Shortest Disjoint Paths on a Grid

Mathieu Mari[*] Anish Mukherjee[†] Michał Pilipczuk[‡] Piotr Sankowski[§]

**Abstract**

The well-known $k$-disjoint paths problem involves finding pairwise vertex-disjoint paths between $k$ specified pairs of vertices within a given graph if they exist. In the shortest $k$-disjoint paths problem one looks for such paths of minimum total length. Despite nearly 50 years of active research on the $k$-disjoint paths problem, many open problems and complexity gaps still persist. A particularly well-defined scenario, inspired by VLSI design, focuses on infinite rectangular grids where the terminals are placed at arbitrary grid points. While the decision problem in this context remains NP-hard, no prior research has provided any positive results for the optimization version. The main result of this paper is a fixed-parameter tractable (FPT) algorithm for this scenario. It is important to stress that this is the first result achieving the FPT complexity of the shortest disjoint paths problem in any, even very restricted classes of graphs where we do not put any restriction on the placements of the terminals.

## 1 Introduction

The $k$-disjoint paths problem has been actively studied for almost 50 years already [20]. It is also one of the key problems studied in the algorithmic graph minor theory [44, 45, 46]. The problem can be formally defined as follows: Given a graph $G$ and a set of $2k$ vertices $s_1, t_1, \ldots, s_k, t_k$ called *terminals*, the goal is to find a set of $k$ vertex disjoint paths $P_1, \ldots, P_k$ where each $P_i$ joins $s_i$ and $t_i$, for $i \in [k]$. While we are primarily interested in the vertex disjoint version here, the edge-disjoint version of this problem is very well studied as well. For a survey on the extensive study on this problem see e.g., [24, 30].

Since the very beginning, the central application for this problem was VLSI design. Disjoint paths are not only key primitive in this setup, but they also allow for some simplifications. As formalized by [50] for the case of VLSI design, the problem can be considered on rectangular grid graphs. One of the earliest papers on this topic proves NP-completeness of $k$-disjoint paths problem in the case of infinite rectangular grids [34]. In this problem, we are given pairs of terminals with integer coordinates on an infinite grid and we only need to decide whether vertex disjoint paths connecting these terminals exist. Some flaws in this paper have been later corrected in [21]. Following these papers, the rectangular grid setup has been considered in numerous papers with additional extensions or restrictions [28, 38, 27, 53]. For example, [3] extends the problem to multi-layer meshes where the goal is to optimize both of the layers as well as their number. Pinter [42] studies routing disjoint paths on two parallel lines and shows several structural and algorithmic results for this scenario. Tompa [51] previously studied the minimization version of a similar problem where the paths must be separated by a fixed distance. In [29] edge-disjoint paths in finite grids is studied where the terminals can be placed on the inner or the outer face of the grid. An efficient algorithm for finding the edge-disjoint paths in a rectangular grid where the terminals lie on the boundary of the grid is given in [39]. Multicommodity flow [23] and other flow problems [8] have also been a subject of study in rectangular grids.

In more recent years, [41] proposes an algorithm for the shortest non-crossing rectilinear paths in a 2D grid that avoid obstacles placed on the boundary of the grid. Whereas [25] study bounds on the *path-pairability* number in an infinite grid number which is the largest value $k$ for which there exist $k$ edge-disjoint paths. A series of works have made significant progress, especially in the context of grids [10, 13, 12], where the goal is to maximize the number of demand pairs satisfied (a path $P_i$) by a set of disjoint paths.

[*]Université de Montpellier, LIRMM, France. Partially supported by the ERC CoG grant TUgbOAT no 772346.

[†]University of Warwick, United Kingdom. Research supported in part by the Centre for Discrete Mathematics and its Applications (DIMAP) and by EPSRC award EP/V01305X/1.

[‡]University of Warsaw, Poland. This work is a part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, Grant Agreement No 948057 — BOBR.

[§]University of Warsaw, IDEAS NCBR, and MIM Solutions, Poland. Partially supported by the ERC CoG grant TUgbOAT no 772346 and the National Science Centre (NCN) grant no. 2020/37/B/ST6/04179

Despite this long line of effort, there are still many white areas in its complexity map – especially when the aim is to find the *shortest* disjoint paths. In the shortest $k$-disjoint paths (also referred to as the *min-sum* $k$-disjoint paths) problem we also want the total length of the solution to be minimized. This minimization problem is of central importance within algorithmic graph theory and combinatorial optimization, as it finds applications in many different contexts including VLSI design, transportation networks, as well as virtual circuit routing. While the decision version discussed above is fairly well understood, the optimization version is not so much. In particular, none of the papers mentioned above considered the optimization version of the problem introduced in [34], i.e., computing the shortest disjoint paths on infinite grids. This is the cleanest setup that one can consider for the general $k$ case. Here we focus on the scenario where the given instance is an infinite grid and there is no restriction on the placements of the terminals. The main result of this paper is an FPT algorithm for this setup.

THEOREM 1.1. *Given a set $\mathcal{T} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ of $k$ pairs of terminals placed arbitrarily on a grid, we can find the shortest $k$-disjoint paths between $\mathcal{T}$ or attest that there is no feasible solution in time $k^{2^{O(k)}} \cdot O([\mathcal{T}])$, where $[\mathcal{T}]$ is the number of bits required to encode $\mathcal{T}$ in binary.*

To the best of our knowledge, no XP algorithm was even known before this work. It is important to stress that this is the first result stating the FPT bound for the shortest disjoint path problem in any, even very restricted, classes of graphs where we do not put any restriction on the placements of the terminals. In our opinion, it strongly indicates that similar results shall hold for general planar graphs, as graph minor theory [46] identifies grids as the core complex cases for planar graphs. Our result is obtained using a variant of the *irrelevant vertex technique* [46] where we prove that vertices far away from all of the terminals can be removed, without modifying the length of the shortest solution and reintroduced after the optimal solution was found. The irrelevant vertex technique has been previously used for planar graphs [1]. However, to our knowledge, this is the first time that it has been used for the minimization version of the problem. In general, the minimization version of the problem is challenging as it is already W[1]-hard in general undirected graphs, parameterized by $k$, even when each path needs to be the shortest path [35].

**Related works on disjoint paths.** As indicated at the beginning, the classic disjoint paths problem is NP-hard in general graphs [20], and remains so even in very restricted settings. When the graph is directed, the problem is NP-hard for $k = 2$ [22]. It is one of Karp's NP-hard problems [26] (when $k$ is part of the input) and remains so when restricted to planar graphs [37]. [40] extends this result to the edge-disjoint variant as well which remains NP-hard even when the graph is planar and the terminals are incident on the outer face [49]. Furthermore, the problem remains NP-hard even on grid graphs [34]. Marx [38] shows that the problem is NP-hard on a rectangular grid even if the union of the supply and the demand graphs is Eulerian.

However, some positive results are also known for special classes of directed graphs such as in directed acyclic graphs [22], in directed planar graphs [48], and in tournaments [9]. Robertson and Seymour study the $k$-disjoint paths problem in a planar graph, where the terminals lie on the boundary of one or two faces [44] as part of the celebrated *Graph Minors* series. They extended the results to graphs on a surface [45] to give a fixed-parameter tractable (FPT) algorithm, where the parameter is the number $k$ of terminal pairs, and later even to general undirected graphs [46]. A solution to this problem was central to the Graph Minors Project and added to the importance of the corresponding optimization version. The dependency on the number of vertices $n$ has been further improved in more recently [31]. Furthermore, improved FPT algorithms have been obtained in planar graphs for various cases in recent years and this is still an active area of research, see e.g., [1, 16, 36, 54].

Significant progress has also been made in the approximation front to overcome the intractability. In particular, for the version where one wants to maximize the number of demands that can be satisfied (i.e., an $s_i t_i$ path) recent works have focused on the cases where the given instance is a grid or planar graph and the sources lie on the outer face and obtain improved upper bounds [10, 11, 12] as well as results on the hardness side [10, 14, 13].

**Related works on optimization versions.** The optimization variants of the $k$-disjoint paths problem are considerably harder. A version of the problem is called length-bounded disjoint paths, where each of the paths needs to have a length bounded by some integer $b$. This problem is NP-hard even when the $k$ terminals (where $k$ is part of the input) lie on the outer face [52]. The problem of finding two disjoint paths, one of which is the shortest path, is also NP-hard [18].

For the shortest $k$-disjoint paths problem that we focus on in this paper, where the goal is to minimize the total length of the paths, very few instances are known to be solvable in polynomial time. For unrestricted placement of the terminals in a graph only positive result known is for $k = 2$. Namely, Björklund and Husfeldt [5] give a randomized polynomial time algorithm in general undirected graphs. A deterministic polynomial time bound for the same and also, for $k \geq 3$ whether the problem admits a polynomial time solution remain as tantalizing open questions.

In planar graphs, Colin de Verdière and Schrijver [15] and Kobayashi and Sommer [32] give polynomial time algorithms for the shortest $k$-disjoint paths problem in some very special cases. An $O(kn \log n)$ time algorithm is given in [15] for the case when the sources are incident on one face and the sinks on another. In [32] an $O(n^4 \log n)$ time and $O(n^3 \log n)$ time algorithm is given when the terminal vertices are on one face for $k \leq 3$ or on two faces for $k = 2$, respectively. Borradaile et al. [6] give an $O(kn^5)$ time algorithm when the sources and the sinks lie in sequence on the boundary of the outer face. The case where all the terminals lie on the outer face in any order admits an FPT algorithm [17, 33]. Björklund and Husfeldt [4] presented a deterministic algorithm for $k = 2$ in subcubic planar graphs. On the negative side, Brandes et. al [7] show that the shortest edge-disjoint paths problem is NP-hard even in max degree 4 planar graphs, that even fulfills the Eulerian condition.

For arbitrary $k$, a linear time algorithm is known for bounded tree-width graphs [47]. Polynomial-time algorithms are also known through reducing the problems to the minimum cost flow problem when all the sources (or sinks) coincide or when the terminal vertices lie on a face in the order $s_1, s_2, \ldots, s_k, t_k, \ldots, t_2, t_1$ [52]. Lochet [35] considers the $k$-disjoint *shortest paths* problem where the goal is to find $k$ vertex-disjoint paths *each* of which is the shortest path and gives an XP-algorithm running in time $n^{O(k^{5^k})}$ in an $n$ vertex graph and also show W[1]-hardness. Another related problem, called shortest *non-crossing walks* was studied in planar graphs [19], which presents an FPT algorithm parameterized by the number of faces the terminals lie on.

**1.1 Technical Overview** Suppose we are given a set of $2k$ terminals in an infinite grid and that there exists a family $\mathcal{P}$ of disjoints paths connecting each pair of terminals, such that the sum of the lengths of the paths in $\mathcal{P}$ is minimum. The goal of our algorithm is to "guess" this solution $\mathcal{P}$. By guessing, we mean that the algorithm will enumerate all possible ways of connecting the terminals, and return the one with minimum total length. Of course, in an infinite grid, there is a priori infinitely many potential solutions. To reduce the number of solutions to enumerate, we show that $\mathcal{P}$ can be locally transformed into another solution of minimum total length that does not use a specific set of vertices of the grid, called *irrelevant vertices*, whose location only depends on the position of the terminals. This is formalized in our Structural Lemma, proved in Section 3.4. For an illustration of the irrelevant vertices, see Fig. 2. There are two types of irrelevant vertices. The first type are the vertices that are outside a bounding box of the terminals, whose border is at distance $\Theta(2^k)$ from the extreme terminals. The second type are the vertices inside the bounding box that are at distance $\Theta(k2^k)$ from the horizontal and vertical lines containing the terminals. Thanks to the structural lemma, we can thus already considerably reduce the number of solutions to enumerate. However, the number of non-irrelevant vertices may still be as large as $\Theta(k2^k \cdot L)$ where $L$ is the maximum distance between two terminals, which can be arbitrarily larger than any function of $k$. Then, to further reduce the exploration space, we show that the subpaths of an optimal solution, that are induced by vertices that are "far" from the terminals, are actually shortest paths. Intuitively, there is no benefit for a path to do a "U-turn" far from the terminals, or equivalently, the only reason for a path to do a U-turn is to bypass a terminal. This is formalized in Lemma 3.4. Hence, since the parts of the sought solution that are far from the terminals consist of a union of shortest paths, we show that it is actually enough to guess their endpoints. Overall, we have reduced the number of guesses to $k^{2^{O(k)}}$.

**Organization** After some preliminaries in Section 2, we prove the Structural Lemma in Section 3 and describe our algorithm in Section 4. We conclude in Section 5 with some open questions.

## 2 Preliminaries

For two integers $x, y$ with $x \leq y$, we denote $[x, y] = \{z \in \mathbb{Z} \mid x \leq z \leq y\}$. For a set $S \in \mathbb{Z}^2$, we denote $\overline{S} := \{u \in \mathbb{Z}^2 \mid u \notin S\}$ the complement of $S$. Given two sets of integers $A, B$, we denote $X \times Y = \{(x, y) \in \mathbb{Z}^2 \mid x \in X, y \in Y\} \subseteq \mathbb{Z}^2$ the cartesian product of $X$ and $Y$.

A *vertex* is a point $u = (x(u), y(u)) \in \mathbb{Z}^2$, where $x(u)$ is the $x$-coordinate of $u$, and $y(u)$ is its $y$-coordinate. Given two vertices $u, v$, we denote by $d(u, v) = |x(u) - x(v)| + |y(u) - y(v)|$ their (Manhattan) distance. Given a vertex $u$ and a set of vertices $S$, we denote $d(u, S) = \min_{v \in S} d(u, v)$. A *path* is a sequence of vertices $(u_0, \ldots, u_s)$

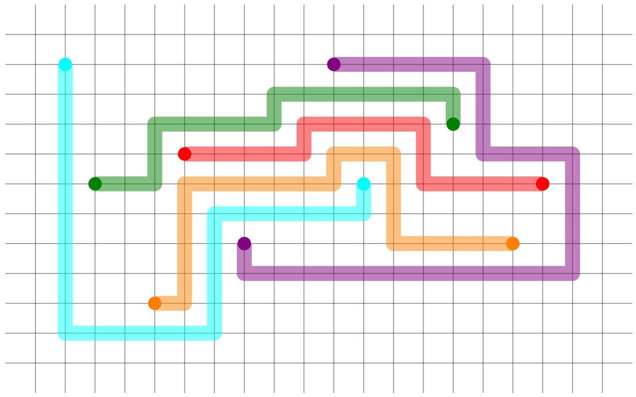

Figure 1: A routing. The colored vertices represent the pairs of terminals.

such that for all $i$, $i = 0, \ldots, s-1$, we have $d(u_i, u_{i+1}) = 1$. The *length* of a path $P = (u_0, \ldots, u_s)$ is $\ell(P) = s$, and its *endpoints* are the vertices $u_0$ and $u_s$. We say that $P$ *connects* $u$ and $v$, if $P$ is a path and $u$ and $v$ are its endpoints. We say that two paths $P$ and $P'$ are *disjoint* if $P \cap P' = \varnothing$.

Definition 2.1. (The $k$-SPG problem.) *We are given a set of $2k$ vertices $\mathcal{T} = \{s_1, t_1, \ldots, s_k, t_k\}$ called* terminals *organized into pairs. A* routing *of $\mathcal{T}$ is a family $\mathcal{P} = (P_1, \ldots, P_k)$ of pairwise-disjoint paths such that for each $i \in [1, k]$, the path $P_i$ connects $s_i$ and $t_i$. The* length *of a routing is the sum of the lengths of the paths $\ell(\mathcal{P}) \coloneqq \sum_{i=1}^{k} \ell(P_i)$. The goal is to compute an* optimal *routing of $\mathcal{T}$, i.e., a routing with minimum total length or attests that there is no routing of $\mathcal{T}$.*

**Paths.** We say that a path $P = (u_0, \ldots, u_s)$ is *simple* if for every $i, j \in [0, s]$ such that $|i - j| \geq 2$, we have $d(u_i, u_j) \geq 2$. It is not difficult to see that if a solution of $k$-SPG exists, then all its paths are simple. Otherwise, we could replace a non-simple path $P$ of the routing with a strictly shorter path whose vertices are a strict subset of the vertices of $P$. Notice that if $P$ is a simple path, then $P$ can simply be given as a set of vertices. Indeed, each internal vertex has exactly two vertices at distance 1, so there are exactly two manners (that correspond to both orientations) to order the vertices of a path. Thus, in this paper, to simplify, we often interpret $P$ as a set of vertices, non-necessarily ordered. In particular, we say that $P \subseteq \mathbb{Z}^2$ is a path if vertices of $P$ can be ordered such that the corresponding sequence is a path. Given a simple path $P$, we say that $Q$ is a *subpath* of $P$ if $Q \subseteq P$ and $Q$ is a path.

Given a path $P$ and a $S \subseteq \mathbb{Z}^2$, we say that $Q$ is a *hair of $P$ in $S$* if $Q \subseteq P \cap S$, $Q$ is a path, and there is no path $Q' \subseteq P \cap S$, such that $Q$ is a strict subset of $Q'$. It is easy to see that the hairs of $P$ in $S$ form a partition of $P \cap S$. If $\mathcal{P}$ is a routing, then the hairs of $\mathcal{P}$ in $S$ is the union of all hairs of $P$ in $S$, for all $P \in \mathcal{P}$. We say that a path $P$ (a routing $\mathcal{P}$) *uses* a vertex $u$ if $P$ (there is a path $P \in \mathcal{P}$ that, resp.) contains $u$.

## 3 Structural Lemma

In this section, we present our structural lemma on which is based our algorithm. The lemma says there is a way of "re-arranging" a routing, in a way that the total length does not increase, and the new routing does not use any vertex from a specific set of vertices, called *irrelevant vertices*, which depends only on the position of the terminals.

There are two types of irrelevant vertices: the ones that are not outside a specific bounding box of the terminals; and the ones that are contained inside the bounding box, but whose $x$- and $y$-coordinates are distant enough for the coordinates of the terminals. See Fig. 2 for an illustration.

We now define precisely these sets. Let $\mathcal{T} = \{s_1, t_1, \ldots, s_k, t_k\}$ be a set of $2k$ terminals. Let

$$\mathcal{B}_{\mathcal{T}} \coloneqq [x_{\min} - 2^{k+1}, x_{\max} + 2^{k+1}] \times [y_{\min} - 2^{k+1}, y_{\max} + 2^{k+1}],$$

where $x_{\min}, x_{\max}, y_{\min}, y_{\max}$ denote the minimum and maximum $x$ and $y$ coordinates in $\mathcal{T}$. Let $\mathcal{H}_{\mathcal{T}}$ and $\mathcal{V}_{\mathcal{T}}$ respectively denote the set of the horizontal and vertical lines that contain a terminal. Formally,

$$\mathcal{H}_{\mathcal{T}} \coloneqq \bigcup_{t \in \mathcal{T}} \{u \in \mathbb{Z}^2 \mid y(u) = y(t)\} \text{ and } \mathcal{V}_{\mathcal{T}} \coloneqq \bigcup_{t \in \mathcal{T}} \{u \in \mathbb{Z}^2 \mid x(u) = x(t)\}.$$

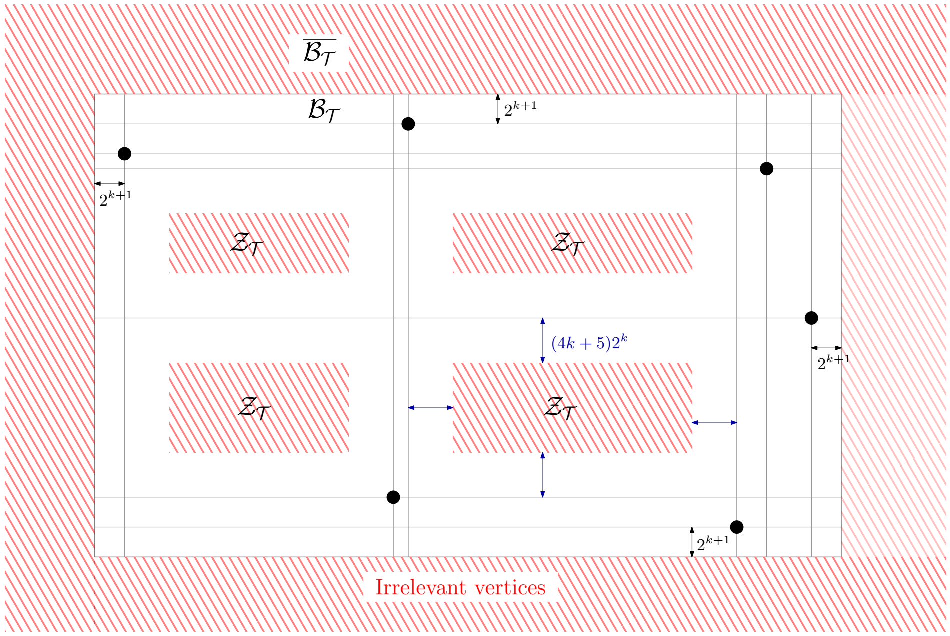


Figure 2: The set of irrelevant vertices is represented by red-shaded areas. Black points are terminals. We prove that there is an optimal routing that does not use any irrelevant vertex.

Let $\mathcal{Z}_\mathcal{T} := \{u \in \mathbb{Z}^2 \mid d(u, \mathcal{V}_\mathcal{T} \cup \mathcal{H}_\mathcal{T}) > (4k+5)2^k\}$. We call $\overline{\mathcal{B}_\mathcal{T}} \cup \mathcal{Z}_\mathcal{T}$ the set of *irrelevant vertices for* $\mathcal{T}$.

Lemma 3.1. (Structural Lemma) *If there exists a routing of a set of terminals $\mathcal{T}$, then there exists an optimal routing of $\mathcal{T}$ that does not use any irrelevant vertex.*

In the rest of the section, we prove Lemma 3.1. In the whole section we fix a set $\mathcal{T} = \{s_1, t_1, \dots, s_k, t_k\}$ of $2k$ terminals, and we suppose that there exists a routing $\mathcal{P} = (P_1, \dots, P_k)$ of $\mathcal{T}$ (otherwise there is nothing to prove). Without loss of generality, we assume that all paths in $\mathcal{P}$ are simple. We construct a routing $\mathcal{P}' = (P'_1, \dots, P'_k)$ such that $\ell(\mathcal{P}') \le \ell(\mathcal{P})$ and for all $i \in [1,k]$, $P'_i \subseteq \mathcal{B}_\mathcal{T} \setminus \mathcal{Z}_\mathcal{T}$. To achieve this, we use two shortening strategies presented in Section 3.1. The idea is that if there exists a set $W$ of vertices with some special properties (see Definitions 3.1 and 3.2), then we are able to shorten the solution by re-arranging the routing locally around $W$.

Then, in Section 3.2, we study the intersection of $\mathcal{P}$ routing with some special subsets of vertices called strips. We show that if $\mathcal{P}$ has some hairs in the strip that are not shortest paths, then we can locate and apply one of the two shortening operations, in order to reduce the length of the routing. We show that this implies that vertices outside the bounding box $\mathcal{B}_\mathcal{T}$ are indeed irrelevant. Next, in Section 3.3, we introduce a technique that consists of modifying a routing, without modifying its length, by "pushing-down" its paths; and prove some useful properties of pushed-down routings.

Finally, in Section 3.4, we complete the proof of the structural lemma, by showing that if a pushed-down routing uses some irrelevant vertex in $\mathcal{Z}_\mathcal{T}$, then we can again locate and apply one of our shortening strategies.

**3.1 Shortening a routing** We first present a rather straightforward shortening strategy, called *simple shortcut*, and then present a more involved shortening strategy that relies on the existence of a *shortenable window*.

Definition 3.1. (Simple Shortcut) *We say that a routing $\mathcal{P}$ has a* simple shortcut *if there exists a path $P \in \mathcal{P}$ and two vertices $u, v \in P$ such that:*

1. $x(u) = x(v)$ *or* $y(u) = y(v)$,
2. $d(u,v) \ge 2$,
3. *no vertices in the (unique) shorter path from $u$ to $v$ (except $u$ and $v$) belong to a path in $\mathcal{P}$.*

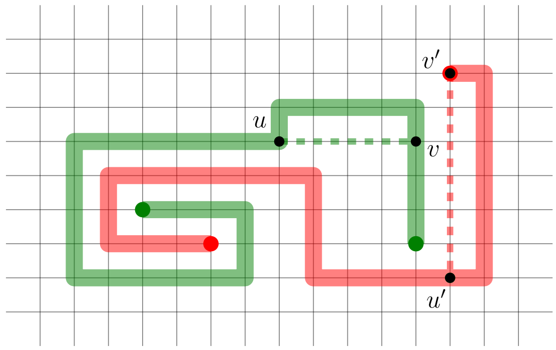


Figure 3: A non-optimal routing. This is attested by the presence of two simple shortcuts (dashed segments).

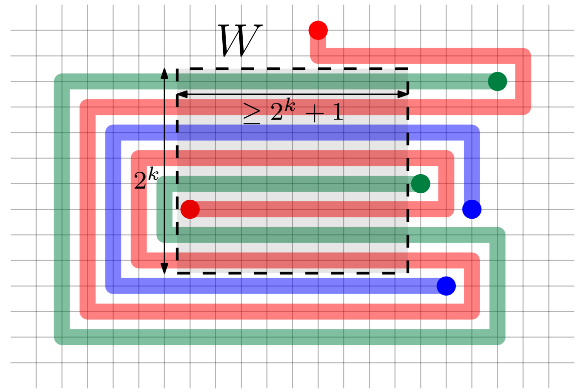


Figure 4: A routing with a horizontal shortenable window (here, $k = 3$).

It is clear that if a routing has a simple shortcut, then it is not optimal. Indeed, replacing the subpath of $P$ between $u$ and $v$ by the shortest path between $u$ and $v$ creates a strictly shorter routing (see Fig. 3).

Let us now present the second, this time non-trivial, way of shortening a routing. A *window* is a set of vertices $[x^-, x^+] \times [y^-, y^+] \subset \mathbb{Z}^2$ for some integers $x^-, y^-, x^+, y^+$ such that $x^- < x^+$ and $y^- < y^+$.

DEFINITION 3.2. *Given a routing $\mathcal{P}$ of $\mathcal{T}$, we say that a window $W = [x^-, x^+] \times [y^-, y^+]$ is* horizontally shortenable *(with respect to $\mathcal{P}$) if the following conditions hold:*

*(a) the height is $y^+ - y^- + 1 = 2^k$,*

*(b) the width is $x^+ - x^- \geq 2^k + 1$, and*

*(c) for each integer $y \in [y^-, y^+]$, the set $[x^-, x^+] \times \{y\}$ is a subpath of some path in $\mathcal{P}$.*

See Fig. 4 for an illustration. We define *vertically shortenable* windows symmetrically. A window is *shortenable* if it is either vertically shortenable or horizontally shortenable.

LEMMA 3.2. *If $\mathcal{P}$ is a routing of $\mathcal{T}$, and $W$ is a shortenable window with respect to $\mathcal{P}$, then there exists a routing $\mathcal{P}'$ of $\mathcal{T}$ with $\ell(\mathcal{P}') < \ell(\mathcal{P})$.*

In particular, this implies that an optimal solution has no shortenable windows. The proof is highly inspired by a technique introduced in [2], sometimes referred to as the "reflection trick". The plan of the proof is to first identify a sub-window $W' \subset W$ that has the property that each path in $\mathcal{P}$ intersects $W'$ an even number of times. Then, we delete the subpaths of $\mathcal{P}$ that are in $W'$ and then reconnect the disconnected terminals by adding pairwise disjoint paths in $W'$ of strictly shorter total length.

To identify $W'$, we use the following combinatorial result. A *word* on an alphabet $A$, is an ordered sequence $a_1 \dots a_n$ (with possibly repetitions), where for all $i \in [1, n]$, $a_i \in A$. A *subword* of a word $a_1 \dots a_n$ is a word of the form $a_i \dots a_j$, with $1 \leq i \leq j \leq n$. A word is *even* if each letter appears an even number of times.

LEMMA 3.3. *Let $w = a_1 \dots a_n$ be a word of length $n$ on an alphabet $A$. If $n \geq 2^{|A|}$, then $w$ has an non-empty even subword.*

We remark that the bound $2^{|A|}$ on the size of $w$ is tight. Indeed, for each $n \geq 1$, there is a word $w_n$ of length $2^n - 1$ on the alphabet $\{1, \dots, n\}$ that does not have any even subword.[1]

*Proof.* [Proof of Lemma 3.3.] Without loss of generality, we assume that $A = \{1, \dots, m\}$. We define $f : \{0, 1, \dots, n\} \to \{0, 1\}^m$ such that for each $i \in [1, n]$, and each $a \in A$, the $a$-th coefficient of $f(i)$ is 1 if and only if $a$ appears an even number of times in the prefix $a_1 \dots a_i$. If $n \geq 2^m$, then by the pigeonhole principle, there exist two distinct indices $i, j$ with $0 \leq i < j \leq n$, such that $f(i) = f(j)$. It is easy to see that the subword $a_{i+1} \dots a_j$ is even, and not empty. ∎

[1] define recursively $w_m = w_{m-1} m w_{m-1}$.

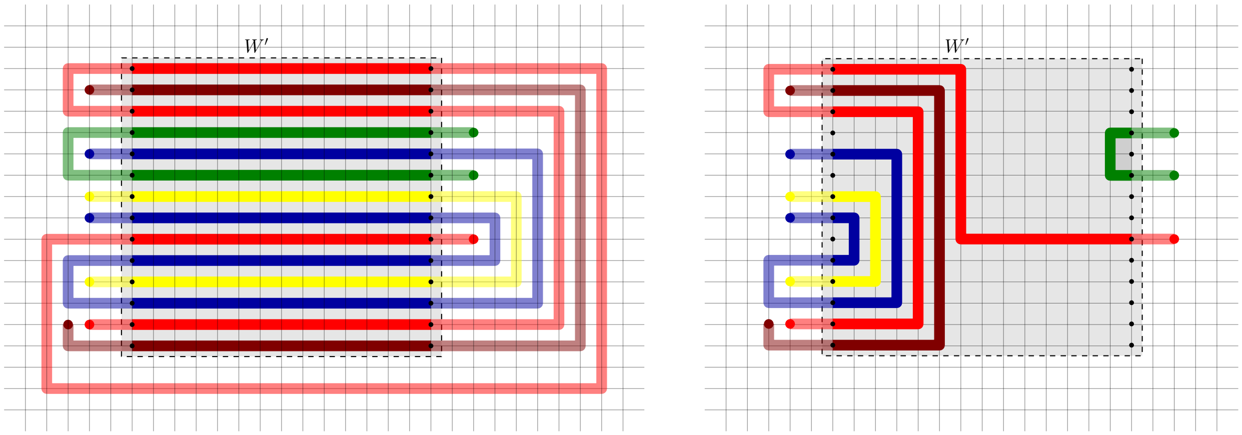


Figure 5: Shortening a routing via a shortenable window. All hairs of the window $W'$ are replaced by new pairwise disjoint paths inside $W'$ connecting the same terminals. After these steps, some subpaths outside $W'$ are no longer needed and are removed from the routing.

We treat the case when $W$ is horizontal; the other case is symmetric. Notice that the sequence of hairs of $\mathcal{P}$ in $W$ from top to bottom can be seen as a word where each letter corresponds to a path in $\mathcal{P}$ (or a pair of terminals). We deduce from Lemma 3.3 that there exist two integers $y^- < y^+$ such that for each path $P \in \mathcal{P}$, the number of hairs of $P$ in the window $W' := W \cap \{(x, y) \mid x \in \mathbb{Z}, y \in [y^-, y^+]\}$ is even. To facilitate the presentation, we assume that $W' = [0, h] \times [0, \ell]$. Notice that $h$ is necessarily an odd number. Since $W$ is a shortenable window, we know that $\ell \geq 2^k + 1$.

Let $L = \{(0, y) \mid y \in \mathbb{Z}, 0 \leq y \leq h\}$ and $R = \{(\ell, y) \mid y \in \mathbb{Z}, 0 \leq y \leq h\}$ be the left side and the right side of $W'$, respectively. For each path $P \in \mathcal{P}$, let $n(P) := |P \cap (L \cup R)|$ that corresponds to the number of times that $P$ enters or leaves $W'$. Notice that for each $P \in \mathcal{P}$, $n(P)$ is a multiple of 4, and let $n'(P) = n(P)/4$. Let $P[1], \ldots, P[n(P)]$ denote the intersection vertices of $P$ with $L \cup R$ ordered in increasing distance (along $P$) from the source of $P$.

In particular, for each $j \in [1, n(P) - 1]$ the subpath of $P$ between $P[j]$ and $P[j + 1]$ is contained in $W'$ (in $\overline{W'}$) if and only if $j$ is odd (even, resp.). To construct $\mathcal{P}'$, we replace, for each $P \in \mathcal{P}$ and each $i \in [1, n'(P)]$, the subpath of $P$ between $P[4i - 3]$ and $P[4i]$ by a subpath $Q^i_P$ (with endpoints $P[4i - 3]$ and $P[4i]$) that is contained in $W'$. We will ensure that these new paths are pairwise disjoint. This is illustrated in Fig. 5.

The key property to guarantee that new paths are pairwise-disjoint is to show that vertices on $L \cup R$ to be reconnected are pairwise *non-crossing* in the following sense.

Let us order the vertices in $L \cup R$ as follows:

$$(0, 0) \prec (0, 1) \prec \cdots \prec (0, h) \prec (\ell, h) \prec (\ell, h - 1) \prec \cdots \prec (\ell, 0).$$

We say that two pairs of vertices in $L \cup R$ are *crossing* if no two vertices from the same pair are consecutive (in the ordered sequence that corresponds to these four vertices). Intuitively, two pairs of vertices are crossing, if any two topological paths connecting the pairs are intersecting. See Fig. 6.

CLAIM 3.1. *Let $P, P'$ be any two paths in $\mathcal{P}$ (with possibly $P = P'$), and let $i, i'$ be two integers such that $1 \leq i \leq n'(P)$ and $1 \leq i' \leq n'(P')$. Then, the pairs $(P[4i - 3], P[4i])$ and $(P'[4i' - 3], P'[4i'])$ are non-crossing.*

*Proof.* We first remark that $(P[4i-3], P[4i])$ and $(P'[4i'-3], P'[4i'])$ are crossing if and only if $(P[4i-2], P[4i-1])$ and $(P'[4i' - 2], P'[4i' - 1])$ are crossing. Indeed, the pairs $(P[4i - 2], P[4i - 1])$ and $(P'[4i' - 2], P'[4i' - 1])$ are obtained by vertical symmetry from the pairs $(P[4i - 3], P[4i])$ and $(P'[4i' - 3], P'[4i'])$; a vertical symmetry results in inverting the order of the vertices of $L \cup R$; and the defining of crossing is invariant by inverting the order.

Let $H$ be the subpath of $P$ that connects $P[4i - 2]$ and $P[4i - 1]$. Similarly, let $H'$ be the subpath of $P'$ that connects $P'[4i' - 2]$ and $P'[4i' - 1]$. The paths $H$ and $H'$ are outside $W$, except their endpoints (that are in $L \cup R$). If $(P[4i - 2], P[4i - 1])$ and $(P'[4i' - 2], P'[4i' - 1])$ are crossing, then using the Jordan curve theorem, we obtain

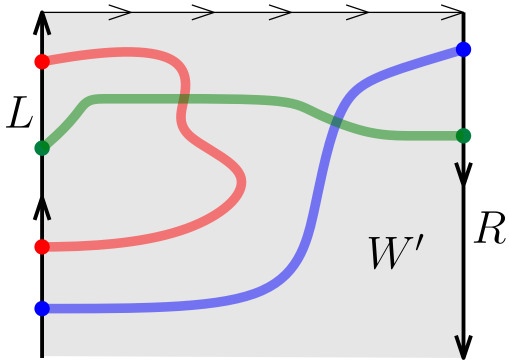


Figure 6: In this example, the green pair is crossing with the red pair and with the blue pair. However, the blue pair and the red pair are not crossing; this is the case since red vertices are between the blue vertices, according to the order (arrows) defined on $L \cup R$.

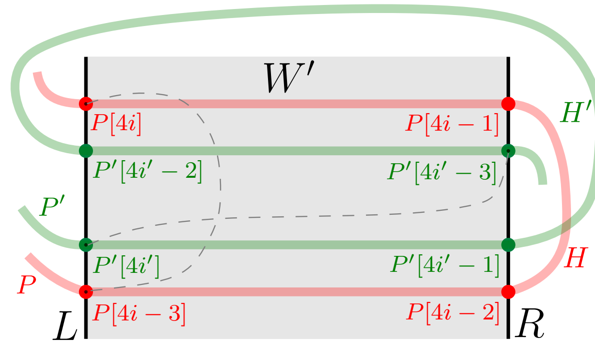


Figure 7: This situation is impossible. If $(P[4i-3], P[4i])$ (red) and $(P'[4i'-3], P'[4i'])$ (green) are crossing (gray dashed lines), then by symmetry, $(P[4i-2], P[4i-1])$ and $(P'[4i'-2], P'[4i'-1])$ are also crossing, which implies that $H$ and $H'$ (and thus also $P$ and $P'$) are not disjoint.

that $H$ and $H'$ are non-disjoint (see Fig. 7). This can also be seen by observing that there is a continuous function of the plane that maps $W'$ to $\overline{W'}$ and that preserves the boundary of $W'$; thus crossing crossing inside $W'$ implies a crossing outside $W'$. Thus our assumption contradicts the fact that $\mathcal{P}$ is a routing. Hence, $(P[4i-3], P[4i])$ and $(P'[4i'-3], P'[4i'])$ are non-crossing. □

We now construct a family of paths $\{Q_P^i \mid P \in \mathcal{P}, i \in [1, n'(P)]\}$. Let

$$\begin{aligned}
\mathcal{L} &:= \{(P,i) \mid P \in \mathcal{P}, i \in [1, n'(P)], P[4i-3] \in L \text{ and } P[4i] \in L\},\\
\mathcal{R} &:= \{(P,i) \mid P \in \mathcal{P}, i \in [1, n'(P)], P[4i-3] \in R \text{ and } P[4i] \in R\},\\
\mathcal{M} &:= \{(P,i) \mid P \in \mathcal{P}, i \in [1, n'(P)], P[4i-3] \in L \text{ and } P[4i] \in R \text{ or vice-versa}\}.
\end{aligned}$$

To define the paths, we first need to define the level of each element of these families. Intuitively, when reconnecting the disconnected terminals, we use paths that are "nested" in each other. This induces an ordering of the pairs, formally captured by the notion of levels, defined below, that enables us to precisely define the new paths.

We define the level of elements in $\mathcal{L}$ recursively as follows. Let $(P,i) \in \mathcal{L}$. If there is no $(P', i') \in \mathcal{L}$ such that

$$P'[4i'-3] \text{ and } P'[4i'] \text{ are between } P[4i-3] \text{ and } P[4i], \tag{3.1}$$

then $l(P,i) = 1$. Otherwise, $l(P,i) = 1 + \max(l(P',i'))$ where the max is over all elements $(P',i') \in \mathcal{L}$ such that (3.1). We define levels of elements of $\mathcal{R}$ similarly. Let $l_{\max}$ and $r_{\max}$ be the maximum levels in $\mathcal{L}$ and $\mathcal{R}$, respectively.

We now define the levels of elements in $\mathcal{M}$. Let $(P,i) \in \mathcal{M}$. The level of $(P,i)$ is $l(P,i) := 1 + n$ where $n$ is the number of elements $(P',i') \in \mathcal{M}$ such that $y(p)$ is between $y(P[4i-3])$ and $y(P[4i])$ where $p = \{P'[4i'-3], P'[4i']\} \cap L$.

We can now construct the paths.

- For each $(P,i) \in \mathcal{L}$, we define $Q_P^i$ as the path that connects $P[4i-3]$ and $P[4i]$ with corners $(l(P,i), y(P[4i-3]))$ and $(l(P,i), y(P[4i]))$.
- For each $(P,i) \in \mathcal{R}$, we define $Q_P^i$ as the path that connects $P[4i-3]$ and $P[4i]$ with corners $(\ell - l(P,i), y(P[4i-3]))$ and $(\ell - l(P,i), y(P[4i]))$.
- For each $(P,i) \in \mathcal{M}$, we define $Q_P^i$ as the path that connects $P[4i-3]$ and $P[4i]$ with corners $(l_{\max} + l(P,i), y(P[4i-3]))$ and $(l_{\max} + l(P,i), y(P[4i]))$.

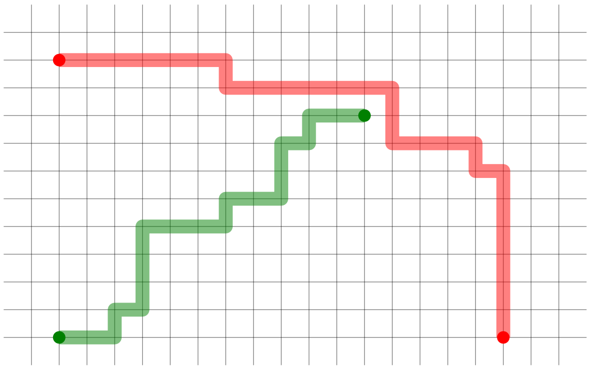

Figure 8: Monotone paths. The red path is (⌞⌝)-monotone, while the green one is (⌜⌟)-monotone.

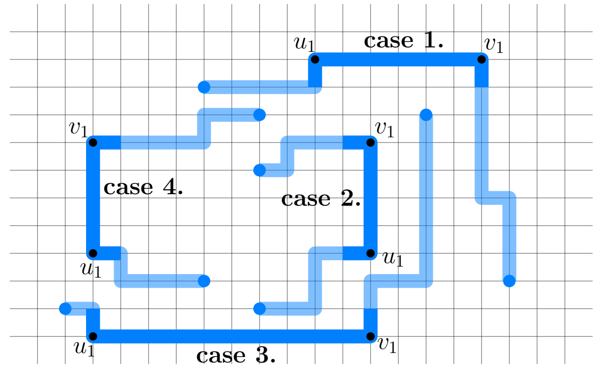


Figure 9: Non-monotone paths.

It is not difficult to see that these paths are pairwise disjoint and that they are contained in $W'$ (see Fig. 5).

We can now formally construct the new routing $\mathcal{P}'$. For each path $P \in \mathcal{P}$ that intersects $W$, and for $0 \le i \le n'(P)$, we replace the subpath of $P$ from $P[4i-3]$ to $P[4i]$ by $Q_P^i$. We have shown that $\mathcal{P}'$ is a routing, and it is clear that its length is strictly smaller than the length of $\mathcal{P}$ (this follows from the fact that in $W'$ some vertices are not used by any path in $\mathcal{P}'$, while all vertices in $W'$ are used by $\mathcal{P}$). This complete the proof of Lemma 3.2.

**3.2 Monotone Hairs in Strips** Given a path $P = (u_0, \dots, u_s)$, we say that a vertex $u_i \in P$ is a *corner* of $P$ if $1 \le i \le s-1$ and if the vertices $u_{i-1}$ and $u_{i+1}$ have distinct $x$-coordinates and distinct $y$-coordinates. There are four types of corners : ⌞, ⌜, ⌝ and ⌟ (see e.g. vertices denoted by letters $u$ and $v$ on Fig. 9). We denote ⌞$(P)$ the subset of $P$ that corresponds to ⌞-corners of $P$. We define similarly ⌜$(P)$, ⌝$(P)$ and ⌟$(P)$.

We say that a simple path $P = (u_0, \dots, u_s)$ is *monotone* if it is the shortest path between its endpoints. On a grid, a path $P$ is a shortest path if and only if either (i) ⌞$(P) = \varnothing$ and ⌝$(P) = \varnothing$, or (ii) ⌜$(P) = \varnothing$ and ⌟$(P) = \varnothing$ (see Fig. 8). We refer to the former type as (⌜⌟)-monotone and the latter as (⌞⌝)-monotone. The subpath of $P$ between two consecutive corners is called a *segment* of $P$.

We say that a set $S$ of vertices is *connected* if for each pair of vertices in $S$, there exists a path in $S$ that connects them. A *connected component* of $S$ is a maximal (for inclusion) connected subset of $S$. Recall that $\mathcal{H}_\mathcal{T}$ and $\mathcal{V}_\mathcal{T}$ respectively denote the set of the horizontal and vertical lines that contains a terminal. Let $\mathcal{S}_v = \{u \in \mathbb{Z}^2 \mid d(u, \mathcal{V}_\mathcal{T}) \ge 2^{k+1} + 1\}$. We refer to the connected components of $\mathcal{S}_v$ as *vertical strips*. Similarly, we refer to the connected components of $\mathcal{S}_h := \{u \in \mathbb{Z}^2 \mid d(u, \mathcal{H}_\mathcal{T}) \ge 2^{k+1} + 1\}$ as *horizontal strips*. A subset of vertices is a *strip* if it is either a horizontal strip or a vertical strip. We remark that any vertex in a strip is at a distance at least $2^{k+1} + 1$ from any terminal.

Lemma 3.4. *Let $\mathcal{P}$ be a routing of $\mathcal{T}$. Suppose that there is a strip $S$ and a path $P \in \mathcal{P}$ such that a hair of $P$ in $S$ is not monotone. Then, $\mathcal{P}$ has either a simple shortcut or a shortenable window.*

*Proof.* We treat the case when $S$ is a horizontal strip. The other case is symmetric. Again, we assume without loss of generality that paths in $\mathcal{P}$ are simple. Suppose that there exists a hair $P_1$ in $S$ that is not monotone. This implies that $P_1$ has two consecutive corners $u_1$ and $v_1$ such that one of the following conditions holds.

**Case 1.** $u_1$ is a ⌜-corner and $v_1$ is a ⌝-corner.

**Case 2.** $u_1$ is a ⌟-corner and $v_1$ is a ⌝-corner.

**Case 3.** $u_1$ is a ⌞-corner and $v_1$ is a ⌟-corner.

**Case 4.** $u_1$ is a ⌞-corner and $v_1$ is a ⌜-corner.

See Fig. 9. Cases 1 and 3 are symmetrical, and so are cases 2 and 4. We first treat case 1 and then case 2.

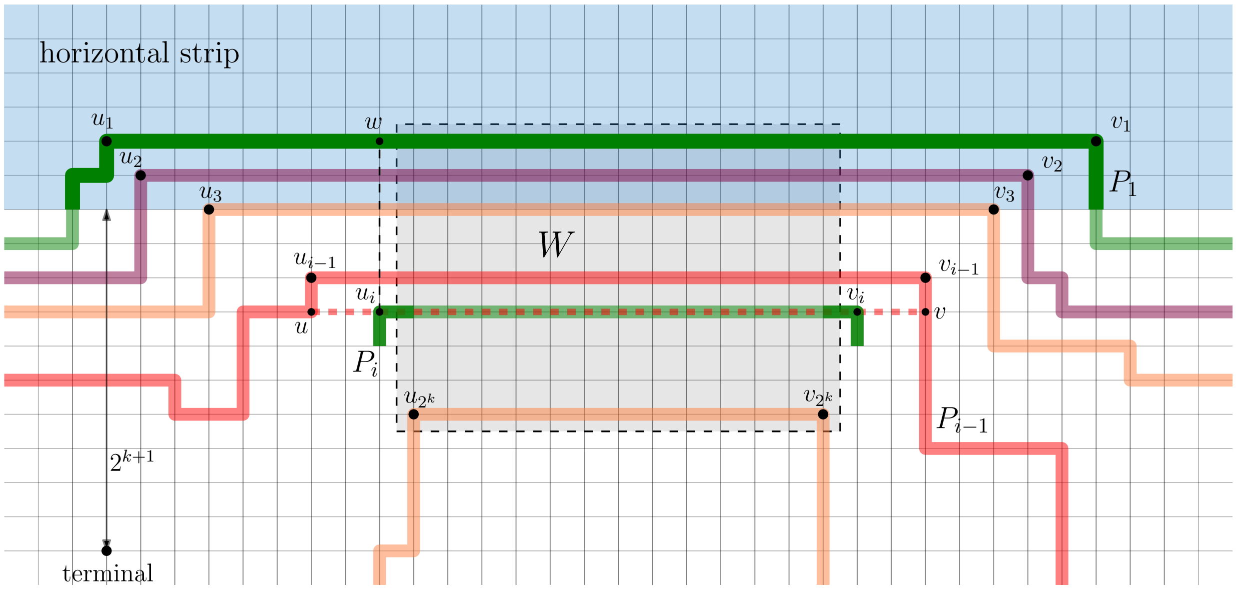


Figure 10: Proof of case 1 (Lemma 3.4). We identify a shortenable window $W$ in the case that the routing has a non-monotone hair (opaque green path) with respect to a strip (light blue area). To show the existence of this window, we construct a long enough nested sequence of non-monotone subpaths or show that there exists a simple shortcut.

**Proof in case 1.** We assume that $\mathcal{P}$ does not have any simple shortcut and we construct a horizontal shortenable window for $\mathcal{P}$. Notice that we have $x(u_1) < x(v_1)$ and $y(u_1) = y(v_1)$. We show by induction that there exist vertices $(u_i)_{i=1}^{2^{k+1}}$ and $(v_i)_{i=1}^{2^{k+1}}$ such that, for all $i \in [1, 2^{k+1}]$, it holds that

(i) $u_i$ and $v_i$ are two consecutive corners of some path $P_i$ in $\mathcal{P}$,

(ii) $u_i$ is a $\ulcorner$-corner and $v_i$ is a $\urcorner$-corner,

(iii) if $i \geq 2$, then $y(u_i) = y(v_i) = y(u_{i-1}) - 1$ and $x(u_{i-1}) + 1 \leq x(u_i) < x(v_i) \leq x(v_{i-1}) - 1$.

See Fig. 10. The case $i = 1$ is true by assumption. Suppose that we have constructed $u_1, \ldots, u_{i-1}$ and $v_1, \ldots, v_{i-1}$ that satisfy conditions (i)-(iii) for some $i \in [2, 2^{k+1}]$. Let $u = (x(u_{i-1}), y(u_{i-1}) - 1)$ and $v = (x(v_{i-1}), y(v_{i-1}) - 1)$. As $u_{i-1}$ and $v_{i-1}$ are respectively $\ulcorner$-corner and $\urcorner$-corner of $P_{i-1}$, $u$ and $v$ are in $P_{i-1}$. Since by assumption, there is no simple shortcut, the shortest path between them (red dashed line in Fig. 10) must contain a vertex from another path $P_i \in \mathcal{P}$ (see Definition 3.1). Notice that $P_i \neq P_{i-1}$, since $P_i$ is simple.

Let $u_i$ denote the leftmost point on $P_i \cap \{(x, y(u)) \mid x(u) < x < x(v)\}$. We claim that $u_i$ is a $\ulcorner$-corner of $P_i$. First, assume that $u_i$ is a terminal. Then, we know by applying (iii) recursively, that the vertex $w = (x(u_i), y(u_i) + i - 1)$ belongs to $P_1$, and we have $d(w, u_i) = i - 1 < 2^{k+1}$. But by assumption, $P_1$ is contained in a strip, which by definition, implies that $d(w, \mathcal{T}) \geq 2^{k+1}$. Thus, $u_i$ is not a terminal, and since $P_i$ and $P_{i-1}$ are disjoint, $u_i$ is necessarily a $\ulcorner$-corner of $P_i$. Now, let $v_i \in P_i$ be the corner of $P_i$ consecutive to $u_i$ such that $y(v_i) = y(u_i)$. Since $P_i$ and $P_{i-1}$ are disjoint, $v_i$ is necessarily a $\urcorner$-corner and is contained in $C$. We have established the recurrence.

Let $(u_i)_{i=1}^{2^{k+1}}$ and $(v_i)_{i=1}^{2^{k+1}}$ the sequences of vertices obtained. We define $W = [x(u_{2^k}), x(v_{2^k})] \times [y(u_1), y(u_{2^k})]$. We claim that $W$ is a horizontal shortenable window for $\mathcal{P}$ (see Definition 3.2). Indeed, by (iii), we have $y(u_{2^k}) = y(u_1) - 2^k + 1$, so the height of the window is $2^k$. Moreover, by (iii), we have

$$x(v_{2^k}) - x(u_{2^k}) \geq 2 + x(v_{2^k+1}) - x(u_{2^k+1}) \geq \cdots \geq 2 \cdot 2^k + x(v_{2^{k+1}}) - x(u_{2^{k+1}}) \geq 2^k + 1.$$

Hence, the width of the window is at least $2^k + 1$. Finally, it follows from (iii) that for all $i \in [1, 2^k]$, $x(u_i) \leq x(u_{2^k})$ and $x(v_i) \geq x(v_{2^k})$. Using (i) and (iii), we know that the segment from $u_i$ to $v_i$ is a subpath of $P_i \in \mathcal{P}$, and that these subpaths cover $W$ completely. Thus, $W$ is a horizontal shortenable window, what we wanted to prove.

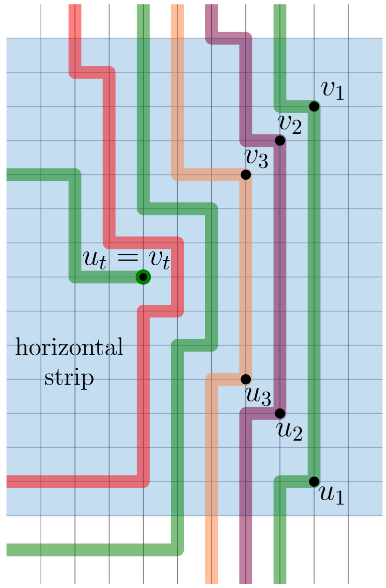


Figure 11: Proof of case 2 (Lemma 3.4). In an optimal routing, no hair of a horizontal strip can be "vertically" non-monotone (case 2 and 4 in Figure 9). That would imply there is a terminal (vertex $u_t$) contained in a strip (blue area), which is impossible by the definition of a strip.

**Proof in case 2.** We show that $\mathcal{P}$ has a simple shortcut. Suppose for a contradiction that this is not the case. Similar to case 1, we construct two sequences of vertices $(u_i)_{i=1}^t$ and $(v_i)_{i=1}^t$, for some integer $t \geq 1$ such that, $u_t = v_t$ is a terminal and for all $i \in [2, t]$,

(i) if $i < t$, then $u_i$ and $v_i$ are two consecutive corners of some path $P_i$ in $\mathcal{P}$;

(ii) if $i < t$, then $u_i$ is a $\lrcorner$-corner and $v_i$ is a $\urcorner$-corner;

(iii) $x(u_i) = x(v_i) = y(u_{i-1}) - 1$ and $y(u_{i-1}) + 1 \leq y(u_i) \leq y(v_i) \leq x(v_{i-1}) - 1$.

See Fig. 11. By (iii), we deduce that $y(u_1) < y(u_t) < y(v_1)$. This implies that $u_t \in S$, which is a contradiction with the fact that a strip does not contain any terminal. This shows that in case 2, $\mathcal{P}$ necessarily has a simple shortcut. This concludes the proof of Lemma 3.4. ∎

With Lemma 3.4, we can already show that there vertices outside the bounding box $\mathcal{B}_\mathcal{T}$ are not used by any optimal routing. Recall that $\mathcal{B}_\mathcal{T} = [x_{\min} - 2^{k+1}, x_{\max} + 2^{k+1}] \times [y_{\min} - 2^{k+1}, y_{\max} + 2^{k+1}]$ where $x_{\min}, x_{\max}, y_{\min}, y_{\max}$ denote the maximum and minimum $x$- and $y$-coordinates in $\mathcal{T}$.

Corollary 3.1. *Let $\mathcal{G}$ be a routing that uses a vertex $u \notin \mathcal{B}_\mathcal{T}$. Then, $\mathcal{P}$ is not optimal.*

*Proof.* See Fig. 12. The complement of $\mathcal{B}_\mathcal{T}$ is contained in the union of four strips. Suppose without loss of generality that $u$ is contained in the strip $S = \{v \in \mathbb{Z}^2 \mid y(v) \geq y_{\max} + 2^{k+1}\}$. Let $P \in \mathcal{P}$ be the path that contains $u$. Since $P$ connects two terminals, that are inside $\mathcal{B}_\mathcal{T}$ (and thus also outside $S$), it has to intersect the set $\{z \in \mathbb{Z}^2 \mid y(z) = y_{\max} + 2^{k+1}\}$ in at least two vertices $v$ and $w$, such that the subpath of $P$ that connects $v$ and $w$ contains $u$. This subpath is a hair with respect to $S$ and is not monotone. Thus, by Lemma 3.4, either $\mathcal{P}$ has a simple shortcut or has a shortenable window. In both cases, we deduce using Lemma 3.2 that $\mathcal{P}$ is not optimal. ∎

**3.3 Pushed-down routing** We say that a vertex is a *top-corner* of a path if it is either a $\urcorner$-corner or a $\ulcorner$-corner. Given a $\ulcorner$-corner $u$ of a path, we say that the vertex $(x(u) + 1, y(u) - 1)$ is the *inside vertex* of $u$. We define symmetrically the *inside vertex* of a $\urcorner$-corner. See Fig. 13.

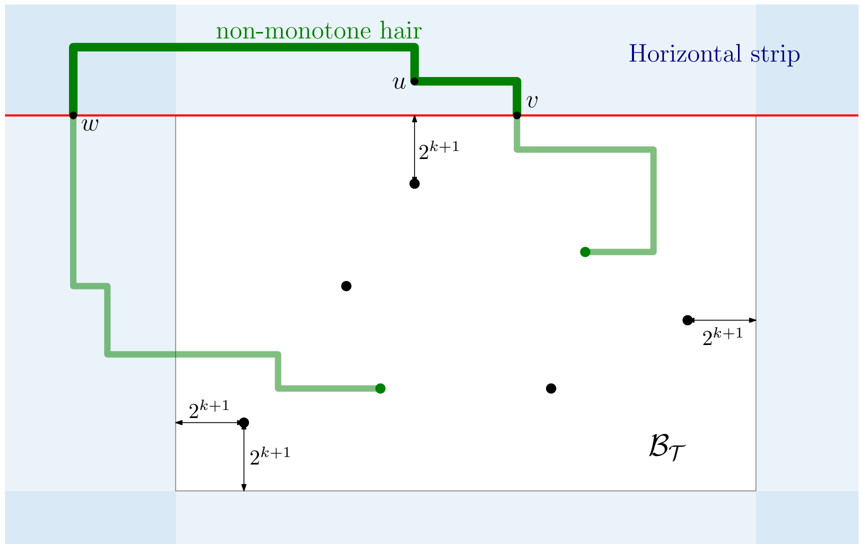


Figure 12: Vertices that are not in $\mathcal{B}_{\mathcal{T}}$ are irrelevant: if a routing uses one of these vertices, then this routing is not optimal. Otherwise, there would be a non-monotone hair in one of the extreme strips, which would contradict Lemma 3.4.

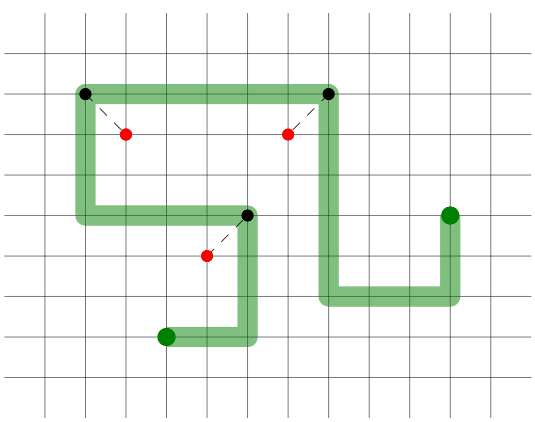

Figure 13: The red points indicate the inside vertex or their corresponding top-corner (black points).

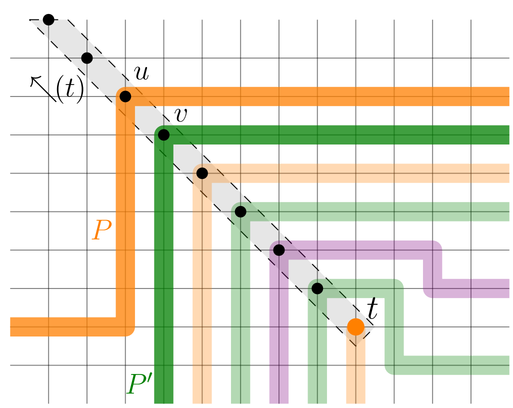


Figure 14: Proof of Lemma 3.5. The orange dot is a terminal $t$, and the set of dots represents the set $\nwarrow(t)$.

Definition 3.3. (Pushed-down Routing) *We say that a routing $\mathcal{P}$ is* pushed-down *if the inside vertex of every top-corner in $\mathcal{P}$ is on some path in $\mathcal{P}$.*

The routing shown in Fig. 1 is pushed-down.

Observation 3.1. *If there is a routing $\mathcal{P}$ of $\mathcal{T}$, then there exists a pushed-down routing $\mathcal{P}'$ for $\mathcal{T}$ with the same length as $\mathcal{P}$.*

*Proof.* Suppose that there is a path $P = (u_0, \dots, u_s) \in \mathcal{P}$ that has a top-corner $u_i$, for some $i$ with $1 \le i \le s-1$, such that its inside vertex $u_i'$ is not contained in any path in $\mathcal{P}$. Then, replace $P$ by the path $(u_0, \dots, u_{i-1}, u_i', u_{i+1}, \dots, u_s)$. This new family of paths is a routing and has the same length. Repeat this swap until for each top-corner, the corresponding inside vertex is in some path of the routing. We now argue that this procedure eventually stops. Indeed, defining the potential of a routing as the sum of the $y$-coordinates of the vertices of the paths in the routing. Then, each swap diminishes the potential by one. This implies that the shortest routing with minimum potential is pushed-down. □

Given a vertex $t$, we denote $\nwarrow(t) := \{(x(t) - i, y(t) + i), i \in \mathbb{N}\} \subset \mathbb{Z}^2$ and $\nearrow(t) := \{(x(t) + i, y(t) + i), i \in \mathbb{N}\} \subset \mathbb{Z}^2$. See Fig. 14.

Lemma 3.5. *Let $\mathcal{P}$ be a pushed-down routing with simple paths. If $u$ is a $\ulcorner$-corner (a $\urcorner$-corner) of some path $P \in \mathcal{P}$, then there exists a terminal $t \in \mathcal{T}$, such that $u \in \nwarrow(t)$ ($u \in \nearrow(t)$, resp.).*

*Proof.* We only treat the case when $u$ is a $\ulcorner$-corner, the other case is symmetric. See Fig. 14 for an illustration of the proof. Let $v$ denote the inside vertex of $u$. Since $\mathcal{P}$ is pushed-down, there exists $P' \in \mathcal{P}$ such that $v \in P'$. Since $P$ is simple we have $P \neq P'$. If $v \in \mathcal{T}$, then we have proved what we wanted. Otherwise, since $P$ and $P'$ are disjoint, $v$ is necessarily a $\ulcorner$-corner. Notice that $u \in \nwarrow(t)$ for some vertex $t$ if and only if $v \in \nwarrow(t)$. Thus, we repeat the same argument with $u = v$. Since $\mathcal{P}$ has a finite length, we will eventually reach a terminal. □

**3.4 Proof of Lemma 3.1** We have already proved (Corollary 3.1) that if a routing uses a vertex that is not in $\mathcal{B}_\mathcal{T}$, then this routing is not optimal. In this section, we show that it is also the case if the routing uses a point in $\mathcal{Z}_\mathcal{T}$. Recall that $\mathcal{Z}_\mathcal{T} = \{u \in \mathbb{Z}^2 \mid d(u, \mathcal{V}_\mathcal{T} \cup \mathcal{H}_\mathcal{T}) > (4k+5)2^k\}$, where $\mathcal{V}_\mathcal{T}$ and $\mathcal{H}_\mathcal{T}$ are respectively the sets of horizontal and vertical lines containing terminals. Unlike in Section 3.2, we need to assume here that the routing considered is pushed-down.

Lemma 3.6. *Let $\mathcal{P}$ be a pushed-down routing that uses a vertex in $\mathcal{Z}_\mathcal{T}$. Then, $\mathcal{P}$ is not optimal.*

This lemma, together with Observation 3.1 and Corollary 3.1 implies the Structural Lemma.

*Proof.* [Proof of Lemma 3.6.] First, we can assume that paths in $\mathcal{P}$ are simple. Otherwise we can easily obtain a strictly shorter routing. Let $u \in \mathcal{Z}_\mathcal{T}$ denote the vertex used by $\mathcal{P}$. We prove that $\mathcal{P}$ is not optimal by showing that there is either a simple shortcut or a shortenable window. See Figure 15 for visual support of the proof.

The vertex $u$ is contained in the intersection of a horizontal strip $S_h$ and a vertical strip $S_v$. Indeed, notice that for all $k \ge 1$, we have $(4k+5)2^k > 2^{k+1}$. Let $\mathcal{Z}'$ be the subset of $S_h \cap S_v$ formed by the vertices that are at distance of at least $2^{k+1}$ from the boundary of $S_h \cap S_v$. Notice that $u \in \mathcal{Z}'$. Let $Q$ be the hair in $\mathcal{Z}'$ that contains $u$. We may assume that $Q$ is monotone, otherwise by Lemma 3.4, we already know that $\mathcal{P}$ is not optimal.

We now show that $Q$ has at most $2k$ top-corners. We treat the case when $Q$ is a $(\llcorner\urcorner)$-monotone path; the other case is symmetric. In this case, all top-corners of $Q$ are $\urcorner$-corners. Since $\mathcal{P}$ is pushed-down, we know from Lemma 3.5 that for each $\urcorner$-corner $v$ of $Q$, there exists a terminal $t_v$ such that $v \in \nearrow(t_v)$. Since $Q$ is monotone, for any two $\urcorner$-corners of $Q$, $v$ and $v'$, we have $t_v \neq t_{v'}$. Thus, since $|\mathcal{T}| = 2k$, we conclude that $Q$ has at most $2k$ top-corners. In particular, since corners of $Q$ are alternating between top and bottom ($\llcorner$ or $\urcorner$) corners, $Q$ has at most $4k$ corners, and thus at most $4k+1$ segments. Since $Q$ connects two points in the boundary of $\mathcal{Z}'$ and contains $u$, that is at a distance at least $(4k+1)2^k$ from the boundary, the length of $Q$ is at least $(4k+1)2^{k+1}$. Thus, by the pigeonhole principle, there is a segment $C$ of $Q$ of length at least $2^{k+1}$.

Based on this long segment $C$, we now construct a long sequence of stacked segments (similar as the proof of Lemma 3.4) in order to identify a shortenable window. If the endpoints of $C$ are corners of $Q$, then we define

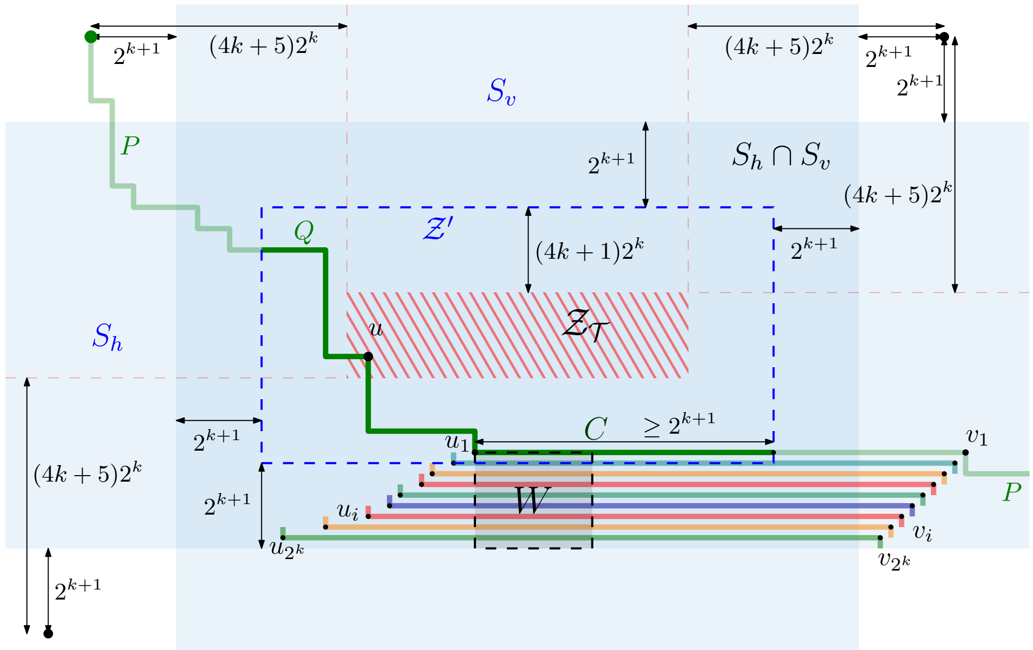


Figure 15: Proof of Lemma 3.6. The goal is to show the existence of a shortenable window $W$. To lighten the figure, some paths in $\mathcal{P}$ are not represented.

$u_1$ and $v_1$ to be these vertices, where $v_1$ is a top-corner. Otherwise, one endpoint of $C$ is an endpoint of $Q$ (like in Fig. 15). Let $P \in \mathcal{P}$ be the path that contains $Q$. Let $u_1$ and $v_1$ denote the consecutive corners of $P$ such that $C$ is contained in the segment between $u_1$ and $v_1$ (see Fig. 15). Without loss of generality, we assume that $v_1$ is the top-corner. Notice that one of strips $S_h$ or $S_v$ contains both $u_1$ and $v_1$. Also notice that either $u_1$ or $v_1$ is contained in $\mathcal{Z}'$ (or both); suppose $u_1 \in \mathcal{Z}'$; then for each terminal $t \in \mathcal{T}$, we have $|x(t) - x(u_1)| \geq 2^{k+2}$ and $|y(t) - y(u_1)| \geq 2^{k+2}$.

We now construct the shortenable window. We assume that $y(u_1) = y(v_1)$ and $x(u_1) < x(v_1)$. Other cases are treated symmetrically. Notice that this implies that $u_1$ and $v_1$ are respectively a $\llcorner$ and a $\neg$-corner, and are both contained in $S_h$. Let $W = [x(u_1), x(u_1) + 2^k + 1] \times [y(u_1) - 2^k, y(u_1)]$. We claim either $\mathcal{P}$ is not optimal, or $W$ is a horizontal shortenable window for $\mathcal{P}$. By Lemma 3.2, the second case implies that $\mathcal{P}$ is not optimal.

The proof that $W$ is an horizontal shortenable window for $\mathcal{P}$ is similar to the proof of Lemma 3.4. Suppose that $\mathcal{P}$ is optimal. We construct two sequences of vertices $(u_i)_{i=1}^{2^k}$ and $(v_i)_{i=1}^{2^k}$ such that, for all $i$, $1 \leq i \leq 2^k$,

(i) $u_i$ and $v_i$ are two consecutive corners of some path $P_i$ in $\mathcal{P}$,

(ii) $u_i$ is a $\llcorner$-corner and $v_i$ is a $\neg$-corner,

(iii) if $i \geq 2$, then $y(u_i) = y(v_i) = y(u_{i-1}) - 1$,

(iv) if $i \geq 2$, then $x(u_i) \leq x(u_{i-1}) - 1$ and $x(v_i) = x(v_{i-1}) - 1$.

It is not difficult to see that the existence of two such sequences implies that $W$ is a horizontal shortenable window for $\mathcal{P}$ (see Definition 3.2).

For $i = 1$, properties (i)-(iv) hold by assumption. Suppose that we have constructed $u_1, \ldots, u_{i-1}$ and $v_1, \ldots, v_{i-1}$ that satisfy conditions (i)-(iv) for some $i \geq 2$. Since $v_{i-1}$ is a $\neg$-corner, and the routing is pushed-down, there is a path $P_i \in \mathcal{P}$ that contains the inside vertex $v_i := (x(v_{i-1}) - 1, y(u_{i-1}) - 1)$ of $v_{i-1}$. Since $P_i$ is simple, we have $P_i \neq P_{i-1}$. Moreover, $v_i \notin \mathcal{T}$, as otherwise, applying (iii) for $j = 1, \ldots, i$, that would imply that there is a terminal $t$, such that $|y(t) - y(u_1)| = i - 1 < 2^{k+1}$, which is a contradiction with what precedes (specifically that $u_1 \in \mathcal{Z}'$). Therefore, since $P_i$ and $P_{i-1}$ are disjoint, $v_i$ is necessarily a $\neg$-corner of $P_i$. Let $u_i$ denote the corner of $P_i$ that is consecutive to $v_i$ on its left, i.e., $x(u_i) < x(v_i)$ and $y(u_i) = y(v_i)$ (notice that $u_i \notin \mathcal{T}$, as otherwise that would again

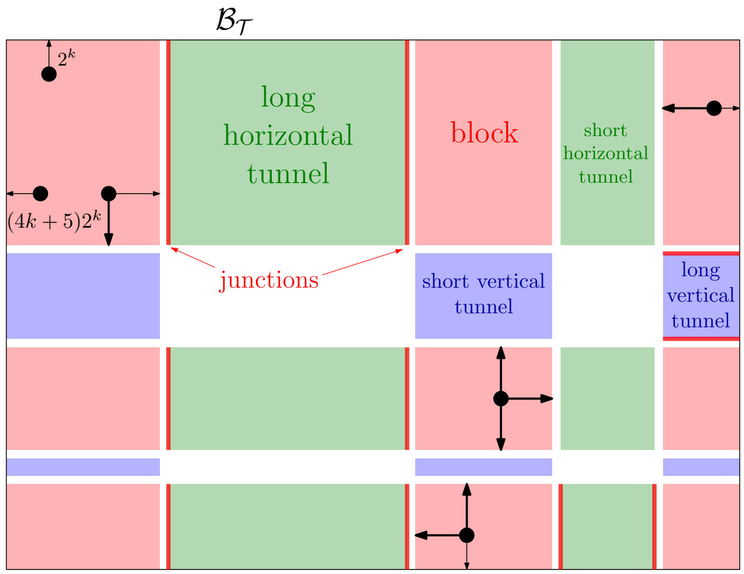


Figure 16: The partition of the relevant vertices into blocks (pink), horizontal tunnels (green), and vertical tunnels (blue). Red lines indicate the junctions of long tunnels. Arrows indicate distances: thin ones for distance $2^k$, and thick ones for distance $(4k+5)2^k$.

contradict the fact that $u_1 \in \mathcal{Z}'$). We know by applying (iii), that $u_i$ and $v_i$ are both contained in $S_h$ and thus by Lemma 3.4, the hair of $P_i$ in $S_h$ is monotone. This implies that $u_i$ is a $\llcorner$-corner of $P_i$. In particular, since $P_i$ and $P_{i-1}$ are disjoint, we must have $x(u_i) \leq x(u_{i-1}) - 1$. We have established the recurrence. This finishes the proof of Lemma 3.6. $\square$

This completes the proof of the Structural Lemma.

## 4 The Algorithm

In this section, we present a fixed-parameter algorithm that given a set $\mathcal{T} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ of $k$ pairs of terminals, computes an optimal routing for $\mathcal{T}$.

THEOREM 4.1. *There is an algorithm that computes an optimal routing for $\mathcal{T}$ in time $k^{2^{O(k)}} \cdot O([\mathcal{T}])$, where $[\mathcal{T}]$ is the number of bits to encode $\mathcal{T}$ in binary.*

We remark that our structural Lemma implies that the treewidth of the grid induced by the relevant vertices is bounded by a function of $k$. This follows from the fact that the face-vertex incidence graph diameter of this planar graph is bounded by a function of $k$ [43]. Since the classical dynamic program for the classic $k$-disjoint paths problem on bounded treewidth graphs can easily be adapted to the shortest version, we directly obtain an FPT algorithm where the polynomial term in the running time corresponds to the number of relevant vertices. In this section, we improve this bound by showing an algorithm where the polynomial term in the running time is actually linear in the total number of bits used to encode the terminals.

To describe the algorithm, we first partition the set $\mathcal{B}_{\mathcal{T}} \setminus \mathcal{Z}_{\mathcal{T}}$ into *tunnels* and *blocks*. Let a *horizontal street* be a connected component of the set

$$\mathcal{R}_h \coloneqq \mathcal{B}_{\mathcal{T}} \cap \{v \in \mathbb{Z}^2 \mid \exists t \in \mathcal{T}, |y(t) - y(v)| \leq (4k+5)2^k\}.$$

Similarly, let a *vertical street* be a connected component of the set

$$\mathcal{R}_v \coloneqq \mathcal{B}_{\mathcal{T}} \cap \{v \in \mathbb{Z}^2 \mid \exists t \in \mathcal{T}, |x(t) - x(v)| \leq (4k+5)2^k\}.$$

The intersection between a vertical street and a horizontal street is called a *block*. We call a *horizontal tunnel*, a connected component of $\mathcal{R}_h \setminus \mathcal{R}_v$ and *vertical tunnel*, a connected component of $\mathcal{R}_v \setminus \mathcal{R}_h$. It is easy to see that blocks, horizontal tunnels and vertical tunnels form a partition of $\mathcal{B}_{\mathcal{T}} \setminus \mathcal{Z}_{\mathcal{T}}$.

We say that a horizontal (vertical) tunnel is *long* if its width (height, resp.) is at least its height (width, resp.). Otherwise, we say that it is *short*.

For each long horizontal tunnel, we call *left junction* (*right junction*), the set of vertices of the tunnel with leftmost (rightmost, resp.) $x$-coordinates. We define similarly the *top junction* and the *bottom junction* of a long vertical tunnel. See Fig. 16.

We now define a configuration. Let $V'$ denote the set of vertices that are either contained in a block, in a short tunnel or in a junction of a long tunnel. A *configuration* is a map $c : V' \to \{0, 1, \dots, k\}$.

Let $c$ be a configuration and let $J$ and $J'$ be the two junctions of a given long horizontal tunnel (long vertical tunnel). Let us order the sets $\{v \in J \mid c(v) > 0\} = \{v_1, \dots, v_t\}$ and $\{v' \in J' \mid c(v') > 0\} = \{v'_1, \dots, v'_{t'}\}$ from top to bottom (from left to right, resp.). See Fig. 17. We say that $J$ and $J'$ are *compatible* (with respect to $c$), if $t = t'$ and if for each $i \in [1, t]$, we have $c(v_i) = c(v'_i)$. In this case we say that for each $i \in [1, t]$, the vertices $v_i$ and $v'_i$ are *linked*.

Claim 4.1. *Let $J$ and $J'$ be the two junctions of a given long horizontal tunnel (long vertical tunnel). Let $t$ be an integer smaller than the height (width, resp.) of the tunnel. Let $\{v_1, \dots, v_t\} \subseteq J$ a sequence of vertices of $J$ ordered from top to bottom (from left to right, resp.), and let $\{v'_1, \dots, v'_t\} \subseteq J'$ a sequence of vertices of $J'$ ordered from top to bottom (from left to right, resp.).*

*Then, there exists a family $(Q_j)_{j=1}^t$ of pairwise disjoint paths, such that for each $j$, $1 \le j \le t$: (i) the endpoints of $Q_j$ are $v_j$ and $v'_j$, (ii) $Q_j$ is contained in the tunnel, (iii) $Q_j$ is a shortest path, (iv) $Q_j$ has at most two corners, and (v) the internal vertices of $Q_j$ are not contained in the junctions.*

*Proof.* We prove the Claim when the tunnel is horizontal and $J$ is the left junction. Other cases are symmetric. For each $j \in [1, t]$, let $c_j = (x_j, y(v_j))$ and $c'_j = (x_j, y(v'_j))$ where $x_j = x(v_j) + j$ if $y(v_j) \le y(v'_j)$ and $x_j = x(v'_j) + t + 1 - j$ otherwise. See Fig. 17. Let $Q_j$ be the path from $v_j$ to $v'_j$ with corners $c_j$ and $c'_j$. It is easy to see that this path is well-defined and satisfies the desired properties. Moreover, it is clear that these paths are pairwise disjoints and contained inside the tunnel. ∎

We say that a configuration $c$ is *valid*, if the two junctions of each long tunnel are compatible and if for each $i \in [1, k]$, the set of vertices $V'_i := \{v \in V' \mid c(v) = i\} = \{v_0, \dots, v_s\}$ can be ordered in a way that $v_0 = s_i$, $v_s = t_i$ and for each $j \in [0, \ell - 1]$, $v_j$ and $v_{j+1}$ are either at distance 1 or are linked (within some long tunnel). In this case we say that the *length* of $P_i$ is $\ell(P_i) = \sum_{j=0}^{s-1} d(v_j, v_{j+1})$. The *length* of a valid configuration is $\sum_{i=1}^{k} \ell(P_i)$.

**The algorithm.** The algorithm of Theorem 4.1 works as follows: enumerate all possible configurations and return a valid configuration of shorter length, if such routing exists; otherwise, return that there is no routing for $\mathcal{T}$.

Notice that, as described, the algorithm returns an integer (that we later prove to be the minimum length of a routing for $\mathcal{T}$, if such a routing exists). We explain how to construct the paths associated to that valid configuration. Then, we show the correctness of the algorithm and finally analyze its running time.

**Construction of the paths.** Let $c$ be a valid configuration. We construct a routing $\mathcal{P} = (P_1, \dots, P_k)$ from $c$ as follows. Let $\mathcal{Q}$ be the union of the paths obtained from Claim 4.1 over each long tunnel, where the sequences $\{v_1, \dots, v_t\}$ and $\{v'_1, \dots, v'_t\}$ for each pair of junctions $J, J'$ correspond to the sets $\{v \in J \mid c(v) > 0\}$ and $\{v' \in J' \mid c(v') > 0\}$. If a path $Q \in \mathcal{Q}$ connects two vertices $v$ and $v'$ such that $c(v) = c(v') = i$ for some $i$ with $1 \le i \le k$, we say that $Q$ is of color $i$. For each $i$, $1 \le i \le k$, we construct $P_i$ as the union of $\{v \in V' \mid c(v) = i\}$ and the union of paths $Q \in \mathcal{Q}$ of color $i$. It is not difficult to see that $\mathcal{P}$ is a routing for $\mathcal{T}$ whose length is the same as the length of $c$.

**Correctness of the algorithm.** Assume that $\mathcal{P}^* = (P_1, \dots, P_k)$ is an optimal routing of $\mathcal{T}$. We construct a valid configuration $c$, of the same length.

From the Structural Lemma, we know that we can assume that the paths of $\mathcal{P}^*$ do not contain any irrelevant vertex. In particular, these paths are contained in the union of blocks and tunnels. We define $c$ as follows. Let $i \in [1, k]$ and $v \in V' \cap P_i$. If $v$ is contained in a block, a short tunnel, or is contained in a junction and has a neighbor in $P_i$ that is contained in a block, then $c(v) = i$, otherwise, $c(v) = 0$.[2] We argue that $c$ is a valid configuration and

[2]the intuition here is that we want to avoid two consecutive vertices on the same junction to have color $i$, otherwise this may results in two non-compatible junctions.

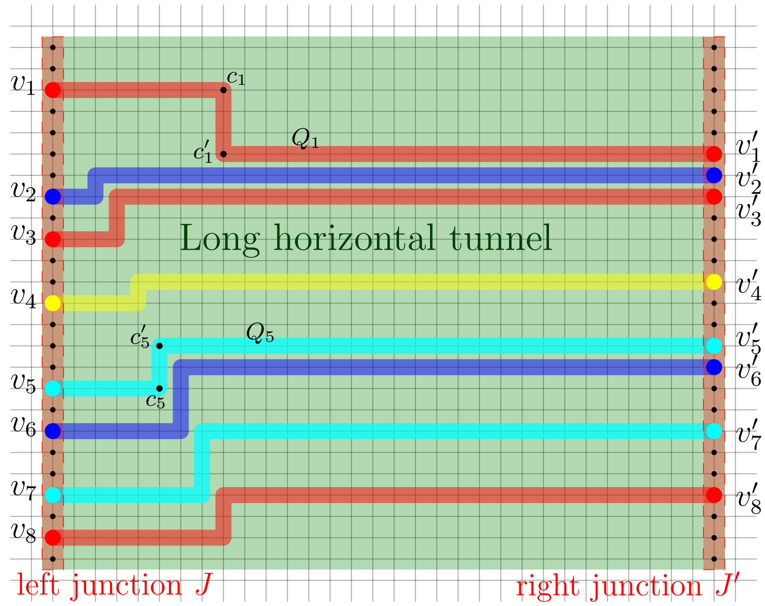


Figure 17: Two compatible junctions. Pairs of linked vertices can be connected with a family of disjoint shortest paths contained in the tunnel (see Claim 4.1).

that it has the same length as $\mathcal{P}$.

Fix $i \in [1, k]$, and let $V'_i = \{v_1, \dots, v_s\} \subseteq P_i$ be the set of vertices such that $c(v) = i$, ordered from the source of $P_i$ to its sink. Suppose that there is an index $j \in [1, s-1]$ such that $d(v_j, v_{j+1}) > 1$. We show that $v_j$ and $v_{j+1}$ are linked (within some tunnel). First, it is clear that the interior of the path between $v_j$ and $v_{j+1}$ is contained in some tunnel (since $P_i$ does not contain any irrelevant vertex). Suppose for a contradiction that $v_j$ and $v_{j+1}$ are contained in the same junction of that tunnel. Let $u$ and $u'$ be the vertices on $P_i$ and are respectively adjacent to $v_j$ and $v_{j+1}$ and that are contained in a block. These vertices exist by definition of $c$ and are contained in the same block (this follows from our assumption). Further, they have the same $x$-coordinate (if the tunnel is horizontal), or the same $y$-coordinate (otherwise). This implies that the subpath of $P_i(u, u')$ between $u$ and $u'$ is not a shortest path (or equivalently not monotone). This is a contradiction with Lemma 3.4 since the path $P_i(u, u')$ is contained in a strip. Thus, $v_j$ and $v_{j+1}$ are linked. In particular, the length of $P_i(v_j, v_{j+1})$ is equal to $d(v_j, v_{j+1})$. We have shown that $c$ is a valid configuration of the same length as $\mathcal{P}^*$.

**Running time.** To give the running time of our algorithm, we need to calculate the size of $V'$. The sum of the widths (heights) of the blocks of distinct $x$-coordinates (distinct $y$-coordinates) is at most $k \cdot (1 + 2 \cdot (4k+5)2^k) = 2^{O(k)}$. Thus there are at most $(2^{O(k)})^2 = 2^{O(k)}$ vertices contained into blocks. Further, there are at most $(2^{O(k)})^2 = 2^{O(k)}$ vertices contained into long tunnels. Finally, the number of vertices contained in junctions is $2^{O(k)}$. Thus $|V'| = 2^{O(k)}$. This implies that the number of configurations is $(k+1)^{|V'|} = k^{2^{O(k)}}$. All vertices in $V'$ are contained in a bounding box that is at a distance at most $2^k$ from the coordinates of each vertex and can be encoded using $O(2(b + \log(2^k))) = O([\mathcal{T}])$ bits, where $b$ is the maximum number of bits of a coordinate of a point in $\mathcal{T}$. For each configuration, we can decide in time $\mathrm{poly}(2^{O(k)})$ whether this configuration is valid or not. Hence, the overall running time is $k^{2^{O(k)}} O([\mathcal{T}])$.

This completes the proof of Theorem 4.1.

## 5 Conclusion

In this paper, we consider the shortest $k$-disjoint paths problem in a grid and show that the problem admits a fixed parameter tractable algorithm parameterized by the number of terminals. Although our result is primarily stated for the case of an infinite rectangular grid our result applies to any finite rectangular grid as well. While we resolve the complexity question of the shortest $k$-disjoint paths problem in a grid, there are several intriguing questions that remain open. The main open question arising from our work would be to resolve the complexity status of shortest $k$-disjoint paths in general planar graphs. The only known positive results, in this case, either consider a very small number of terminals or put restrictions on their placement in the graph.

Another interesting question is to extend our algorithm for the case of grids *with holes*. As presented, our techniques rely on the assumption that we work on a complete grid. For instance, the definition of a simple shortcut, the definition of pushed-down routings, and the procedure to identify shortenable windows. That said, we believe that our techniques potentially could be extended to argue that, there exist irrelevant vertices even in the case of holes. For instance, one might still be able to prove the existence of a bounding box at the distance $O(2^k)$ around the terminals and some holes: first discard holes that are far from the bounding box of the terminals since they do not perturb the solution, and then take a bounding box that is far enough from the terminals and the remaining holes. Inside this bounding box, points with $x$-coordinates (and $y$-coordinates) distant enough from the $x$-coordinates (and $y$-coordinates) of the terminals and the holes would still be irrelevant. Still, in the analysis of the running time, one difficulty with the holes is how to encode them in the problem instance. An interesting future work could be to study the parameterized complexity of the problem when parameterized by the number of terminals plus the number of corners of the holes.

## References


[1] I. Adler, S. G. Kolliopoulos, P. K. Krause, D. Lokshtanov, S. Saurabh, and D. M. Thilikos, *Irrelevant vertices for the planar disjoint paths problem*, Journal of Combinatorial Theory, Series B, 122 (2017), pp. 815–843.

[2] I. Adler, S. G. Kolliopoulos, and D. M. Thilikos, *Planar disjoint-paths completion*, Algorithmica, 76 (2016), pp. 401–425.

[3] A. Aggarwal, J. Kleinberg, and D. P. Williamson, *Node-disjoint paths on the mesh and a new trade-off in VLSI layout*, SIAM Journal on Computing, 29 (2000), pp. 1321–1333.

[4] A. Björklund and T. Husfeldt, *Counting shortest two disjoint paths in cubic planar graphs with an NC algorithm*, in 29th International Symposium on Algorithms and Computation, ISAAC 2018, vol. 123 of LIPIcs, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2018, pp. 19:1–19:13.

[5] ———, *Shortest two disjoint paths in polynomial time*, SIAM Journal on Computing, 48 (2019), pp. 1698–1710.

[6] G. Borradaile, A. Nayyeri, and F. Zafarani, *Towards single face shortest vertex-disjoint paths in undirected planar graphs*, in 23rd Annual European Symposium on Algorithms, ESA, 2015, pp. 227–238.

[7] U. Brandes, G. Neyer, and D. Wagner, *Edge-disjoint paths in planar graphs with short total length*. 1996.

[8] W. Chan and F. Y. L. Chin, *Efficient algorithms for finding the maximum number of disjoint paths in grids*, J. Algorithms, 34 (2000), pp. 337–369.

[9] M. Chudnovsky, A. Scott, and P. D. Seymour, *Disjoint paths in unions of tournaments*, Journal of Combinatorial Theory, Series B, 135 (2019), pp. 238–255.

[10] J. Chuzhoy and D. H. K. Kim, *On approximating node-disjoint paths in grids*, in International Conferences on Approximation, Randomization, and Combinatorial Optimization, APPROX/RANDOM 2015, vol. 40 of LIPIcs, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2015, pp. 187–211.

[11] J. Chuzhoy, D. H. K. Kim, and S. Li, *Improved approximation for node-disjoint paths in planar graphs*, in Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing, STOC 2016, ACM, 2016, pp. 556–569.

[12] J. Chuzhoy, D. H. K. Kim, and R. Nimavat, *Improved approximation for node-disjoint paths in grids with sources on the boundary*, in 45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, vol. 107 of LIPIcs, 2018, pp. 38:1–38:14.

[13] J. Chuzhoy, D. H. K. Kim, and R. Nimavat, *Almost polynomial hardness of node-disjoint paths in grids*, Theory of Computing, 17 (2021), pp. 1–57.

[14] J. Chuzhoy, D. H. K. Kim, and R. Nimavat, *New hardness results for routing on disjoint paths*, SIAM Journal on Computing, 51 (2022), pp. 17–189.

[15] É. Colin de Verdière and A. Schrijver, *Shortest vertex-disjoint two-face paths in planar graphs*, ACM Transactions on Algorithms, 7 (2011), pp. 19:1–19:12.

[16] M. Cygan, D. Marx, M. Pilipczuk, and M. 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, IEEE Computer Society, 2013, pp. 197–206.

[17] S. Datta, S. Iyer, R. Kulkarni, and A. Mukherjee, *Shortest k-disjoint paths via determinants*, in 38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2018, vol. 122 of LIPIcs, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2018, pp. 19:1–19:21.

[18] T. Eilam-Tzoreff, *The disjoint shortest paths problem*, Discrete Applied Mathematics, 85 (1998), pp. 113 – 138.

[19] J. Erickson and A. Nayyeri, *Shortest non-crossing walks in the plane*, in 22nd Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, SIAM, 2011, pp. 297–208.

[20] S. Even, A. Itai, and A. Shamir, *On the complexity of time table and multi-commodity flow problems*, in 16th Annual Symposium on Foundations of Computer Science, SFCS 1975, 1975, pp. 184–193.

[21] M. Formann and F. Wagner, *The VLSI layout in various embedding models*, in 16rd International Workshop on Graph-Theoretic Concepts in Computer Science, WG 1990, vol. 484 of Lecture Notes in Computer Science, Springer, 1990, pp. 130–139.

[22] S. Fortune, J. E. Hopcroft, and J. Wyllie, *The directed subgraph homeomorphism problem*, Theoretical Computer Science, 10 (1980), pp. 111–121.

[23] A. Frank, *Disjoint paths in a rectilinear grid*, Combinatorica, 2 (1982), pp. 361–371.

[24] A. Frank, *Packing paths, circuits, and cuts — a survey*, in Paths, Flows and VLSI-Layouts, B. Korte, L. Lovasz, H. Proemel, and A. Schrijver, eds., Springer, Berlin, 1990, pp. 47–100.

[25] A. S. Jobson, A. E. Kézdy, and J. Lehel, *Path-pairability of infinite planar grids*, Period. Math. Hung., 83 (2021), pp. 220–232.

[26] R. M. Karp, *On the computational complexity of combinatorial problems*, Networks, 5 (1975), pp. 45–68.

[27] M. Kaufmann, *A linear-time algorithm for routing in a convex grid*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 9 (1990), pp. 180–184.

[28] M. Kaufmann and K. Mehlhorn, *Routing problems in grid graphs*, in Paths, Flows and VLSI-Layout, Springer, Heidelberg, 1990, pp. 165–184.

[29] M. Kaufmann and K. Mehlhorn, *On local routing of two-terminal nets*, Journal of Combinatorial Theory, Series B, 55 (1992), pp. 33–72.

[30] K. Kawarabayashi, *The disjoint paths problem: Algorithm and structure*, in 5th International Workshop on Algorithms and Computation, WALCOM 2011, vol. 6552 of Lecture Notes in Computer Science, Springer, 2011, pp. 2–7.

[31] K. Kawarabayashi, Y. Kobayashi, and B. A. Reed, *The disjoint paths problem in quadratic time*, Journal of Combinatorial Theory, Series B, 102 (2012), pp. 424–435.

[32] Y. Kobayashi and C. Sommer, *On shortest disjoint paths in planar graphs*, Discrete Optimization, 7 (2010), pp. 234–245.

[33] Y. Kobayashi and T. Terao, *One-face shortest disjoint paths with a deviation terminal*, in 33rd International Symposium on Algorithms and Computation, ISAAC 2022, vol. 248 of LIPIcs, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2022, pp. 47:1–47:15.

[34] M. Kramer and J. van Leeuwen, *The complexity of wire routing and finding minimum area layouts for arbitrary vlsi circuits*, Advances in Computer Research, 2 (1984), pp. 129–146.

[35] W. Lochet, *A polynomial time algorithm for the k-disjoint shortest paths problem*, in 32nd ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, SIAM, 2021, pp. 169–178.

[36] D. Lokshtanov, P. Misra, M. Pilipczuk, S. Saurabh, and M. Zehavi, *An exponential time parameterized algorithm for planar disjoint paths*, in 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, ACM, 2020, pp. 1307–1316.

[37] J. F. Lynch, *The equivalence of theorem proving and the interconnection problem*, SIGDA Newsl., 5 (1975), pp. 31–36.

[38] D. Marx, *Eulerian disjoint paths problem in grid graphs is np-complete*, Discrete Applied Mathematics, 143 (2004), pp. 336–341.

[39] K. Mehlhorn and F. P. Preparata, *Routing through a rectangle*, Journal of the ACM, 33 (1986), pp. 60–85.

[40] M. Middendorf and F. Pfeiffer, *On the complexity of the disjoint paths problems*, Combinatorica, 13 (1993), pp. 97–107.

[41] S. Nagaraja, *Non-crossing rectilinear shortest minimum bend paths in the presence of rectilinear obstacles*, (2018), p. 82 to 91.

[42] R. Y. Pinter, *On routing two-point nets across a channel*, in 19th Design Automation Conference, DAC 1982, ACM/IEEE, 1982, pp. 894–902.

[43] N. Robertson and P. D. Seymour, *Graph minors. III. Planar tree-width*, Journal of Combinatorial Theory, Series B, 36 (1984), pp. 49–64.

[44] ———, *Graph minors. VI. Disjoint paths across a disc*, Journal of Combinatorial Theory, Series B, 41 (1986), pp. 115–138.

[45] ———, *Graph minors. VII. Disjoint paths on a surface*, Journal of Combinatorial Theory, Series B, 45 (1988), pp. 212–254.

[46] ———, *Graph minors. XIII. The disjoint paths problem*, Journal of Combinatorial Theory, Series B, 63 (1995), pp. 65 – 110.

[47] P. Scheffler, *A practical linear time algorithm for disjoint paths in graphs with bounded tree-width*, Technical Report 396, (1994).

[48] A. Schrijver, *Finding k disjoint paths in a directed planar graph*, SIAM Journal on Computing, 23 (1994), pp. 780–788.

[49] W. Schwärzler, *On the complexity of the planar edge-disjoint paths problem with terminals on the outer boundary*, Comb., 29 (2009), pp. 121–126.
[50] C. D. Thompson, *A Complexity Theory for VLSI*, PhD thesis, Carnegie Mellon University, USA, 1980.
[51] M. Tompa, *An optimal solution to a wire-routing problem (preliminary version)*, in 12th Annual ACM Symposium on Theory of Computing, STOC 1980, ACM, 1980, pp. 161–176.
[52] H. van der Holst and J. de Pina, *Length-bounded disjoint paths in planar graphs*, Discrete Applied Mathematics, 120 (2002), pp. 251 – 261.
[53] F. Wagner and B. Wolfers, *Short wire routing in convex grids*, in 2nd International Symposium on Algorithms, ISA 1991, vol. 557 of Lecture Notes in Computer Science, Springer, 1991, pp. 72–82.
[54] M. Włodarczyk and M. Zehavi, *Planar disjoint paths, treewidth, and kernels*, in 64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, IEEE, 2023, pp. 649–662.