order. Since $|W|$ is finite, this curve visits the same point more than once, and hence it contains a simple closed curve C^*. Since C^* is simple and visits vertices on $P_i, P_{i+1}, \dots$ in this order, C^* surrounds S exactly once in the clockwise direction; see Figure 13. In particular, C^* contains exactly one vertex on P_i for each $i \in [k]$. Let U be the set of vertices in V contained in C^*. Then, $|U| = k$ and $G \setminus U$ has no path between $\{s_1, \dots, s_k\}$ and $\{t_1, \dots, t_k\}$ by the choice of C^*. Furthermore, U contains no terminals, because $U \subseteq W$ and W contains no terminals. Therefore, U is a terminal separator of size k, which contradicts the assumption. ◀

As long as $\mathcal{P} \neq \mathcal{Q}$, we apply this lemma and replace $\mathcal{P}$ with $\mathcal{P}'$, repeatedly. Then, this procedure terminates when $\mathcal{P} = \mathcal{Q}$, and gives a reconfiguration sequence from $\mathcal{P}$ to $\mathcal{Q}$. This completes the proof for the case when $\mathcal{P} \preceq \mathcal{Q}$.

We now give a remark on the length of the reconfiguration sequence. For $\hat{S}$-$\hat{T}$ paths P and Q with the same endpoints, we see that $L(Q) \setminus L(P)$ contains $O(|V|^2)$ faces of $\hat{G}$. Therefore, in the reconfiguration sequence from $\mathcal{P}$ to $\mathcal{Q}$ obtained above, each path in $\mathcal{P}$ is replaced with another path $O(|V|^2)$ times, which shows that the number of applications of Lemma 15 is $O(k|V|^2)$ in total.

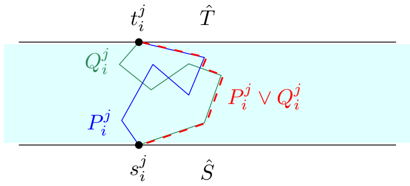

Figure 14 Construction of $P_i^j \vee Q_i^j$.

5.3 General Case

In this subsection, we consider the case when $\mathcal{P} \preceq \mathcal{Q}$ does not necessarily hold. For $i \in [k]$ and $j \in \mathbb{Z}$, define $P_i^j \vee Q_i^j$ as the s_i^j-t_i^j path in $\hat{G}$ with maximal $L(P_i^j \vee Q_i^j)$ subject to $P_i^j \vee Q_i^j \subseteq P_i^j \cup Q_i^j$; see Figure 14. Note that such a path is uniquely determined, $P_i^j \preceq P_i^j \vee Q_i^j$, and $Q_i^j \preceq P_i^j \vee Q_i^j$. Since $\hat{G}$ is periodic, for any $j \in \mathbb{Z}$, $P_i^j \vee Q_i^j$ corresponds to a common s_i-t_i walk $P_i \vee Q_i$ in G. Actually, $\mathcal{P} \vee \mathcal{Q} := (P_1 \vee Q_1, \dots, P_k \vee Q_k)$ is a linkage in G.

▶ **Lemma 16.** *$\mathcal{P} \vee \mathcal{Q}$ is a linkage in G.*

Proof. We first show that $P_i \vee Q_i$ is a path for each $i \in [k]$. Assume to the contrary that $P_i \vee Q_i$ visits a vertex $v \in V$ more than once. Then, for $j \in \mathbb{Z}$, there exist $j_1, j_2 \in \mathbb{Z}$ with $j_1 < j_2$ such that $P_i^j \vee Q_i^j$ contains both v^{j_1} and v^{j_2}. Since the path $P_i^j \vee Q_i^j$ is contained in the subgraph $P_i^j \cup Q_i^j$, without loss of generality, we may assume that P_i^j contains v^{j_2}. This shows that $v^{j_1} \in L(P_i^j) \subseteq L(P_i^j \vee Q_i^j)$, which contradicts that v^{j_1} is contained in $P_i^j \vee Q_i^j$.

We next show that $P_1 \vee Q_1, \dots, P_k \vee Q_k$ are pairwise vertex-disjoint. Assume to the contrary that $P_i \vee Q_i$ and $P_{i'} \vee Q_{i'}$ contain a common vertex $v \in V$ for distinct $i, i' \in [k]$. Since $\hat{G}$ is periodic, there exist $j, j' \in \mathbb{Z}$ such that $P_i^j \vee Q_i^j$ and $P_{i'}^{j'} \vee Q_{i'}^{j'}$ contain v^0 (i.e., the copy of v in R^0). We may assume that (j, i) is smaller than (j', i') in the lexicographical ordering, that is, either $j < j'$ holds or $j = j'$ and $i < i'$ hold. Since $P_i^j \vee Q_i^j \subseteq P_i^j \cup Q_i^j$, we may also assume that $v^0 \in P_i^j$ by changing the roles of P_i^j and Q_i^j if necessary. Then, we obtain $v^0 \in P_i^j \subseteq L(P_{i'}^{j'}) \subseteq L(P_{i'}^{j'} \vee Q_{i'}^{j'})$, which contradicts that v^0 is contained in $P_{i'}^{j'} \vee Q_{i'}^{j'}$. ◀

We also see that $\mu(P_i \vee Q_i, C) = 0$ for $i \in [k]$ by definition, and hence $\mu(\mathcal{P}, \mathcal{P} \vee \mathcal{Q}) = 0$. Since $\mathcal{P} \preceq \mathcal{P} \vee \mathcal{Q}$ and $\mu(\mathcal{P}, \mathcal{P} \vee \mathcal{Q}) = 0$, $\mathcal{P}$ is reconfigurable to $\mathcal{P} \vee \mathcal{Q}$ as described in Section 5.2. Similarly, $\mathcal{Q}$ is reconfigurable to $\mathcal{P} \vee \mathcal{Q}$, which implies that $\mathcal{P} \vee \mathcal{Q}$ is reconfigurable to $\mathcal{Q}$. By combining them, we see that $\mathcal{P}$ is reconfigurable to $\mathcal{Q}$, which completes the proof of the sufficiency in Theorem 14.

Note that the reconfiguration sequence from $\mathcal{P}$ to $\mathcal{Q}$ can be constructed in polynomial time by the discussion in Section 5.2.

6 Concluding Remarks

Although Disjoint Paths Reconfiguration and Disjoint s-t Paths Reconfiguration are decision problems, the proofs for our positive results (Theorems 3, 4, 5, and 14) show that we can find a reconfiguration sequence in polynomial time if it exists.

We leave several open problems for future research. We proved that Disjoint Paths Reconfiguration can be solved in polynomial time when the problem is restricted to the two-face instances. On the other hand, we do not know whether Disjoint Paths Reconfiguration in planar graphs can be solved in polynomial time for fixed k, and even when $k = 2$, if we drop the requirement that inputs are two-face instances.

We did not try to minimize the number of reconfiguration steps when a reconfiguration sequence exists. It is an open problem whether a shortest reconfiguration sequence can be found in polynomial time for Disjoint Paths Reconfiguration restricted to planar two-face instances.

A natural extension of our studies is to consider a higher-genus surface. As a preliminary result, in the full version [20], we give a proof (sketch) to show that when the number k of curves is two, the reconfiguration is *always* possible for any connected orientable closed surface Σ_g of genus $g \geq 1$. Note that this result does not refer to graphs embedded on Σ_g, but only refers to the case when curves can pass through any points on the surface. It is not clear what we can say for Disjoint Paths Reconfiguration for graphs embedded on Σ_g, $g \geq 1$.

References

1 Isolde Adler, Stavros G Kolliopoulos, Philipp Klaus Krause, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M Thilikos. Irrelevant vertices for the planar disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 122:815–843, 2017. doi:10.1016/j.jctb.2016.10.001.

2 Saeed Akhoondian Amiri, Szymon Dudycz, Mahmoud Parham, Stefan Schmid, and Sebastian Wiederrecht. On polynomial-time congestion-free software-defined network updates. In *2019 IFIP Networking Conference, Networking 2019, Warsaw, Poland, May 20-22, 2019*, pages 1–9. IEEE, 2019. doi:10.23919/IFIPNetworking.2019.8816833.

3 Saeed Akhoondian Amiri, Szymon Dudycz, Stefan Schmid, and Sebastian Wiederrecht. Congestion-free rerouting of flows on DAGs. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 143:1–143:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.143.

4 John Asplund, Kossi D. Edoh, Ruth Haas, Yulia Hristova, Beth Novick, and Brett Werner. Reconfiguration graphs of shortest paths. *Discret. Math.*, 341(10):2938–2948, 2018. doi:10.1016/j.disc.2018.07.007.

5 John Asplund and Brett Werner. Classification of reconfiguration graphs of shortest path graphs with no induced 4-cycles. *Discret. Math.*, 343(1):111640, 2020. doi:10.1016/j.disc.2019.111640.

6 Paul S. Bonsma. The complexity of rerouting shortest paths. *Theor. Comput. Sci.*, 510:1–12, 2013. doi:10.1016/j.tcs.2013.09.012.

7 Paul S. Bonsma. Rerouting shortest paths in planar graphs. *Discret. Appl. Math.*, 231:95–112, 2017. doi:10.1016/j.dam.2016.05.024.

8 Glencora Borradaile, Amir Nayyeri, and Farzad Zafarani. Towards single face shortest vertex-disjoint paths in undirected planar graphs. In *Proceedings of 23rd Annual European Symposium on Algorithms (ESA)*, volume 9294 of *Lecture Notes in Computer Science*, pages 227–238, 2015. doi:10.1007/978-3-662-48350-3_20.

9 Éric Colin de Verdière and Alexander Schrijver. Shortest vertex-disjoint two-face paths in planar graphs. *ACM Transactions on Algorithms*, 7(2):1–12, 2011. doi:10.1145/1921659.1921665.

10 Marek Cygan, Dániel Marx, Marcin Pilipczuk, and Michal Pilipczuk. The planar directed k-vertex-disjoint paths problem is fixed-parameter tractable. In *54th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013)*, pages 197–206, 2013. doi:10.1109/FOCS.2013.29.

11 Samir Datta, Siddharth Iyer, Raghav Kulkarni, and Anish Mukherjee. Shortest k-disjoint paths via determinants. In *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2018)*, volume 122, pages 19:1–19:21, 2018. doi:10.4230/LIPIcs.FSTTCS.2018.19.

12 Julian Enoch, Kyle Fox, Dor Mesica, and Shay Mozes. A faster algorithm for maximum flow in directed planar graphs with vertex capacities. In Hee-Kap Ahn and Kunihiko Sadakane, editors, *32nd International Symposium on Algorithms and Computation, ISAAC 2021, December 6-8, 2021, Fukuoka, Japan*, volume 212 of *LIPIcs*, pages 72:1–72:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ISAAC.2021.72.

13 Benson Farb and Dan Margalit. *A primer on mapping class groups*, volume 49 of *Princeton Mathematical Series*. Princeton University Press, Princeton, NJ, 2012.

14 Lester R. Ford and Delbert R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956. doi:10.4153/CJM-1956-045-5.

15 Steven Fortune, John Hopcroft, and James Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10(2):111–121, 1980. doi:10.1016/0304-3975(80)90009-2.

16 András Frank. Packing paths, circuits, and cuts – A survey. In *Paths, Flows, and VLSI-Layout*, pages 47–100. Springer, 1990.

17 Kshitij Gajjar, Agastya Vibhuti Jha, Manish Kumar, and Abhiruk Lahiri. Reconfiguring shortest paths in graphs. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022*, pages 9758–9766. AAAI Press, 2022.

18 Refael Hassin. On multicommodity flows in planar graphs. *Networks*, 14(2):225–235, 1984. doi:10.1002/net.3230140204.

19 Takehiro Ito, Erik D. Demaine, Nicholas J. A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theor. Comput. Sci.*, 412(12-14):1054–1065, 2011. doi:10.1016/j.tcs.2010.12.005.

20 Takehiro Ito, Yuni Iwamasa, Naonori Kakimura, Yusuke Kobayashi, Shun ichi Maezawa, Yuta Nozaki, Yoshio Okamoto, and Kenta Ozeki. Rerouting planar curves and disjoint paths. arXiv:2210.11778, 2022. doi:10.48550/arXiv.2210.11778.

21 Takehiro Ito, Naonori Kakimura, Naoyuki Kamiyama, Yusuke Kobayashi, and Yoshio Okamoto. Reconfiguration of maximum-weight b-matchings in a graph. *J. Comb. Optim.*, 37(2):454–464, 2019. doi:10.1007/s10878-018-0289-3.

22 Takehiro Ito, Jun Kawahara, Yu Nakahata, Takehide Soh, Akira Suzuki, Junichi Teruyama, and Takahisa Toda. ZDD-based algorithmic framework for solving shortest reconfiguration problems. arXiv:2207.13959, 2022. doi:10.48550/arXiv.2207.13959.

23 Marcin Kaminski, Paul Medvedev, and Martin Milanic. Shortest paths between shortest paths. *Theor. Comput. Sci.*, 412(39):5205–5210, 2011. doi:10.1016/j.tcs.2011.05.021.

24 Haim Kaplan and Yahav Nussbaum. Maximum flow in directed planar graphs with vertex capacities. *Algorithmica*, 61(1):174–189, 2011. doi:10.1007/s00453-010-9436-7.

25 Richard M Karp. On the computational complexity of combinatorial problems. *Networks*, 5(1):45–68, 1975. doi:10.1002/net.1975.5.1.45.

26 Samir Khuller and Joseph Naor. Flow in planar graphs with vertex capacities. *Algorithmica*, 11(3):200–225, 1994. doi:10.1007/BF01240733.

27 Yusuke Kobayashi and Kensuke Otsuki. Max-flow min-cut theorem and faster algorithms in a circular disk failure model. In *2014 IEEE Conference on Computer Communications (INFOCOM)*, pages 1635–1643, 2014. doi:10.1109/INFOCOM.2014.6848100.

28 Yusuke Kobayashi and Christian Sommer. On shortest disjoint paths in planar graphs. *Discrete Optimization*, 7(4):234–245, 2010. doi:10.1016/j.disopt.2010.05.002.

29 Mark R Kramer. The complexity of wire-routing and finding minimum area layouts for arbitrary VLSI circuits. *Advances in Computing Research*, 2:129–146, 1984.

30 James F Lynch. The equivalence of theorem proving and the interconnection problem. *ACM SIGDA Newsletter*, 5(3):31–36, 1975. doi:10.1145/1061425.1061430.

31 Colin J. H. McDiarmid, Bruce A. Reed, Alexander Schrijver, and F. Bruce Shepherd. Non-interfering network flows. In *Proceedings of the Third Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 245–257, 1992. doi:10.1007/3-540-55706-7_21.

32 Colin J. H. McDiarmid, Bruce A. Reed, Alexander Schrijver, and F. Bruce Shepherd. Induced circuits in planar graphs. *Journal of Combinatorial Theory, Series B*, 60(2):169–176, 1994. doi:10.1006/jctb.1994.1011.

33 Karl Menger. Zur allgemeinen Kurventheorie. *Fundamenta Mathematicae*, 10:96–115, 1927.

34 Naomi Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4):52, 2018. doi:10.3390/a11040052.

35 Haruko Okamura. Multicommodity flows in graphs. *Discrete Applied Mathematics*, 6(1):55–62, 1983. doi:10.1016/0166-218X(83)90100-2.

36 Haruko Okamura and Paul D. Seymour. Multicommodity flows in planar graphs. *Journal of Combinatorial Theory, Series B*, 31(1):75–81, 1981. doi:10.1016/S0095-8956(81)80012-3.

37 Kensuke Otsuki, Yusuke Kobayashi, and Kazuo Murota. Improved max-flow min-cut algorithms in a circular disk failure model with application to a road network. *Eur. J. Oper. Res.*, 248(2):396–403, 2016. doi:10.1016/j.ejor.2015.07.035.

38 Bruce Reed. Rooted routing in the plane. *Discrete Applied Mathematics*, 57:213–227, 1995. doi:10.1016/0166-218X(94)00104-L.

39 Bruce Reed, Neil Robertson, Alexander Schrijver, and Paul D. Seymour. Finding disjoint trees in planar graphs in linear time. In *Contemporary Mathematics 147*, pages 295–301. American Mathematical Society, 1993. doi:10.1090/conm/147/01180.

40 Neil Robertson and Paul D. Seymour. Graph minors. VI. Disjoint paths across a disc. *Journal of Combinatorial Theory, Series B*, 41(1):115–138, 1986. doi:10.1016/0095-8956(86)90031-6.

41 Neil Robertson and Paul D. Seymour. Graph minors. VII. Disjoint paths on a surface. *Journal of Combinatorial Theory, Series B*, 45(2):212–254, 1988. doi:10.1016/0095-8956(88)90070-6.

42 Neil Robertson and Paul D. Seymour. Graph minors. XIII. The disjoint paths problem. *Journal of combinatorial theory, Series B*, 63(1):65–110, 1995. doi:10.1006/jctb.1995.1006.

43 Alexander Schrijver. Finding k disjoint paths in a directed planar graph. *SIAM Journal on Computing*, 23(4):780–788, 1994. doi:10.1137/S0097539792224061.

44 Paul D. Seymour. Disjoint paths in graphs. *Discrete Mathematics*, 29:293–309, 1980. doi:10.1016/0012-365X(80)90158-2.

45 Yossi Shiloach. A polynomial solution to the undirected two paths problem. *Journal of the ACM*, 27(3):445–456, 1980. doi:10.1145/322203.322207.

46 Anand Srinivas and Eytan Modiano. Finding minimum energy disjoint paths in wireless ad-hoc networks. *Wireless Networks*, 11(4):401–417, 2005. doi:10.1007/s11276-005-1765-0.

47 Hitoshi Suzuki, Takehiro Akama, and Takao Nishizeki. Algorithms for finding internally disjoint paths in a planar graph. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 72(10):55–67, 1989. doi:10.1002/ecjc.4430721006.

48 Hitoshi Suzuki, Takehiro Akama, and Takao Nishizeki. Finding Steiner forests in planar graphs. In *Proceedings of the first annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 444–453, 1990.

49 Carsten Thomassen. 2-linked graphs. *European Journal of Combinatorics*, 1:371–378, 1980. doi:10.1016/S0195-6698(80)80039-4.

50 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.

51 Yipu Wang. Max flows in planar graphs with vertex capacities. *ACM Trans. Algorithms*, 18(1), 2022. doi:10.1145/3504032.

52 Marcin Wrochna. Reconfiguration in bounded bandwidth and tree-depth. *J. Comput. Syst. Sci.*, 93:1–10, 2018. doi:10.1016/j.jcss.2017.11.003.