Tight Bounds for Linkages in Planar Graphs

Isolde Adler[1] Stavros G. Kolliopoulos[2] Philipp Klaus Krause[1]
Daniel Lokshtanov[3] Saket Saurabh[4] Dimitrios Thilikos[2]

[1] Goethe-Universität, Frankfurt am Main
[2] National and Kapodistrian University of Athens
[3] University of California, San Diego
[4] Institute of Mathematical Sciences, Chennai

Abstract. The Disjoint-Paths Problem asks, given a graph G and a set of pairs of terminals $(s_1, t_1), \ldots, (s_k, t_k)$, whether there is a collection of k pairwise vertex-disjoint paths linking s_i and t_i, for $i = 1, \ldots, k$. In their $f(k) \cdot n^3$ algorithm for this problem, Robertson and Seymour introduced the *irrelevant vertex technique* according to which in every instance of treewidth greater than $g(k)$ there is an "irrelevant" vertex whose removal creates an equivalent instance of the problem. This fact is based on the celebrated *Unique Linkage Theorem*, whose – very technical – proof gives a function $g(k)$ that is responsible for an immense parameter dependence in the running time of the algorithm. In this paper we prove this result for planar graphs achieving $g(k) = 2^{O(k)}$. Our bound is radically better than the bounds known for general graphs. Moreover, our proof is new and self-contained, and it strongly exploits the combinatorial properties of planar graphs. We also prove that our result is optimal, in the sense that the function $g(k)$ cannot become better than exponential. Our results suggest that any algorithm for the Disjoint-Paths Problem that runs in time better than $2^{2^{o(k)}} \cdot n^{O(1)}$ will probably require drastically different ideas from those in the irrelevant vertex technique.

Keywords: Disjoint-Paths Problem, Planar Graphs, Linkages, Treewidth, Parameterized Algorithms.

1 Introduction

One of the most studied problems in graph algorithms is the Disjoint-Paths Problem (DPP): *Given a graph G, and a set of k pairs of terminals, $(s_1, t_1), \ldots, (s_k, t_k)$, decide whether G contains k vertex-disjoint paths $P_1, \ldots, P_k$ where P_i has endpoints s_i and t_i, $i = 1, \ldots, k$.* In addition to its numerous applications in areas such as network routing and VLSI layout, this problem has been the catalyst for extensive research in algorithms and combinatorics [22]. DPP is NP-complete, along with its edge-disjoint or directed variants, even when the input graph is planar [23,16,13,15]. The celebrated algorithm of Roberson and Seymour solves it however in $f(k) \cdot n^3$ steps, where f is some computable function [17]. This implies that when we parameterize DPP by the number k of terminals, the problem is fixed-parameter tractable. The Robertson-Seymour algorithm is the central algorithmic result of the Graph Minors series of papers, one of the deepest and most influential bodies of work in graph theory.

The basis of the algorithm in [17] is the so called *irrelevant-vertex technique* which can be summarized very roughly as follows. As long as the input graph G violates certain structural conditions, then it is possible to find a vertex v that is *solution-irrelevant:* every collection of paths certifying a solution to the problem can be rerouted to an *equivalent* one, that links the same pairs of terminals, but in which the new paths avoid v. One then iteratively removes such irrelevant vertices until the structural conditions are met. By that point the graph has been simplified enough so that the problem can be attacked via dynamic programming.

The following two structural conditions are used by the algorithm in [17]: (i) G excludes a clique, whose size depends on k, as a minor and (ii) G has treewidth bounded by some function of k. When it comes to enforcing Condition (ii), the aim is to prove that in graphs without big clique-minors and with treewidth at least $g(k)$ there is always a solution-irrelevant vertex. This is the most complicated part of the proof and it was postponed until the later papers in the series [18,19]. The bad news is that the complicated proofs also imply an *immense* dependence, as expressed by the function f, of the running time on the parameter k. This puts the algorithm outside the realm of feasibility even for elementary values of k.

The ideas above were powerful enough to be applicable also to problems outside the context of the Graph Minors series. During the last decade, they have been applied to many other combinatorial problems and now they constitute a basic paradigm in parameterized algorithm design (see, e.g., [3,4,7,9,10,12]). However, in most applications, the need for overcoming the high parameter dependence emerging from the structural theorems of the Graph Minors series, especially those in [18,19], remains imperative. Hence two natural directions of research are: simplify parts of the original proof for the general case or focus on specific graph classes that may admit proofs with better parameter dependence. An important step in the first direction was taken recently by Kawarabayashi and Wollan in [11] who gave an easier and shorter proof of the results in [18,19]. While the parameter dependence of the new proof is certainly much better than the previous, immense, function, it is still huge: a rough estimation from [11] gives a lower bound for $g(k)$ of magnitude $2^{2^{2^{\Omega(k)}}}$ which in turn implies a lower bound for $f(k)$ of magnitude $2^{2^{2^{2^{\Omega(k)}}}}$.

In this paper we offer a solid advance in the second direction, focusing on planar graphs. We prove that, for planar graphs, $g(k)$ is singly exponential. In particular we prove the following result.

Theorem 1. *There is a constant c such that every n-vertex planar graph G with treewidth at least c^k contains a vertex v such that every solution to* DPP *with input G and k pairs of terminals can be replaced by an equivalent one avoiding v.*

Given the above result, our Theorem 6 shows how to reduce, in $O(n^2)$ time, an instance of DPP to an an equivalent one whose graph G' has treewidth $2^{O(k)}$. Then, using dynamic programming, a solution, if one exists, can be found in $k^{O(\text{treewidth}(G'))} \cdot n = 2^{2^{O(k)}} \cdot n$ steps.

The proof of Theorem 1 deviates significantly from those in [18,19,11]. It is self-contained and exploits extensively the combinatorics of planar graphs. Moreover, we give strong evidence that a parameterized algorithm for DPP with singly exponential dependence, if one exists, should require entirely different techniques. Indeed, in that sense, the result in Theorem 1 is tight:

Theorem 2. *There exists an instance of the* DPP *on a $2^{\Omega(k)}$-treewidth planar graph G that has a unique solution spanning all the vertices of G.*

Notice that, due to the recent lower bounds in [14], the Disjoint-Paths Problem cannot be solved in $2^{o(w \log w)} \cdot n^{O(1)}$ for graphs of treewidth at most w, unless the Exponential Time Hypothesis (ETH) fails. This result, along with Theorem 2, reveals the limitations of the irrelevant vertex technique: any algorithm for the Disjoint-Paths Problem whose parameter dependence that is better than doubly exponential, will probably require drastically different techniques.

2 Preliminaries

Graphs are finite, undirected and simple. We denote the vertex set of a graph G by $V(G)$ and the edge set by $E(G)$. Every edge is a two-element subset of $V(G)$. A graph H is a *subgraph* of a graph G, denoted by $H \subseteq G$, if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. A *path* in a graph G is a sequence $P = v_1, \ldots, v_n$ of pairwise distinct vertices of G, such that $v_i v_{i+1} \in E(G)$ for all $1 \leq i \leq n-1$. For a graph G with $e = vw \in E(G)$ let G/e denote the graph obtained from G by *contracting* e, i.e. $V[G/e] := (V(G) \setminus \{v, w\}) \cup \{x_e\}$, where x_e is a new vertex, and $E(G/e) := \big(E(G) \setminus \{uu' \mid uu' \cap e \neq \emptyset\}\big) \cup \big\{ux_e \mid uv \in E(G) \text{ or } uw \in E(G)\big\}$. A graph H is a *minor* of a graph G, if H can be obtained from a subgraph of G by a sequence of edge contractions. We use standard graph terminology as in [5]. The *disjoint paths problem* (DPP) is the following problem.

DPP

Input: A graph G, and pairs of terminals $(s_1, t_1), \ldots, (s_k, t_k) \in V(G)^{2k}$
Question: Are there k pairwise vertex disjoint paths $P_1, \ldots, P_k$ in G such that P_i has endpoints s_i and t_i?

We will call such a sequence $P_1, \ldots, P_k$ a *solution* of the DPP.

Given an instance $(G, (s_1, t_1), \ldots, (s_k, t_k))$ of DPP we say that a non-terminal vertex $v \in V(G)$ is *irrelevant*, if $(G, (s_1, t_1), \ldots, (s_k, t_k))$ is a YES-instance if and only if $(G \setminus v, (s_1, t_1), \ldots, (s_k, t_k))$ is a YES-instance. From now on G will always be an instance to DPP accompanied by k terminal pairs.

Definition 1 (Grid). *Let $m, n \geq 1$. The $(m \times n)$* grid *is the Cartesian product of a path of length $m-1$ and a path of length $n-1$. In case of a square grid where $m=n$, then we say that n is the* size *of the grid.*

A *subdivided grid* is a graph obtained from a grid by replacing some edges of the grid by pairwise internally vertex disjoint paths of length at least one. *Embeddings* of graphs in the plane, *plane graphs*, *planar graphs* and *faces* are defined in the usual way. A graph is *outerplanar*, if it has an embedding in the plane where all vertices are incident to the infinite face.

A *cycle* in a graph G is a subgraph $H \subseteq G$, such that $V(H) = \{x_0, x_1, \ldots, x_{k-1}\}$, $E(H) = \{x_0x_1, x_1x_2, \ldots, x_{k-2}x_{k-1}, x_{k-1}x_0\}$ for some $k \in \mathbb{N}$, $k \neq 1$, and $i \neq j \Rightarrow x_i \neq x_j$. Notice that we allow cycles to consist of a single vertex.

We use the fact that a subdivided grid has a unique embedding in the plane (up to homeomorphism). For (1×1) grids and subdivided (2×2) grids this is clear, and for $(n \times n)$ grids with $n \geq 2$ this follows from Tutte's Theorem stating that 3-connected graphs have unique embeddings in the plane (up to homeomorphism). This implies that subdivisions of $(n \times n)$ grids have unique embeddings as well. The *perimeter* of a subdivided grid H is the cycle in H that is incident to the outer face (in an/every planar embedding of H).

A *directed graph* is a pair $D = (V, E)$ where V is a set and $E \subseteq V \times V$. We call the elements of E *directed edges*. For a directed edge $(u, v) \in E$ we say that u is the *tail* of (u, v), $u = \mathrm{tail}(u, v)$, and v is the *head* of (u, v), $v = \mathrm{head}(u, v)$.

3 Upper Bounds

The main result of this section is Theorem 6 stating that there is an $O(n^2)$-step algorithm that, given an instance G of DPP of treewidth $2^{\Omega(k)}$, can find a set of irrelevant vertices whose removal from G creates an equivalent instance of treewidth $2^{O(k)}$.

3.1 Basic Definitions

For our proofs we need first some definitions.

Observation 1 *Let G, $(s_1, t_1, \ldots, s_k, t_k)$ be a planar instance of* DPP *and let $h \in \mathbb{N}$. If G contains a subdivided $((h\sqrt{2k+1}) \times (h\sqrt{2k+1}))$ grid, then G contains a subdivided $h \times h$ grid H such that in every embedding of H all terminals lie outside the open disc bounded by the perimeter of H.*

Notice that Observation 1 ensures that once we have a large grid we can also assume that we have a large grid *that does not contain any terminal vertices.* Next we define a specific kind of embedding of cycles that helps us enforce structure in the proof.

Definition 2 (Tight concentric cycles). *Let G be a plane graph and let $C_0, \ldots, C_n$ be a sequence of cycles in G such that each cycle bounds a closed disc D_i in the plane. We call $C_0, \ldots, C_n$* concentric*, if for all $i \in \{0, \ldots, n-1\}$, the cycle C_i is contained in the interior of D_{i+1}. The concentric cycles $C_0, \ldots, C_n$ are* tight*, if, in addition, C_0 is a single vertex and for every $i \in \{0, \ldots, n-1\}$, $D_{i+1} \setminus D_i$ does not contain a cycle C bounding a disc D in the plane with $D_{i+1} \supsetneq D \supseteq D_i$.*

Two simple remarks are in order about tight concentric cycles.

Remark 1. Let G be a plane graph and let $C_0, \ldots, C_n$ be tight concentric cycles in G bounding closed discs $D_1, \ldots, D_n$, respectively, in the plane. Let P be a path connecting vertices u and v with $u, v \notin D_n$. If a vertex of P is contained in the interior of D_i (i.e. in $D_i \setminus C_i$), then P has a vertex on C_{i-1}.

Remark 2. If a graph contains a $((2n+1) \times (2n+1))$ grid minor, it contains a sequence $C_0, \ldots, C_n$ of tight concentric cycles.

A *linkage* in a graph G is a family of pairwise disjoint paths in G. The *endpoints* of a linkage L are the endpoints of the paths in L, and the *pattern* of L is the matching on the endpoints induced by the paths, i.e. the pattern is the set $\big\{\{s,t\} \mid L$ contains a path from s to $t\big\}$.

Definition 3 (Segment & Handle). *Let G be a plane graph, let C be a cycle in G bounding a closed disc D in the plane and let P be a path in G such that its endpoints are outside of D. We say that a path P_0 is a D-*segment *(resp. D-*handle*) of P, if P_0 is a non-empty maximal subpath of P whose endpoints are on C, and $P_0 \subseteq D$ (resp. $P_0 \cap D$ contains only the endpoints of P_0). For a linkage $\mathcal{P}$ in G we say that a path P_0 is a D-*segment *(D-*handle*) of $\mathcal{P}$, if P_0 is a D-*segment *(D-*handle*) of some path P of $\mathcal{P}$.*

Remark 3. Let G be a plane graph, let C be a cycle in G bounding a closed disc D in the plane and let $P = su_1 \ldots u_q t$ be a path in G such that s and t are outside D. Suppose $x_a, x_b, x_c, \ldots, x_j$ is the order in which the vertices of path P appear on the cycle C when we traverse it from s to t. Then the subpath of P between x_a and x_b is a D-segment while the subpath of P between x_b and x_c is a D-handle, and D-segments and D-handles alternate.

From now on we will assume that G is a plane graph containing a sequence $C_0, \ldots, C_n$ of concentric cycles bounding closed discs $D_0, \ldots, D_n$, respectively, in the plane. Furthermore there are no terminals contained in D and G is an YES-instance. That is, there are paths between s_i and t_i such that they are mutually disjoint. These paths form a linkage that will be denoted by $\mathcal{P}$. From now onwards whenever we say *linkage* we mean a set of disjoint paths between pairs (s_i, t_i). We will often refer to the D_n-segments (D_n-handles) of $\mathcal{P}$ simply as the *segments (handles)* of $\mathcal{P}$.

Definition 4 (I(Handle) and β(Handle)). *Let G be a plane graph containing a sequence $C_0, \ldots, C_n$ of concentric cycles bounding closed discs $D_0, \ldots, D_n$, respectively, in the plane and $\mathcal{P}$ be a linkage. Let P be a D_n-handle and let its endpoints be x and y. Let $C_n[x,y]$ denote the path between x, y on the cycle C such that the finite face bounded by $P \cup C_n[x,y]$ does not contain the interior of D_n. By $I(P)$ we denote the subgraph of G that has boundary $P \cup C_n[x,y]$, and we let $\beta(P) := C_n[x,y]$.*

Definition 5 (Cheap solution). *Let G be a plane graph containing a sequence $C_0, \ldots, C_n$ of concentric cycles bounding closed discs $D_0, \ldots, D_n$, respectively, in the plane. For a linkage $\mathcal{P}$ of G, define its *cost *$c(\mathcal{P})$ as the number of edges of $\mathcal{P}$ that do not belong to $\bigcup_{i=0}^{n} C_i$. A linkage $\mathcal{P}$ is called *cheap*, if there is no other linkage $\mathcal{Q}$, such that $c(\mathcal{Q}) < c(\mathcal{P})$.*

Observe that the contribution of a D_n-handle of $\mathcal{P}$ to $c(\mathcal{P})$ is always positive. Edges of D_n-segments contribute to $c(\mathcal{P})$ whenever they do not belong to a concentric cycle. We assume for the remainder of Section 3 that we are given a cheap solution $\mathcal{P}$ to our input instance and we explore its structure.

3.2 Simple properties of a cheap solution $\mathcal{P}$

Lemma 1 ($\star$). *If $\mathcal{P}$ is a cheap solution to the input instance then there is no segment P of $\mathcal{P}$ with vertices appearing in the order $\dots, v_0, \dots, v_1, \dots, v_2, \dots$ where v_0 and v_2 are vertices of C_ℓ, and v_1 is a vertex of C_j, for $n \geq j > \ell \geq 0$.*

Lemma 2 ($\star$). *Let $\mathcal{P}$ be a cheap solution to the input instance and Q be a handle. Then there is terminal inside $I(Q)$.*

We remark that Lemma 2 is true in more general setting. We will use the generalized version in a proof later. Let D be a disc with the boundary cycle C and $\mathcal{T}$ be a subpath of a path in $\mathcal{P}$ with endpoints x and y on the disc and no points in the interior of the disc. Let $C[x, y]$ denote the path between x, y on cycle C such that the finite face bounded by $\mathcal{T} \cup C[x, y]$ does not contain the interior of D. By $I(\mathcal{T})$ we denote the subgraph that has boundary $\mathcal{T} \cup C[x, y]$. A proof similar to the one in Lemma 2 gives us the following.

Lemma 3. *Let $\mathcal{P}$ be a cheap solution to the input instance and $\mathcal{T}$ be a subpath of a path in $\mathcal{P}$ with endpoints on the disc and no points in the interior of the disc. Then there is a terminal inside $I(\mathcal{T})$.*

3.3 Bounding the number of segment types

In this section we define a notion of segment types and obtain an upper bound on the number of segment types.

Definition 6 (Segment Type). *Let $\mathcal{P}$ be a solution to the input instance. Let R and S be two D_n-segments. Let Q and Q' be the two paths on C_n connecting an endpoint of R with an endpoint of S and passing through no other endpoint of R or S. We say that R and S are* equivalent*, and we write $R \parallel S$, if no D_n-segment of $\mathcal{P}$ has both endpoints on Q and no D_n-segment has both endpoints on Q'. A* type *of D_n-segments is an equivalence class of D_n-segments under the relation $\parallel$.*

Definition 7 (Segment graph). *We start with the subgraph of G contained in D_n. Retain only the edges and vertices of $\bigcup \mathcal{P} \cup C_n$. Choose an edge. If it is part of C_n, contract it unless it connects endpoints of segments of different type. If it is not part of C_n, contract it unless it connects endpoints of segments. Repeat until there are no contractable edges left. Remove duplicate edges and loops, such that the graph becomes simple again. The resulting graph is the* segment graph *of D_n.*

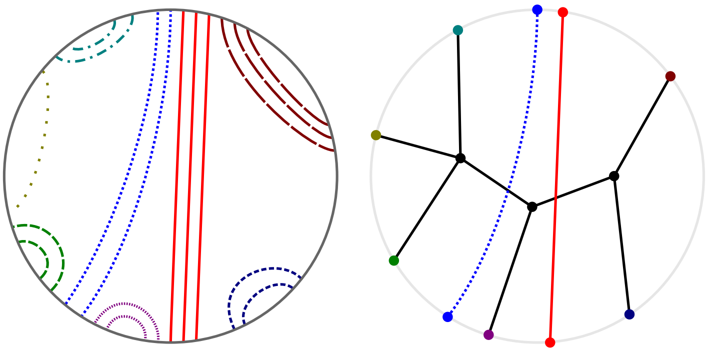
Segment graphs are outerplanar graphs. An example can be seen in Figure 1.

Definition 8 (Tongue tip). *A D_n-segment type is called* tongue tip*, if it is a single vertex in the segment graph of D_n.*

Definition 9 (Segment dual graph). *We take the dual graph of the segment graph of D_n. Delete the vertex that represents the infinite face. Add the vertices representing the tongue tips of the segment graph and connect them to the vertices representing neighboring faces in the segment graph. The resulting graph is the* segment dual graph *of D_n.*

It is easy to see that the segment dual graph is a tree. See Figure 1(b) for an example.

Remark 4. Since the segment graph is outerplanar, the segment dual graph is a tree. All inner nodes of the segment dual graph have degree at least 3.



(a) Segments of 8 Types, with 6 of the types being tongue tips

(b) Segment graph (colored) and segment dual graph (black)

Fig. 1. Segments

The next lemma is based on Lemmata 2 and 3 and is one of the main ingredients of our proof. It.

Lemma 4 (Tongue-taming ($\star$)). *Let $\mathcal{P}$ be a cheap solution to the input instance. Then there are at most $2k-1$ tongue tips.*

Theorem 3. *Let $\mathcal{P}$ be a cheap solution to the input instance then $\mathcal{P}$ has at most $4k-3$ different types of D_n-segments.*

Proof. The segment types correspond to the edges in the segment dual graph. The tongue tips correspond to the leaves of the segment dual graph. According to Lemma 4 this tree has at most $2k$ leaves, and according to Remark 4 all inner nodes have degree at least three. Thus the segment dual graph has at most $4k-3$ edges. □

3.4 Bounding the size of segment types

In this section we find a bound on the size of segment types in cheap solutions and we combine it with the bound on the number of segment types obtained in the previous section to find irrelevant vertices. Indeed, we find that cheep solutions only pass through a bounded number of concentric cycles.

We find the bound on the size of segment types by rerouting in the presence of a large segment type. In a first step, we allow ourselves to freely reroute in a disc (making sure that the graph remains planar), and we bound the number of segments of solution paths in the disc. In a second step, we realize our rerouting in a sufficiently large grid.

Lemma 5 ($\star$). *Let Σ be an alphabet of size $|\Sigma| = k$. Let $w \in \Sigma^*$ be a word over Σ. If $|w| > 2^k$, then w contains an infix y with $|y| \geq 2$, such that every letter occurring in y occurs an even number of times in y.*

The following lemma is essentially the main combinatorial result from [1]. The proof is included here for the sake of completeness.

Lemma 6 (Rerouting in a disc ($\star$)). *Let G be a plane graph with k pairs of terminals such that the DPP has a solution $\mathcal{P}$. Let G contain a cycle C bounding a closed disc D in the plane, such that no terminal lies in D. Assume that*

- *every D-segment of $\mathcal{P}$ is simply an edge,*
- *besides vertices and edges of D-segments, the interior of D contains no other vertices or edges of G.*

If there is a segment type that contains more than 2^k segments, then we can replace the outerplanar graph O consisting of all D-segments of $\mathcal{P}$ by a new outerplanar graph O' such that in $(G \setminus O) \cup O'$ the DPP (with the original terminals) has a solution and $|E(O')| < |E(O)|$.

Definition 10. *Let $n, m \in \mathbb{N}$. An* untidy $(n \times m)$ grid *is a graph obtained from a set $\mathcal{H}$ of n pairwise vertex-disjoint (*horizontal*) paths and a set $\mathcal{V} = \{V_1, \dots, V_m\}$ of m pairwise vertex-disjoint (*vertical*) paths as follows: Every path in $\mathcal{V}$ intersects every path in $\mathcal{H}$ in precicely one non-empty path, and each path $H \in \mathcal{H}$ consists of m vertex-disjoint segments such that V_i intersects H only in its ith segment (for every $i \in \{1, \dots, m\}$). A* subdivided untidy $(n \times m)$ grid *is obtained from an $(n \times m)$ grid by subdividing edges.*

Let τ be a segment type in the plane graph. Recall that all the segments in a type are “parallel” to each other. We say that segments $S_1, \dots, S_n \in \tau$ are *consecutive*, if they appear in this order (or in the reverse order) in the plane. Segment types that go far into the concentric cycles yield subdivided untidy grids. More precisely, we show the following that is an easy consequence of Lemma 1.

Lemma 7. *Let $l, n, r \in \mathbb{N}$ with $n \geq l-1$. Let $\mathcal{P}$ be a cheap solution to the input instance. If there is a type τ of D_n-segments of $\mathcal{P}$ with $|\tau| \geq r$ such that r consecutive segments of τ each contain a vertex of D_{n-l+1}, then G contains a subdivided untidy $(2l \times r)$ grid as a subgraph, with the r consecutive segments of τ as vertical paths, and suitable subpaths of $C_n, \dots, C_{n-l+1}, C_{n-l+1}, \dots, C_n$ (in this order) as horizontal paths.* □

The following lemma shows that we can reroute a sufficiently large segment type in the case that many segments of the type go far into the concentric cycles.

Lemma 8 (Rerouting in an untidy grid). *Let $n, k \in \mathbb{N}$ with $n \geq 2^{k-1} - 1$. Let $\mathcal{P}$ be a cheap solution to the input instance. Then $\mathcal{P}$ has no type τ of D_n-segments with $|\tau| \geq 2^k + 1$, such that each of $2^k + 1$ consecutive segments in τ contains a vertex in $D_{n-2^{k-1}+1}$.*

Proof. Towards a contradiction, assume that τ is a type of D_n-segments of $\mathcal{P}$ with $|\tau| \geq 2^k + 1$, such that each of $2^k + 1$ consecutive segments in τ contains a vertex in $D_{n-2^{k-1}+1}$. Let $r := 2^k + 1$ and let $l := 2^{k-1}$. Let $H \subseteq G$ be a subdivided untidy $(2l \times r)$ grid as in Lemma 7. The proof that follows is similar to the proof of Lemma 6, that is, we reroute some segments of τ. In addition, we make sure that we can realize the rerouted segments in H. If we let $D := D_n$, contract every D_n-segment of $\mathcal{P}$ to a single edge and remove all other vertices and edges of G in the interior of D, then we can apply Lemma 6 to the segment τ_0 obtained from τ by the contraction. We do this as follows: order the segments of τ (and hence of τ_0) according to their occurence in the plane. Colour the first $2^k + 1$ segments by the number of the path of $\mathcal{P}$ they belong to. Among these consecutive paths, find an infix d of colours as in Lemma 5. Then $2 \leq |d| \leq 2^k$. Let H' be the subdivided unitidy $(2l \times |d|)$ subgrid of H with vertical paths corresponding only to the segments in d, and the horizontal paths shortened accordingly, as much as possible.

Reroute the segments in τ that correspond to the the letters of d as in Lemma 6. Then we obtain $\frac{|d|}{2}$ new segments (indeed, in the lemma each of them is a single edge, but now we simply subdivide them if necessary) and a new solution $\mathcal{P}'$ of the DPP. For routing the new segments in H', we use the paths of the old segments as follows: For routing a new segment, we use at most two old segments corresponding to a letter in d (two vertical paths in H'), and, for crossing horizontally, one horizontal path in H. Notice that all D_n-segments of $\mathcal{P}'$ use subpaths of $\mathcal{P}$ and horizontal paths of H' only. Since $2 \leq |d| \leq 2^k$ it follows that $1 \leq \frac{|d|}{2} \leq 2^{k-1}$. But H' has $2l = 2^k$ horizontal paths and horizontal crossings of $\mathcal{P}'$ use at most $\frac{|d|}{2} < 2^k$ of them. Hence one of them, h say, is not used by any horizontal crossing of $\mathcal{P}'$. But then h has a crossing with at least $1 \leq \frac{|d|}{2}$ vertical path in H' that is not used by any path in $\mathcal{P}'$. With this it is easy to see that $c(\mathcal{P}') < c(\mathcal{P})$, a contradiction. □

The following remark says that if we have a sufficiently large segment type, then many segments will go far into the concentric cycles.

Lemma 9. *Let $n, l, r \in \mathbb{N}$. Let $\mathcal{P}$ be a cheap solution to the input instance. Let τ be a type of D_n-segments with $|\tau| \geq 2l + r$. Then $n \geq l - 1$ and τ contains r consecutive segments such that each of them has a vertex in D_{n-l+1}.*

Proof. Order the segments in τ according to ther occurence in the planar embedding. Remove the l first and the l last segments. From Lemma 1 it follows that $n \geq l - 1$ and each of the remaining segments contains a vertex in D_{n-l+1}. □

Theorem 4 (Bounding the number of segments). *Let $\mathcal{P}$ be a cheap solution to the input instance. Then there are at most $(8k-6) \cdot 2^k + 4k - 3$ D_n-segments of $\mathcal{P}$.*

Proof. We first prove that every type of D_n-segments of $\mathcal{P}$ contains less than $2 \cdot 2^k + 1$ segments. Let $l := 2^{k-1}$ and $r := 2^k + 1$. Towards a contradiction, assume that τ is a type of D_n-segments with $|\tau| \geq 2 \cdot 2^k + 1 = 2l + r$. Then, by Lemma 9, τ contains $r = 2^k + 1$ consecutive segments such that each of them has a vertex in D_{n-l+1}, but this is not possible by Lemma 8, a contradiction.

Now, by Theorem 3 the number of types of D_n-segments of $\mathcal{P}$ is at most $4k - 3$, so the total number of segments is at most $(8k-6) \cdot 2^k + 4k - 3$, concluding the proof. □

Theorem 5 (Irrelevant Vertex). *Let G be a plane graph with k pairs of terminals, $n = (8k-6) \cdot 2^k + 4k - 2$, and let G contain a sequence $C_0, \ldots, C_n$ of concentric cycles bounding closed discs $D_0, \ldots, D_n$, respectively, in the plane, such that no terminal of the* DPP *lies in D_n. Let $C_0 = \{v\}$, and assume that the* DPP *has a solution. Then the* DPP *has a solution that avoids v.*

Let G be a plane graph with k pairs of terminals, $n = (8k-6) \cdot 2^k + 4k - 2$, and let G contain a sequence $C_0, \ldots, C_n$ of tight concentric cycles bounding closed discs $D_0, \ldots, D_n$, respectively, in the plane, such that no terminal of the DPP *instance lies in D_n. Assume that* DPP *has a solution. Then there is a vertex $v \in V(G)$ such that* DPP *has a solution that avoids v.*

Proof. Without loss of generality we assume that the cycles $C_0, \ldots, C_n$ form a sequence of tight concentric cycles around some vertex $v \in V(G)$. Let $\mathcal{P}$ be a cheap solution to DPP. We argue that $\mathcal{P}$ avoids v. By Theorem 4 the number of D_n-segments of $\mathcal{P}$ is at most $n - 1$.

Consider a D_n-segment P of $\mathcal{P}$. Let i be the lowest integer such that $P \cap C_i \neq \emptyset$. We say that P *peaks* at C_i. Note that P peaks at exactly one cycle C_i. Suppose $\mathcal{P}$ does not avoid v. Since the number of D_n-segments of $\mathcal{P}$ is at most $n - 1$, there is an i such that no D_n-segment of $\mathcal{P}$ peaks at C_i and some D_n-segment P of $\mathcal{P}$ peaks at C_{i-1}. Let P' be the subpath of P with endpoints in C_i and internal vertices in $D_i \setminus C_i$, in particular P' contains $P \cap C_{i-1}$. Let Q be the path between the endpoints of P' such that $Q \subseteq C_i$ and v is not contained in a cycle formed by $Q \cup P$. Since no segment peaks at C_i, Lemma 1 implies that no D_n-segments of $\mathcal{P}$ contains an interior vertex of Q. Hence we can reroute P along Q rather than along P', contradicting that $\mathcal{P}$ is a cheap solution. □

Given a plane graph G and a vertex v we show how to check whether a particular vertex v satisfies the conditions of Theorem 5. We set $C_0 = \{v\}$ and given C_i we construct C_{i+1} by performing a depth first search from a neighbor u of a vertex in C_i, always chosing the rightmost edge leaving the vertex we are visiting. This search will either output an innermost cyclic walk (which then can be pruned to a cycle) around C_i or determine that no such walk exists. In the case that a cycle C_{i+1} is output, we check whether the cyclic walk contained a terminal s_i or t_i. If it did, it means that this terminal lies on C_{i+1} or in its interior. At this point (or when the search outputs that no cycle around C_i exists), we have determined that there are i tight concentric cycles around v with no terminal in the interior of C_i. If $i > (8k-6) \cdot 2^k + 4k - 2$ this implies that v satisfies the conditions of Theorem 5. Clearly this procedure can be implemented to run in linear time. This yields the following theorem.

Theorem 6. *Let G be a plane graph with k pairs of terminals, there is an $O(|V(G)^2|)$ time algorithm that outputs an induced subgraph G' of G such that $tw(G') \leq 72\sqrt{2}k^{\frac{3}{2}} \cdot 2^k$ and G is a YES-instance for* DPP *if and only if G' does.*

Proof. W.l.o.g. we assume that G is connected. For each vertex $v \in V(G)$ we check whether v satisfies the conditions of Theorem 5 in linear time. If it does, we delete v. The resulting graph is G'. By Theorem 5 G has a DPP solution if and only if G' does. Observe that no vertex of G' satisfies the conditions of Theorem 5 because deleting a vertex u from a graph cannot increase the number of concentric cycles around a vertex v. It remains to argue that $tw(G') \leq 72\sqrt{2}k^{\frac{3}{2}} \cdot 2^k$. Suppose not, we will prove that G' contains a vertex v and a sequence $C_0, \dots, C_n$ of concentric cycles bounding closed discs $D_0, \dots, D_n$, respectively, in the plane, such that no terminal of the DPP lies in D_n. This will contradict the construction of G'.

By [21],[8, Theorem 1], G contains a grid minor of size $\eta \times \eta$, where $\eta = 16\sqrt{2}k^{\frac{3}{2}} \cdot 2^k$. Since there are only $2k$ terminals, there is a $(16k \cdot 2^k \times 16k \cdot 2^k)$ subgrid embedded in a disc D which does not contain any terminals. Thus, in D there is a sequence of at least $(8k-1) \cdot 2^k$ concentric cycles. The vertex v in the innermost of these cycles satisfies the conditions of Theorem 5, contradicting the construction of G'. □

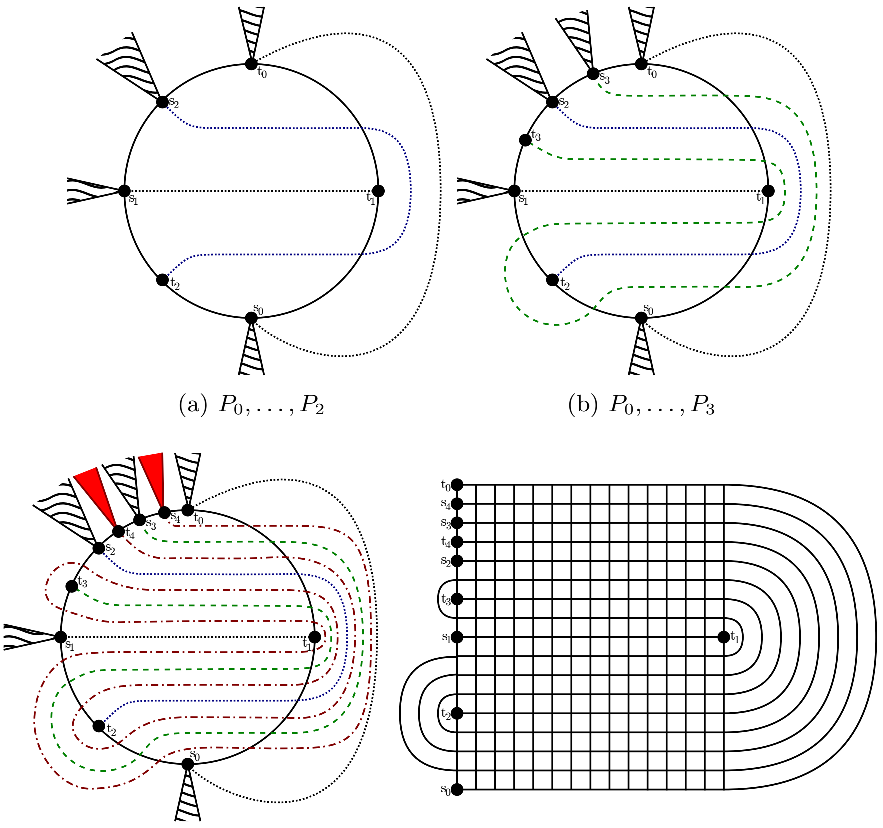
4 The Lower Bound

Let $H \subseteq G$ be a subgraph of the plane graph G. An *inner vertex* of H is a vertex that is not part of the boundary of H.

Definition 11 (Crossing). *Let $H \subseteq G$ be a subgraph of the plane graph G. We say that a path* crosses *the subgraph H if it contains an inner vertex of H and its endpoints are not inner vertices of H. For $k \in \mathbb{N}$ we say that a path $P = p_0, p_1, \dots, p_n$* crosses *H k times, if it can be split into k paths $P_0 = p_0, p_1, \dots, p_{i_1}, P_1 = p_{i_1}, p_{i_1+1}, \dots p_{i_2}, \dots, P_{k-1} = p_{i_{k-1}}, p_{i_{k-1}+1}, \dots, p_n$ with each $P_i, i = 0, \dots k-1$ crossing H. The parts of the P_i that do not lie outside of H are called crossings of H.*

Intuitively, we construct our example from a grid H of sufficient size. We add endpoints s_0 and t_0 on the boundary of the grid, mark the areas opposite to the grid as not part of the graph and connect s_0 to t_0 without crossing the grid. Now we continue to mark vertices by s_i and t_i in such a way that P_i has to cross H as often as possible (in order to avoid crossing $P_j, j < i$). Once s_i and t_i have been added we remove the area opposite to the grid from s_i from the graph. Figure 2(a) shows the situation after doing this for i up to 2. In this construction P_0 does not cross the grid at all, while P_1 crosses it once and P_{i+1} crosses it twice as often as P_i for $i > 0$: Let k_i be the number of times P_i crosses the grid. $k_0 = 0, k_1 = 1, k_{i+1} = 2k_i, k_i = 2^{i-1}, i > 0$. After the last P_i has been added, the areas opposite to the grid from both s_i and t_i are removed from the graph as seen in Figure 2(c).

Formally, to construct problem and graph with $k+1$ terminals, we use a $((2^k + 1) \times (2^k + 1))$ grid. Let the vertices on the left boundary of the grid be $n_0, \ldots, n_{2^k}$. Terminals are assigned as follows: t_0 is the topmost vertex on the left boundary on the grid, t_1 the middle vertices on the right boundary. For all other terminals: $s_i := n_{2^{k-i}}, t_i := n_{3 \cdot 2^{k-i}}$. Then add edges going around the t_i to the graph: For $i > 1, t_i = n_j$ add $n_{j-1}n_{j+1}, n_{j-2}n_{j+1}, \ldots, n_{j-2^{k-i}-1}n_{j+2^{k-i}-1}$, and on the right boundary of H do the analogue for t_1. See Figure 2(d) for a graph constructed this way.



(a) $P_0, \ldots, P_2$

(b) $P_0, \ldots, P_3$

(c) $P_0, \ldots, P_4$

(d) Actual graph with grid

Fig. 2. Construction of graph and solution

Theorem 7 ($\star$). *There is only one solution to the constructed DPP, all vertices of the graph lie on paths of the solution and the grid is crossed $2^k - 1$ times by such paths.*

In particular, H has no *irrelevant* vertex in the sense of [19].

Corollary 1. *There is a planar graph G with $k + 1$ pairs of terminals such that*

- *G contains a $((2^k + 1) \times (2^k + 1))$ grid as a subgraph,*
- *the disjoint paths problem on this input has a unique solution,*
- *the solution uses all vertices of G; in particular, no vertex of G is irrelevant.*

Vital linkages and tree-width We refer the reader to [2] for the definitions of *tree-width* and *path-width.* A linkage L in a graph G is a *vital linkage* in G, if $V(\bigcup L) = V(G)$ and there is no other linkage $L' \neq L$ in G with the same pattern as L.

Theorem 8 (Robertson and Seymour [20]). *There are functions f and g such that if G has a vital linkage with k components then G has tree-width at most $f(k)$ and path-width at most $g(k)$.*

Recall that the $(n \times n)$ grid has path-width n and tree-width n. Our example yields a lower bound for f and g:

Corollary 2. *Let f and g be as in Theorem 8. Then $2^{k-1}+1 \leq f(k)$ and $2^{k-1}+1 \leq g(k)$.*

Proof. Looking at the graph G and DPP constructed above the solution to the DPP is, due to its uniqueness, a vital linkage for the graph G. G contains a $((2^k + 1) \times (2^k + 1))$ grid as a minor. The tree-width of such a grid is $2^k + 1$, its path-width $2^k + 1$ [6]. Thus we get lower bounds $2^{k-1} + 1 \leq f(k), g(k)$ for the functions f and g. □

Acknowledgment. We thank Ken-ichi Kawarabayashi and Paul Wollan for providing details on the bounds in [11].

References

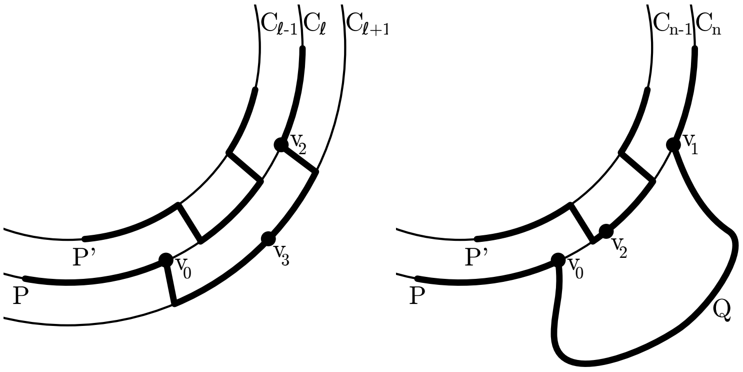
1. I. Adler, S. G. Kolliopoulos, and D. Thilikos. Planar disjoint paths completion. Submitted for publication, 2011.
2. H. L. Bodlaender. A tourist guide through treewidth. *Acta Cybernet.*, 11(1-2):1–21, 1993.
3. Anuj Dawar, Martin Grohe, and Stephan Kreutzer. Locally excluding a minor. In *LICS'07*, pages 270–279. IEEE Computer Society, 2007.
4. Anuj Dawar and Stephan Kreutzer. Domination problems in nowhere-dense classes. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2009)*, pages 157–168, 2009.
5. R. Diestel. *Graph Theory.* Springer, 2005.
6. John Ellis and Robert Warren. Lower Bounds on the Pathwidth of some Grid-like Graphs. *Discrete Applied Mathematics*, 156(5):545–555, 2008.
7. Petr A. Golovach, M. Kaminski, D. Paulusma, and D. M. Thilikos. Induced packing of odd cycles in a planar graph. In *Proceedings of the 20th International Symposium on Algorithms and Computation (ISAAC 2009)*, volume 5878 of *Lecture Notes in Comput. Sci.*, pages 514–523. Springer, Berlin, 2009.
8. Qian-Ping Gu and Hisao Tamaki. Improved bounds on the planar branchwidth with respect to the largest grid minor size. In *ISAAC (2)*, volume 6507 of *Lecture Notes in Computer Science*, pages 85–96, 2010.

9. Ken-ichi Kawarabayashi and Yusuke Kobayashi. The induced disjoint path problem. In *Proceedings of the 13th Conference on Integer Programming and Combinatorial Optimization (IPCO 2008)*, volume 5035 of *Lect. Notes Comp. Sc.*, pages 47–61. Springer, Berlin, 2008.
10. Ken-ichi Kawarabayashi and Bruce Reed. Odd cycle packing. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC 2010)*, pages 695–704, New York, NY, USA, 2010. ACM.
11. Ken-ichi Kawarabayashi and Paul Wollan. A shorter proof of the graph minor algorithm: the unique linkage theorem. In *Proc. of the 42nd annual ACM Symposium on Theory of Computing (STOC 2010)*, pages 687–694, 2010.
12. Yusuke Kobayashi and Ken-ichi Kawarabayashi. 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)*, pages 1146–1155. ACM-SIAM, 2009.
13. M. R. Kramer and J. van Leeuven. The complexity of wire-routing and finding minimum area layouts for arbitrary VLSI circuits. *Advances in Comp. Research*, 2:129–146, 1984.
14. Daniel Lokshtanov, Daniel Marx, and Saket Saurabh. Slightly superexponential parameterized problems. In *22st ACM–SIAM Symposium on Discrete Algorithms (SODA 2011)*, pages 760–776, 2011.
15. J. F. Lynch. The equivalence of theorem proving and the interconnection problem. *ACM SIGDA Newsletter*, 5:31–36, 1975.
16. Matthias Middendorf and Frank Pfeiffer. On the complexity of the disjoint paths problem. *Combinatorica*, 13(1):97–107, 1993.
17. Neil Robertson and P. D. Seymour. Graph minors. XIII. The disjoint paths problem. *J. Combin. Theory Ser. B*, 63(1):65–110, 1995.
18. Neil Robertson and Paul Seymour. Graph minors. XXI. Graphs with unique linkages. *J. Combin. Theory Ser. B*, 99(3):583–616, 2009.
19. Neil Robertson and Paul D. Seymour. Graph Minors. XXII. Irrelevant vertices in linkage problems. *Journal of Combinatorial Theory, Series B.* (to appear).
20. Neil Robertson and Paul D. Seymour. Graph minors. XXI. Graphs with unique linkages. *Journal of Combinatorial Theory, Series B*, 99(3):583–616, 2009.
21. Neil Robertson, Paul D. Seymour, and Robin Thomas. Quickly excluding a planar graph. *J. Comb. Theory Series B*, 62:323–348, 1994.
22. Alexander Schrijver. *Combinatorial optimization. Polyhedra and efficiency. Vol. A.* Springer-Verlag, Berlin, 2003.
23. Jens Vygen. NP-completeness of some edge-disjoint paths problems. *Discrete Appl. Math.*, 61(1):83–90, 1995.

Appendix

Proofs of the lemmata of Subsection 3.2

Proof (of Lemma 1). Assume that such a segment exists. Then there is an innermost one, i.e. one for which ℓ is minimal. Let P be such an innermost segment. Let v_0 and v_2 be the vertices nearest to v_1 in P, such that v_0 and v_2 are part of C_ℓ. P has vertices $\ldots, v_0, \ldots, v_3, \ldots, v_2, \ldots$, in this order, where v_3 is in $C_{\ell+1}$: P has to go through $C_{\ell+1}$ to reach C_j from C_ℓ. The part of P between v_0 and v_2 cannot be replaced by the part of C_ℓ that connects them, since $\mathcal{P}$ is cheap: It contains vertices and these vertices are used in the solution. This means that vertices in C_ℓ between v_0 and v_2 are part of some other segment P'. However to reach these vertices P' has to share a vertex with C_{l-1} according to Remark 1, since there is no terminal in D_n and P blocks all other ways to leave (Figure 3(a)). This contradicts the choice of P. □



(a) Lemma 1

(b) Lemma 2

Fig. 3. Constructions in proofs of Lemmata 1 and 2

Proof (of Lemma 2). Assume that there is no such terminal. Let Q be a handle of an innermost such path P, i.e., there is no other such path that contains a vertex of $I(Q)$. Let v_0 and v_1 be the endpoints of Q. Q cannot be replaced by the part of C_n that connects v_0 to v_1, since $\mathcal{P}$ is cheap: this part contains a vertex v_2, and v_2 is used in the solution. This means that v_2 is used by some segment P'. Since no terminal can be reached from v_2 without using a vertex in C_{n-1} according to Remark 1 (Q together with v_0 and v_1 block any way out). P' contains vertices in C_{n-1} before and after v_2, and v_2 is part of C_n (Figure 3(b)). However, this is not possible according to the previous lemma. □

Proof of the Lemma of Subsection 3.3

Proof (of the Tongue Taming Lemma (Lemma 4)). We represent each tongue tip by the outermost segment r that it contains, that is, r is the segment that has a vertex in common with C_n, but not with C_{n-1}. Throughout this proof, we call these outermost segments of tongue tips *representatives*, and we will denote them by lowercase letters even though they correspond to paths in the graph G.

Let P be the path to which r belongs, P breaks into segments, handles and at most two pieces which are neither handles nor segments, namely the subpaths of P which connect the endpoints of P with the disc D. We call these the parts of P. If a part directly preceding or succeeding r on P is a handle, we associate that handle with r. If a part directly preceding or succeeding r on P is not a handle and goes to a terminal, we associate the terminal with r. We denote the two handles associated with r as $Left(r)$ and $Right(r)$. Here $Left(r)$ is a handle that starts in the endpoint of r that we reach by starting at the center of r and moving counterclockwise along C_n. By abusing the term "endpoint" we will also refer to r as an endpoint of the two handles associated with it.

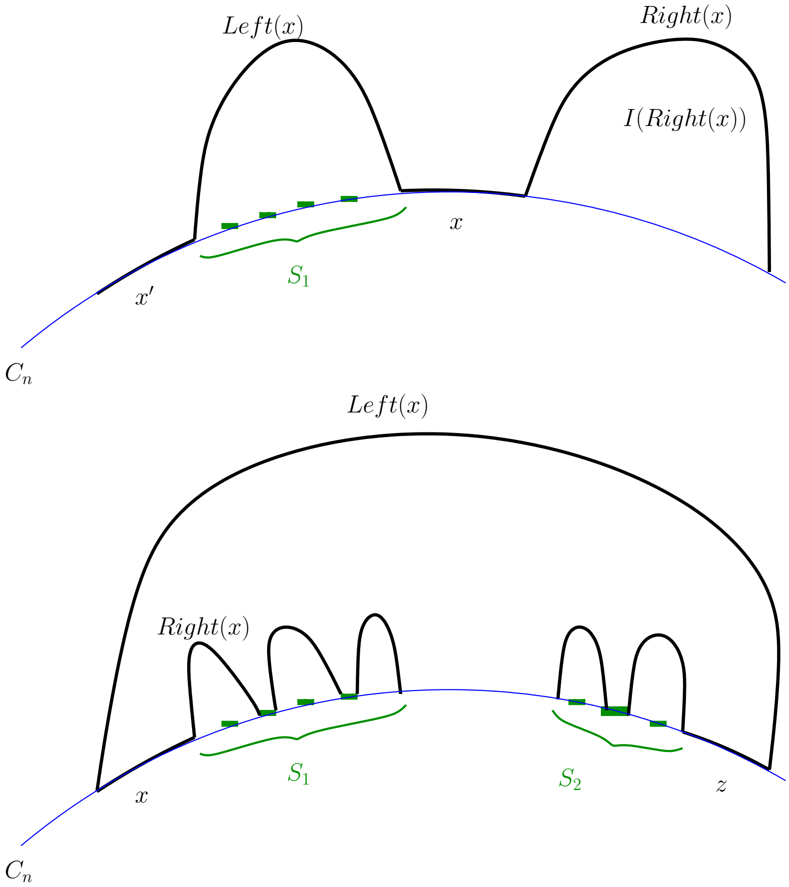


Fig. 4. An illustration of some cases from the proof of Lemma 4. On the top: Case 2a. On the bottom, an illustration of the Case 2b, with $S_2 \neq \emptyset$.

We prove the lemma by showing the existence of a matching in an auxiliary bipartite graph. More precisely we construct the following two graphs.

- A bipartite graph T with bipartitions A and B. The part A contains a vertex for each representative and B contains the set of terminals. We add an edge between a vertex denoting the representative r and a vertex denoting the terminal t if: t is associated with r, $t \in I(Left(r))$ or $t \in I(Right(r))$. We will abuse notation by referring with the same symbol both to vertices in T and to the object (terminal or representative) that the vertices correspond to.

- A graph H containing a vertex for each representative. Two representatives are adjacent if they *share* a handle, that is, if there is a handle associated with both representatives. Observe that the graph H is a set of disjoint paths.

We show that $|A| \leq |B| - 1 \leq 2k - 1$ by showing that in T, for each $S \subseteq A$, $|N(S)| > |S|$. Since $|B|$ is bounded by $2k$ we obtain the desired bound for $|A|$. Our proof is by contradiction. Specifically, suppose there is a set $S \subseteq A$ such that $|S| \geq |N(S)|$. Among all such sets we pick one with **smallest** $|N(S)|$, and among the ones with smallest $|N(S)|$ we pick one with **largest** $|S|$. We now define a partial order among representatives. We say that x is *below* y if $x \in \beta(Left(y)) \cup \beta(Right(y))$ and $y \notin \beta(Left(x)) \cup \beta(Right(x))$. Next we make a few observations about the set S.

1. The set $T[S \cup N(S)]$ is connected. Otherwise each component of $T[S \cup N(S)]$ contains at least one vertex from A and at least one vertex from B. Hence there is an $S' \subseteq S$ such that $T[S' \cup N(S')]$ is a connected component of $T[S \cup N(S)]$ and $|S'| \geq |N(S')|$ and $|N(S')| < |N(S)|$. This contradicts our choice of S.
2. Let Z be the set of all representatives that are below some representatives in S. Then by our choice of S we have that $Z \subseteq S$.
3. For any $r \in S$ and $t \in N(r)$ there must be an $r' \in S$ different from r such that $t \in N(r')$. Otherwise $N(S)$ contains t while $N(S \setminus \{r\})$ does not, and hence $|N(S \setminus \{r\})| \leq |S \setminus \{r\}|$, contradicting the choice of S. In other words, no representative in S has a private neighbor.

Let $S' = \{u \mid u \in S$ and for all $w \in S$, $u \neq w, u$ is not below $w\}$. Essentially no representative in S' is below any other representative in S. Observe that $N(S') = N(S)$ and hence $T[S' \cup N(S')]$ is connected. Notice that since no two vertices in S' are below each other, the only way two vertices in S can be adjacent to the same terminal is by sharing a handle. Thus $H[S']$ is a connected subpath of a path P^* of H. Observe that P^* may contain vertices that are not in S'.Let $P_{xy} = H[S']$ and we denote the endpoints of P_{xy} by x and y respectively.

At this point we distinguish between a few different cases based on the way the handles of x are embedded. See Fig. 4 for an illustration.

1. There is a terminal associated to x. Assume that there is a terminal t associated with x. Then t is a private neighbor of x, a contradiction.
2. There are two handles associated with x. We consider two subcases according to whether the handles go in "opposite direction" or in the "same direction". The *direction* of a handle is h defined as follows. The handle h goes *left* from r if $h = Left(r)$ and $r \notin \beta(h)$ or $h = Right(r)$ and $r \in \beta(h)$, and h goes *right* otherwise.
   (a) The handles of x go in opposite directions. Since x is an endpoint of P_{xy} and the handles of x go in opposite directions, at least one of the two handles associated with x must end in a tongue tip $x' \notin S$. Without loss of generality $Left(x)$ starts in x and ends in x'. Let $S_1 = S \cap \beta(Left(x))$ and $S_2 = S \cap \beta(Right(x))$. By Lemma 2 both $I(Left(x))$ and $I(Right(x))$ contain at least one terminal each. Furthermore, if $S_1 \neq \emptyset$ we have that $|N(S_1)| > |S_1|$ by the choice of S. But then $|N(S \setminus (S_1 \cup \{x\}))| \leq |S \setminus (S_1 \cup \{x\})|$. We need to argue that $S \setminus (S_1 \cup \{x\}) \neq \emptyset$, however since $I(Right(x))$ contains at least one terminal and this terminal is not a private neighbor of x it follows that $S \setminus (S_1 \cup \{x\}) \neq \emptyset$.

(b) Finally, we have the case that the handles of x go in the same direction. This case requires a bit more care. Without loss of generality we assume that both handles of x go to the right. Observe that the handle $Right(x)$ is drawn inside the handle $Left(x)$. Let z be the other endpoint of $Left(x)$. Now we define S^* to contain all vertices in S which are below both x and z, but not below any other vertex in S.

Let P_1 be the subpath of $H[S^*]$ that contains a neighbor of x in H. Note that x has a neighbor in S^* in H, namely the other endpoint of $Right(x)$. Let S_1 be the set of representatives in S^* which are on the path P_1 or below a representative on P_1. Similarly, if z has a neighbor in S^* in H then let P_2 be the subpath of $H[S^*]$ that contains a neighbor of z in H. Let S_2 be the set of representatives in S^* which are on the path P_2 or below a representative on P_2. If z has no neighbor in H from S^*, set P_2 to be an empty path and $S_2 = \emptyset$. Finally, set $S_3 = S^* \setminus (S_1 \cup S_2)$. Observe that $N(S_1)$, $N(S_2)$ and $N(S_3)$ are disjoint sets. Furthermore, $|N(S_1)| > |S_1|$, if $S_2 \neq \emptyset$ then $|N(S_2)| > |S_2|$ and if $S_3 \neq \emptyset$ then $|N(S_3)| > |S_3|$.

The path P_1, x, z, P_2 in Z corresponds to a path Q in G with enpoints on C_n and no vertices in the interior of D_n. Thus, by Lemma 3, there is a terminal $t \in I(Q)$. Since $t \in I(Q)$ it follows that $t \in I(Left(x))$ and $t \notin N(S_1 \cup S_2)$. Hence, if $S_3 = \emptyset$ we have that $|N(S \setminus (\{x, z\} \cup S_1))| \leq |S \setminus (\{x, z\} \cup S_1)|$.

On the other hand, if $S_3 \neq \emptyset$ we have that $|N(S \setminus (\{x, z\} \cup S_1 \cup S_3))| \leq |S \setminus (\{x, z\} \cup S_1 \cup S_3)|$. It remains to argue that $S \setminus (\{x, z\} \cup S_1 \cup S_3) \neq \emptyset$. Consider the handle W associated with z which does not end in x. By Lemma 2 there is a terminal t' in $I(W)$. If t' is adjacent to x in T, then the other endpoint of W is in S_2 and hence S_2 is non-empty. If, on the other hand, t' is not adjacent to x, there must be a representative $z' \in S$ different from z, and not in $I(Left(x))$ such that $t' \in N(z')$ because otherwise t' is a private neighbor of z. Since $z' \in S \setminus (\{x, z\} \cup S_1 \cup S_3)$ we have that $S \setminus (\{x, z\} \cup S_1 \cup S_3) \neq \emptyset$, concluding the proof.

This completes the proof. □

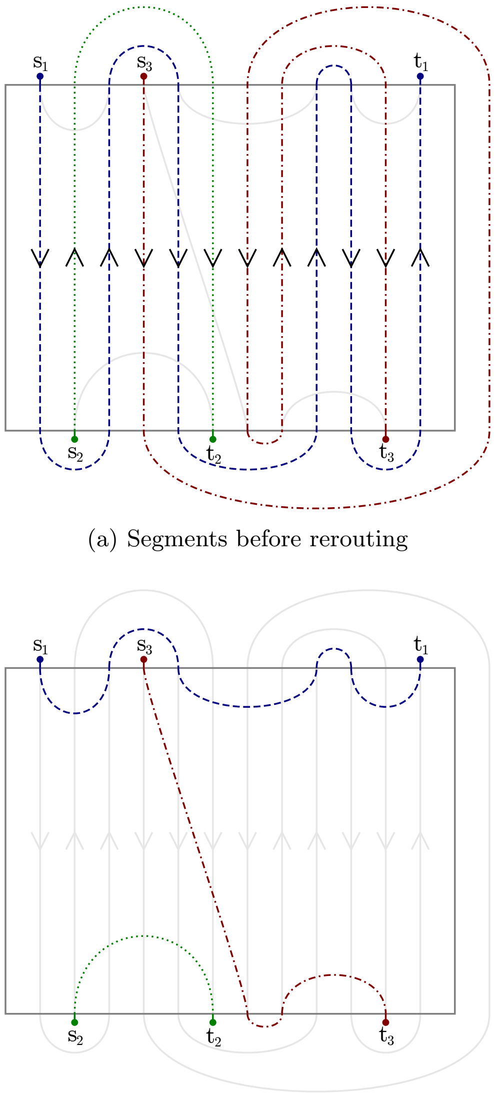
Proofs of the Lemmata of Subsection 3.4

Proof (of Lemma 5). Let $\Sigma = \{a_1, \ldots, a_k\}$, and let $w = w_1 \cdots w_n$ with $n > 2^k$. Define vectors $z_i \in \{0,1\}^k$ for $i \in \{1, \ldots, n\}$, and we let the jth entry of vector z_i be 0 if and only if letter a_j occurs an even number of times in the prefix $w_1 \cdots w_i$ of w and 1 otherwise. Since $n > 2^k$, there exist $i, i' \in \{1, \ldots, n\}$ with $i \neq i'$, such that $z_i = z_{i'}$. Then $y = w_i \cdots w_{i'}$ proves the lemma. □

Proof (of Lemma 6). Assume there is a segment type in O that contains more than 2^k edges. Choose an ordering of the edges of this segment type that corresponds to their order in the plane. Label each of these edges by a word in $\{1, \ldots, k\}$ according to the number i of the path $P_i \in \mathcal{P}$ that it belongs to. Hence we obtain a word w over the alphabet $\{1, \ldots, k\}$ with $|w| > 2^k$, and by Lemma 5 we know that w contains an infix y with $|y| \geq 2$, such that every letter occurring in y occurs an even number of times in y. Let $Y := \{e_1, e_2, \ldots, e_{|y|}\}$ be the set of edges of O that correspond to the letters in y. For every path $P_i \in \mathcal{P}$ we orient all edges of P_i from s_i to t_i. For a path $P_i \in \mathcal{P}$ with $E(P_i) \cap Y \neq \emptyset$, let $e^i_1, \ldots, e^i_{2n_i}$ be the (even number of) edges of Y appearing on P_i in this order when moving from s_i to t_i. We introduce a shortcut for P_i as follows:

For every odd number $j \in \{1, \ldots, 2n_i\}$, we replace the subpath of P_i from tail(e^i_j) to head(e^i_{j+1}) by a new edge f^i_j in the disc D. Having done this for all odd numbers

$j \in \{1, \ldots, 2n_i\}$, we obtain a new path P'_i from s_i to t_i that uses strictly less edges in D than P_i. Let O' be the graph obtained from O by modifying all paths $P \in \mathcal{P}$ with $E(P) \cap Y \neq \emptyset$ in this way (see Figure 4). Since $|y| \geq 2$ we have $|E(O')| < |E(O)|$. Ovbiously, the DPP has a solution in $(G \setminus O) \cup O'$.



(a) Segments before rerouting

(b) Segments after rerouting

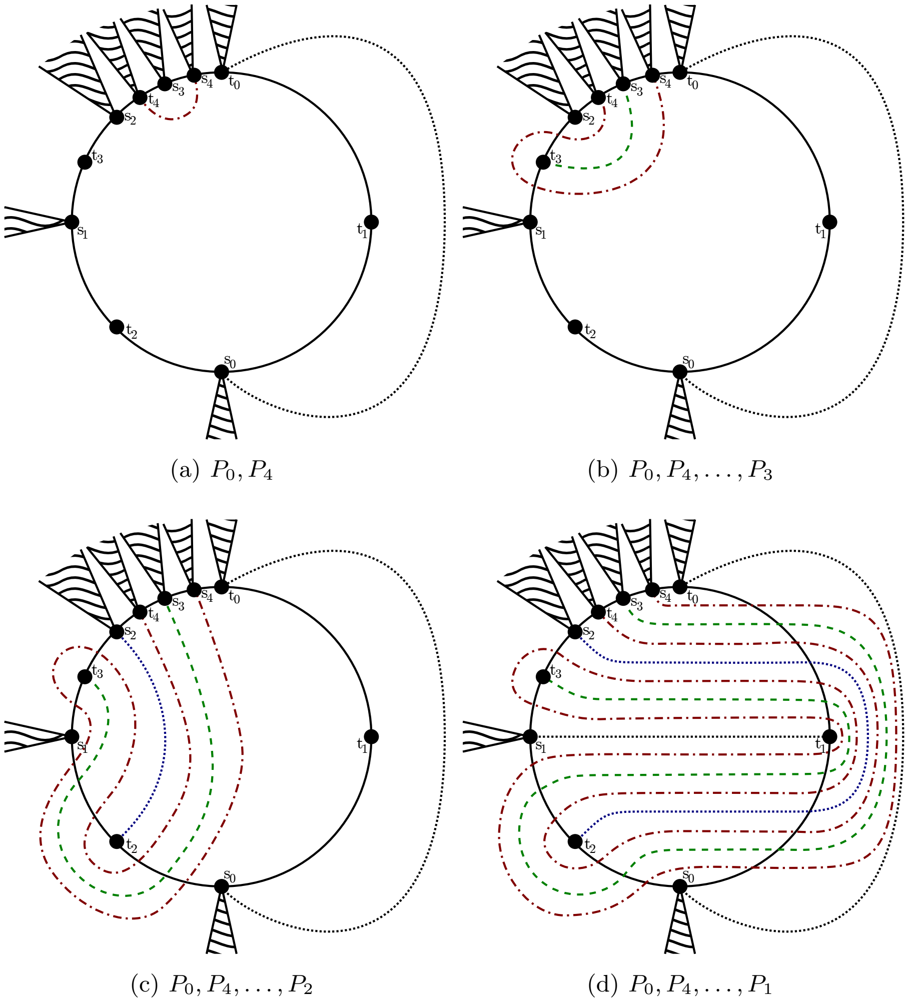
Fig. 5. Construction in proof of Lemma 6

Finally, we give a geometric argument showing that O' is outerplanar: Given the planar embedding of G as above, we transform D homeomorphically into a large rectangle R. The upper side of R lies on a straight line U, and the lower side lies on a straight line L. Let S be the straight line parallel to U and L, that divides D into halves. We assume that every edge in Y is represented by a vertical straight line segment from U to L. We now take a smaller rectangle $R_0 \subseteq R$ bounded by U and L, such that the only D_n-segments it contains are the segments in Y. Let ∂R_0 be the boundary of R_0.

For every path P_i using an edge of Y, let F^i_j denote the subpath of P_i from head(e^i_j) to tail(e^i_{j+1}) (for $j \in \{1, \ldots, 2n_i\}$, j odd). We replace F^i_j by a single edge c^i_j. Then the graph H with vertex set $V(O)$ and edge set

$$\{c^i_j \mid i \in \{1, \ldots, k\},\ E(P_i) \cap Y \neq \emptyset,\ j \in \{1, \ldots, 2_{n_i}\},\ j \text{ odd}\}$$

is outerplanar. Take the embedding of H obtained from the embedding of G using the



(a) P_0, P_4

(b) $P_0, P_4, \ldots, P_3$

(c) $P_0, P_4, \ldots, P_2$

(d) $P_0, P_4, \ldots, P_1$

Fig. 6. Optimality of solution

rectangle as above, where we embed the edges c^i_j along the subpaths F^i_j. We regard the embedded c^i_j as simple curves. Now we reflect the curves c^i_j inward at ∂R_0. We may assume that after reflection, none of the c_j^i crosses S or the drawing of an edge in $E(O) \setminus Y$. (If not, transform the drawing accordingly.) After reflection, the curves still have their endpoints on ∂R_0 and they are pairwise non-crossing. In a second step, we now reflect the curves at S. Let $(c_j^i)'$ denote the curve obtained from c_j^i by these two reflections. Now $(c_j^i)'$ connects $\text{tail}(e_j^i)$ to $\text{head}(e_{j+1}^i)$ (while c_j^i connects $\text{head}(e_j^i)$ to $\text{tail}(e_{j+1}^i)$). Due to symmetry, the $(c_j^i)'$ are pairwise non-crossing, and none of them crosses a drawing of an edge in $E(O) \setminus Y$. Hence the $(c_j^i)'$ together with the drawing of edges in $E(O) \setminus Y$ provide an outerplanar drawing of O' (where $(c_j^i)'$ is the drawing of f_j^i). Hence O' is outerplanar. This concludes the proof of the lemma. □

Proof of Theorem 7 from Section 4

Proof (of Theorem 7). To connect s_k to t_k we need to cross H at least once (Figure 6(a)). However, a solution in which P_K crosses H only once would block P_{k-1}. To connect s_{k-1} to t_{k-1} P_k has to be routed around t_{k-1}, which requires leaving and reentering H (Figure 6(b)). Thus inductively constructing a solution each $P_i, i > 0$ requires a crossing of H and doubles the number of crossings in each $P_j, j > i$. The solution uses all edges on the left side of H and uses all but one of the edges on the right side. Thus the only way to connect s_0 to t_0 is without crossing the grid. □