# Efficient Graph Minors Theory and Parameterized Algorithms for (Planar) Disjoint Paths

Daniel Lokshtanov[1], Saket Saurabh[2,3], and Meirav Zehavi[4](✉)

[1] University of California, Santa Barbara, USA
daniello@ucsb.edu
[2] IRL 2000 ReLaX, Indian Institute of Mathematical Sciences, Chennai, India
saket@imsc.res.in
[3] University of Bergen, Bergen, Norway
[4] Ben-Gurion University of the Negev, Beersheba, Israel
meiravze@bgu.ac.il

**Abstract.** In the Disjoint Paths problem, the input consists of an $n$-vertex graph $G$ and a collection of $k$ vertex pairs, $\{(s_i, t_i)\}_{i=1}^{k}$, and the objective is to determine whether there exists a collection $\{P_i\}_{i=1}^{k}$ of $k$ pairwise vertex-disjoint paths in $G$ where the end-vertices of $P_i$ are $s_i$ and $t_i$. This problem was shown to admit an $f(k)n^3$-time algorithm by Robertson and Seymour *Graph Minors XIII, The Disjoint Paths Problem, JCTB*. In modern terminology, this means that Disjoint Paths is fixed parameter tractable (FPT) with respect to $k$. Remarkably, the above algorithm for Disjoint Paths is a cornerstone of the entire Graph Minors Theory, and conceptually vital to the $g(k)n^3$-time algorithm for Minor Testing (given two undirected graphs, $G$ and $H$ on $n$ and $k$ vertices, respectively, determine whether $G$ contains $H$ as a minor).

In this semi-survey, we will first give an exposition of the Graph Minors Theory with emphasis on efficiency from the viewpoint of Parameterized Complexity. Secondly, we will review the state of the art with respect to the Disjoint Paths and Planar Disjoint Paths problems. Lastly, we will discuss the main ideas behind a new algorithm that combines treewidth reduction and an algebraic approach to solve Planar Disjoint Paths in time $2^{k^{\mathcal{O}(1)}} n^{\mathcal{O}(1)}$ (for undirected graphs).

**Keywords:** Disjoint paths · Planar disjoint paths · Graph minors · Treewidth

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 819416 and no. 715744). The second author also acknowledges the support of Swarnajayanti Fellowship grant DST/SJF/MSA-01/2017-18. The third author acknowledges the support of ISF grant no. 1176/18. The first and third authors also acknowledge the support of BSF grant no. 2018302.

F. V. Fomin et al. (Eds.): Bodlaender Festschrift, LNCS 12160, pp. 112–128, 2020.
https://doi.org/10.1007/978-3-030-42071-0_9

## 1 Background on Graph Minors Theory with Emphasis on Efficiency

Arguably, the origin of Parameterized Complexity is the *graph minors project of Robertson and Seymour*. Recollecting the birth of Parameterized Complexity, Downey [21] stated not only that "a real inspiration was the theorem of Robertson and Seymour", but also that in the early years of the field, "many listeners thought that what we were doing was basically applying Robertson-Seymour." The concept of a minor originated already in the early 20th century. Formally, for any two graphs $G$ and $H$, we say that $H$ is a *minor* of $G$ if there exists a series of edge deletions, edge contractions and vertex deletions in $G$ that yields $H$. One of the most famous results in Graph Theory is Kuratowski's theorem [42], which states that a graph is planar if and only if it does not contain the graphs $K_{3,3}$ and $K_5$ as minors. That is, the class of planar graphs is characterized by a set of two forbidden minors. Robertson and Seymour set out to prove a vast generalization of Kuratowski's theorem, namely, Wagner's conjecture [60]: *Any infinite sequence of graphs contains two graphs such that one is a minor of the other (that is, the class of all graphs is well-quasi ordered by the minor relation).* Equivalently, Wagner's conjecture states that *any* minor-closed family of graphs can be characterized by a finite set of forbidden minors.

In perhaps one of the most amazing feats of modern mathematics, Robertson and Seymour managed to prove Wagner's conjecture. The endeavour of Robertson and Seymour to prove this conjecture spans a series of over 23 papers, published from 1983 to 2004. One of the main reasons why this theory has had such a great impact is the sheer number of novel algorithms and algorithmic techniques that were developed as a part of it. A few notable algorithmic highlights are their parameterized algorithms for Minor Testing (given two graphs, $G$ on $n$ vertices and $H$ on $k$ vertices, decide whether $H$ is a minor of $G$), Disjoint Paths (given a graph $G$ and a collection of $k$ terminal pairs, $\{(s_i,t_i)\}|_{i=1}^{k}$, decide whether $G$ has $k$ pairwise vertex-disjoint paths, $\{P_i\}|_{i=1}^{k}$, where for every $i \in \{1,2,\ldots,k\}$, the endpoints of $P_i$ are $s_i$ and $t_i$) and a constant-factor approximation parameterized algorithm to compute the treewidth of a given graph. Their project introduced key definitions and concepts such as those of an *excluded grid* and a *tree decomposition* (a decomposition of a graph into a tree-like structure), along with key structural results such as duality theorems (e.g., the characterization of treewidth in the terms of a family of connected graphs called a *bramble*). Additionally, their project presented new methods such as the so-called *irrelevant vertex technique*. Notably, *all* of these intermediate results have found applications and implications across a wide range of research domains.

Unfortunately, the hidden constants in the results above, both in terms of time complexities and in the structural theorems themselves, are *really, really bad*. In fact, the immense parameter dependence of algorithms based on the graph minors project earned them their own name—"galactic algorithms" [43]. As phrased by Johnson [35], "for any instance $G = (V, E)$ that one could fit into the known universe, one would easily prefer $|V|^{70}$ to even constant time, if that constant had to be one of Robertson and Seymour's". In light of this, Downey [21] stated that "in retrospect, it might have been a bit unfortunate to tie the FPT material to the Robertson-Seymour material when we spoke". Indeed, keeping in mind that the primary objective of the paradigm of Parameterized Complexity is to cope with computational intractability, we are facing a blatant discrepancy:

> **While Parameterized Complexity does provide an extremely rich toolkit to design efficient parameterized algorithms, one of its foundations and still most powerful tools yields algorithms that are wildly impractical.**

In 1989, Fellows [24] noted that "it is likely to be many years before the practical significance of Robertson-Seymour theorems is fully understood". Nevertheless, for some of the algorithms, substantial advances have been made. In particular, Grohe et al. [34] gave an algorithm for computing *Robertson and Seymour's structural decomposition* (stating that all graphs excluding some fixed graph as a minor have a tree decomposition with bags that are almost embeddable in a fixed surface), that runs in time $f(k)n^2$ (for some function $f$ of $k$), improving over the $f(k)n^3$-time algorithm of Robertson and Seymour. Prior to this result, Kawarabayashi and Wollan [38] gave a simplified proof of correctness of the graph minors algorithm. This proof yields a parametrized algorithm for Minor Testing with the best currently known parameter dependence. Here, $f(k)$ is a "tower of powers of 2 of height at most 5, with $k^{1000}$ on top" [61]. From the work of Kawarabayashi et al. [37] on Disjoint Paths, we also know that Minor Testing is solvable in time $f(k)n^2$. Chuzhoy [10], building upon the seminal work of Chekuri and Chuzhoy [9], gave an improved algorithm for a weaker variant of Robertson and Seymour's structural decomposition. However, here the improvement is in the quality of the output decomposition, not in the time it takes to compute it.

Both the algorithm for Minor Testing and the structural theorem of Robertson and Seymour were discovered as consequences of the entire graph minors theory that they built. In retrospect, however, the "converse" also holds true: almost the entire graph minors theory would have had to be built in order to achieve either one of these goals, that is, the algorithm for Minor Testing as well as the structural theorem of Robertson and Seymour! In plain words, the design of an efficient algorithm for Minor Testing as a goal *necessitates* to devise efficient versions of large parts of the whole graph minors theory. A problem that is tightly linked to Minor Testing, yet seemingly more difficult than it, is Topological Minor Testing: given two graphs, $G$ on $n$ vertices and $H$ on $k$ vertices, decide whether $H$ is a *topological minor* of $G$. That is, the objective is to determine whether there exists a series of operations that delete an edge, *dissolve* an edge (i.e. delete a degree-2 vertex and make its two neighbors adjacent) and delete a vertex in $G$ that yields $H$. We remark that Kuratowski's theorem [42] was, in fact, originally phrased in the terms of topological minors rather than minors. Unlike Minor Testing and Disjoint Paths, the question of the parameterized complexity of Topological Minor Testing was not resolved by Robertson and Seymour, and was first stated explicitly as an open problem by Downey and Fellows in [22]. Since then, this question was restated as an open problem many times, until it was positively resolved by Grohe et al. [33], who designed an $f(k)n^3$-time (galactic) algorithm.

**Some Central Applications**
Graph minors in general, and Minor Testing in particular, have enjoyed numerous applications over the past 30 years. These applications span a wide range of areas, including (but not limited to) Approximation Algorithms, Exact Exponential and Polynomial-Time Algorithms, Parameterized Complexity, Logic, Computational Geometry and Property Testing. It is highly conceivable that algorithmically (and structurally) efficient Graph Minors Theory, being a core engine behind all of these applications, will have great impact on all of these areas simultaneously. For the sake of illustration, we briefly discuss three important discoveries (with emphasis on Parameterized Complexity) that build upon graph minors.

**Classification.** By Robertson and Seymour's theorem, every minor-closed family of graphs can be characterized by a finite set of forbidden minors. In particular, for any minor-closed family of graphs $\mathcal{G}$, the family of graphs obtained from the graphs in $\mathcal{G}$ by adding at most $k$ vertices is also minor-closed. As Minor Testing is solvable in time $f(k)n^2$ [37], this observation immediately shows that a vast range of parametrized problems, such as Feedback Vertex Set, Planar Vertex Deletion and Graph Genus, are *non-uniformly FPT*: for every integer $k$, the set of forbidden minors might be different. Furthermore, the result is non-constructive as long as we do not know how to compute the finite set of forbidden minors. Nevertheless, this result provides a very useful classification tool (for whether a problem is FPT or not), known since the early days of Parameterized Complexity.

**Bidimensionality.** The Graph Minors Theory laid the foundation for studying how computational problems that are hard on general graphs behave when restricted to minor-free graphs. The theory of *bidimensionality* [19] builds upon this knowledge, particularly on the relationship between grids and treewidth. While most NP-hard graph problems remain NP-hard even on planar graphs [30], many problems that are fixed-parameter intractable on general graphs are FPT on planar graphs, and even on graph classes excluding a fixed $H$ as a minor. Bidimensionality simultaneously yields linear-time parameterized algorithms with subexponential parameter dependence [19], polynomial-time approximation schemes [20,26], and linear kernels [28] for many problems on minor-free graphs, with applications even in computational geometry [18,27]. Nevertheless, there are fundamental graph problems on planar and minor-free graphs for which bidimensionality seems insufficient. Examples of such problems include the Longest Path problem on directed graphs, Steiner Tree, Odd Cycle Transversal, and many others [25,46,51]. More information can be found in two other chapters in this volume (one by D. Marx, and the other by Ma. Pilipczuk).

**Irrelevant Vertices.** The irrelevant vertex technique originated from Robertson and Seymour's algorithm for the Disjoint Paths problem [54]. Since then, this technique has found several other applications [4,17,32,33,48]. Roughly speaking, as long as the treewidth of the graph is large, the technique is applied by repeatedly finding an *irrelevant vertex*—a vertex whose deletion does not change the answer to the problem. For an illustrative application of this technique in textbook level of detail, we refer to Chapter 7.8 in [16].

## 2 (Planar) Disjoint Paths: State of the Art

Conceptually vital to the algorithm for Minor Testing of Robertson and Seymour, and the source of the irrelevant vertex technique, is their $f(k)n^3$-time algorithm for the Disjoint Paths problem [54]. The current state-of-the-art is the algorithm developed by Kawarabayashi et al. [37] in 2012, which runs in time $f(k)n^2$. Just like the case of Minor Testing, the parameter dependence of this algorithm on $k$ renders it a "galactic algorithm". The Disjoint Paths problem is important on its own right due to its applications in the contexts of transportation networks, VLSI layout and virtual circuit routing [29,50,58,59]. It was shown to be $\mathsf{NP}$-complete by Karp in [36], being one of Karp's original $\mathsf{NP}$-complete problems. In fact, it remains NP-complete even if the input graph is restricted to be a grid [41].

### 2.1 Minor Testing and Disjoint Paths on Planar Graphs

We first remark that the Minor Testing, Disjoint Paths and Topological Minor Testing problems are well known to be $\mathsf{NP}$-hard also when restricted to the class of planar graphs [47]. Moreover, it is easy to see that if (Topological) Minor Testing is solvable in time $2^{k^{\mathcal{O}(1)}}n^c$ (for some $c > 0$) on $H$-minor free graphs, then Disjoint Paths is also solvable in time $2^{k^{\mathcal{O}(1)}}n^c$ (for the same $c$) on this class of graphs. To see this, consider an instance $(G, \{(s_i, t_i)\}|_{i=1}^{k}, k)$ of Disjoint Paths on $H$-minor free graphs. Briefly, the idea is to attach, to each terminal $s_i$ or $t_i$, a "large enough" clique (on $\mathcal{O}(|V(H)|k)$ vertices) of unique size, and define the graph to be sought as a minor as an appropriate combination of these cliques. Unfortunately, this reduction idea is clearly tailored specifically to $H$-minor free graphs—in particular, it is inapplicable to planar graphs and general graphs.

Apart from being important problems on their own right, the design of algorithms for Minor Testing, Disjoint Paths and Topological Minor Testing on planar graphs also serves as a critical building block for the design of algorithms for these problems on general graphs in view of the way Robertson and Seymour's Graph Minors Theory is structured. Without delving into technical details, we note that all known algorithms for (Topological) Minor Testing and Disjoint Paths (on general graphs) are based on the distinction between the case where the input graph $G$ contains a large clique as a minor, and the case where it does not. Already at this stage, we see that the resolution of (Topological) Minor Testing and Disjoint Paths on $H$-minor-free graphs is a building block towards the resolution of these problems on general graphs. Moreover, when the input graph does not contain a large clique as a minor, known algorithms distinguish between the case where the treewidth of $G$ is small, and the case where the treewidth of $G$ is large. In the latter case, $G$ contains a so-called *flat wall* that further motivates, or even necessitates, the study of these problems on planar and "almost planar" graph classes.

With respect to known algorithms, for the Minor Testing problem on planar graphs, the design of a $2^{k^{\mathcal{O}(1)}}n$-time algorithm is folklore: if the input graph $G$ has treewidth larger than some function linear in $k$, then it necessarily contains the sought graph as a minor (this is a property holds only for planar graphs!), and otherwise it is possible to solve the problem in time $2^{k^{\mathcal{O}(1)}}n$ via dynamic programming over tree decompositions (e.g., using [2]). We also remark that, for Minor Testing on planar graphs, Adler et al. [3] developed an algorithm that runs in time $\mathcal{O}(2^{\mathcal{O}(k)}n + n^2 \log n)$.

For the Disjoint Paths problem on planar graphs [52,53], and even on graphs of bounded genus [23,40,52], there already exist algorithms with running times whose dependency on $n$ in linear, but whose dependency on $k$ is prohibitive. Additionally, for the Disjoint Paths problem on planar graphs, Adler et al. [5] developed a $2^{2^{\mathcal{O}(k)}}n^2$-time algorithm; in particular, towards that end, they presented a so-called *unique linkage theorem* that states that, in every instance of Disjoint Paths on planar graphs where the treewidth is larger than $2^{ck}$ (for some $c > 0$), there exists an irrelevant vertex and it is computable in linear time. Such a relation with single exponential dependency of the treewidth bound on $k$ also holds for graphs of bounded genus [49]. However, for the more general $H$-minor-free graphs, the dependency becomes a tower of exponents [31,39] (prior to these works—that is, from the project of Robertson and Seymour—not even the computability of the bound was not known).

The Planar Disjoint Paths problem is intensively studied also from the perspective of approximation algorithms, with a burst of activity in recent years [11–15]. Highlights of this work include an approximation algorithm with approximation factor $\mathcal{O}(n^{9/19} \log^{\mathcal{O}(1)} n)$ [12] and, under reasonable complexity-theoretic assumptions, the proof of hardness of approximating the problem within a factor of $2^{\Omega(\frac{1}{(\log \log n)^2})}$ [14]. For the Directed Disjoint Paths problem on planar graphs, Schrijver [57] gave an algorithm with running time $n^{\mathcal{O}(k)}$, in contrast to the NP-hardness for $k = 2$ on general directed graphs. Almost 20 years later, Cygan et al. [17] improved over the algorithm of Schrijver and showed that Directed Disjoint Paths on planar graphs is FPT by giving an algorithm with running time $2^{2^{\mathcal{O}(k^2)}}n^{\mathcal{O}(1)}$.

## 2.2 General Structure of Algorithms for (Planar) Disjoint Paths

All known algorithms for both Disjoint Paths and Planar Disjoint Paths have the same high level structure. In particular, given a graph $G$ we distinguish between the cases of $G$ having "small" or "large" treewidth. In case the treewidth is large, we distinguish between two further cases: either $G$ contains a "large" clique minor or it does not. This results in the following case distinctions.

1. **Treewidth is small.** Let the treewidth of $G$ be $w$. Then, we use the known dynamic programming algorithm with running time $2^{\mathcal{O}(w \log w)} n^{\mathcal{O}(1)}$ [56] to solve the problem. It is important to note that, assuming the Exponential Time Hypothesis (ETH), there is neither an algorithm for Disjoint Paths running in time $2^{o(w \log w)} n^{\mathcal{O}(1)}$ [44], nor an algorithm for Planar Disjoint Paths running in time $2^{o(w)} n^{\mathcal{O}(1)}$ [7].
2. **Treewidth is large and $G$ has a large clique minor.** In this case, we use the good routing property of the clique to find an irrelevant vertex and delete it without changing the answer to the problem. Since this case does not arise for graphs embedded on a surface or for planar graphs, we do not discuss it in more detail.
3. **Treewidth is large and $G$ has no large clique minor.** Using a fundamental structure theorem for minors called the flat wall theorem, we can conclude that $G$ contains a large planar piece and a vertex $v$ that is sufficiently insulated in the middle of it. Applying the unique linkage theorem [55] to this vertex, we conclude that it is irrelevant and remove it. For planar graphs, one can use the unique linkage theorem of Adler et al. [5]:

   > Any instance of Disjoint Paths consisting of a planar graph with treewidth at least $82k^{3/2}2^k$ and $k$ terminal pairs contains a vertex $v$ such that every solution to Disjoint Paths can be replaced by an equivalent one whose paths avoid $v$.

   This result says that if the treewidth of the input planar graph is (roughly) $\Omega(2^k)$, then we can find an irrelevant vertex and remove it. A natural question is whether we can guarantee an irrelevant vertex even if the treewidth is $\Omega(\mathsf{poly}(k))$. Adler and Krause [6] exhibited a planar graph $G$ with $k+1$ terminal pairs such that $G$ contains a $(2^k+1) \times (2^k+1)$ grid as a subgraph, Disjoint Paths on this input has a unique solution, and the solution uses all vertices of $G$; in particular, *no vertex of $G$ is irrelevant.* This implies that the irrelevant vertex technique can only guarantee a treewidth of $\Omega(2^k)$, even if the input graph is planar.

Combining items (1) and (3), observe that the known methodology for Disjoint Paths can only guarantee an algorithm with running time $2^{2^{\mathcal{O}(k)}} n^2$ even when restricted to planar graphs. Thus, a $2^{k^{\mathcal{O}(1)}} n^{\mathcal{O}(1)}$-time algorithm for Planar Disjoint Paths appears to require entirely new ideas. As this obstacle was known to Adler et al. [1], it is likely to be the main motivation for Adler to pose the existence of a $2^{k^{\mathcal{O}(1)}} n^{\mathcal{O}(1)}$ time algorithm for Planar Disjoint Paths as an open problem.

# 3 Recent Development: Combination of Treewidth Reduction and an Algebraic Approach

Recent joint work of the authors of this paper with Misra and Pilipczuk [45] led to the development of an FPT algorithm for Planar Disjoint Paths whose parameter dependency on $k$ is bounded by $2^{\mathcal{O}(k^2)}$. Specifically, we proved the following theorem.

**Theorem 1** **([45]).** *The* Planar Disjoint Paths *problem is solvable in time* $2^{\mathcal{O}(k^2)}n^{\mathcal{O}(1)}$.

Our algorithm is based on a novel combination of two techniques that do not seem to give the desired outcome when used on their own. The first ingredient is the treewidth reduction theorem of Adler et al. [5] that proves that given an instance of Planar Disjoint Paths, the treewidth can be brought down to $2^{\mathcal{O}(k)}$ (see item (3) in Sect. 2.2). This by itself is sufficient for an FPT algorithm (this is what Adler et al. [5] do), but as explained above, it seems hopeless that it will bring a $2^{k^{\mathcal{O}(1)}}n^{\mathcal{O}(1)}$-time algorithm.

We circumvent the obstacle by using an algorithm for a more difficult problem with a worse running time, namely, Schrijver's $n^{\mathcal{O}(k)}$-time algorithm for Disjoint Paths on directed planar graphs [57]. Schrijver's algorithm has two steps: a "guessing" step where one (essentially) guesses the homology class of the solution paths, and then a surprising homology-based algorithm that, given a homology class, finds a solution in that class (if one exists) in polynomial time. Our key insight is that for Planar Disjoint Paths, if the instance that we are considering has been reduced according to the procedure of Adler et al. [5], then we only need to iterate over $2^{\mathcal{O}(k^2)}$ homology classes in order to find the homology class of a solution, if one exists. The proof of this key insight is highly non-trivial, and builds on a cornerstone ingredient of the FPT algorithm of Cygan et al. [17] for Directed Disjoint Paths on planar graphs. To the best of our knowledge, this is the first algorithm that finds the exact solution to a problem that exploits that the treewidth of the input graph is small in a way that is different from doing dynamic programming.

In what follows, we begin with an explanation of the statement of the main technical result of Schrijver [57]. Then, we will introduce a special Steiner tree that is one of the key components in the design of our algorithm, in order to explain one of the arguments where treewidth reduction is critical.

## 3.1 Schrijver's Main Technical Result

The starting point of our algorithm is Schrijver's view [57] of a collection of "non-crossing" (but possibly not vertex- or even edge-disjoint) sets of walks as flows. To work with flows (defined immediately), we deal with directed graphs. (In this context, undirected graphs are treated as directed graphs by replacing each edge by two parallel arcs of opposite directions.) Specifically, we denote an instance of Directed Planar Disjoint Paths as a tuple $(D, S, T, g, k)$ where $D$ is a directed plane graph, $S, T \subseteq V(D)$, $k = |S|$ and $g : S \rightarrow T$ is bijective. Then, a *solution* is a set $\mathcal{P}$ of pairwise vertex-disjoint directed paths in $D$ containing, for each vertex $s \in S$, a path directed from $s$ to $g(s)$. In the language of flows, each arc of $D$ is assigned a word with letters in $T \cup T^{-1}$ (that is, we treat the set of vertices $T$ also as an alphabet), where $T^{-1} = \{t^{-1} : t \in T\}$. A word is *reduced* if, for all $t \in T$, the letters $t$ and $t^{-1}$ do not appear consecutively. Then, a *flow* is an assignment of reduced words to arcs that satisfies two constraints. First, when we concatenate the words assigned to the arcs incident to a vertex $v \notin S \cup T$ in clockwise order, where words assigned to ingoing arcs are reversed and their letters negated, the result (when reduced) is the empty word (see Fig. 1). This is an algebraic interpretation of the standard flow-conservation constraint. Second, when we do the same operation with respect to a vertex $v \in S \cup T$, then when the vertex is in $S$, the result is $g(s)$ (rather than the empty word), and when it is in $T$, the result is $t$. There is a natural association of flows to solutions: for every $t \in T$, assign the letter $t$ to all arcs used by the path from $g^{-1}(t)$ to $t$.

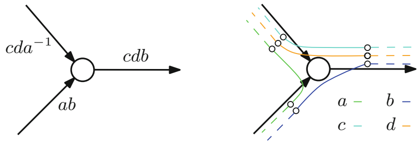

**Fig. 1.** Flow at a vertex and its reduction.

Roughly speaking, Schrijver proved that if a flow $\phi$ is given along with the instance $(D, S, T, g, k)$, then in *polynomial time* we can either find a solution or determine that there is no solution "similar to $\phi$". Specifically, two flows are *homologous* (which is the notion of similarity) if one can be obtained from the other by a *set* of "face operations" defined as follows.

**Definition 2.** *Let $D$ be a directed plane graph with outer face $f$, and denote the set of faces of $D$ by $\mathcal{F}$. Two flows $\phi$ and $\psi$ are* homologous *if there exists a function $h : \mathcal{F} \rightarrow (T \cup T^{-1})^*$ such that* (i) *$h(f) = 1$, and* (ii) *for every arc $e \in A(D)$, $h(f_1)^{-1} \cdot \phi(e) \cdot h(f_2) = \psi(e)$ where $f_1$ and $f_2$ are the faces at the left-hand side and the right-hand side of $e$, respectively.*

Then, a slight modification of Schrijver's theorem [57] yields the following corollary.

**Corollary 3.** *There is a polynomial-time algorithm that, given an instance $(D, S, T, g, k)$ of* Directed Planar Disjoint Paths*, a flow $\phi$, and a subset $X \subseteq A(D)$, either finds a solution of $(D - X, S, T, g, k)$ or decides that there is no solution of it such that the "flow associated with it" and $\phi$ are homologous in $D$.*

**Discrete Homotopy and Our Objective.** The language of flows brings several technicalities such as having different sets of non-crossing walks corresponding to the same flow (see Fig. 2). Instead, we may define a notion of *discrete*

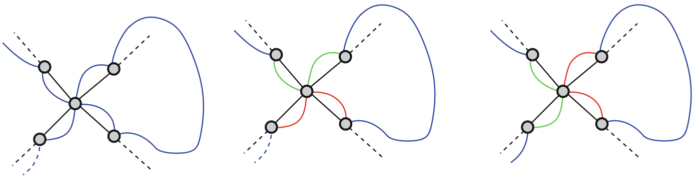

**Fig. 2.** Two different ways of extracting a walk from a flow.

*homotopy*, which is an equivalence relation that consists of three face operations. Then, we deal only with collections of non-crossing edge-disjoint walks, called *weak linkages*. Roughly speaking, two weak linkages are *discretely homotopic* if one can be obtained from the other by using "face operations" that push/stretch its walks across faces and keep them non-crossing and edge-disjoint (see Fig. 3). We note that the order in which face operations are applied is important in discrete homotopy (unlike homology)—e.g., we cannot stretch a walk across a face if no walk passes its boundary, but we can execute operations that will move a walk to that face, and then stretch it. We can translate Corollary 3 to discrete homotopy (and undirected graphs) to derive the following result.

**Lemma 4.** *There is a polynomial-time algorithm that, given an instance* $(G, S, T, g, k)$ *of* Planar Disjoint Paths*, a weak linkage* $\mathcal{W}$ *in* $G$ *and a subset* $X \subseteq E(G)$*, either finds a solution of* $(G - X, S, T, g, k)$ *or decides that no solution of it is discretely homotopic to* $\mathcal{W}$ *in* $G$*.*

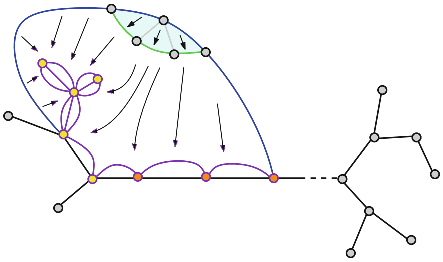

**Fig. 3.** Moving a weak linkage (having one walk) with "face operations".

In light of this result, our objective is reduced to the following task.

Compute a collection of weak linkages such that if there exists a solution, then there also exists a solution (*possibly a different one!*) that is discretely homotopic to one of the weak linkages in our collection. To prove Theorem 1, the size of the collection should be upper bounded by $2^{\mathcal{O}(k^2)}$.

This task turns out to be very challenging, and our current manuscript spans roughly 80 pages to achieve it. In the next section, we will describe one of the main ingredients, which also allows us to hint at the necessity of treewidth reduction at preprocessing.

### 3.2 Key Player: Steiner Tree

A key to the proof of our theorem is a very careful construction (done in three steps) of a so-called *Backbone Steiner tree* $R$. We use the term Steiner tree to refer to any tree in the *radial completion* of $G$ (the graph obtained by placing a vertex on each face and making it adjacent to all vertices incident to the face) whose set of leaves is precisely $S \cup T$. Having the aforementioned Backbone Steiner tree $R$ at hand, we have a more focused goal: we will zoom into weak linkages that are "pushed onto $R$", and we will only generate such weak linkages to construct our collection. Informally, a weak linkage is *pushed onto* $R$ if all of the edges used by all of its walks are *parallel to* edges of $R$. We do not demand that the edges belong to $R$ itself, because then the aforementioned goal cannot be achieved—some edges will have to be used several times, which prevents satisfying the edge disjointness requirement. Instead, we make $\Theta(n)$ parallel copies of each edge in the radial completion (the precise number arises from considerations in the "pushing process"), and then impose the weaker demand of being parallel. Now, our goal is to show the following statement: If there exists a solution, then there also exists one that can be pushed onto $R$ by applying face operations (in discrete homotopy) so that it becomes *identical* to one of the weak linkages in our collection (see Fig. 3).

At this point, one remark is in place. Our Steiner tree $R$ is a subtree of the radial completion of $G$ rather than $G$ itself. Thus, if there exists a solution discretely homotopic to one of the weak linkages that we generate, it might not be a solution in $G$. We easily circumvent this issue by letting the set $X$ in Lemma 4 contain all "fake" edges.

**Example of the Necessity of Treewidth Reduction.** To be able to generate a collection of only $2^{k^{\mathcal{O}(1)}}$ weak linkages so that the aforementioned statement can be proven, we carefully construct our Backbone Steiner tree $R$ in three steps so that it will satisfy several critical properties. (The third step is the most technical one, and will not be presented here). The first step is merely an initialization step, where we consider an arbitrary Steiner tree as $R$. Then, in the second step we modify $R$ as follows. For each "long" maximal degree-2 path $P$ of $R$ with endpoints $u$ and $v$, we will compute two minimum-size vertex sets, $S_u$ and $S_v$, such that $S_u$ separates (i.e., intersects all paths between) the following two subgraphs in the radial completion of $G$: *(i)* the subtree of $R$ that contains $u$ after the removal of a vertex $u_1$ of $P$ that is "very close" to $u$, and *(ii)* the subtree of $R$ that contains $v$ after the removal of a vertex $u_2$ that is "close" to $u$. The condition satisfied by $S_v$ is symmetric (see Fig. 4). Here, "very close" refers to distance $2^{c_1 k}$ and "close" refers to distance $2^{c_2 k}$ for some constants $c_1 < c_2$. (The selection of $u'$ not to be $u$ itself is of use in the third modification of $R$.)

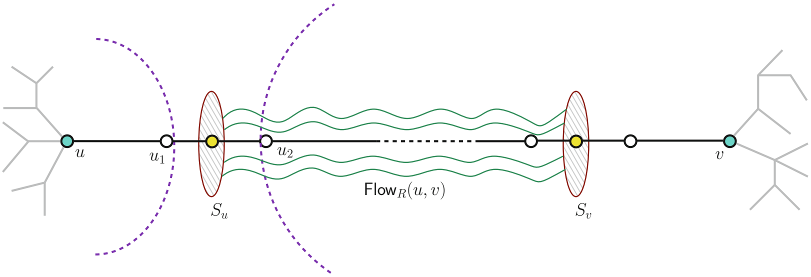

**Fig. 4.** Separators and flows for a long maximal degree-2 path $P$ in $R$.

To utilize these separators, we need their sizes to be upper bounded by $2^{\mathcal{O}(k)}$. For our initial Steiner tree $R$, such small separators may not exist. However, the modification we will present now can be shown to guarantee their existence. Specifically, we will ensure that $R$ does not have any *detour*, which roughly means that each of its maximal degree-2 paths is a shortest path connecting the two subtrees obtained once it is removed. More formally, we define a detour as follows (see Fig. 5).

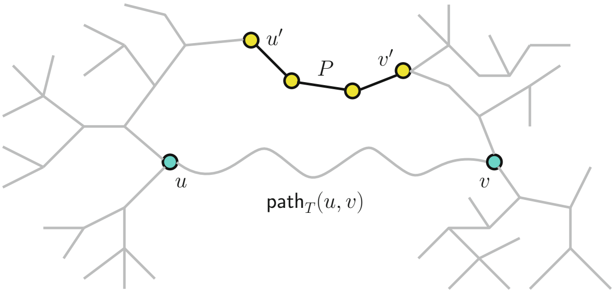

**Fig. 5.** A detour in a Steiner tree $T$.

**Definition 5.** *A* detour *in $R$ is a pair of vertices $u, v \in V_{\geq 3}(R) \cup V_{=1}(R)$ that are endpoints of a maximal degree-2 path $L$ of $R$, and a path $P$ in the radial completion of $G$, such that* (i) *$P$ is shorter than $L$,* (ii) *one endpoint of $P$ belongs to the component of $R - V(L) \setminus \{u, v\}$ containing $u$, and* (iii) *one endpoint of $P$ belongs to the component of $R - V(L) \setminus \{u, v\}$ containing $v$.*

By repeatedly "short-cutting" $R$, a process that terminates in a linear number of steps, we obtain a new Steiner tree $R$ with no detour. Now, if the separator $S_u$ is large, then there is a large number of vertex-disjoint paths that connect the two subtrees separated by $S_u$, and all of these paths are "long", namely, of length at least $2^{c_2 k} - 2^{c_1 k}$. Based on a result by Bodlaender et al. [8] (whose application requires to work in the radial completion of $G$ rather than $G$ itself), we can show that the existence of these paths implies that the treewidth of $G$ is large. Thus, if the treewidth of $G$ were small, all of our separators would have also been small. Fortunately, to guarantee this, we just need to invoke the known treewidth reduction for Planar Disjoint Paths in a preprocessing step:

**Proposition 6** **([5]).** *There is a $2^{\mathcal{O}(k)}n^2$-time algorithm that, given an instance $(G, ST, g, k)$ of* Planar Disjoint Paths*, outputs an equivalent instance $(G'$, $S, T, g, k)$ of* Planar Disjoint Paths *where $G'$ is a subgraph of $G$ whose treewidth is upper bounded by $2^{ck}$ for some constant $c$.*

Intuitively, having separators of size $2^{\mathcal{O}(k)}$ is useful with respect to the proof of sufficiency of generating a collection of only $2^{k^{\mathcal{O}(1)}}$ weak linkages as follows. When we consider a solution $\mathcal{P}$ (if one exists), then from the fact that the paths in $\mathcal{P}$ are vertex disjoint, we immediately have the property that the paths in $\mathcal{P}$ can "go across" different maximal degree-2 paths of $R$ only $2^{\mathcal{O}(k)}$ many times (because when going across different maximal degree-2 paths, either the path needs to intersect a $2^{\mathcal{O}(k)}$-sized separator or a "short" maximal degree-2 path). Having any solution $\mathcal{P}$ satisfy this property helps in proving that when we push any solution $\mathcal{P}$ onto $R$, at least those parts of the paths in $\mathcal{P}$ that go across different maximal degree-2 paths are only $2^{\mathcal{O}(k)}$ in number, and therefore they can be accommodated using only $2^{\mathcal{O}(k)}$ parallel edges for each edge of $R$.

## References

1. Adler, I.: List of open problems. In: 6th Workshop on Graph Classes, Optimization, and Width Parameters (2013). http://www.cs.upc.edu/~sedthilk/grow/Open_Problems_GROW_2013.pdf
2. Adler, I., Dorn, F., Fomin, F.V., Sau, I., Thilikos, D.M.: Faster parameterized algorithms for minor containment. Theor. Comput. Sci. **412**(50), 7018–7028 (2011). https://doi.org/10.1016/j.tcs.2011.09.015
3. Adler, I., Dorn, F., Fomin, F.V., Sau, I., Thilikos, D.M.: Fast minor testing in planar graphs. Algorithmica **64**(1), 69–84 (2012). https://doi.org/10.1007/s00453-011-9563-9
4. Adler, I., Kolliopoulos, S.G., Krause, P.K., Lokshtanov, D., Saurabh, S., Thilikos, D.: Tight bounds for linkages in planar graphs. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011. LNCS, vol. 6755, pp. 110–121. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22006-7_10
5. Adler, I., Kolliopoulos, S.G., Krause, P.K., Lokshtanov, D., Saurabh, S., Thilikos, D.M.: Irrelevant vertices for the planar disjoint paths problem. J. Comb. Theory Ser. B **122**, 815–843 (2017). https://doi.org/10.1016/j.jctb.2016.10.001
6. Adler, I., Krause, P.K.: A lower bound for the tree-width of planar graphs with vital linkages. CoRR abs/1011.2136 (2010). http://arxiv.org/abs/1011.2136
7. Baste, J., Sau, I.: The role of planarity in connectivity problems parameterized by treewidth. Theor. Comput. Sci. **570**, 1–14 (2015). https://doi.org/10.1016/j.tcs.2014.12.010
8. Bodlaender, H.L., Fomin, F.V., Lokshtanov, D., Penninkx, E., Saurabh, S., Thilikos, D.M.: (Meta) kernelization. J. ACM **63**(5), 44:1–44:69 (2016). https://doi.org/10.1145/2973749

9. Chekuri, C., Chuzhoy, J.: Polynomial bounds for the grid-minor theorem. In: Symposium on Theory of Computing, STOC 2014, New York, NY, USA, 31 May–03 June 2014, pp. 60–69 (2014)
10. Chuzhoy, J.: Improved bounds for the flat wall theorem. In: Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, pp. 256–275 (2015)
11. Chuzhoy, J., Kim, D.H.K.: On approximating node-disjoint paths in grids. In: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, Princeton, NJ, USA, 24–26 August 2015. LIPIcs, vol. 40, pp. 187–211. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2015)
12. Chuzhoy, J., Kim, D.H.K., Li, S.: Improved approximation for node-disjoint paths in planar graphs. In: Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, 18–21 June 2016, pp. 556–569. ACM (2016)
13. Chuzhoy, J., Kim, D.H.K., Nimavat, R.: New hardness results for routing on disjoint paths. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, 19–23 June 2017, pp. 86–99. ACM (2017)
14. Chuzhoy, J., Kim, D.H.K., Nimavat, R.: Almost polynomial hardness of node-disjoint paths in grids. In: Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, 25–29 June 2018, pp. 1220–1233. ACM (2018)
15. Chuzhoy, J., Kim, D.H.K., Nimavat, R.: Improved approximation for node-disjoint paths in grids with sources on the boundary. In: 45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, Prague, Czech Republic, 9–13 July 2018. LIPIcs, vol. 107, pp. 38:1–38:14 (2018)
16. Cygan, M., et al.: Parameterized Algorithms. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21275-3
17. Cygan, M., Marx, D., Pilipczuk, M., Pilipczuk, M.: The planar directed k-vertex-disjoint paths problem is fixed-parameter tractable. In: 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, Berkeley, CA, USA, 26–29 October 2013, pp. 197–206 (2013). https://doi.org/10.1109/FOCS.2013.29
18. Demaine, E.D., Fomin, F.V., Hajiaghayi, M.T., Thilikos, D.M.: Fixed-parameter algorithms for $(k, r)$-center in planar graphs and map graphs. ACM Trans. Algorithms **1**(1), 33–47 (2005)
19. Demaine, E.D., Fomin, F.V., Hajiaghayi, M., Thilikos, D.M.: Subexponential parameterized algorithms on bounded-genus graphs and $H$-minor-free graphs. J. ACM **52**(6), 866–893 (2005)
20. Demaine, E.D., Hajiaghayi, M.T.: Bidimensionality: new connections between FPT algorithms and PTASs. In: Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Canada, pp. 590–601 (2005)
21. Downey, R.: The birth and early years of parameterized complexity. In: Bodlaender, H.L., Downey, R., Fomin, F.V., Marx, D. (eds.) The Multivariate Algorithmic Revolution and Beyond. LNCS, vol. 7370, pp. 17–38. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30891-8_2
22. Downey, R.G., Fellows, M.R.: Fixed-parameter intractability. In: Proceedings of the Seventh Annual Structure in Complexity Theory Conference, Boston, Massachusetts, USA, 22–25 June 1992, pp. 36–49 (1992). https://doi.org/10.1109/SCT.1992.215379

23. Dvorak, Z., Král, D., Thomas, R.: Coloring triangle-free graphs on surfaces. In: Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, NY, USA, 4–6 January 2009, pp. 120–129 (2009). http://dl.acm.org/citation.cfm?id=1496770.1496784
24. Fellows, M.R.: The Robertson-Seymour theorems: a survey of applications. In: Graphs and Algorithms: Proceedings of the AMS-IMS-SIAM Joint Summer Research Conference Held 28 June–4 July 1987 with Support from the National Science Foundation, pp. 1–18 (1989)
25. Fellows, M.R., Fomin, F.V., Lokshtanov, D., Rosamond, F.A., Saurabh, S., Villanger, Y.: Local search: is brute-force avoidable? J. Comput. Syst. Sci. **78**(3), 707–719 (2012)
26. Fomin, F.V., Lokshtanov, D., Raman, V., Saurabh, S.: Bidimensionality and EPTAS. In: Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, USA, pp. 748–759 (2011)
27. Fomin, F.V., Lokshtanov, D., Saurabh, S.: Bidimensionality and geometric graphs. In: Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, 17–19 January 2012, pp. 1563–1575 (2012)
28. Fomin, F.V., Lokshtanov, D., Saurabh, S., Thilikos, D.M.: Bidimensionality and kernels. In: Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, USA, pp. 503–510 (2010)
29. Frank, A.: Packing paths, cuts, and circuits-a survey. Paths, Flows and VLSI-Layout, pp. 49–100 (1990)
30. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, New York (1979)
31. Geelen, J., Huynh, T., Richter, R.B.: Explicit bounds for graph minors. J. Comb. Theory Ser. B **132**, 80–106 (2018). https://doi.org/10.1016/j.jctb.2018.03.004
32. Golovach, P.A., van't Hof, P.: Obtaining planarity bycontracting few edges. Theor. Comput. Sci. **476**, 38–46 (2013). https://doi.org/10.1016/j.tcs.2012.12.041
33. Grohe, M., Kawarabayashi, K., Marx, D., Wollan, P.: Finding topological subgraphs is fixed-parameter tractable. In: Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6–8 June 2011, pp. 479–488 (2011). https://doi.org/10.1145/1993636.1993700
34. Grohe, M., Kawarabayashi, K., Reed, B.A.: A simple algorithm for the graph minor decomposition - logic meets structural graph theory. In: Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, pp. 414–431 (2013)
35. Johnson, D.S.: The NP-completeness column: an ongoing guide. J. Algorithms **8**(2), 285–303 (1987). https://doi.org/10.1016/0196-6774(87)90043-5
36. Karp, R.M.: On the computational complexity of combinatorial problems. Networks **5**(1), 45–68 (1975). https://doi.org/10.1002/net.1975.5.1.45
37. Kawarabayashi, K., Kobayashi, Y., Reed, B.A.: The disjoint paths problem in quadratic time. J. Comb. Theory Ser. B **102**(2), 424–435 (2012). https://doi.org/10.1016/j.jctb.2011.07.004
38. Kawarabayashi, K., Wollan, P.: A shorter proof of the graph minor algorithm: the unique linkage theorem. In: Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5–8 June 2010, pp. 687–694 (2010)
39. Kawarabayashi, K., Wollan, P.: A simpler algorithm and shorter proof for the graph minor decomposition. In: Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6–8 June 2011, pp. 451–458 (2011)

40. Kobayashi, Y., Kawarabayashi, K.: Algorithms for finding an induced cycle in planar graphs and bounded genus graphs. In: Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, NY, USA, 4–6 January 2009, pp. 1146–1155 (2009). http://dl.acm.org/citation.cfm?id=1496770.1496894
41. Kramer, M.R., van Leeuwen, J.: The complexity of wirerouting and finding minimum area layouts for arbitrary VLSI circuits. Adv. Comput. Res. **2**, 129–146 (1984)
42. Kuratowski, K.: Sur le problème des courbes gauches en topologie. Fund. Math. **15**, 271–283 (1930). (in French)
43. Lipton, R.J., Regan, K.W.: People, Problems, and Proofs: Essays from Gödel's Lost Letter: 2010. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41422-0
44. Lokshtanov, D., Marx, D., Saurabh, S.: Slightly superexponential parameterized problems. SIAM J. Comput. **47**(3), 675–702 (2018). https://doi.org/10.1137/16M1104834
45. Lokshtanov, D., Misra, P., Pilipczuk, M., Saurabh, S., Zehavi, M.: An exponential time parameterized algorithm for planar disjoint paths. Manuscript in Preparation (2019)
46. Lokshtanov, D., Saurabh, S., Wahlström, M.: Subexponential parameterized odd cycle transversal on planar graphs. In: IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, India, pp. 424–434 (2012)
47. Lynch, J.F.: The equivalence of theorem proving and the interconnection problem. ACM SIGDA Newslett. **5**(3), 31–36 (1975)
48. Marx, D.: Chordal deletion is fixed-parameter tractable. Algorithmica **57**(4), 747–768 (2010). https://doi.org/10.1007/s00453-008-9233-8
49. Mazoit, F.: A single exponential bound for the redundant vertex theorem on surfaces. arXiv preprint arXiv:1309.7820 (2013)
50. Ogier, R.G., Rutenburg, V., Shacham, N.: Distributed algorithms for computing shortest pairs of disjoint paths. IEEE Trans. Inf. Theory **39**(2), 443–455 (1993). https://doi.org/10.1109/18.212275
51. Pilipczuk, M., Pilipczuk, M., Sankowski, P., van Leeuwen, E.J.: Network sparsification for Steiner problems on planar and bounded-genus graphs. In: 55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, USA, pp. 276–285 (2014)
52. Reed, B.A.: Rooted routing in the plane. Discrete Appl. Math. **57**(2–3), 213–227 (1995). https://doi.org/10.1016/0166-218X(94)00104-L
53. Reed, B.A., Robertson, N., Schrijver, A., Seymour, P.D.: Finding disjoint trees in planar graphs in linear time. In: Graph Structure Theory, Proceedings of a AMS-IMS-SIAM Joint Summer Research Conference on Graph Minors held June 22 to July 5, 1991, at the University of Washington, Seattle, USA, pp. 295–301 (1991)
54. Robertson, N., Seymour, P.D.: Graph minors. XIII. The disjoint paths problem. J. Comb. Theory Ser. B **63**(1), 65–110 (1995). https://doi.org/10.1006/jctb.1995.1006
55. Robertson, N., Seymour, P.D.: Graph minors. XXII. Irrelevant vertices in linkage problems. J. Comb. Theory Ser. B **102**(2), 530–563 (2012). https://doi.org/10.1016/j.jctb.2007.12.007
56. Scheffler, P.: A practical linear time algorithm for disjoint paths in graphs with bounded tree-width. TU, Fachbereich 3 (1994)

57. Schrijver, A.: Finding k disjoint paths in a directed planar graph. SIAM J. Comput. **23**(4), 780–788 (1994). https://doi.org/10.1137/S0097539792224061
58. Schrijver, A.: Combinatorial Optimization: Polyhedra and Efficiency, vol. 24. Springer, Heidelberg (2003)
59. Srinivas, A., Modiano, E.: Finding minimum energy disjoint paths in wireless ad-hoc networks. Wirel. Netw. **11**(4), 401–417 (2005). https://doi.org/10.1007/s11276-005-1765-0
60. Wagner, K.: Über eine eigenschaft der ebenen komplexe. Math. Ann **114**(1), 570–590 (1937)
61. Wollan, P.: Personal communication, January 2015