The planar directed k-Vertex-Disjoint Paths problem is fixed-parameter tractable

Marek Cygan*, Dániel Marx†, Marcin Pilipczuk* and Michał Pilipczuk‡
*Institute of Informatics, University of Warsaw, Poland. Email: {cygan,malcin}@mimuw.edu.pl
† Institute for Computer Science and Control, Hungarian Academy of Sciences
(MTA SZTAKI), Hungary. Email: dmarx@cs.bme.hu
‡Department of Informatics, University of Bergen, Norway. Email: michal.pilipczuk@ii.uib.no

***Abstract*—Given a graph G and k pairs of vertices (s_1, t_1), …, (s_k, t_k), the k-Vertex-Disjoint Paths problem asks for pairwise vertex-disjoint paths P_1, …, P_k such that P_i goes from s_i to t_i. Schrijver [SICOMP'94] proved that the k-Vertex-Disjoint Paths problem on planar directed graphs can be solved in time $n^{O(k)}$. We give an algorithm with running time $2^{2^{O(k^2)}} \cdot n^{O(1)}$ for the problem, that is, we show the fixed-parameter tractability of the problem.**

***Keywords*-disjoint paths; fixed parameter tractability; planar graphs; directed graphs;**

I. Introduction

A classical problem of combinatorial optimization is finding disjoint paths with specified endpoints:

> k-Vertex-Disjoint Paths Problem (k-DPP)
> **Input:** A graph G and k pairs of vertices (s_1, t_1), …, (s_k, t_k).
> **Question:** Do there exist k pairwise vertex-disjoint paths P_1, …, P_k such that P_i goes from s_i to t_i?

We consider only the vertex-disjoint version of the problem in this paper; disjoint means vertex disjoint if we do not specify otherwise. If the number k of paths is part of the input, then the problem is NP-hard even on undirected planar graphs [2]. However, for every fixed k, Robertson and Seymour showed that there is a cubic-time algorithm for the problem in general undirected graphs [3]. Their proof uses the structure theory of graphs excluding a fixed minor and is therefore extremely complicated. More recently, a significantly simpler, but still very complex algorithm was announced by Kawarabayashi and Wollan [4]. Obtaining polynomial running time for fixed k is significantly simpler in the special case of planar graphs [5]; see also the self-contained presentations of Reed et al. [6] or Adler et al. [7].

The problem becomes dramatically harder for directed graphs: it is NP-hard even for $k = 2$ in general directed graphs [8]. Therefore, we cannot expect an analogue of the undirected result of Robertson and Seymour [3] saying that the problem is polynomial-time solvable for fixed k. For directed planar graphs, however, Schrijver gave an algorithm with polynomial running time for fixed k:

Theorem I.1 (Schrijver [9])**.** *The k-Vertex-Disjoint Paths Problem on directed planar graphs can be solved in time $n^{O(k)}$.*

The algorithm of Schrijver is based on enumerating all possible homology types of the solution and checking in polynomial time whether there is a solution for a fixed type. Therefore, the running time is mainly dominated by the number $n^{O(k)}$ of homology types. Our main result is improving the running time by removing k from the exponent of n:

Theorem I.2. *The k-Vertex-Disjoint Paths Problem on directed planar graphs can be solved in time $2^{2^{O(k^2)}} \cdot n^{O(1)}$.*

In other words, we show that the k-Disjoint Paths Problem is fixed-parameter tractable on directed planar graphs. The fixed-parameter tractability of this problem was asked as an open question by Bodlaender, Fellows, and Hallett [10] already in 1994, in one of the earliest papers on parameterized complexity. The question was reiterated in the open problem list of the classical monograph of Downey and Fellows [11] in 1999. Note that, for undirected planar graphs, the algorithm with best dependence on k is due to Adler et al. [7] and has running time $2^{2^{O(k)}} \cdot n^{O(1)}$. Therefore, for the more general directed version of the problem, we cannot expect at this point a running time with better than double-exponential dependence on k.

For general undirected graphs, the algorithm of Robertson and Seymour [3] relies heavily on the structure theory of graphs excluding a fixed minor; in fact, this algorithm is one of the core achievements of the Graph Minors series. More recent results on finding subdivisions [12] or parity-constrained disjoint paths [13] also build on this framework. Even in the much simpler planar case, the algorithm presented by Adler et al. [7] uses the concepts and tools developed in the study of excluded minors. In a nutshell, their algorithm has three main components. First, if treewidth (a measure that plays a crucial role in graph structure theory) is bounded, then standard algorithmic techniques can be used to solve the k-Vertex-Disjoint Paths Problem. Second, if

The full version of this work can be found in [1]. The first and third author were supported in part by the NCN grant N206567140. The second author was supported by the ERC Starting Grant PARAMTIGHT, reference 280152. The fourth author was partially supported by the ERC grant "Rigorous Theory of Preprocessing", reference 267959.

treewidth is large, then (planar version of) the Excluded Grid Theorem [14]–[17] implies that the graph contains a subdivision of a large wall, which further implies that there is a vertex enclosed by a large number of disjoint concentric cycles, none of them enclosing any terminals. Finally, Adler [7] et al. show that such a vertex is irrelevant, in the sense that it can be removed without changing the answer to the problem. Thus by iteratively removing such irrelevant vertices, one eventually arrives at a graph of bounded treewidth.

Can we apply a similar deep and powerful theory in the directed version of the problem? There is a notion of directed treewidth [18] and an excluded grid theorem holds at least for planar graphs (and more generally, for directed graphs whose underlying undirected graph excludes a fixed minor [19]). However, the other two algorithmic components are missing: it is not known how to solve the k-Vertex Disjoint Paths problem in $f(k) \cdot n^{O(1)}$ time on directed graphs having bounded directed treewidth and the directed grids excluded by these theorems do not seem to be suitable for excluding irrelevant vertices. There are other notions that try to generalize treewidth to directed graphs, but the algorithmic applications are typically quite limited [20]–[25]. In particular, the k-Vertex-Disjoint Paths Problem is known to be W[1]-hard on directed acyclic graphs [26], which is strong evidence that any directed graph measure that is small on acyclic graphs is not likely to be of help.

Our algorithm does not use any tool from the structure theory of undirected graphs, or any notion of treewidth for directed graphs. The only previous results that we use are the results of Ding, Schrijver, and Seymour [27], [28] on various special cases of the directed disjoint paths problem, the cohomology feasibility algorithm of Schrijver [9], and a self-contained combinatorial argument from Adler et al. [7]. Therefore, we have to develop our own tools and in particular a new type of decomposition suitable for the problem. A concept that appears over and over again in this paper is the notion of alternation: we are dealing with sequences of paths and cycles having alternating orientation (i.e., each one has an orientation that is the opposite of the next one), we measure the "width" of a sequence of arcs by the number of alternations in the sequence, and we measure "distance" between faces by the minimum alternation on any sequence of arcs between them. Section II gives a high-level overview of the algorithm, while all formal arguments are postponed to the full version [1]. Let us highlight here the most important steps and the main ideas.

Irrelevant vertices: Analogously to Adler et al. [7], we prove that a vertex enclosed by a large set of concentric cycles having *alternating orientation* and not enclosing any terminals is irrelevant. As expected, the proof is more complicated and technical than in the undirected case.

Duality of alternation: We show that alternation has properties that are similar to the classical properties of undirected planar graphs. We prove an approximate duality between alternating paths and the minimum alternation size of a cut (reminiscent of max-flow min-cut duality), and between concentric cycles and alternation distance (reminiscent of the fact that two faces far away in a planar graph are separated by many disjoint cycles).

Decomposition: We present a novel kind of decomposition into "disc" and "ring" components. The crucial property of the decomposition is that the set of arcs leaving a component has bounded alternation. That is, the components are connected by a bounded number of *bundles,* each containing a set of "parallel" arcs with the same orientation.

Handling ring components: Ring components pose a particular challenge: we have to understand how many turns a path of the solution does when connecting the inside and the outside. We prove a rerouting argument showing that only a bounded number of possibilities has to be taken into account for the winding numbers of these paths.

Guessing bundle words: Given a decomposition, a path of the solution can be described by a word consisting of a sequence of symbols representing the bundles visited by the path, in the order they appear in the path. Note that a bundle can be used several times by a path of the solution, thus the word can be very long. Our goal is to enumerate a bounded number of possible bundle words for each path of the solution. These words, together with our understanding of what is going on inside the rings, allow us to guess the homology type of the solution, and then invoke Schrijver's cohomology feasibility algorithm to check if there is a solution with this homology type.

The techniques introduced in this paper were developed specifically with the k-Vertex-Disjoint Paths Problem in mind. It is likely that some of the duality arguments or decomposition techniques can have applications for other problems involving planar directed graphs.

In general directed graphs, vertex-disjoint and edge-disjoint versions of the disjoint paths problems are equivalent: one can reduce the problems to each other by simple local transformations (e.g., splitting a vertex into an in-vertex and an out-vertex). However, such local transformations do not preserve planarity. Therefore, our result has no implications for the edge-disjoint version of the problem on planar directed graphs. Let us note that in planar graphs the edge-disjoint version seems very different from the vertex-disjoint version: as the paths can cross at vertices, the solution does not have a topological structure of the type that is exploited by both Schrijver's algorithm [9] and our algorithm. The complexity of the planar edge-disjoint version for fixed k remains an open problem; it is possible that, similarly to general graphs [8], it is NP-hard even for $k = 2$.

One can define a variant of the planar edge-disjoint problem where crossings are not allowed. That is, in the noncrossing edge-disjoint version paths are allowed to share vertices, but if edge e_1 entering v is followed by e_2, and edge f_1 entering v is followed by f_2, then the cyclic order of these edges cannot be (e_1, f_1, e_2, f_2) or (e_1, f_2, e_2, f_1) around v. It is easy to see that this version can be reduced (in a planarity-preserving way) to the vertex disjoint version by replacing each vertex by a large bidirected grid. Therefore, our algorithm can solve the noncrossing edge-disjoint version of the k-Disjoint Paths Problem as well.

II. Overview of the algorithm

The goal of this section is to give an informal overview of our main result — the fixed-parameter algorithm for finding k disjoint paths in directed planar graphs.

A. *Irrelevant vertex rule*

Let us first recall how to solve the k-disjoint paths problem in the undirected (even non-planar) case. The algorithm of Robertson and Seymour [3] considers two cases. If the treewidth of the input graph G is bounded by a function of the parameter (k, the number of terminal pairs), then the problem can be solved by a standard dynamic programming techniques on a tree decomposition of small width of G. Otherwise, by the Excluded Grid Theorem [14]–[17], G contains a large grid as a minor.

The idea now is to distinguish a vertex v of G, whose deletion does not change the answer of the problem; that is, there exist the required k disjoint paths in G if and only if they exist in $G \setminus v$. Note that the disjoint paths problem can become only harder if we delete a vertex; thus, to pronounce v irrelevant, one needs to prove that any solution using the vertex v can be redirected to a similar one, omitting v.

In the case of planar graphs one may apply the following quite intuitive reasoning. Assume that G contains a large grid as a minor; as there are at most $2k$ terminals, a large part of this grid does not enclose any terminal. In such a part, a vertex v hidden deep inside the grid seems irrelevant: any solution using v needs to traverse a large part of the grid to actually contain v, and it should be possible to "shift" the paths a little bit to omit v. This reasoning can be made formal, and Adler et al. [7] proved that, in undirected planar graphs, the middle vertex of a grid of exponential (in k) size is irrelevant. In fact, they show a bit stronger statement: if we have sufficiently many (around 2^k) concentric cycles on the plane, such that the outermost cycle does not enclose any terminal, then any vertex on the innermost cycle is irrelevant.

One of the main arguments in the proof of Adler et al. [7] is as follows. Assume that there are many pairwise disjoint segments of the solution that cross sufficiently many orthogonal paths (henceforth called *chords*) in the graph; see Figure 1. Assume moreover that the aforementioned segments are the only parts of the solution that appear in the area enclosed by the outermost segments and chords (i.e., in the part of the plane depicted in Figure 1). Then, if the number of segments is more than 2^k, one can redirect some of them, using the chords, and shortening the solution. Thus,

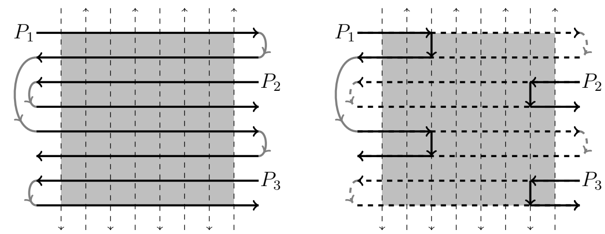

Figure 1: A situation where a shortcut can be made and how it can be made. There are more than 2^k horizontal segments of the paths, crossed by sufficiently many vertical chords, that, in the directed setting, are required to be of alternating orientation. Moreover, it is assumed that no other part of any path intersects the gray area, so that the paths remain pairwise disjoint after rerouting.

in a minimal (in some carefully chosen sense) solution, a set of more than 2^k paths cannot go together for a longer period of time.

The argument of Adler et al. [7] described in the previous paragraph redirects the paths of the solution using the chords in an undirected way, and hence the direction in which a chord is used is unpredictable, depending on the order in the which the segments appear on the paths of the solution. Hence, if we want to transfer this argument to the directed setting, then we need to make some assumption on the direction of the chords. It turns out that what we need is that the chords are directed paths with alternating orientation. This ensures that we always have a chord going in the right direction at any place we would possibly need it.

If a set of paths intersect the innermost cycle, then they need to traverse all cycles. Adler et al. [7] show how to find a subset of these paths and how to cut out chords from the cycles in a way that satisfies the conditions of the rerouting argument. In the directed setting, in order to obtain chords of alternating orientation, we need to assume that the cycles have alternating orientation too, that is the cycles form a *sequence of concentric cycles with alternating orientation*.

Luckily, it turns out that such a sequence of cycles is sufficient for the following irrelevant vertex rule.

Theorem II.1 (Irrelevant vertex rule)**.** *For any integer k, there exists $d = d(k) = 2^{O(k^2)}$ such that the following holds. Let G be an instance of k-*DPP *and let $C_1, C_2, C_3, \dots, C_d$ bet a sequence of concentric cycles in G with alternating orientation, where C_1 is the outermost cycle. Assume moreover that C_1 does not enclose any terminal. Then any vertex of C_d is irrelevant.*

At the heart of the proof of Theorem II.1 lies the rerouting argument described above, which states that a solution can be rerouted and shortened if a set of more than 2^k paths travel together through sufficiently many (exponential in k) chords cut out from the alternating cycles C_i. However, it is

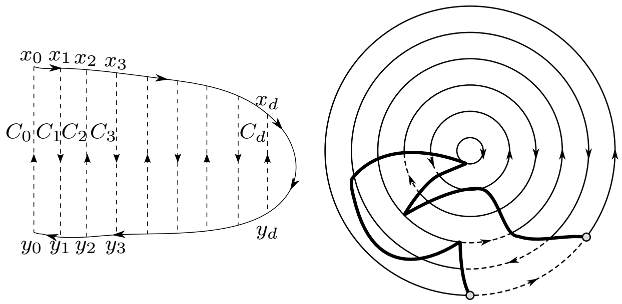

Figure 2: A d-bend B with chords $C_0, C_1, \ldots, C_d$, and how it can be cut out from concentric cycles, using parts of the cycles as chords.

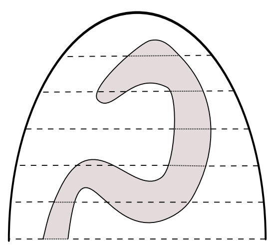

Figure 3: A part of a path creates a bend inside another bend.

much harder to prove the existence of these paths and chords needed for the rerouting argument than in the undirected case, and we now sketch how it could be done.

Consider the situation assumed in Theorem II.1 and assume we have a solution where one path, say P, intersects the innermost cycle. On one side of P we obtain a structure we call a *bend*, depicted in Figure 2. The parts of the cycles are called *chords*, a bend with d chords is a d-*bend*. Moreover, *the type of the bend* is the number of different paths from the solution that intersect the interior or the boundary of the bend; our initial bend is of type at most k. Our main technical claim in the proof of Theorem II.1 is that in a (somehow defined) minimal solution there do not exist d-bends of type t, for $d > f(k,t)$ and some function $f(k,t) = 2^{O(kt)}$, that do not enclose any terminals.

Assume we have a d-bend B of type t, for some large d, enclosed by a part of a path P_i of the solution. We analyze the *segments* of the solution: the maximal subpaths of the paths $P_1, \ldots, P_k$ in the interior of the bend. If the interval vertices of the last two chords are not intersected by any segment, then one of these two chords has the right orientation to serve as a shortcut for the path P_i, contradicting the minimality of the solution. Therefore, we can assume that all but the last two chords are intersected by segments. If any segment of $P_{i'}$ intersects the j-th chord of B, then it itself induces a j'-bend B' inside B, for some $j' = j - O(1)$ (see Figure 3). Hence, if the path P_i itself does not intersect the interior of the d-bend B, any bend inside B is of strictly smaller type, and the claim is proven by induction on t.

Otherwise, we can argue that several segments of P_i enter the interior of the d-bend B. Our goal is to prove that there is a large set of segments of P_i entering B that form a nested sequence and they travel together through a large number of chords deep inside the bend, with no other segment of P_i between them. Then we can argue that any other segment of some $P_{i'}$ with $i' \neq i$ intersecting these chords is also nested with these segments, otherwise they would create a large bend of strictly smaller type, and induction could be applied. Therefore, we get a large set of paths travelling together through a large number of chords, and the rerouting argument described above can be invoked.

We would like to note that we can test in polynomial time if the irrelevant vertex rule applies: if we guess one of the faces enclosed by C_r and the orientation of C_r, we can construct the cycles in a greedy manner, packing the next cycle as close as possible to the previously constructed one. However, we do not use this property in our algorithm: the decomposition algorithm, described in the next subsection, returns an irrelevant vertex situation if it fails to produce a suitable decomposition.

We would also like to compare the assumptions of Theorem II.1 with the conjectured canonical obstruction for small directed treewidth, depicted in Figure 4. It has been shown that a planar graph [29], or, more generally, a graph excluding a fixed undirected minor [19], has small directed treewidth unless it contains a large directed grid (as in Figure 4), in some minor-like fashion, and this statement is conjectured to be true for general graphs [18]. Although the assumption of bounded directed treewidth may be easier to use than the bounded-alternation decomposition presented in the next subsection, we do not know how to argue about irrelevancy of some vertex or arc in the directed grid. Thus, we need to stick with our irrelevant vertex rule with relatively strong assumptions (a large number of alternating cycles), and see in the rest of the proof what can be deduced if such a situation does not occur.

B. *Decomposition and duality theorems*

Once we have proven the irrelevant vertex rule (Theorem II.1), we may see what can be deduced about the structure of the graph if the irrelevant vertex rule does not apply. Recall that in the undirected case the absence of an irrelevant vertex implied a bound on the treewidth of the graph, and hence the problem can be solved by a standard dynamic programming algorithm.

In our case the situation is significantly different. As we shall see, the assumptions in Theorem II.1 are rather strong, and, if the irrelevant vertex rule is not applicable, the problem does not become as easy as in the bounded-treewidth case. Recall that Theorem II.1 assumed a large number of cycles of alternating orientation, and these alternations were

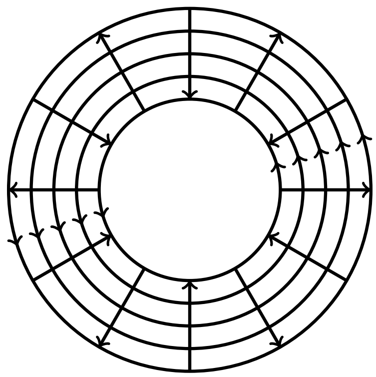

Figure 4: A directed grid — a conjectured canonical obstacle for small directed treewidth.

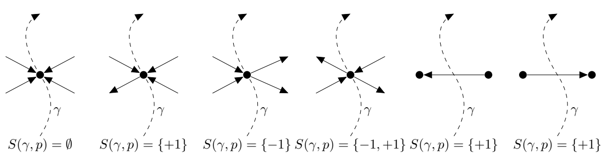

Figure 5: An illustration of the definition of $S(\gamma, p)$ for $p \in \gamma \cap G$.

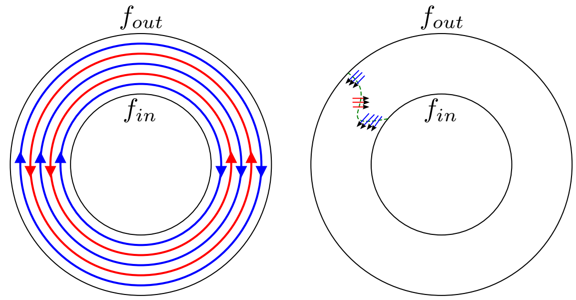

Figure 6: Two cases in Lemma II.2(1): cycles of alternating orientation between f_{in} and f_{out} or a curve of bounded alternation connecting f_{in} and f_{out}.

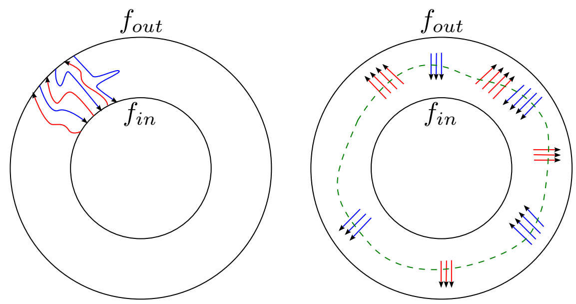

Figure 7: Two cases in Lemma II.2(2): paths of alternating orientation connecting f_{in} and f_{out} or a curve of bounded alternation separating f_{in} and f_{out}.

crucial for the rerouting argument. It turns out that, if such cycles cannot be found, we can decompose the graph into relatively simple pieces using cuts of bounded alternation.

Consider a directed curve γ on the plane that intersects the plane graph G only in a finite number of points (i.e., γ does not "slide" along any arc of G). For any point $p \in \gamma \cap G$ we define $S(\gamma, p) \subseteq \{-1, +1\}$ as follows: $-1 \in S(\gamma, p)$ if it is possible for a path in G to cross γ in p from left to right, and $+1 \in S(\gamma, p)$ if it possible to cross γ from right to left (see Figure 5). The *alternation* of γ is the length of the longest sequence of alternating $+1$ and -1s that is embeddable (in a natural way) into the sequence $S(\gamma, p)_{p \in \gamma \cap G}$.

Note that the existence of a curve γ with alternation $\mathfrak{a}$ connecting faces f_1 and f_2 proves that f_1 and f_2 cannot be separated by a sequence of more than $\mathfrak{a}$ concentric cycles of alternating orientation. Thus, a curve of bounded alternation is in some sense dual to the notion of concentric cycles of bounded alternation. It turns out that this duality is tight: such a curve of bounded alternation is the only obstacle that prevents the existence of these concentric cycles. One can also formulate a duality statement similar to the classical max-flow min-cut duality, with a set of paths of alternating orientation playing the role of the flow and a curve of bounded alternation playing the role of a cut. The following lemma states both types of duality in an informal way (see Figures 6 and 7 for illustration).

Lemma II.2 (Alternation dualities, informal statement.)**.** *Let G be a graph embedded in a subset of a plane homeomorphic to a ring, and let f_{in} and f_{out} be the two faces of G that contain the inside and the outside of the ring, respectively. Let r be an even integer. Then, in polynomial time, one can in G:*

1) *either find a sequence of r cycles of alternating orientation, separating f_{in} from f_{out}, or find a curve connecting f_{in} with f_{out} with alternation at most r (Figure 6); and*
2) *either find a sequence of r paths, connecting f_{in} and f_{out}, with alternating orientation, or find a closed curve separating f_{in} from f_{out} with alternation at most $r + 4$ (Figure 7).*

Let us now give intuition on how to prove statements like Lemma II.2. If we identify f_{in} and f_{out}, or more intuitively, extend the surface with a handle connecting f_{in} and f_{out}, we can perceive G as a graph on a torus. After some gadgeteering, we may use the following result of Ding, Schrijver, and Seymour [27]: if one wants (in a graph G on a torus) to route a set of vertex-disjoint cycles with prescribed homotopy class and directions, a canonical obstacle is a face-vertex curve γ (of some other homotopy class), where the sequence $S(\gamma, p)_{p \in \gamma \cap G}$ does not contain the expected subpattern of $+1$ and -1s. Note that such a curve is not far from the curves promised by Lemma II.2.

Equipped with this understanding of alternation, we prove

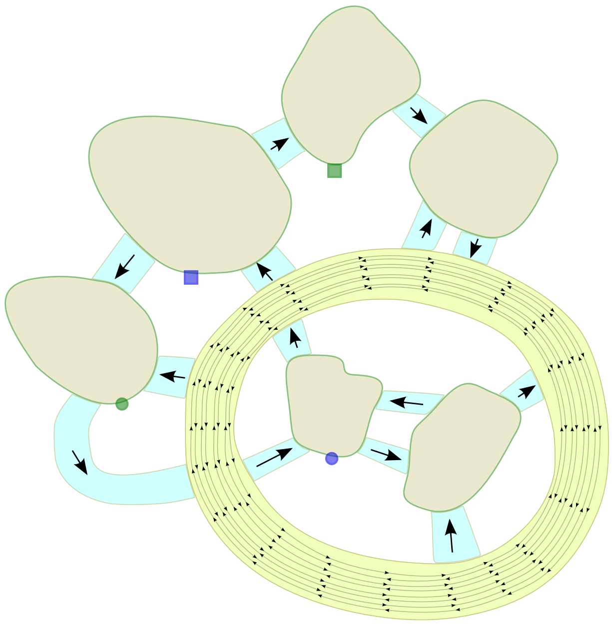

Figure 8: An example of a decomposition with six disc components and a single ring component.

a decomposition theorem that is crucial for our algorithm. We state this theorem here only informally (see Figure 8 for an illustration).

Theorem II.3 (Decomposition theorem, informal statement)**.** *Assume that G is a plane graph with k terminal pairs to which the irrelevant vertex rule is not applicable. Then one can partition the graph G into a bounded (in k) number of* disc *and* ring components*, using cuts of bounded total alternation; a disc (resp., ring) component occupies a subset of the plane that is isomorphic to a disc (resp., ring). Moreover, each terminal lives on the border of a disc component, and each ring component contains many concentric cycles of alternating orientation, separating the inside from the outside.*

The decomposition of Theorem II.3 is obtained by iteratively refining a decomposition, moving a terminal to the boundary of a component in each step. If a disc component contains a terminal such that there is a curve of bounded alternation from the terminal to the boundary of the component, then the terminal can be moved to the boundary by removing the arcs intersected by the curve. This operation increases the alternation of the cut separating the component from the rest of graph only by a bounded number, thus we can afford to perform one such step for each terminal. Otherwise, if there is no such curve, then Lemma II.2(1) implies that there is a large set of concentric cycles of alternating orientations separating all the terminals in the component from the boundary. We again consider two cases. If there is large set of paths with alternating orientations crossing these cycles, then the paths and cycles together form some kind of grid, and we can easily identify a vertex that is separated from all the terminals by a large set of concentric cycles with alternating orientation. Such a vertex is irrelevant by Theorem II.1, and hence can be removed. On the other hand, if there is no such set of paths, then Lemma II.2(2) implies that there is a curve of bounded alternation separating the terminals of the component from the boundary of the component. We can use this curve to cut away a ring component and we can do this in such a way that after removing the ring component, one of the terminals is close to the boundary of the remaining part of the disc component (in the sense that there is a curve of bounded alternation connecting it to the boundary). Therefore, we can apply the argument described above to move this terminal to the boundary. Iteratively applying these steps until all the terminals are on the boundary of its component produces the required decomposition.

How can we use the decomposition of Theorem II.3 to solve k-DPP? The disc components are promising to work with, as the k-DPP problem is polynomial if all terminals lie on the outer face of the graph [28]. In a topological sense, if the terminals are on the outer face, then the solutions are equivalent, whereas if the terminals are on the inside and outside boundaries of a ring, then the solutions can differ in how many "turns" they do along the ring. This possible difference in the number of turns create particular challenges when we are trying to apply the techniques of Schrijver [9] to find a solution based on a fixed homotopy class.

Theorem II.3 would be nicer and more powerful if we could always obtain a decomposition using only disc components, but as we explain in the full version, this does not seem to be possible in general.

C. Bundles and bundle words

From the previous subsection we know that, if the irrelevant vertex rule is not applicable, one may decompose the graph into a bounded number of disc and ring components, using cuts of bounded alternation. Let us reformulate this statement: we can decompose the graph into a bounded number of disc and ring components, connected by a bounded number of *bundles*; a *bundle* is a set of arcs of G that form a directed path in the dual of G, such that no other arc nor vertex of G is drawn between the consecutive arcs of the bundle. Thus, we obtain something we call *bundled instance* $(G, \mathcal{D}, \mathbb{B})$: a graph G with terminals, a family of components $\mathcal{D}$ and a family of bundles $\mathbb{B}$. In Figure 8 one can see a partition of arcs between components into bundles. With any path P in a bundled instance $(G, \mathcal{D}, \mathbb{B})$ we can associate its *bundle word*, denoted $\mathtt{bw}(P)$: we follow the path P from start to end and append a symbol $B \in \mathbb{B}$ whenever we traverse an arc belonging to a bundle B. That is, $\mathtt{bw}(P)$ is a word over alphabet $\mathbb{B}$; see Figure 9 for an example.

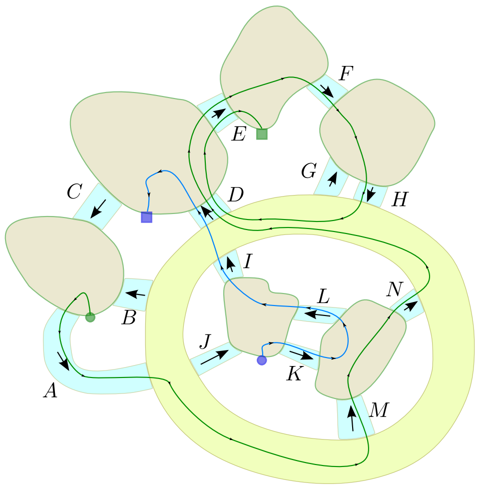

Figure 9: An example of a solution. One path has bundle word $AMNDEFHDE$ and the other path has bundle word $KLID$.

Assume for a moment that there are no ring components in the decomposition; ring components present their own challenges requiring an additional layer of technical work, but they do not alter the main line of reasoning. Assume moreover that we have computed somehow bundle words $\mathtt{bw}(P_i)$, $i = 1, 2, \ldots, k$ for some solution $(P_i)_{i=1}^k$ for k-DPP on G. The important observation is that, given the bundle words, the cohomology feasibility algorithm of Schrijver [9] is able to extract (an approximation of) the paths P_i in polynomial time.

To show this, let us recall the algorithm of Schrijver [9] that solves k-DPP in polynomial time for every fixed k. The heart of the result of Schrijver lies in the proof that k-DPP is polynomial-time solvable if we are given a homotopy class of the solution. In simpler words, given a "pre-solution", where many paths can traverse the same arc, even in wrong direction (but they cannot cross), one can in polynomial time check if the paths can be "shifted" (i.e., modified by a homotopy) so that they become a feasible solution. In such a "shift" (homotopy) one can move a path over a face, but not over a face that contains a terminal.[1] See Figure 10 for an illustration of different homotopy classes of a solution.

In our setting, we note that, in the absence of ring components, two solutions with the same set of bundle words of each paths are homotopical; thus, given bundle words of a solution, one can use the algorithm of Schrijver to verify if there is a solution consistent with the given set of bundle words. However, one should note that the homotopies are allowed to do much more than to only move paths within a bundle; formally, using the Schrijver's algorithm we can either (i) correctly conclude that there is no solution with

[1]Note that we can assume that each terminal is of degree one, and the notion of "face containing a terminal" is well-defined.

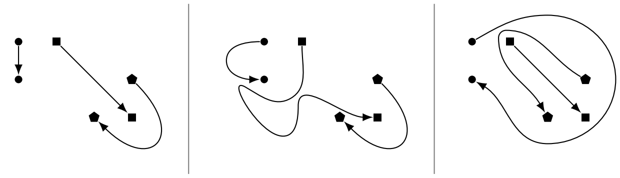

Figure 10: Different homotopy classes of a solution: in the first two figures, the solutions are of the same class, whereas on the third figure the homotopy class is different.

given set of bundle words $(p_i)_{i=1}^k$, or (ii) compute a solution $(P_i)_{i=1}^k$ such that the bundles of $\mathtt{bw}(P_i)$ is a subset (as a multiset) of the bundles of p_i.

Unfortunately, if the decomposition contains ring components as well, then the bundle words of a solution does not describe the homotopy class of the solution. What do we need to learn, apart from bundle words of the solution, to identify the homotopy class of the solution if ring components are present? The answer is not very hard to see: for any subpath of a path in a solution that crosses some ring component (i.e., goes from the inside to the outside of vice versa) we need to know how many times it "turns" inside the ring component; we call it a *winding number* inside a component.

Thus, our goal is to compute a small family of possible bundle words and winding numbers such that, if there exists a solution to k-DPP on G, there exists a solution consistent with one of the elements of the family. In fact, our main goal in the rest of the proof is to show that one can compute such family of size bounded in the parameter k.

D. Guessing bundle words

Assume again that there are no ring components; we are to guess the bundle words of one of the solutions. Recall that the number of bundles, $|\mathbb{B}|$, is bounded in k. Thus, if a bundle word of some path P_i from a solution $(P_i)_{i=1}^k$ is long, it needs to contain many repetitions of the same bundles.

Let us look at one such repetition: let uB be a subword of $\mathtt{bw}(P_i)$, where B is the first symbol of u and u contains each symbol of $\mathbb{B}$ at most once. We call such place a *spiral*. Note that this spiral separates the graph into two parts, the inside and the outside, where any other path can cross the spiral only in a narrow place inside the bundle B (see Figure 11). As the arcs of B go in one direction, any path P_j, $j \neq i$ can cross the spiral only once, in the same direction as P_i, and the path P_i cannot cross the spiral uB again. Note that we know exactly which paths cross the spiral uB: the paths that have terminals on different sides of the spiral uB.

There is also one important corollary of this observation on spirals. If $\mathtt{bw}(P_i) = xuBy$ for some words x, y and spiral uB, then, for any bundle B' that does not appear in u, only one of the words x or y may contain B': the bundle B' is either contained inside the spiral uB or outside it. By

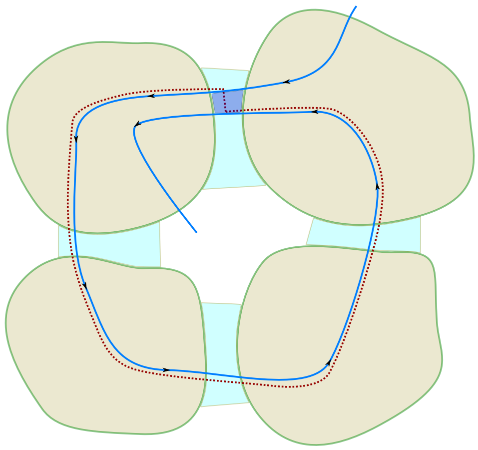

Figure 11: A spiral. Any other path may cross the dotted curve only in a narrow place in the top bundle (highlighted).

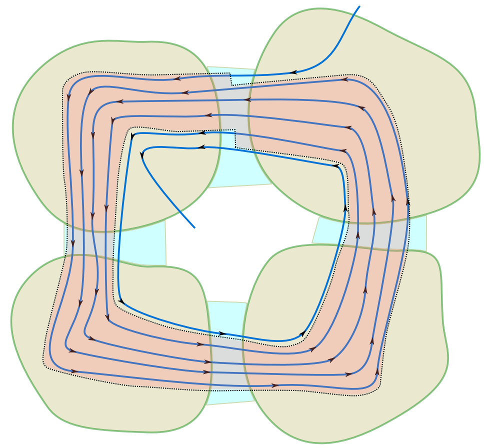

Figure 12: A spiraling ring; the dotted lines are its borders.

some quite simple word combinatorics, we infer that $\mathtt{bw}(P_i)$ can be decomposed as $u_1^{r_1} u_2^{r_2} u_3^{r_3} \dots u_s^{r_s}$, where each word u_i contains each symbol of $\mathbb{B}$ at most once, each r_i is an integer and $s \leq 2|\mathbb{B}|$. Note that if we aim to guess bundle words of some solution $(P_i)_{i=1}^k$, there is only a bounded number of choices for the length s and the words u_i; the difficult part is to guess the exponents r_i, if they turn out to be big (unbounded in k). That is, we can easily guess the global structure of the spirals (how they are nested, etc.), but we cannot easily guess the "width" of the spirals (how many turns of the same type they do). We need further analysis and insight in order to be able to guess these numbers as well.

Let us focus on a place in a graph where a path P_i in the solution $(P_i)_{i=1}^k$ contains a subword $u^r B$ of $\mathtt{bw}(P_i)$ for some large integer r, where B is the first symbol of u. The situation, depicted in Figure 12, looks like a large spiral; the *spiraling ring* is the area between the first and last spiral uB in the subword $u^r B$. Note that any path P_j that enters this area, actually needs to traverse all r spirals uB and $\mathtt{bw}(P_j)$ contains a subword $u^{r-1}B$; let $I \subseteq \{1, 2, \dots, k\}$ be the set of indices j such that P_j traverses uB. Moreover, note that, since B contains arcs going in only one direction, these parts of paths $(P_j)_{j \in I}$ are the only intersections of the solution $(P_i)_{i=1}^k$ with the spiraling ring.

Our main claim is that we can choose r to be as small as possible, just to be able to route the desired paths through the spiraling ring in G. More formally, we prove that if we can route $|I|$ directed paths through the spiraling ring, such that each path traverses B roughly r^* times (but they may start and end in different places than the parts of the solution $(P_i)_{i=1}^k$ traversing the spiraling ring), then we can modify the solution $(P_i)_{i=1}^k$ inside the spiraling ring such that $r \leq r^* + O(1)$. If we choose r^* to be minimal possible, we have $|r - r^*| = O(1)$ and we can guess r.

To prove that the paths can be rerouted, we show the following theorem.

Theorem II.4 (rerouting in a ring, informal statement)**.** *Let G be a plane graph embedded in a ring, with outer face f_{out} and inner face f_{in}. Assume that there exist two sequences of vertex-disjoint paths $(P_i)_{i=1}^s$ and $(Q_i)_{i=1}^s$ connecting f_{in} with f_{out}, such that P_i goes in the same direction (from f_{in} to f_{out} or vice versa) as Q_i, and the endpoints of $(P_i)_{i=1}^s$ lie in the same order on f_{in} as the endpoints of $(Q_i)_{i=1}^s$. Then one can reroute $(P_i)_{i=1}^s$ inside G, keeping the endpoints, such that the winding number of P_1 differs from the winding number of Q_1 by no more than 6.*

How to prove such a rerouting statement? We again make use of the results of Ding et al. [27] on a canonical obstacle for routing a set of directed cycles on a torus (as in the proof of Lemma II.2). We connect f_{in} and f_{out} with a handle, and perceive the paths P_i and Q_i as a cycles on a torus. The winding number w_P of P_1 determines the homotopy class of the cycles $(P_i)_{i=1}^s$, and the winding number w_Q of Q_1 determines the homotopy class of the cycles $(Q_i)_{i=1}^s$. Now we observe that an obstacle for routing the same set of cycles with "homotopy" w for $w_Q + O(1) < w < w_P - O(1)$ (or, symmetrically, $w_P + O(1) < w < w_Q - O(1)$) would be an obstacle for "homotopy" either w_Q or w_P, a contradiction. Hence, almost all "homotopies" between w_Q and w_P are realizable. By some gadgeteering, we may force the cycles to use the same endpoints as the paths $(P_i)_{i=1}^s$, at the cost of $O(1)$ loss in the "homotopy" class.

We would like to note that Theorem II.4 is a cornerstone of our result. The exponential time complexity of the algorithm of Schrijver [9] comes from the fact that the number of homotopy classes of a solution solution cannot be bounded by a function of k, because the number of possibilities for the number of turns of the solution in some ring-like parts of the graph cannot be bounded by a function of k. Theorem II.4 overcomes this obstacle by showing that that for each such ring, one can choose a canonical number of turns (that depends only on the ring, not how it is connected to the outside) and the solution can be assumed to spiral a similar number of turns than the canonical pass. In other words, if one would try to construct a $W[1]$-hardness proof of k-DPP by a reduction from, say, k-CLIQUE, one cannot expect to obtain a gadget that encodes a choice of a vertex or edge of the clique by a number of turns a solution path makes in some ring-like part of the graph — such an encoding seems natural, taking into account the source of the exponential-time complexity of the algorithm of Schrijver [9].

However, it still requires significant work to make use of Theorem II.4. In the case of spiraling rings, the question that remains is how to get minimal exponent r^* such that $|I|$ paths can be routed through a spiraling ring with r^* turns. The idea is to isolate a part of the graph and parts of the bundle words of the solution where only one exponent is unknown, and then apply Schrijver's algorithm for different choices of exponent; the desired value r^* is the smallest exponent for which Schrijver's algorithm returns a solution.

Choosing (close to) minimum possible number of turns in a spiraling ring gives us also one more advantage: the paths output by the Schrijver's algorithm cannot differ much from the bundle words we have fed the algorithm, and which makes the output really meaningful. A significant technical work needs to be also devoted to choosing a proper subgraph of the input graph, so that the invoked algorithm measures only one unknown exponent in the bundle words; here the main trick is to guess the exponents in the order of increasing lengths of their bases.

E. Handling a ring component

In the previous subsection we have sketched how to guess bundle words of the solution in absence of ring components. Recall that, for a ring component, and for any part of a path that traverses a ring component (henceforth called *ring passage*; note that ring passages are visible in bundle words of paths) we need to know its *winding number*: the number of times it turns inside the ring component.

As we have learnt already how to route paths in rings, it is tempting to use the aforementioned techniques to guess winding numbers: guess how many ring passages there are, and find one winding number w^* for which routing is possible (using Schrijver's algorithm)[2]; the actual solution should be reroutable to a winding number w close to w^*.

However, there are two major problems with this approach. First, not only the ring passages of the solution use the arcs and vertices of a ring component, but parts of paths from the solution that start and end on the same side of the ring component (henceforth called *ring visitors*) may also be present. Luckily, we may assume that the ring components contain many concentric cycles of alternating orientation, as otherwise the decomposition algorithm would cut it though to obtain a disc component. If a ring visitor goes too deeply into the ring component, it creates a d-bend for too large d and we can reroute it. Thus, the ring visitors use only a thin layer of the ring component, and we can argue that we can still conduct the rerouting argument in the ring component without bigger loss on the bound on $|w - w^*|$.

Second, we do not have yet any means to control the number of ring passages, and the previous techniques of guessing bundle words have significant technical problems if we try to handle spiraling rings involving ring components. Hence, it is not trivial to guess the set of ring passages traversing a ring component. Here again we may use the concentric cycles hidden inside a ring component, as well as bounded alternation cuts that can be found repeatedly inside the ring component if the irrelevant vertex rule is not applicable. We argue that, if we have many ring passages, a large number of them need to travel together via a large number of concentric cycles of alternating orientation and we can use a rerouting argument in the spirit of the one used by Adler et al. [7].[3] Overall, we obtain that there exists a solution with a bounded number of ring passages, and we are able to guess a good candidate for a winding number inside a ring component. To merge the techniques of the previous and this subsection, we need to handle ring visitors when guessing bundle words: these visitors may take part in some large spiraling ring. Luckily, as ring visitors are shallow in ring components, we can "peel" the ring components: separate a thin layer that may contain ring visitors, cut it through and treat is as disc component for the sake of bundle word guessing algorithm.

F. Summary

We conclude with a summary of the structure of the algorithm. First, we invoke decomposition algorithm of Theorem II.3. If it fails, it exhibits a place where the irrelevant vertex rule is applicable; apply the rule and restart the algorithm. Otherwise, compute bundled instance $(G, \mathcal{D}, \mathbb{B})$, with $|\mathcal{D}|$ and $|\mathbb{B}|$ bounded in k.

Given the bundled instance $(G, \mathcal{D}, \mathbb{B})$, we aim to branch into subcases whose number is bounded by a function of k, in each subcase guessing a set of bundle words for the solution, as well as winding numbers of each ring passage. Our branching will be exhaustive in the following sense: if G is a YES-instance to k-DPP, there will be a guess consistent with some solution (but not *all* solutions will have their consistent branches).

We branch in two phases. First, we guess the bundle words; the hard part is to guess exponents in spiraling rings u^r, where we argue that we can choose an exponent close

[2] Note that all winding numbers of passages in one ring component do not differ by more than one, and in this overview we assume they are equal.

[3] It is worth noting that the bound on directions make it possible to use a simple flow argument instead of the techniques of Adler et al. [7].

to minimal possible number of turns in a spiraling ring, and detect this number using an application of Schrijver's algorithm for a carefully chosen subgraph of G. Second, we guess the winding numbers; here we argue that the winding numbers of the solution can be assumed to be close to a winding number of an arbitrarily chosen way to route ring passages through the ring component, ignoring the ring visitors.

Finally, given bundle words and winding numbers, we deduce the homotopy type of the solution and invoke Schrijver's algorithm on the entire graph to verify whether our guess is a correct one.

References

[1] M. Cygan, D. Marx, M. Pilipczuk, and M. Pilipczuk, "The planar directed k-vertex-disjoint paths problem is fixed-parameter tractable," *CoRR*, vol. abs/1304.4207, 2013.

[2] M. Middendorf and F. Pfeiffer, "On the complexity of the disjoint paths problem," *Combinatorica*, vol. 13, no. 1, pp. 97–107, 1993.

[3] N. Robertson and P. D. Seymour, "Graph minors. XIII. The disjoint paths problem," *J. Combin. Theory Ser. B*, vol. 63, no. 1, pp. 65–110, 1995.

[4] K. Kawarabayashi and P. Wollan, "A shorter proof of the graph minor algorithm: the unique linkage theorem," in *STOC*, 2010, pp. 687–694.

[5] N. Robertson and P. D. Seymour, "Graph minors. VII. Disjoint paths on a surface," *J. Comb. Theory, Ser. B*, vol. 45, no. 2, pp. 212–254, 1988.

[6] B. A. Reed, N. Robertson, A. Schrijver, and P. D. Seymour, "Finding disjoint trees in planar graphs in linear time," in *Graph Structure Theory*, 1991, pp. 295–302.

[7] I. Adler, S. G. Kolliopoulos, P. K. Krause, D. Lokshtanov, S. Saurabh, and D. M. Thilikos, "Tight bounds for linkages in planar graphs," in *ICALP (1)*, 2011, pp. 110–121.

[8] S. Fortune, J. Hopcroft, and J. Wyllie, "The directed subgraph homeomorphism problem," *Theoret. Comput. Sci.*, vol. 10, no. 2, pp. 111–121, 1980.

[9] A. Schrijver, "Finding k disjoint paths in a directed planar graph," *SIAM J. Comput.*, vol. 23, no. 4, pp. 780–788, 1994.

[10] H. L. Bodlaender, M. R. Fellows, and M. T. Hallett, "Beyond NP-completeness for problems of bounded width: hardness for the W hierarchy," in *STOC*, 1994, pp. 449–458.

[11] R. G. Downey and M. R. Fellows, *Parameterized Complexity*, ser. Monographs in Computer Science. New York: Springer, 1999.

[12] M. Grohe, K. Kawarabayashi, D. Marx, and P. Wollan, "Finding topological subgraphs is fixed-parameter tractable," in *STOC*, 2011, pp. 479–488.

[13] K. Kawarabayashi, B. A. Reed, and P. Wollan, "The graph minor algorithm with parity conditions," in *FOCS*, 2011, pp. 27–36.

[14] R. Diestel, T. R. Jensen, K. Y. Gorbunov, and C. Thomassen, "Highly connected sets and the excluded grid theorem," *J. Comb. Theory, Ser. B*, vol. 75, no. 1, pp. 61–73, 1999.

[15] N. Robertson, P. D. Seymour, and R. Thomas, "Quickly excluding a planar graph," *J. Comb. Theory, Ser. B*, vol. 62, no. 2, pp. 323–348, 1994.

[16] N. Robertson and P. D. Seymour, "Graph minors. V. Excluding a planar graph," *J. Comb. Theory, Ser. B*, vol. 41, no. 1, pp. 92–114, 1986.

[17] Q.-P. Gu and H. Tamaki, "Improved bounds on the planar branchwidth with respect to the largest grid minor size," *Algorithmica*, vol. 64, no. 3, pp. 416–453, 2012.

[18] T. Johnson, N. Robertson, P. D. Seymour, and R. Thomas, "Directed tree-width," *J. Comb. Theory, Ser. B*, vol. 82, no. 1, pp. 138–154, 2001.

[19] K. Kawarabayashi and S. Kreutzer, "An excluded grid theorem for digraphs with forbidden minors," 2013, manuscript.

[20] D. Berwanger, A. Dawar, P. Hunter, S. Kreutzer, and J. Obdrzálek, "The dag-width of directed graphs," *J. Comb. Theory, Ser. B*, vol. 102, no. 4, pp. 900–923, 2012.

[21] R. Ganian, P. Hlinený, J. Kneis, D. Meister, J. Obdrzálek, P. Rossmanith, and S. Sikdar, "Are there any good digraph width measures?" in *IPEC*, 2010, pp. 135–146.

[22] R. Ganian, P. Hlinený, J. Kneis, A. Langer, J. Obdrzálek, and P. Rossmanith, "On digraph width measures in parameterized algorithmics," in *IWPEC*, 2009, pp. 185–197.

[23] D. Meister, J. A. Telle, and M. Vatshelle, "Characterization and recognition of digraphs of bounded Kelly-width," in *WG*, 2007, pp. 270–279.

[24] M. A. Safari, "D-width: A more natural measure for directed tree width," in *MFCS*, 2005, pp. 745–756.

[25] S. Kreutzer and S. Ordyniak, "Digraph decompositions and monotonicity in digraph searching," *Theor. Comput. Sci.*, vol. 412, no. 35, pp. 4688–4703, 2011.

[26] A. Slivkins, "Parameterized tractability of edge-disjoint paths on directed acyclic graphs," *SIAM J. Discrete Math.*, vol. 24, no. 1, pp. 146–157, 2010.

[27] G. Ding, A. Schrijver, and P. D. Seymour, "Disjoint cycles in directed graphs on the torus and the Klein bottle," *J. Comb. Theory, Ser. B*, vol. 58, no. 1, pp. 40–45, 1993.

[28] ——, "Disjoint paths in a planar graph - a general theorem," *SIAM J. Discrete Math.*, vol. 5, no. 1, pp. 112–116, 1992.

[29] T. Johnson, N. Robertson, P. D. Seymour, and R. Thomas, unpublished.