Simpler and faster algorithms for detours in planar digraphs [*]

Meike Hatzel[†]   Konrad Majewski[‡]   Michał Pilipczuk[§]   Marek Sokołowski[¶]

Abstract

In the DIRECTED DETOUR problem one is given a digraph G and a pair of vertices s and t, and the task is to decide whether there is a directed simple path from s to t in G whose length is larger than $\mathsf{dist}_G(s, t)$. The more general parameterized variant, DIRECTED LONG DETOUR, asks for a simple s-to-t path of length at least $\mathsf{dist}_G(s, t) + k$, for a given parameter k. Surprisingly, it is still unknown whether DIRECTED DETOUR is polynomial-time solvable on general digraphs. However, for planar digraphs, Wu and Wang [Networks, '15] proposed an $\mathcal{O}(n^3)$-time algorithm for DIRECTED DETOUR, while Fomin et al. [STACS 2022] gave a $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$-time fpt algorithm for DIRECTED LONG DETOUR. The algorithm of Wu and Wang relies on a nontrivial analysis of how short detours may look like in a plane embedding, while the algorithm of Fomin et al. is based on a reduction to the 3-DISJOINT PATHS problem on planar digraphs. This latter problem is solvable in polynomial time using the algebraic machinery of Schrijver [SIAM J. Comp., '94], but the degree of the obtained polynomial factor is huge.

In this paper we propose two simple algorithms: we show how to solve, in planar digraphs, DIRECTED DETOUR in time $\mathcal{O}(n^2)$ and DIRECTED LONG DETOUR in time $2^{\mathcal{O}(k)} \cdot n^4 \log n$. In both cases, the idea is to reduce to the 2-DISJOINT PATHS problem in a planar digraph, and to observe that the obtained instances of this problem have a certain topological structure that makes them amenable to a direct greedy strategy.

[†]National Institute of Informatics, Tokyo, Japan (meikehatzel@nii.ac.jp)

[‡]Institute of Informatics, University of Warsaw, Poland (k.majewski@mimuw.edu.pl)

[§]Institute of Informatics, University of Warsaw, Poland (michal.pilipczuk@mimuw.edu.pl)

[¶]Institute of Informatics, University of Warsaw, Poland (marek.sokolowski@mimuw.edu.pl)

1   Introduction

The complexity status of DIRECTED DETOUR is arguably one of the most tantalizing open questions within the area of graph algorithms. The problem asks to decide, for a given digraph G and two terminals s and t, whether there is a simple path in G from s to t that is not the shortest — has length strictly larger than $\mathsf{dist}_G(s,t)$. It is still unknown whether this problem is polynomial-time solvable on general digraphs. See the work of Fomin et al. [3] for a discussion of relevant literature.

   Given this state of the affairs, it is interesting to study DIRECTED DETOUR on restricted classes of digraphs, in hope for finding more positive results or useful insight. In this vein, a particularly well-motivated idea is to consider the class of planar digraphs. The reason for this is that in planar digraphs, the k-DISJOINT PATHS problem — decide the existence of disjoint directed paths linking given k pairs of terminals — is polynomial-time solvable for every fixed k [8], and even fixed-parameter tractable when parameterized by k [2]. This is not the case in general digraphs, where the problem is NP-hard already for $k = 2$ [5]. This can be used for DIRECTED DETOUR. Namely, Fomin et al. [3] showed that the more general parameterized version of the problem — DIRECTED LONG DETOUR, where we look for a simple s-to-t path of length at least $\mathsf{dist}_G(s,t)+k$, for a given parameter k — can be reduced (with a $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ multiplicative overhead in the complexity) to 3-DISJOINT PATHS, which can be solved in polynomial time in planar digraph using the algorithm of Schrijver [8]. This of course applies also to the basic DIRECTED DETOUR problem by setting $k = 1$, but even earlier Wu and Wang [9] gave a direct $\mathcal{O}(n^3)$-time algorithm for this case.

   While the reduction of Fomin et al. [3] is actually quite simple, the algorithm of Schrijver [8] for 3-DISJOINT PATHS in planar digraphs is not, as it relies on an involved algebraic framework. In particular, the degree of the polynomial bounding the running time is at least a two-digit number. On the other hand, the cubic algorithm of Wu and Wang [9] is also quite complicated and relies on an analysis of different planar configurations that may occur.

   In this work we propose two simple algorithms: one for DIRECTED DETOUR and one for DIRECTED LONG DETOUR, both in planar digraphs. These are summarized below.

Theorem 1.1. DIRECTED DETOUR *in planar digraphs can be solved in time* $\mathcal{O}(n^2)$.

Theorem 1.2. DIRECTED LONG DETOUR *in planar digraphs can be solved in time* $2^{\mathcal{O}(k)} \cdot n^4$ *by a Monte Carlo algorithm, or deterministically in time* $2^{\mathcal{O}(k)} \cdot n^4 \log n$.

   The main idea in the proof of Theorem 1.1 is to perform a reduction to the 2-DISJOINT PATHS problem, roughly similarly as in the work of Fomin et al. [3]. However, we observe that if this reduction is performed carefully, then one essentially obtains an instance of 2-DISJOINT PATHS where three out of four terminals lie on one face, say the outer face. Such instances can be solved very easily: one of the paths — the one with both terminals on the outer face — can be chosen greedily so that it leaves the maximum possible space for the other path. Then the other path can be found by a simple reachability check.

   For Theorem 1.2, as in Fomin et al. [3], we use the result of Bezáková et al. [1] that the EXACT DIRECTED LONG DETOUR problem — finding a shortest s-to-t path of length *exactly* $\mathsf{dist}_G(s,t)+k$ — can be solved in time $2^{\mathcal{O}(k)} \cdot n^2$, even on general digraphs. This allows us to assume, when solving DIRECTED LONG DETOUR, that there are no s-to-t paths of length between $\mathsf{dist}_G(s,t)+k$ and $\mathsf{dist}_G(s,t)+3k$. Having this assumption, the proof of Theorem 1.2 proceeds by expanding the basic idea behind Theorem 1.1 with color coding.

We believe that compared to [3, 9], our algorithms provide a simpler explanation for the tractability of DIRECTED (LONG) DETOUR in planar digraphs. While we do not expect that the gained insight will be directly applicable to the case of general digraphs, we hope that it might be a better starting point for generalizations to less restrictive topological graph classes, for instance to digraphs of bounded genus or digraphs whose underlying undirected graphs exclude a fixed minor.

2   Preliminaries

Graphs.   In this paper we consider planar directed graphs $G = (V(G), E(G))$. For the purposes of our problem, we can assume that the input graphs are simple, that is, they do not have self-loops or multiple arcs connecting two vertices in the same direction. Moreover, we assume that G is weakly connected, that is, the underlying undirected graph is connected. For convenience, we set $n = |V(G)|$ and $m = |E(G)|$. Since G is planar and simple, we have that $m \in \mathcal{O}(n)$.

Given an arc $e = uv \in E(G)$, we say that u is the *tail* of e and v is the *head* of e. Here, we consider e incident to both u and v. A sequence $v_1 v_2 \ldots v_k$ of vertices is called a *walk* in G if for each $i \in \{1, \ldots, k-1\}$, $v_i v_{i+1}$ is an arc in G. The vertex v_1 is called the *origin* of the walk, while v_k is its *destination*. If the vertices of a walk are pairwise different, the walk is a *path*. Given two walks W_1, W_2, if the destination of W_1 coincides with the origin of W_2, then we define $W_1 \circ W_2$ as the concatenation of both walks. The *length* of a path P, denoted $\mathsf{length}(P)$, is the number of arcs it contains. Given a path P and two of its vertices x and y, we denote by $P[x \to y]$ the subpath of P which starts at x, goes along the arcs of P, and ends in y.

Given a pair of vertices $u, v \in V(G)$, the *distance* from u to v in G is the length of the shortest u-to-v-path in G and is denoted by $\mathsf{dist}_G(u, v)$. If there is no directed path from u to v in G, we put $\mathsf{dist}_G(u, v) = +\infty$. If the graph G is known from the context, we may omit G from the notation and simply write $\mathsf{dist}(u, v)$.

Plane embeddings.   Given a planar graph G, its embedding into the plane is a mapping of the vertices of G to pairwise distinct points in the plane and the arcs of G to plane curves, so that:

- each arc $uv \in E(G)$ is mapped to a plane curve whose endpoints coincide with the images of u and v, and such that no other vertex of G is mapped to a point on the curve; and

- the images of the edges of G are pairwise internally disjoint.

A *plane graph* is a planar graph together with a fixed embedding of the graph into the plane. Such an embedding splits the plane in a number of regions, called *faces*. One of the faces is unbounded and called the *outer face*. Given a face F, its boundary ∂F can be described as a cyclic sequence of arcs bounding F (assuming G is connected). If ∂F is isomorphic to a simple polygon (ignoring the orientations of the arcs), we say that F is *simple*.

For algorithmic purposes, we represent planar embeddings combinatorially: each vertex v of the graph stores the anti-clockwise ordering of all arcs of G incident to v. Given a planar graph G, its combinatorial embedding can be computed in time linear with respect to the

Figure 1: (a) Three (v,w)-grounded paths. We have $P_1 \prec_{\text{lex}} P_2 \prec_{\text{lex}} P_3$, $P_1 \prec_{\text{top}} P_2$, $P_1 \prec_{\text{top}} P_3$, but $P_2 \not\prec_{\text{top}} P_3$. (b) A (v,w)-grounded path P and its left area $\Gamma_{\text{L}}(P)$ (filled in blue).

size of the graph [6]. Note that the combinatorial embedding uniquely determines the faces of the planar graph, however, it does not designate the outer face of the embedding. Hence, in our algorithms, we may elect any face to be the outer face.

Comparing paths in plane graphs. Let G be a plane graph and F be its outer face. Assume that F is simple. We say that a path P in G is *v-grounded (with respect to F)* if its origin v is a vertex of F. Moreover, we say that P is *(v,w)-grounded (with respect to F)* if both its origin v and its destination w are vertices of F. We now present two ways to compare (doubly) grounded paths in G. We note that the following definitions are standard.

Definition 2.1. *Assume that P_1 and P_2 are two different v-grounded paths with respect to a face F in a plane graph G. We say that P_1 is* lexicographically left of P_2 *(denoted $P_1 \prec_{\text{lex}} P_2$) if one of the following conditions holds:*

- *P_1 is a prefix of P_2; or*

- *consider a plane graph G_v created by taking G, together with its planar embedding, and adding a fresh vertex v' and a fresh arc $v'v$, mapped to the plane so that the image of v' is placed outside of F (i.e., so that v' is not enclosed by the boundary of F). Let P_1' and P_2' be paths in G_v, constructed by prepending v' to P_1 and P_2, respectively. Let q be the last vertex of the longest common prefix of P_1' and P_2', p be the vertex preceding q on the common prefix, and $r_1 \neq r_2$ be the vertices succeeding q on P_1' and P_2', respectively. Then, the arcs pq, qr_1 and qr_2 are embedded clockwise in this order around q.*

Intuitively, $P_1 \prec_{\text{lex}} P_2$ if at the end of the common prefix of P_1 and P_2, the path P_1 terminates or branches off left of P_2 (Figure 1(a)). Observe that $\prec_{\text{lex}}$ induces a linear order on all paths grounded at a vertex $v \in V(F)$. Thus, given any vertex $w \in V(G)$ reachable

3

from v in G, we define L_{vw}, the *lexicographically leftmost* path from v to w, as the unique minimal path from v to w with respect to $\prec_{\text{lex}}$. Given a combinatorial embedding of G, we can compute the lexicographically leftmost path from v to w in time $\mathcal{O}(n)$: it suffices to run a depth-first search of G in which each vertex considers all its neighbors in the left-to-right order.

We follow with a more restrictive way of comparing paths in G. Given a (v, w)-grounded path P, we define the *area left of P*, denoted $\Gamma_{\mathsf{L}}(P)$, as the region of the plane whose anti-clockwise boundary is the closed walk defined as the concatenation of P and the anti-clockwise segment of the boundary of F from w to v (Figure 1(b)). Note that the interior of $\Gamma_{\mathsf{L}}(P)$ may be disconnected if P internally intersects the boundary of F.

Definition 2.2. *Given two (v, w)-grounded paths P_1 and P_2, we say that P_1 is* topologically left of P_2 *(denoted $P_1 \prec_{\text{top}} P_2$) if $\Gamma_{\mathsf{L}}(P_1) \subsetneq \Gamma_{\mathsf{L}}(P_2)$.*

Note that if $P_1 \prec_{\text{top}} P_2$, then $P_1 \prec_{\text{lex}} P_2$. Thus, $\prec_{\text{top}}$ is a partial order on the set of all (v, w)-grounded paths, and $\prec_{\text{lex}}$ restricted to those paths is a linear extension of $\prec_{\text{top}}$. It is straightforward to observe that the lexicographically leftmost path from v to w — the minimum (v, w)-grounded path in $\prec_{\text{lex}}$ — is also the unique minimum (v, w)-grounded path in $\prec_{\text{top}}$. So if both v and w lie on F, we call L_{vw} simply the *leftmost path* from v to w.

Analogously, we define the *(lexicographically) rightmost path* from v to w, denoted R_{vw}, and the *area right of* a (v, w)-grounded path P, denoted $\Gamma_{\mathsf{R}}(P)$. Note that each (v, w)-grounded path P splits the graph into two parts: the area $\Gamma_{\mathsf{L}}(P)$ left of P and the area $\Gamma_{\mathsf{R}}(P)$ right of P, with both areas intersecting only at P.

Simplifying the outer face. Both the lexicographical and the topological comparisons of grounded paths require the outer face F of a plane graph G to be simple. However, this precondition might not be satisfied in the setting of our problem. To alleviate this issue, we show how to simplify the outer face by adding two directed arcs to the plane graph.

Let u and v be two vertices of F. The (u, v)-*simplification* of the outer face in G, denoted $\mathsf{Simplify}(G, u, v)$, is the digraph obtained from G by adding two arcs f_1, f_2, each from u to v. Both arcs are embedded in the outer face F of G, so that the outer face of $\mathsf{Simplify}(G, u, v)$ is bounded by f_1 and f_2 (see Figure 2). Thus, the outer face of $\mathsf{Simplify}(G, u, v)$ is simple and contains two vertices, u and v. We remark that $\mathsf{Simplify}(G, u, v)$ is not necessarily a simple digraph; however, this does not pose a problem since after the simplification, the number of edges of the graph is still linearly bounded in the number of vertices of the graph.

Cutting arcs of plane graphs. We finally describe an operation on plane graphs that introduces new vertices to the outer face F of the embedding. Assume that F is simple. Take two vertices $u \in V(F)$, $v \notin V(F)$, both incident to an arc $e \in E(G)$. An operation of *cutting the graph along e* produces a new graph G_e which is the result of the following process (see Figure 3):

1. Enumerate all arcs incident to u in anti-clockwise order: $e_1, e_2, \ldots, e_k$, where $e_1, e_k \in E(F)$.

2. Let $\ell \in \{2, 3, \ldots, k-1\}$ be such that $e_\ell = e$.

3. Remove u from the graph, along with all arcs incident to u.

Figure 2: An example plane digraph (left) and its (u, v)-simplification (right).

(a)   (b)   (c)

Figure 3: (a) A directed plane graph G. (b) The result of cutting G along e. (c) The result of cutting G along P.

4. Introduce two new vertices u_1, u_2 to the graph. For each $i \in \{1, 2, \ldots, \ell\}$, add to the graph a new arc e_i^1 which is obtained from the arc e_i by replacing the endpoint u with u_1. Similarly, for each $i \in \{\ell, \ell+1, \ldots, k\}$, add to the graph a new arc e_i^2 which is obtained from the arc e_i by replacing the endpoint u with u_2.

Intuitively, G_e is produced by drawing the graph G on a piece of paper and cutting the piece of paper along e. Note that after this operation, the resulting graph is still planar (Figure 3). Moreover, the outer face F_e of G_e is simple, and $V(F) \setminus V(F_e) = \{u\}$ and $V(F_e) \setminus V(F) = \{u_1, v, u_2\}$. That is, v now lies on the outer face of the new graph.

We can generalize this procedure to paths instead of just edges. Assume that $P = v_1 v_2 \ldots v_k$ is a v_1-grounded path which is disjoint with $V(F)$ except for v_1. Then, cutting the graph along P entails cutting the graph along the arcs $v_1 v_2, v_2 v_3, \ldots, v_{k-1} v_k$, in this order. This way we obtain a plane graph where the destination of the path lies on the outer face. See Figure 3 for an illustration.

It is immediate that given the combinatorial embedding of a plane graph G, one can compute a plane embedding of the graph obtained by cutting G along a path P in time linear with respect to the size of the graph. Thus we again obtain a plane graph.

5

3 Directed Detour

In this section we prove Theorem 1.1.

Let G be a plane digraph, and let $s, t \in V(G)$ be a pair of vertices. Recall that we assume G to be weakly connected and we fix the plane embedding of G. Therefore, from now on for simplicity we identify features in G (vertices, edges, paths, etc.) with their images under the embedding. Our goal is to decide whether there is an s-to-t path in G of length at least $\mathsf{dist}_G(s, t) + 1$. We are going to reduce this question to solving a set of instances of the 2-Disjoint Paths problem. The reduction is similar to the algorithm of Fomin et al. [3] for Directed Long Detour.

First, we run breadth-first search (BFS) starting from s in G. Let L_i denote the i-th layer of this BFS, that is, $L_i = \{v \in V(G) \mid \mathsf{dist}_G(s, v) = i\}$ for $i \in \{0, 1, \ldots, n\}$.

Suppose now that the instance (G, s, t) is a yes-instance, and let P be an s-to-t path witnessing this fact. Since P is a non-shortest path from s to t, there is an index $p \in \{0, 1, \ldots, n\}$ such that the layer L_p contains at least two vertices of P. Let us choose p to be the smallest such an index. Let $G_{\geqslant p}$ be the plane digraph obtained from G by removing all the vertices in $L_0 \cup L_1 \cup \ldots \cup L_{p-1}$. We may assume that $G_{\geqslant p}$ is weakly connected, for otherwise we discard all of its weakly connected components that do not contain the vertex t.

Claim 1. *All vertices of L_p lie on one face of $G_{\geqslant p}$.*

Proof. Consider any vertex v of L_p and any shortest path Q from s to v in G. Observe that all vertices of Q except for v lie in layers $L_0, L_1, \ldots, L_{p-1}$, hence they are removed when constructing $G_{\geqslant p}$ from G. We conclude that v lies on the boundary of the (unique) face of $G_{\geqslant p}$ that contains s. $\qquad\square$

Denote by x and y, respectively, the first and the second vertex on the path P that lie in the layer L_p. In our algorithm we iterate over all possible choices for the vertex y. Note that the choice of y determines the value of p, because $y \in L_p$. Observe that if we guess the vertex y correctly, then in order to find a non-shortest path from s to t it is enough to find a vertex $x \in L_p$ ($x \neq y$) and three paths P_{start}, P_{middle} and P_{end} such that

- P_{start} is a shortest path from s to x in G; and

- P_{middle}, P_{end} are two internally vertex-disjoint paths in $G_{\geqslant p}$ going respectively from x to y and from y to t.

Note that y might be equal to t and then P_{end} is a trivial path only consisting of the single vertex t.

On one hand, vertices x and y divide P into subpaths P_{start}, P_{middle}, and P_{end} satisfying the properties stated above. On the other hand, if we find a vertex x and paths $P_{\mathrm{start}}, P_{\mathrm{middle}}, P_{\mathrm{end}}$ satisfying the above, then their concatenation forms a valid solution. This is because the path P_{start} goes consecutively through layers $L_0, L_1, \ldots, L_p$ due to being a shortest path, and thus it is internally vertex-disjoint from P_{middle} and P_{end}. Also, the concatenation of $P_{\mathrm{start}}, P_{\mathrm{middle}}, P_{\mathrm{end}}$ is a non-shortest path from s to t due to containing at least two different vertices from layer L_p.

It would be natural to also iterate through all possible choices for x, but for the sake of optimizing the running time we do the following instead. Construct a digraph H from the digraph $G_{\geqslant p}$ by adding to it a vertex x_{super} together with arcs $x_{\mathrm{super}}u$ for all $u \in L_p \setminus \{y\}$.

Claim 2. *H is a planar digraph and its implicit embedding can be obtained from the implicit embedding of $G_{\geqslant p}$ in linear time.*

Proof. It suffices to extend the implicit embedding of $G_{\geqslant p}$ by embedding x_{super} anywhere in the unique face whose boundary contains all vertices of L_p (which exists by Claim 1) and draw arcs connecting x_{super} with vertices of $L_p \setminus \{y\}$ within this face. It is straightforward to see that this can be done in the setting of implicit embeddings in linear time. □

Observe that pairs (P_1, P_2) of internally vertex-disjoint paths going respectively from x_{super} to y in H and from y to t in H correspond one-to-one to pairs $(P_{\text{middle}}, P_{\text{end}})$ of internally vertex-disjoint paths going respectively from some vertex $x \in L_p$ $(x \neq y)$ to y in $G_{\geqslant p}$ and from y to t in $G_{\geqslant p}$. We may additionally assume that the face of H containing both x_{super} and y is the outer face of H. This way we reduced our original problem to a set of at most n instances (one instance per each choice of the vertex y) of the following problem:

> PROBLEM A
>
> **Input:** A plane digraph H and three vertices $x, y, t \in V(H)$, where x and y lie on the outer face of H.
>
> **Question:** Are there two internally vertex-disjoint paths P_1 and P_2 such that P_1 is an x-to-y path and P_2 is a y-to-t path in H?

So to conclude Theorem 1.1 it is enough to show the following lemma.

Lemma 3.1. PROBLEM A *can be solved in time $\mathcal{O}(n)$.*

Proof. Let $H^\star$ be a (y, x)-simplification of the outer face of H. Naturally, the outer face of $H^\star$ is simple. Moreover, $(H^\star, x, y, t)$ is a yes-instance if and only if (H, x, y, t) is a yes-instance: $H^\star$ is exactly the graph H with two additional yx arcs, which plainly cannot be a part of the sought solution.

Suppose now that $(H^\star, x, y, t)$ is a yes-instance, and let (P_1, P_2) be a pair of paths in $H^\star$ that form a solution to PROBLEM A on $H^\star$. P_1 connects x to y which lie on the outer face F_O of $H^\star$ (which is a simple cycle), thus it is (x, y)-grounded with respect to F_O. Therefore, it splits the whole digraph H into two parts, the area $\Gamma_{\mathsf{L}}(P_1)$ left of P_1 and the area $\Gamma_{\mathsf{R}}(P_1)$ right of P_1, with both areas intersecting only at P_1. Since P_2 is internally disjoint with P_1, it must be entirely contained within $\Gamma_{\mathsf{L}}(P_1)$ or $\Gamma_{\mathsf{R}}(P_1)$; without loss of generality assume that P_2 is contained in the latter. Then, let $L_{x,y}$ be the leftmost path from x to y in $H^\star$.

Claim 3. $(L_{x,y}, P_2)$ *is also a valid solution to* PROBLEM A.

Proof. By the definition of the leftmost path we know that $\Gamma_{\mathsf{L}}(L_{x,y}) \subseteq \Gamma_{\mathsf{L}}(P_1)$, that is, $L_{x,y}$ cannot use any vertex lying on the right side of P_1. Since all vertices of P_2 (apart from y) lie on the right side of P_1 we conclude the claim. □

An analogous argument shows that if P_2 is contained in $\Gamma_{\mathsf{L}}(P_1)$, then $(R_{x,y}, P_2)$ is a valid solution to PROBLEM A, where $R_{x,y}$ is the rightmost path from x to y in $H^\star$. By the observation above, we know that in order to verify whether $(H^\star, x, y, t)$ is a yes-instance it is enough to find $L_{x,y}$ and $R_{x,y}$ as candidates for P_1, and to check for each candidate whether the vertex t is reachable from y in $H^\star - (V(P_1) \setminus \{y\})$. All these checks can be done in time $\mathcal{O}(n)$ by running a proper depth-first search algorithm. □

7

4   Directed Long Detour

In this section we prove Theorem 1.2.

We are given a plane digraph G, two vertices $s, t \in V(G)$ and an integer $k \in \mathbb{N}$. As in Section 3, we assume that G is weakly connected, and we fix the plane embedding of G. We need to decide whether there is an s-to-t path in G of length at least $\mathsf{dist}_G(s, t) + k$. We are going to reduce this question to solving a set of instances of the 2-DISJOINT PATHS problem (with some additional requirements).

We begin with running the algorithm of Bezáková et al. [1] to check whether there is an s-to-t path of length $\mathsf{dist}_G(s, t) + l$ for some $l \in \{k, k+1, \ldots, 3k-1\}$. From now on, we assume there is no such s-to-t path.

As in Section 3, we run a BFS starting from s in G and set $L_i = \{v \in V(G) \mid \mathsf{dist}_G(s, v) = i\}$ for $i = 0, 1, \ldots, n$. For $i = 1, 2, \ldots, n$, let $G_{\geqslant i}$ be the plane digraph obtained from G by removing all the vertices in $L_0 \cup \ldots \cup L_{i-1}$. Again, without loss of generality, we assume that $G_{\geqslant i}$ is weakly connected.

Now, we iterate through all values for $p = 1, 2, \ldots, n$, and through all choices for $x, y \in L_p$, where $x \neq y$ (there are at most n^2 choices for (p, x, y)). Recall from Section 3 (Section 1) that both x and y lie on the outer face of $G_{\geqslant p}$. Next, let $G_{\geqslant p}^{x,y}$ be the (y, x)-simplification of the outer face of $G_{\geqslant p}$. We are now going to search for three paths P_{start}, P_{middle} and P_{end} satisfying the conditions:

(a) P_{start} is a shortest s-to-x path in G;

(b) P_{middle} is an (x, y)-grounded path in $G_{\geqslant p}^{x,y}$ of length at least $2k$;

(c) P_{end} is a y-to-t path in $G_{\geqslant p}^{x,y}$;

(d) paths P_{middle} and P_{end} are internally vertex-disjoint; and

(e) if $P_{\mathrm{end}} \subseteq \Gamma_{\mathsf{R}}(P_{\mathrm{middle}})$, then there is no (x, y)-grounded path P'_{middle} in $G_{\geqslant p}^{x,y}$ of length at least k such that $P'_{\mathrm{middle}} \prec_{\mathrm{top}} P_{\mathrm{middle}}$; and if $P_{\mathrm{end}} \subseteq \Gamma_{\mathsf{L}}(P_{\mathrm{middle}})$, then there is no such path with $P_{\mathrm{middle}} \prec_{\mathrm{top}} P'_{\mathrm{middle}}$.

We remark that the two yx arcs added in the process of (y, x)-simplification of $G_{\geqslant p}$ cannot be a part of any of the paths P_{start}, P_{middle}, P_{end} and hence their existence can be safely ignored in the following series of claims.

First, we show that the procedure described above is actually equivalent to solving the DIRECTED LONG DETOUR problem on the instance (G, s, t, k).

Claim 4. *Assume that the paths P_{start}, P_{middle} and P_{end} satisfy the properties $(a) - (d)$ above. Then the concatenation*

$$P = P_{start} \circ P_{middle} \circ P_{end}$$

forms a valid solution for the DIRECTED LONG DETOUR problem.

Proof. First, P is a simple s-to-t path. Indeed, P_{middle} and P_{end} are internally vertex-disjoint by property (d). Moreover, since the path P_{start} is a shortest s-to-x path in G, it does not use any vertex of $L_p \cup \ldots \cup L_n = V(G_{\geqslant p})$ apart from x, and thus P_{start} is internally vertex-disjoint from P_{middle} and P_{end} as well.

By property (b) the length of P is at least

$$\mathsf{dist}_G(s,x) + 2k + \mathsf{dist}_G(y,t) = \mathsf{dist}_G(s,y) + 2k + \mathsf{dist}_G(y,t) \geqslant \mathsf{dist}_G(s,t) + 2k,$$

which finishes the proof. $\qquad\square$

Claim 5. *If (G,s,t,k) is a yes-instance, then there exist an integer $p \in \mathbb{N}$ and vertices $x, y \in L_p$ for which there are paths P_{start}, P_{middle} and P_{end} satisfying the properties $(a) - (e)$.*

Proof. Assume that (G,s,t,k) is a yes-instance. Let P be an s-to-t path witnessing this fact. If there are many such paths, we choose P to be a shortest one. As we established that there is no s-to-t path of length $\mathsf{dist}_G(s,t) + l$ for any $l \in \{k, k+1, \ldots, 3k-1\}$, we may assume that the path P is of length at least $\mathsf{dist}_G(s,t) + 3k$. Since P is a non-shortest s-to-t path, we may define $p \in \{1,2,\ldots,n\}$ to be the smallest index such that L_p contains at least two vertices of the path P. Let x and y be, respectively, the first and the second vertex on the path P that lie in the layer L_p.

By definition of p, the subpath $P[s \to x]$ is a shortest s-to-x path, and thus we may set $P_{\mathrm{start}} \coloneqq P[s \to x]$. We also set $P_{\mathrm{end}} \coloneqq P[y \to t]$. We will choose P_{middle} later in the course of the proof.

Let us observe that the length of the subpath $P[x \to y]$ is at least $2k$: otherwise, we consider the path P' being the concatenation of a shortest s-to-y path in G and the subpath $P[y \to t]$. As in Claim 4, we argue that P' is a simple s-to-t path in G, and the length of P' is

$$
\begin{aligned}
\mathsf{dist}_G(s,y) + \mathsf{length}(P[y \to t]) &= \mathsf{dist}_G(s,x) + \mathsf{length}(P[y \to t]) \\
&= \mathsf{length}(P) - \mathsf{length}(P[x \to y]) \\
&> (\mathsf{dist}_G(s,t) + 3k) - 2k = \mathsf{dist}_G(s,t) + k.
\end{aligned}
$$

Hence, P' is also a valid solution for our instance (G,s,t,k) that is shorter than P, which is a contradiction.

Since $P[x \to y]$ is an (x,y)-grounded path in $G_{\geqslant p}^{x,y}$ and is internally disjoint from $P[y \to t]$, we may assume that $P[y \to t] \subseteq \Gamma_{\mathsf{R}}(P[x \to y])$ in $G_{\geqslant p}^{x,y}$; the case $P[y \to t] \subseteq \Gamma_{\mathsf{L}}(P[x \to y])$ is symmetric.

Observe now, that if we set $P_{\mathrm{middle}} \coloneqq P[x \to y]$, then the properties $(a) - (d)$ are satisfied. If the condition (e) holds as well, then we are done. Otherwise, we define P_{middle} as a minimal, with respect to $\prec_{\mathrm{top}}$, (x,y)-grounded path in $G_{\geqslant p}^{x,y}$ of length at least $2k$. Then, $P_{\mathrm{middle}} \prec_{\mathrm{top}} P[x \to y]$, and consequently P_{middle} is internally vertex disjoint from $P_{\mathrm{end}} = P[y \to t]$ as we have $P[y \to t] \subseteq \Gamma_{\mathsf{R}}(P[x \to y])$.

It remains to show that the pair $(P_{\mathrm{middle}}, P_{\mathrm{end}})$ satisfies the property (e). By the definition of P_{middle} we know there is no (x,y)-grounded path of length at least $2k$ which is topologically left of P_{middle}. Suppose now that there exists an (x,y)-grounded path P'_{middle} in $G_{\geqslant p}^{x,y}$ such that $P'_{\mathrm{middle}} \prec_{\mathrm{top}} P_{\mathrm{middle}}$ and $\mathsf{length}(P'_{\mathrm{middle}}) \in [k, 2k-1]$. Consider the following walk P':

$$P' = P[s \to x] \circ P'_{\mathrm{middle}} \circ P[y \to t].$$

P' is a simple path as in $G_{\geqslant p}$ we have $P'_{\mathrm{middle}} \prec_{\mathrm{top}} P_{\mathrm{middle}} \prec_{\mathrm{top}} P[x \to y]$ and $P[y \to t] \subseteq \Gamma_{\mathsf{R}}(P[x \to y])$, therefore P'_{middle} is internally vertex-disjoint from both $P[s \to x]$ and $P[y \to t]$. Moreover, the length of P' is at least

$$\mathsf{dist}_G(s,x) + \mathsf{length}(P'_{\mathrm{middle}}) + \mathsf{dist}_G(y,t) =$$
$$\mathsf{dist}_G(s,y) + \mathsf{dist}_G(y,t) + \mathsf{length}(P'_{\mathrm{middle}}) \geqslant \mathsf{dist}_G(s,t) + \mathsf{length}(P'_{\mathrm{middle}}) \geqslant \mathsf{dist}_G(s,t) + k.$$

Consequently, P' is a valid solution for the instance (G, s, t, k). Finally, since $\mathsf{length}(P'_{\mathrm{middle}}) < 2k \leqslant \mathsf{length}(P[x \to y])$, the path P' is shorter than the path P which is a contradiction.

<div style="text-align: right">□</div>

It remains to show how we can find paths P_{start}, P_{middle} and P_{end} satisfying the above-mentioned conditions. Let us assume that we guess the values of p, x and y correctly. By Claim 4, we know that P_{start} can be set to any shortest s-to-x path, and we see that in order to find the desired paths P_{middle} and P_{end} we can restrict ourselves to the digraph $G^{x,y}_{\geqslant p}$. Let us call a pair of paths $(P_{\mathrm{middle}}, P_{\mathrm{end}})$ *special* for $(G^{x,y}_{\geqslant p}, x, y, t, k)$ if they satisfy the properties $(b) - (e)$ stated above. This reduces our instance of the DIRECTED LONG DETOUR problem to solving the set of at most n^2 instances (one for each choice of x and y) of the following problem (with $H := G^{x,y}_{\geqslant p}$).

PROBLEM B

Input: A plane digraph H, three vertices $x, y, t \in V(H)$, and an integer $k \in \mathbb{N}$, where x and y lie on the outer face of H. The outer face of H is simple and contains only the vertices x and y.

Question: Does there exist a special pair of paths (P_1, P_2) for (H, x, y, t, k)?

To finish the proof of Claim 1.2 it is enough to show the following lemma.

Lemma 4.1. PROBLEM B *can be solved in time $2^{\mathcal{O}(k)} \cdot n^2$ by a Monte Carlo algorithm, or deterministically in time $2^{\mathcal{O}(k)} \cdot n^2 \log n$.*

Proof. We use a variant of the standard color coding technique [4]. Let us color independently every vertex of H with one of two colors, say green and blue, each with probability $\frac{1}{2}$.

Assume for now that (H, x, y, t, k) is a yes-instance, and let (P_1, P_2) be a special pair of paths for our instance. As per our previous considerations, we know that $P_2 \subseteq \Gamma_{\mathsf{L}}(P_1)$ or $P_2 \subseteq \Gamma_{\mathsf{R}}(P_1)$. Without loss of generality assume that $P_2 \subseteq \Gamma_{\mathsf{R}}(P_1)$. Recall that $\mathsf{length}(P_1) \geqslant 2k$. Therefore, we may define x' to be the k-th vertex on P_1 and y' to be the k-th vertex from the end of P_1. Then, $\mathsf{length}(P_1[x \to x']) = \mathsf{length}(P_1[y' \to y]) = k - 1$ and the paths $P_1[x \to x']$ and $P_1[y' \to y]$ are vertex-disjoint.

Suppose that the colors are assigned in such a way that all vertices of the path $P_1[x \to x']$ are green, and all vertices of the path $P_1[y' \to y]$ are blue. The probability of such an event occurring is $(1/2)^{2k}$. Let $\mathcal{G}$ be the set of green vertices of H in the random coloring. Note that we assume that $V(P_1[x \to x']) \subseteq \mathcal{G}$.

Claim 6. $P_1[x \to x']$ *is the lexicographically leftmost x-to-x' path in H entirely contained in $\mathcal{G}$.*

Proof. Suppose that there is an x-to-x' path Q in H, entirely contained in $\mathcal{G}$, such that $Q \prec_{\mathrm{lex}} P_1[x \to x']$. Let $q \in V(H)$ be the vertex for which path $P_1[x \to q]$ is the longest common prefix of P_1 and Q. Also, let r be the first vertex on $Q[q \to x']$, not including q, such that $r \in V(P_1)$. Here, r is well-defined as the vertex x' belongs to both $Q[q \to x']$ and P_1. Since r lies on Q, we necessarily have that r is green. Thus, $r \notin V(P_1[y' \to y])$ since $P_1[y' \to y]$ consists only of blue vertices. Hence, the following walk P_1':

$$P_1' = Q[x \to r] \circ P_1[r \to y]$$

<div style="text-align: center">10</div>

is an (x, y)-grounded path in H of length at least

$$\mathsf{length}\big(P_1[y' \to y])\big) + 1 = k.$$

Recall that $Q \prec_{\mathrm{lex}} P_1$ in H. By the definition of r we observe that $Q[x \to r] \prec_{\mathrm{top}} P_1[x \to r]$. Hence, $P_1' \prec_{\mathrm{top}} P_1$. We conclude that the path P_1' has length at least k and is left of P_1, which contradicts the property (e) of the pair (P_1, P_2). □

Note that the path $P_1[x \to x']$ is disjoint from the set of vertices on the outer face of H, apart from its first vertex x. This allows us to define H^{cut} as the plane digraph obtained from H by cutting along the path $P_1[x \to x']$. Next, let $Z \subseteq V(H^{\mathrm{cut}})$ be the set of new vertices introduced to H' by this operation. Let H' be a (y, x')-simplification of the outer face of H^{cut}.

Claim 7. $P_1[x' \to y]$ *is the leftmost path in H' among all the paths between x' and y which do not contain any vertex of Z.*

Proof. Suppose that there is an x'-to-y path Q in H' which avoids Z and satisfies $Q \prec_{\mathrm{lex}} P_1[x' \to y]$. Let $q \in V(H)$ be the vertex of H such that $Q[x' \to q]$ is the longest common prefix of Q and $P_1[x' \to y]$. Let r be the first vertex on $Q[q \to y]$ (not including q) such that $r \in V(P_1)$.

Analogously as in the proof of Claim 6, we show that the walk P_1' defined in H as follows:

$$P_1' = P_1[x \to x'] \circ Q[x' \to r] \circ P_1[r \to y]$$

is a simple path of length at least k that lies lexicographically left of P_1 in H. This leads to a contradiction to property (e) of (P_1, P_2). □

The observations above lead to an algorithm for PROBLEM B. First, consider the case where in the solution (P_1, P_2), we have that $P_2 \subseteq \Gamma_{\mathsf{R}}(P_1)$. After the random assignment of colors to vertices of H, we iterate through all vertices $u \in V(H) \setminus \{x, y, t\}$ as candidates for the vertex x'. Then, we find the leftmost path S from x to u in H whose all vertices are green. Next, we construct H^{cut} and H', and find the leftmost path T from u to y in H' that avoids any new vertices introduced to H' by the split. Finally, we set $P_1 = S \circ T$. Then it only remains to verify whether there exists any path P_2 from y to t in H that is internally disjoint with P_1. All these checks can be done by running proper depth-first searches in total linear time.

Next, we consider the case where $P_2 \subseteq \Gamma_{\mathsf{L}}(P_1)$. Hence, we repeat all the above steps with any leftmost paths in H replaced with the corresponding rightmost paths. To obtain the constant probability of error, we repeat the entire process above $2^{\mathcal{O}(k)}$ times.

To derandomize this algorithm, instead of assigning the colors to vertices at random, we construct and use an $(n, 2k)$-universal set [7]. In our case this universal set is a family $\mathcal{U}$ of subsets of $V(H)$ such that for any subset A of $V(H)$ of size $2k$ the family $\{A \cap U \mid U \in \mathcal{U}\}$ contains all 2^{2k} subsets of A. We interpret each element of $\mathcal{U}$ as a single coloring, that is, a subset $A \in \mathcal{U}$ corresponds to the coloring c which assigns color green to the vertices of A, and blue to the rest.

We know that there is such family $\mathcal{U}$ of size $2^{\mathcal{O}(k)} \cdot \log n$ and it can be constructed in time $2^{\mathcal{O}(k)} \cdot n \log n$ [7]. We construct it once in the algorithm, and instead of drawing random colorings, we iterate through all elements of the constructed universal set $\mathcal{U}$. By the property

11

of $\mathcal{U}$ there is an element $U \in \mathcal{U}$ such that $V(P_1[x \to x']) \subseteq U$ and $V(P_1[y' \to y]) \cap U = \emptyset$ because $|V(P_1[x \to x']) \cup V(P_1[y' \to y])| = 2k$. This guarantees the correctness of our deterministic algorithm. $\qquad \square$

References

[1] Ivona Bezáková, Radu Curticapean, Holger Dell, and Fedor V. Fomin. Finding detours is fixed-parameter tractable. *SIAM J. Discret. Math.*, 33(4):2326–2345, 2019.

[2] Marek Cygan, Dániel Marx, Marcin Pilipczuk, and Michał Pilipczuk. The Planar Directed k-Vertex-Disjoint Paths problem is fixed-parameter tractable. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013*, pages 197–206. IEEE Computer Society, 2013.

[3] Fedor V. Fomin, Petr A. Golovach, William Lochet, Danil Sagunov, Kirill Simonov, and Saket Saurabh. Detours in Directed Graphs. In *39th International Symposium on Theoretical Aspects of Computer Science, STACS 2022*, volume 219 of *LIPIcs*, pages 29:1–29:16. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2022.

[4] Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Long directed (s, t)-path: FPT algorithm. *Inf. Process. Lett.*, 140:8–12, 2018.

[5] Steven Fortune, John E. Hopcroft, and James Wyllie. The directed subgraph homeomorphism problem. *Theor. Comput. Sci.*, 10:111–121, 1980.

[6] John E. Hopcroft and Robert Endre Tarjan. Efficient Planarity Testing. *J. ACM*, 21(4):549–568, 1974.

[7] Moni Naor, Leonard J. Schulman, and Aravind Srinivasan. Splitters and near-optimal derandomization. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23-25 October 1995*, pages 182–191. IEEE Computer Society, 1995.

[8] Alexander Schrijver. Finding k disjoint paths in a directed planar graph. *SIAM J. Comput.*, 23(4):780–788, 1994.

[9] Bang Ye Wu and Hung-Lung Wang. The next-to-shortest path problem on directed graphs with positive edge weights. *Networks*, 65(3):205–211, 2015.