On the complexity of minimum-link path problems

ON THE COMPLEXITY OF MINIMUM-LINK PATH PROBLEMS[*]

Irina Kostitsyna,[†] *Maarten Löffler,*[‡] *Valentin Polishchuk,*[§] *and Frank Staals*[¶]

Abstract. We revisit the minimum-link path problem: Given a polyhedral domain and two points in it, connect the points by a polygonal path with minimum number of edges. We consider settings where the vertices and/or the edges of the path are restricted to lie on the boundary of the domain, or can be in its interior. Our results include bit complexity bounds, a novel general hardness construction, and a polynomial-time approximation scheme. We fully characterize the situation in 2 dimensions, and provide first results in dimensions 3 and higher for several variants of the problem.

Concretely, our results resolve several open problems. We prove that computing the minimum-link *diffuse reflection path*, motivated by ray tracing in computer graphics, is NP-hard, even for two-dimensional polygonal domains with holes. This has remained an open problem [28] despite a large body of work on the topic. We also resolve the open problem from [41] mentioned in the handbook [29] (see Chapter 27.5, Open problem 3) and The Open Problems Project [17] (see Problem 22): "What is the complexity of the minimum-link path problem in 3-space?" Our results imply that the problem is NP-hard even on terrains (and hence, due to discreteness of the answer, there is no FPTAS unless P=NP), but admits a PTAS.

1 Introduction

The minimum-link path problem is fundamental in computational geometry [5, 27, 30, 33, 35, 38, 41, 49]. It concerns the following question: given a polyhedral domain D and two points s and t in D, what is the polygonal path connecting s to t that lies in D and has as few links as possible?

In this paper, we revisit the problem in a general setting which encompasses several specific variants that have been considered in the literature. First, we nuance and tighten results on the bit complexity involved in optimal minimum-link paths. Second, we present and apply a novel generic NP-hardness construction. Third, we extend a simple polynomial-time approximation scheme.

Concretely, our results resolve several open problems. We prove that computing the minimum-link *diffuse reflection path* in polygons with holes [28] is NP-hard, and we prove

[*]An abridged version of this paper appeared in the proceedings of the 32nd International Symposium on Computational Geometry in 2016.

[†]*Université libre de Bruxelles*, irina.kostitsyna@ulb.ac.be
[‡]*Utrecht University*, m.loffler@uu.nl
[§]*Linköping University*, valentin.polishchuk@liu.se
[¶]*Aarhus University*, f.staals@cs.au.dk

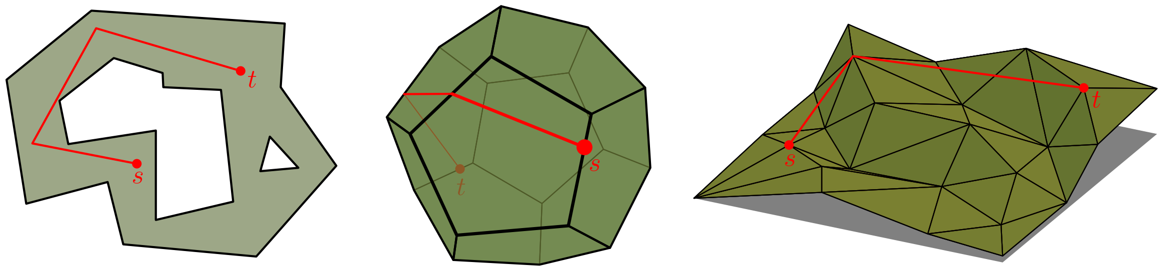

Figure 1: Left: $\mathsf{MinLinkPath}_{2,2}$ in a polygon with holes. Middle: $\mathsf{MinLinkPath}_{1,2}$ on a polyhedron. Right: $\mathsf{MinLinkPath}_{0,3}$ on a polyhedral terrain.

that the minimum-link path problem in 3-space [29] (Chapter 27.5, Open problem 3) is NP-hard (even for terrains). In both cases, there is no FPTAS unless P=NP, but there is a PTAS.

We use terms *links* and *bends* for edges and vertices of the path, saving the terms *edges* and *vertices* for those of the domain (also historically, minimum-link paths used to be called *minimum-bend* [51–53]).

1.1 Problem Statement, Domains and Constraints

Due to their diverse applications, many different variants of minimum-link paths have been considered in the literature. These variants can be categorized by two aspects. Firstly, the *domain* can take very different forms. We select several common domains, ranging from a simple polygon in 2D to complex scenes in full 3D or even in higher dimensions. Secondly, the links and bends of the solution paths are sometimes *constrained* to lie on the boundary of the domain, or bends may be restricted to vertices or edges of the domain. We now survey these settings in more detail.

Problem Statement. Let D be a closed connected d-dimensional polyhedral domain. For $0 \le a \le d$ we denote by $D|^a$ the a-skeleton of D, that is, its a-dimensional subcomplex. For instance, $D|^{d-1}$ is the boundary of D; $D|^0$ is the set of vertices of D. Note that $D|^a$ is not necessarily connected.

Definition 1. *We define $\mathsf{MinLinkPath}_{a,b}(D, s, t)$, for $0 \le a \le b \le d$ and $1 \le b$, to be the problem of finding a minimum-link polygonal path in D between two given points s and t, where the bends of the solution (and s and t) are restricted to lie in $D|^a$ and the links of the solution are restricted to lie in $D|^b$.*

Figure 1 illustrates several instances of the problem in different domains.

Domains. We recap the various settings that have been singled out for studies in computational geometry. We remark that we will not survey the rich field of path planning in rectilinear, or more generally, C-oriented worlds [1]; all our paths will be assumed to be unrestricted in terms of orientations of their links.

One classical distinction between working setups in 2D is *simple polygons* vs. *polygonal domains*. The former are a special case of the latter: simple polygons are domains without holes. Many problems admit more efficient solutions in simple polygons—loosely speaking, the golden standard is running time of $O(n)$ for simple polygons and of $O(n \log n)$ for polygonal domains of complexity n. This is the case, e.g., for the shortest path problem [31, 32]. For minimum-link paths, $O(n)$-time algorithms are known for simple polygons [27, 33, 49], but for polygonal domains with holes the fastest known algorithm runs in nearly quadratic time [41], which may be close to optimal due to 3SUM-hardness of the problem [38]. Even more striking is the difference in the watchman route problem (find a shortest path to see all of the domain), which combines path planning with visibility: in simple polygons the optimal route can be found in polynomial time [15, 19] while for domains with holes the problem cannot be approximated to within a logarithmic factor unless P=NP [40]. Finding minimum-link watchman route is NP-hard even for simple polygons [4].

In 3D, a *terrain* is a polyhedral surface (often restricted to a bounded region in the xy-projection) that is intersected only once by any vertical line. Terrains are traditionally studied in GIS applications and are ubiquitous in computational geometry [11, 39]. Minimum-link paths are closely related to visibility problems, which have been studied extensively on terrains [8, 9, 22, 34, 36, 48]. One step up from terrains, we may consider *simple* polyhedra (surfaces of genus 0), or *full 3D* scenes. Visibility has been studied in full 3D as well [20, 42, 50]. To our knowledge, minimum-link paths in higher dimensions have not been studied before (with the exception of [10] that considered rectilinear paths).

Constraints. In path planning on polyhedral surfaces or terrains, it is standard to restrict paths to the (terrain) surface. Minimum-link paths, on the other hand, have various geographic applications, ranging from feature simplification [30] to visibility in terrains [22]. In some of these applications, paths are allowed to live in free space, while bends are still restricted to the terrain. In the GIS literature, out of simplicity and/or efficiency concerns, it is common to constrain bends even further to vertices of the domain (or, even more severely, the terrain itself may restrict vertices to grid points, as in the popular *digital elevation map* (DEM) model; this, however, may lead to an arbitrarily high increase in the link distance).

In a vanilla minimum-link path problem the location of vertices (bends) of the path are unconstrained, i.e., they can occur anywhere in the free space. In the *diffuse reflection* model [5–7, 12, 28, 45] the bends are restricted to occur on the boundary of the domain. Studying this kind of paths is motivated by ray tracing in realistic rendering of 3D scenes in graphics, as light sources that can reach a pixel with fewer reflections make higher contributions to intensity of the pixel [11, 23]. Despite the 3D graphics motivation, all work on diffuse reflection has been confined to 2D polygonal domains, where the path bends are restricted to edges of the domain.

1.2 Representation and Computation

In computational geometry, the standard model of computation is the *real RAM*, which represents data as an infinite sequence of storage cells which can store any real number or integer. The model supports standard operations (such as addition, multiplication, or taking square-roots) in constant time. The real RAM is preferred for its elegance, but may not always be the best representation of physical computers. For example, the *floor* function is often allowed, which can be used to truncate a real number to the nearest integer, but points at a flaw in the model: if we were allowed to use it arbitrarily, the real RAM could solve PSPACE-complete problems in polynomial time [47]. In contrast, the *word RAM* stores a sequence of w-bit words, where $w \geq \log n$ (and n is the problem size). Data can be accessed arbitrarily, and standard operations, such as Boolean operations (`and`, `xor`, `shl`, ...), addition, or multiplication take constant time. There are many variants of the word RAM, depending on precisely which instructions are supported in constant time. The general consensus seems to be that any function in AC^0 is acceptable.[1] However, it is always preferable to rely on a set of operations as small, and as non-exotic, as possible. Note that multiplication is not in AC^0 [25]. Nevertheless, it is usually included in the word RAM instruction set [24]. The word RAM is much closer to reality, but complicates the analysis of geometric problems.

In many cases, the difference between the models is unimportant, as the real numbers involved in solving geometric problems are in fact algebraic numbers of low degree in a bounded domain, which can be described exactly with constantly many words. Path planning is notoriously different in this respect. Indeed, in the real RAM both the Euclidean shortest paths and the minimum-link paths in 2D can be found in optimal times. On the contrary, much less is known about the complexity of the problems in other models. For L_2-shortest paths the issue is that their length is represented by the sum of square roots and it is not known whether comparing the sum to a number can be done efficiently (if yes, one may hope that the difference between the models vanishes). Slightly more is known about minimum-link paths, for which the models are *provably* different: Kahan and Snoeyink [35] observed that the region of points reachable by k-link paths may have vertices needing $\Omega(k \log n)$ bits to describe. One of the results in this paper is the matching upper bound on the bit complexity of minimum-link paths.

Relatedly, when studying the computational complexity of geometric problems, it is often not trivial to show a problem is in NP. Even if a potential solution can be verified in polynomial time, if such a solution requires real numbers that cannot be described succinctly, the set of solutions to try may be too large. Recently, there has been some interest in computational geometry in showing problems are in NP [21] (see also [46]).

A common practical approach to avoiding bit complexity issues is to approximate the problem by restricting solutions to use only vertices of the input. In minimum-link paths, this corresponds to $\mathsf{MinLinkPath}_{0,b}$. In this case, one can easily compute a minimum-link path by a breadth-first search in the visibility graph of the vertices. This results in an $O(n^2)$ time algorithm in 2D (using [43]), and an $O(n^{7/3} \operatorname{polylog} n)$ time algorithm in 3D (using [2]; for terrains this can be improved slightly [16]). In both cases the running time is dominated

[1] AC^0 is the class of all functions $f : \{0,1\}^* \to \{0,1\}^*$ that can be computed by a family of circuits $(C_n)_{n\in\mathbb{N}}$ with the following properties: (i) each C_n has n inputs; (ii) there exist constants a, b, such that C_n has at most an^b gates, for $n \in \mathbb{N}$; (iii) there is a constant d such that for all n the length of the longest path from an input to an output in C_n is at most d (i.e., the circuit family has bounded depth); (iv) each gate has an arbitrary number of incoming edges (i.e., the *fan-in* is unbounded).

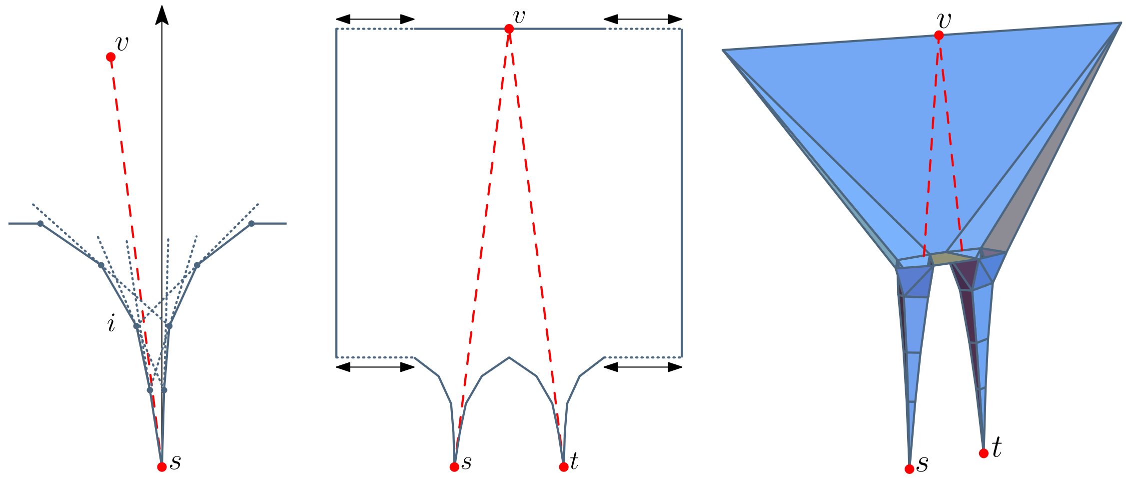

Figure 2: $\mathsf{MinLinkPath}_{0,b}$ may be a factor $\Omega(n)$ worse than $\mathsf{MinLinkPath}_{1,b}$. Left: A construction of a "trench". The only vertices visible from the vertices in layer i are in the previous layer, $i-1$, and in the next layer, $i+1$.; Middle: A polygon with two trenches; horizontal edges are wide enough such that the ends of the top edge are not visible from the inner vertices in the trenches. $\mathsf{MinLinkPath}_{0,2}(s,t)$ requires $\Omega(n)$ vertices whereas $\mathsf{MinLinkPath}_{1,2}(s,t)$ has two links; Right: The 3D construction of the trenches: $\mathsf{MinLinkPath}_{0,3}(s,t)$ requires $\Omega(n)$ vertices whereas $\mathsf{MinLinkPath}_{1,3}(s,t)$ has two links.

by the time it takes to construct the visibility graph. However, a simple example in Figure 2 shows that the number of links in $\mathsf{MinLinkPath}_{0,b}$ may be a linear factor higher than when considering less restricted geometric versions.

In this paper we explore the computational and algebraic complexity of the minimum-link path problems in 2D and 3D under the word RAM computational model, and the issues rising from the clash of geometry and the limited capacity of the word RAM for storing precise numbers.

1.3 Results

We give hardness results and approximation algorithms for various versions of the minimum-link path problem. Specifically,

- In Section 2 we give an $\Omega(n \log n)$ lower bound on the bit complexity of some bends of minimum-link paths in 2D. In Section 2 we show a general lower bound on the bit complexity of minimum-link paths of $\Omega(n \log n)$ bits for some coordinates. (This was previously claimed, but not proven, by Kahan and Snoeyink [35].) We show that the bound is tight in 2D and we argue that this implies that $\mathsf{MinLinkPath}_{a,2}$ is in NP. In Section 5, we argue that in 3D the boundary of the region reachable with k links can consist of k-th order algebraic curves, potentially leading to exponential bit complexity.
- In Section 3.1 we present a blueprint for showing NP-hardness of minimum link problems. We apply it to prove NP-hardness of the diffuse reflection path problem

$\mathsf{MinLinkPath}_{a,b}$	$b = 1$	$b = 2$	$b = 3$
$a = 0$	$O(n)$	$O(n^2)$	$O(n^{7/3} \operatorname{polylog} n)$
$a = 1$	$O(n)$	Simple Polygon: $O(n^9)$ [5] Full 2D: NP-hard$\star$ PTAS$\star$	NP-hard$\star$ (even in terrains) PTAS$\star$
$a = 2$	N/A	Simple Polygon: $O(n)$ [49] Full 2D: $O(n^2\alpha(n)\log^2 n)$ [41] PTAS$\star$	NP-hard$\star$ (even in terrains) PTAS$\star$
$a = 3$	N/A	N/A	Terrains: $O(n)$ Full 3D: NP-hard$\star$ PTAS$\star$

Table 1: Computational complexity of $\mathsf{MinLinkPath}_{a,b}$ for $a \le b \le 3$. Results presented in this paper are marked with $\star$.

($\mathsf{MinLinkPath}_{1,2}$) in 2D polygonal domains with holes in Section 3.2. In Section 6, we use the same blueprint to prove that all interesting versions of minimum-link problems in 3D are weakly NP-hard. The two remaining versions are $\mathsf{MinLinkPath}_{0,3}$, which can be solved using the simple visibility graph approach sketched above, and $\mathsf{MinLinkPath}_{3,3}$ on terrains, which is trivial: any pair of points can be connected by a path with a single bend at height ∞, so we only have to check if the points are pairwise visible. We also note that the minimum-link problems that we prove NP-hard have no FPTAS and no additive approximation (unless P=NP).

- In Section 4 we extend the 2-approximation algorithm from [29, Ch. 27.5], based on computing weak visibility between sets of potential locations of the path's bends, to provide a simple PTAS for $\mathsf{MinLinkPath}_{2,2}$, which we also adapt to $\mathsf{MinLinkPath}_{1,2}$. In Section 7 we give simple constant-factor approximation algorithms for higher-dimensional minimum-link path versions, which can then be used in the same way to show that all versions admit PTASes.
- In Section 7.3 we focus on $\mathsf{MinLinkPath}_{2,3}$ (diffuse reflection in 3D) on terrains—the version that is most important in practice. We give a 2-approximation algorithm that runs faster than the generic algorithm from [29, Ch. 27.5]. We also present an $O(n^4)$-size data structure encoding visibility between points on a terrain and argue that the size of the structure is asymptotically optimal.

Our results are charted and compared to existing results in Table 1.

2 Algebraic Complexity in 2D

2.1 Lower bound on the Bit complexity

Kahan and Snoeyink [35] claim to "give a simple instance in which representing path vertices with rational coordinates requires $\Theta(n^2 \log n)$ bits". In fact, they show that the boundary

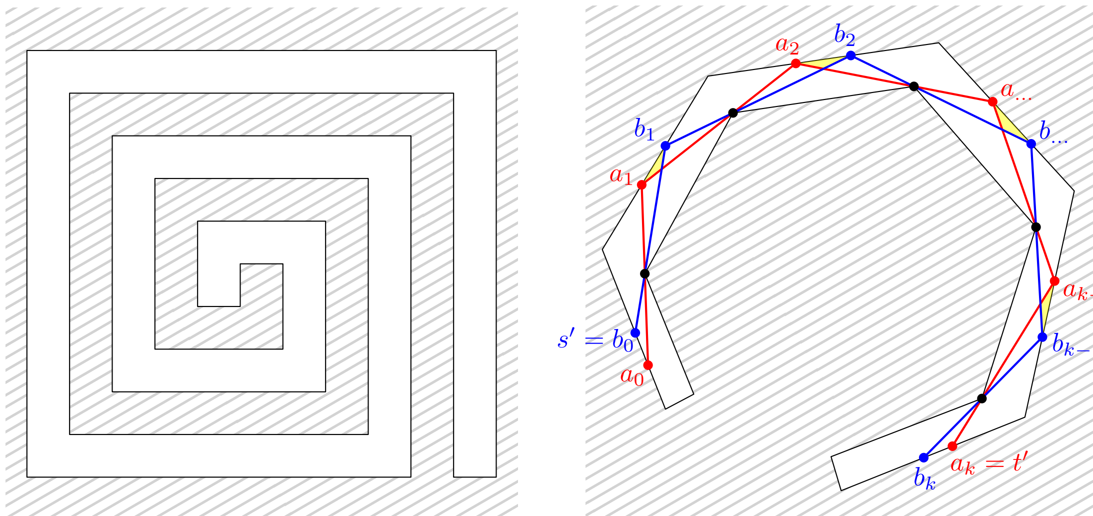

Figure 3: (a) A spiral, as used in the construction by Kahan and Snoeyink. It uses integer coordinates with $O(\log n)$ bits. (b) The general idea.

of the region reachable from s (a point with integer coordinates specified with $O(\log n)$ bits) with k links may have vertices whose coordinates have bit complexity $k \log n$. Note however, that this does not directly imply that a minimum-link path from s to another point t with low-complexity (integer) coordinates must necessarily have such high-complexity bends (i.e., if t itself is not a high-complexity vertex of a region reachable with k links, one potentially could hope to avoid placing the internal vertices of a minimum-link path to t on such high-complexity points as well). Below we present a construction where the intermediate vertices must actually use $\Omega(k \log n)$ bits to be described, even if s and t can be specified using only $\log n$ bits each. We first prove this for the $\mathsf{MinLinkPath}_{1,2}$ variant of the problem, and then extend our results to paths that may bend anywhere within the polygon, i.e., $\mathsf{MinLinkPath}_{2,2}$.

Lemma 1. *There exists a simple polygon P, and points s and t in P such that: (i) all the coordinates of the vertices of P and of s and t can be represented using $O(\log n)$ bits, and (ii) any s-t minimum-link path that bends only on the edges of P has vertices whose coordinates require $\Omega(k \log n)$ bits, where k is the length of a minimum-link path between s and t.*

Proof. We will refer to numbers with $O(\log n)$ bits as *low-complexity*. The general idea in our construction is as follows. We start with a low-complexity point $s' = b_0$ on an edge e_0 of the polygon. We then consider the furthest point b_{i+1} on the boundary of P that is reachable from b_i. More specifically, we require that any point on the boundary of P between s' and b_i is reachable by a path of at most i links, and that any point on the boundary of P beyond b_i requires at least $i+1$ links. We will obtain b_{i+1} by projecting b_i through a vertex c_i. Each such step will increase the required number of bits for b_{i+1} by $\Theta(\log n)$. Eventually, this yields a point b_k on edge e_k. Let t' be the point on e_k that is closest to b_k among the points reachable with k links and having low complexity. Since all points along the boundary from s' to b_k are reachable, and the vertices of P have low complexity, such a point is guaranteed

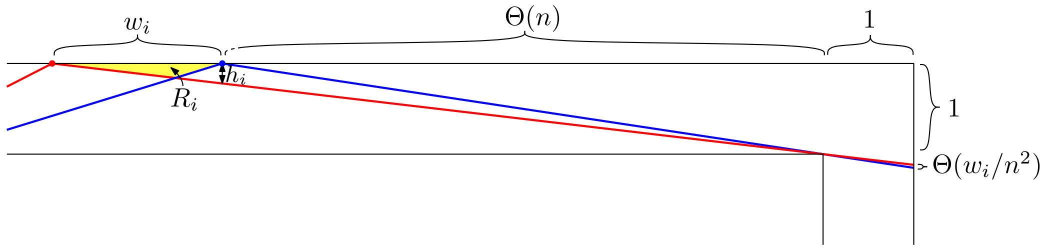

Figure 4: The interval I_i of length w_i produces an interval I_{i+1} of length at most $w_{i+1} = h_i/\Theta(n) = \Theta(w_i/n^2)$, where $h_i = w_i/(w_i + \Theta(n))$. When the i^{th} link can be anywhere in region R_i (shown in yellow), it follows that R_i has height at most h_i, and width at most w_i.

to exist. We set $a_k = t'$ and project a_i through c_{i-1} to a_{i-1} to give us the furthest point (from t') reachable by $k - i$ links. See Figure 3 for an illustration.

The points in the interval $I_i = [a_i, b_i]$, with $1 \leq i < k$, are reachable from s' by exactly i links, and reachable from t' by exactly $k - i$ links. So, to get from s' to t' with k links, we need to choose the i^{th} bend of the path to be within the interval $[a_i, b_i]$. By construction, the intervals for i close to 1 or close to k must contain low-complexity points. We now argue that we can build the construction in such a way that the $(k/2)^{\text{th}}$ interval contains no low-complexity points.

Observe that, if an interval contains no points that can be described with fewer than m bits, its length can be at most 2^{-m}. So, we have to ensure that the $(k/2)^{\text{th}}$ interval has length at most $2^{-k \log n}$.

By construction, the interval I_k has length at most one. Similarly, the length of I_0 can be chosen to be at most one (if it is larger, we can adjust $s' = b_0$ to be the closest integer point to a_0). Now observe that in every step, we can reduce the length w_i of the interval I_i by a factor $\Theta(n^2)$, using a construction like in Figure 4. Our overall construction is then shown in Figure 5.[2]

It follows that $I_{k/2}$ cannot contain two low-complexity points that are close to each other. Note however, that it may still contain one such a point. However, it is easy to see that there is a sub-interval $J_{k/2} = [\ell_{k/2}, r_{k/2}] \subseteq I_{k/2}$ of length $w_{k/2}/2$ that contains no points with fewer than $k \log n$ bits. We enforce the $(k/2)^{\text{th}}$ bend to occur in $J_{k/2}$. This also restricts the possible positions for the i^{th} bend to an interval $J_i \subseteq I_i$. We find these intervals by projecting $\ell_{k/2}$ and $r_{k/2}$ through the vertices of P. Note that s' and t' may not be contained in J_0 and J_k, respectively, so we pick a new start point $s \in J_0$ and end point $t \in J_k$ as follows. Let $m_{k/2}$ be the mid point of $J_{k/2}$ and project m_i through the vertices of P. Now, choose s to be a low-complexity point in the interval $[m_0, r_0]$, and t to be a low-complexity point in the interval $[\ell_k, m_k]$. Observe that $[m_0, r_0]$ and $[\ell_k, m_k]$ have length $\Theta(1)$—as $[\ell_{k/2}, m_{k/2}]$ and $[m_{,k/2}, r_{k/2}]$ have length $w_{k/2}/4$—and thus contain low complexity points. Furthermore, observe that t is indeed reachable from s by a path with $k - 1$ bends (and thus k links), all

[2]The polygon in the figure is not technically simple as it touches itself on the outside. The polygon can easily be modified to be simple while keeping the same min-link path, but the figure would become more cluttered.

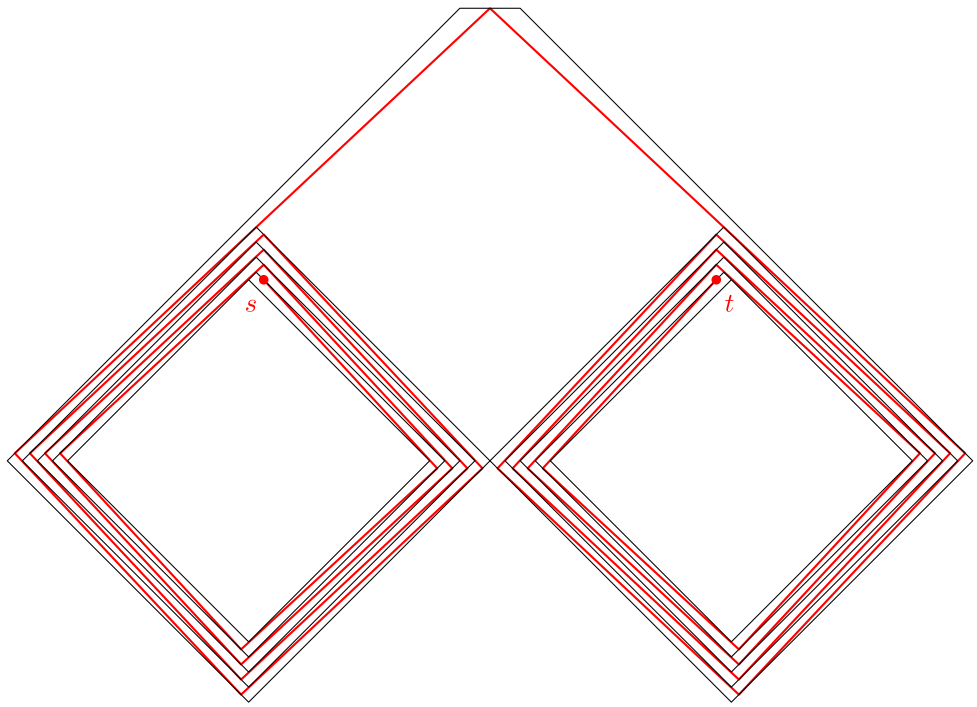

Figure 5: An overview of our polygon P and the minimum-link path that has high-complexity coordinates.

of which much lie in the intervals J_i, $1 \le i < k$ (for example using the path that uses all points m_i). Thus, we have that t is reachable from s by a minimum-link path of k links, and we need $\Omega(k \log n)$ bits to describe the coordinates of the vertices in such a path. □

Lemma 2. *There exists a simple polygon P, and points s and t in P such that: (i) all the coordinates of the vertices of P and of s and t can be represented using $O(\log n)$ bits, and (ii) any s-t minimum-link path has vertices whose coordinates require $\Omega(k \log n)$ bits, where k is the length of a minimum-link path between s and t.*

Proof. We extend the construction from Lemma 1 to the case in which the bends may also lie in the interior of P. Let B_i denote the region in P that is reachable from s' by exactly i links, let A_i be the region reachable from t' by exactly $k - i$ links, and let $R_i = B_i \cap A_i$. To get from s' to t' with k links, the i^{th} bend has to lie in R_i. Now observe that this region is triangular, and incident to the interval I_i (see e.g. Figure 4 for an illustration). This region R_i has width at most w_i and height at most $h_i = w_i/(w_i + \Theta(n))$. Therefore, we can again argue that $R_{k/2}$ is small, and thus contains at most one low-complexity point p. We then again choose a region $R'_{k/2} \subseteq R_{k/2}$ of diameter $w_{k/2}/2$ that avoids point p. The remainder of the argument is analogous to the one before: we can pick points s and t in the restricted regions R'_0 and R'_k that are reachable by a minimum-link path of $k - 1$ bends, all of which have to lie in the regions R'_i. It follows that we again need $\Omega(k \log n)$ bits to describe the coordinates of the vertices in such a path. □

2.2 Upper bound on the Bit complexity

We now show that the bound of Kahan and Snoeyink [35] on the complexity of k-link reachable regions is tight: representing the regions $\mathcal{R}$ as polygons with rational coordinates requires $O(n^2 \log n)$ for any polygon P, assuming that representation of the coordinates of any vertex of P requires at most $c_0 \log n$ bits for some constant c_0. Thus, we have a matching lower and upper bound on the bit complexity of a minimum-link path in 2D.

Consider a simple polygon P with n vertices, and a point $s \in P$. Analogous to [35], define a sequence of regions $\mathcal{R} = \{R_1, R_2, R_3, \dots\}$, where R_1 is a set of all points in P that see s, and R_{i+1} is a region of points in P that see some point in R_i for $i \geq 1$. In other words, region R_{i+1} consists of all the points of P that are illuminated by region R_i.

Construction of region R_{i+1}. If P is a simple polygon, then R_{i+1} is also a simple polygon, consisting of $O(n)$ vertices. We will bound the bit complexity of a single vertex of R_{i+1}. The vertices of such a region are either

- original vertices of P,
- intersection points of P's boundary with lines going through reflex vertices of P, or
- intersection points of P's boundary with rays emanating from the vertices of R_i and going through reflex vertices of P.

Only the last type of vertices can lead to an increase in bit complexity. Each of these vertices is defined as an intersection point of two lines: one of the lines passes through two vertices of P, say $a = (x_a, y_a)$ and $b = (x_b, y_b)$, and, therefore, has a $O(\log n)$ bit representation. The other line passes through one vertex of P, say $c = (x_c, y_c)$, with coordinates of $O(\log n)$ bit complexity, and one vertex of region R_i, say $d = (x_d, y_d)$, with coordinates of potentially higher complexity. The coordinates of the intersection can be calculated by the following formula:

$$\begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} \dfrac{(x_b y_a - x_a y_b + x_a y_c - x_b y_c) x_d + (x_b x_c - x_a x_c) y_d + x_a y_b x_c - y_a x_b x_c}{(y_a - y_b) x_d - (x_a - x_b) y_d + x_a y_c - y_a x_c - x_b y_c + y_b x_c} \\ \dfrac{(y_a y_c - y_b y_c) x_d + (x_b y_a - x_c y_a - x_a y_b + x_c y_b) y_d + x_a y_b y_c - y_a x_b y_c}{(y_a - y_b) x_d - (x_a - x_b) y_d + x_a y_c - y_a x_c - x_b y_c + y_b x_c} \end{pmatrix}$$

$$= \begin{pmatrix} \dfrac{A'_1 x_d + B'_1 y_d + C'_1}{E' x_d + F' y_d + G'} \\ \dfrac{A'_2 x_d + B'_2 y_d + C'_2}{E' x_d + F' y_d + G'} \end{pmatrix}, \text{ for some constants } A'_j, B'_j, C'_j, E', F', \text{ and } G'.$$

Point d lies on the boundary of P. Denote the end points of the side it belongs to as u and v. Then the following relation between the coordinates of d holds:

$$y_d = \frac{(y_u - y_v) x_d + x_u y_v - y_u x_v}{x_u - x_v}.$$

Thus, the equation for the coordinates of the intersection point can be rewritten as:

$$\begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} \dfrac{A_1 x_d + B_1}{C x_d + D} \\ \dfrac{A_2 x_d + B_2}{C x_d + D} \end{pmatrix},$$

where each of A_1, A_2, B_1, B_2, C, and D has bit complexity not greater than $c \log n$ for some constant c (here, it is enough to choose $c = 4c_0$). Let x_d be represented as a rational number p/q, where p and q are relatively prime integers. Then the number of bits required to represent x_d is $sp(x_d) = \lceil \log(p+1) \rceil + \lceil \log(q+1) \rceil \geq \log(p+1) + \log(q+1) \geq 2\log(p+q)$, the last inequality holds for all $p \geq 1$ and $q \geq 1$. Therefore, the number of bits required to represent x^* is

$$\begin{aligned} sp(x^*) = \lceil \log(A_1 p + B_1 q + 1) \rceil + \lceil \log(Cp + Dq + 1) \rceil \leq 2\lceil \log(E(p+q) + 1) \rceil \leq \\ \leq 2\log E + 2\log(p+q) + 2 \leq 2 + 2c\log n + sp(x_d)\,, \end{aligned}$$

where $E = \max\{A_1, B_1, C, D\}$. Analogously for y^*, $sp(y^*) \leq 2 + 2c\log n + sp(x_d)$. Therefore, at every step, the bit complexity of the coordinates grows no more than by an additive value $2 + 2c\log n$. After k steps, the bit-complexity of the regions' vertices is $O(k \log n)$.

Theorem 3. *Representing the regions $\mathcal{R}$ as polygons with rational coordinates requires $O(nk \log n)$ bits.*

Corollary 4. *If there exists a solution with k links, there also exists one in which the coordinates of the bends use at most $O(k \log n)$ bits.*

Our bounds hold also in polygons with holes, where the reachable regions may have vertices that are the intersection of two segments whose end points have high complexity. However, such vertices will be reflex and will not contribute to the next step of projections.

Theorem 5. *MinLinkPath$_{a,2}$ is in NP.*

Proof. We need to show that a candidate solution can be verified in polynomial time. A potential solution needs at most n links. By Corollary 4, we only need to verify candidate solutions that consist of bends with $O(n \log n)$-bit coordinates. Given such a candidate, we need to verify pairwise visibility between at most n pairs of points with $O(n \log n)$-bit coordinates, which can be done in polynomial time. □

3 Computational Complexity in 2D

In this section we show that MinLinkPath$_{1,2}$ is NP-hard. To this end, we first provide a blueprint for our reduction in Section 3.1. In Section 3.2 we then show how to "instantiate" this blueprint for MinLinkPath$_{1,2}$ in a polygon with holes.

3.1 A Blueprint for Hardness Reductions

We reduce from the 2-Partition problem: Given a set of integers $A = \{a_1, \ldots, a_m\}$, find a subset $S \subseteq A$ whose sum is equal to half the sum of all numbers. The main idea behind all the hardness reductions is as follows. Consider a 2D construction in Figure 6 (left). Let point s have coordinates $(0,0)$, and t (not in the figure) have coordinates $(\sum a_i/2, 4m-2)$. For now, in this construction, we will consider only paths from s to t that are allowed to bend on horizontal lines with even y-coordinates. Moreover, we will count an intersection with each such horizontal line as a bend. We will place fences along the lines with odd y-coordinates in such a way that an s-t path with $2m-1$ links exists (that bends only on horizontal lines with even y-coordinates) if and only if there is a solution to the 2-Partition instance.

Call the set of horizontal lines $\ell_0 : y = 0$, $\ell_i : y = 4i - 2$ for $1 \le i \le m$ *important* (dashed lines in Figure 6), and the set of horizontal lines $\ell'_i : y = 4i - 4$ for $2 \le i \le m$ *intermediate* (dash-dotted lines in Figure 6). Each important line ℓ_i will "encode" the running sums of all subsets of the first i integers $A_i = \{a_1, \ldots, a_i\}$. That is, the set of points on ℓ_i that are reachable from s with $2i-1$ links will have coordinates $(\sum_{a_j \in S_i} a_j, 4i-2)$ for all possible subsets $S_i \subseteq A_i$.

Call the set of horizontal lines $f_1 : y = 1$, $f_i : y = 4i - 5$ for $2 \le i \le m$ *multiplying*, and the set of horizontal lines $f'_i : y = 4i - 3$ for $2 \le i \le m$ *reversing*. Each multiplying line f_i contains a fence with two 0-width slits that we call 0-slit and a_i-slit. The 0-slit with x-coordinate 0 corresponds to not including integer a_i into subset S_i, and the a_i-slit with x-coordinate $\sum_1^i a_j - a_i/2$ corresponds to including a_i into S_i. Each reversing line f'_i contains a fence with two 0-width slits (reversing 0-slit and reversing a_i-slit) with x-coordinates 0 and $\sum_1^i a_j$ that "put in place" the next bends of potential minimum-link paths, i.e., into points on ℓ_i with x-coordinates equal to running sums of S_i. We add a vertical fence of length 1 between lines ℓ'_i and f'_i at x-coordinate $\sum_1^i a_j/2$ to prevent the minimum-link paths that went through the multiplying 0-slit from going through the reversing a_i-slit, and vice versa.

As an example, consider (important) line ℓ_2 in Figure 6. The four points on ℓ_2 that are reachable from s with 3 links have x-coordinates $\{0, a_1, a_2, a_1 + a_2\}$. The points on line ℓ'_3 that are reachable from s with a path (with 4 links) that goes through the 0-slit on line f_3 have x-coordinates $\{0, -a_1, -a_2, -(a_1 + a_2)\}$, and the points on ℓ'_3 that are reachable from s through the a_3-slit have x-coordinates $\{a_1+a_2+a_3, 2a_1+a_2+a_3, a_1+2a_2+a_3, 2a_1+2a_2+a_3\}$. The reversing 0-slit on line f'_3 places the first four points on ℓ_3 into x-coordinates $\{0, a_1, a_2, a_1 + a_2\}$, and the reversing a_3-slit places the second four points on ℓ_3 into x-coordinates $\{a_3, a_1 + a_3, a_2 + a_3, a_1 + a_2 + a_3\}$.

In general, consider some point p on line ℓ_{i-1} that is reachable from s with $2i-3$ links. The two points on ℓ'_i that can be reached from p with one link have x-coordinates $-p_x$ and $2\sum_1^i a_j - a_i - p_x$, where p_x is the x-coordinate of p. Consequently, the two points on ℓ_i that can be reached from p with two links have x-coordinates p_x and $p_x + a_i$. Therefore, for every line ℓ_i, the set of points on it that are reachable from s with a minimum-link path have x-coordinates equal to $\sum_{a_j \in S_i} a_j$ for all possible subsets $S_i \subseteq A_i$. Consider line ℓ_m and the destination point t on it. There exists a s-t path with $2m-1$ links if and only if the x-coordinate of t is equal to $\sum_{a_j \in S} a_j$ for some $S \subseteq A$. The complexity of the construction is polynomial in the size of the 2-Partition instance. Therefore, finding a minimum-link path

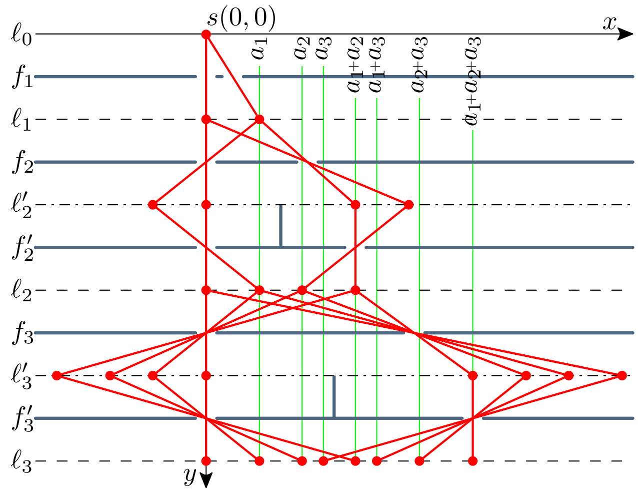

Figure 6: The first few lines of a 2D construction depicting the general idea behind the hardness proofs: important lines ℓ_0–ℓ_3, intermediate lines ℓ'_1–ℓ'_3, multiplying lines f_1–f_3, and reversing lines f'_1–f'_3. The slits in the fences on multiplying and reversing lines are placed in such a way that the locations on ℓ_i that are reachable from s with $2i-1$ links correspond to sums formed by all possible subsets of $\{a_1, \dots, a_i\}$.

from s to t in our 2D construction is NP-hard.

3.2 Hardness of MinLinkPath$_{1,2}$

We can turn our construction from Section 3.1 into a "zigzag" polygon (Figure 7); the fences are turned into obstacles within the corresponding corridors, and slits remain slits—the only free space through which it is possible to go with one link between the polygon edges that correspond to consecutive lines ℓ'_i and ℓ_i (or ℓ_{i-1} and ℓ'_i). This retains the crucial property of the 2D construction: locations reachable with fewest links on the edges of the polygon correspond to sums of numbers in the subsets of A. We conclude:

Theorem 6. *MinLinkPath$_{1,2}$ in a 2D polygonal domain with holes is NP-hard.*

Overall our reduction bears resemblance to the classical *path encoding* scheme [14] used to prove hardness of 3D shortest path and other path planning problems, as we also repeatedly double the number of path homotopy types; however, since we reduce from 2-Partition (and not from 3SAT, as is common with path encoding), our proof(s) are much less involved than a typical path-encoding one.

No FPTAS. Obviously, problems with a discrete answer (in which a second-best solution is separated by at least 1 from the optimum) have no FPTAS. For example, in the reduction in Theorem 6, if the instance of 2-Partition is feasible, the optimal path has $2m-1$ links; otherwise it has $2m$ links. Suppose there exists an algorithm, which, for any $\varepsilon > 0$ finds a $(1+\varepsilon)$-approximate solution in time polynomial in $1/\varepsilon$. Take $\varepsilon = \frac{1}{2m-1}$; note that $1/\varepsilon$ is

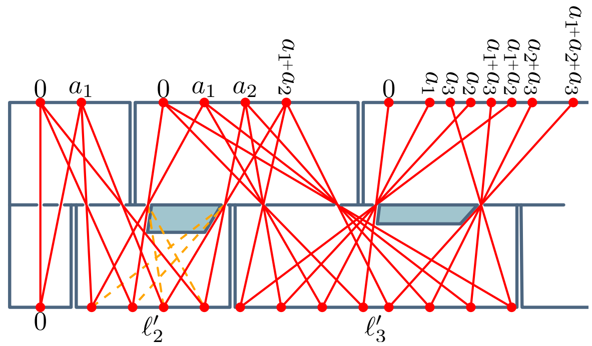

Figure 7: There exists an s-t diffuse reflection path with $2m-1$ links iff 2-Partition instance is feasible.

polynomial, and hence the FPTAS with this ε will complete in polynomial time. For an infeasible instance of 2-Partition the FPTAS would output a path with at least $2m$ links, while for a feasible instance it will output a path with at most $(1+\varepsilon)(2m-1) = 2m-1/2$ links. There is only one such length possible; a path with exactly $\mathsf{opt} = 2m-1$ links. Hence, the FPTAS would be able to differentiate, in polynomial time, between feasible and infeasible instances of 2-Partition.

No additive approximation. We can slightly amplify the hardness results, showing that for any constant K it is not possible to find an additive-K approximation for our problems: Concatenate K instances of the construction from the hardness proof, aligning s in the instance $k+1$ with t from the instance k. Then there is a path with $K(2m-1)$ links through the combined instance if the 2-Partition is feasible; otherwise $K(2m-1)+K-1$ links are necessary, Thus an algorithm, able to differentiate between instances in which the solution has $K(2m-1)$ links and those with $K(2m-1)+K-1$ links in poly(mK) = poly(m) time, would also be able to solve 2-Partition in the same time.

4 Algorithmic Results in 2D

4.1 Constant-factor Approximation

$\mathsf{MinLinkPath}_{2,2}$ can be solved exactly [41]. For $\mathsf{MinLinkPath}_{1,2}$, [28] gives a 3-approximation.

4.2 PTAS

We describe a $(1+\varepsilon)$-approximation scheme for $\mathsf{MinLinkPath}_{1,2}$, based on building a graph of edges of D that are *k-link weakly visible.*

Consider the set F of all edges of D (that is, $\bigcup F = D|^1$). To avoid confusion between edges of D and edges of the graph we will build, we will call elements of F *features* (this will also allow us to extend the ideas to higher dimensions later). Two features $f, f' \in F$ are *weakly visible* if there exist mutually visible points $p \in f, p' \in f'$; more generally, we say f, f'

are *k-link weakly visible* if there exists a k-link path from p to p' (with the links restricted to $D|^1$).

For any constant $k \geq 1$, we construct a graph $G^k = (F, E_k)$, where E_k is the set of pairs of k-link weakly visible features. Let $\pi^k = \{f_0, f_1, \ldots, f_\ell\}$, with $f_0 \ni s$ and $f_\ell \ni t$ be a shortest path in G from the feature containing s to the feature containing t; ℓ is the number of links of π. We describe how to transform π^k into a solution to the $\mathsf{MinLinkPath}_{1,2}$ problem. Embed edges of π into D as k-link paths. This does not necessarily connect s to t since it could be that, inside a feature f_i, the endpoint of the edge $f_{i-1}f_i$ does not coincide with endpoint of the edge $f_i f_{i+1}$; to create a connected path, we observe that the two endpoints can always be connected by two extra links via some feature that is mutually visible from both points (or a single extra link within f_i if we allow links to coincide within the boundary of D).

Lemma 7. *The number of links in* π^k_* *is at most* $(1 + 1/k)\mathsf{opt}$.

Proof. Split opt into pieces of k links each (the last piece may have fewer than k links); the algorithm will find k-link subpaths between endpoints of the pieces. In details, suppose that $\mathsf{opt} = mk + r$ where m, r are the quotient and the remainder from division of opt by k; let $s = v_0, v_1, \ldots, v_{\mathsf{opt}} = t$ be the vertices (bends) of opt, and let f_i be the feature to which the ik-th bend v_{ik} belongs. Since the link distance between $v_{(i-1)k}$ and v_{ik} is k, our algorithm will find a k-link subpath from f_{i-1} to f_i, as well as an r-link subpath from f_m to t. The total number of links in the approximate path is thus at most $mk + m + r \leq (1 + 1/k)(mk + r) = (1 + 1/k)\mathsf{opt}$ (if $r = 0$, our algorithm will find path with at most $mk + m - 1 < (1 + 1/k)mk = (1 + 1/k)\mathsf{opt}$ links; if $r > 0$, our algorithm will find path with at most $mk + r + m \leq (1 + 1/k)(mk + r) = (1 + 1/k)\mathsf{opt}$ links). $\square$

We now argue that the weak k-link visibility between features can be determined in polynomial time using the staged illumination: starting from each feature f, find the set $W(f)$ of points on other features weakly visible from f, then find the set weakly visible from $W^2(f) = W(W(f))$, repeat k times to obtain the set $W^k(f)$ reachable from f with k links; feature f' can be reached from f in k links iff $W^k(f) \cap f' \neq \varnothing$. For constant k, building $W^k(f)$ takes time polynomial in n, although possibly exponential in k (in fact, for diffuse reflection explicit bounds on the complexity of $W^k(f)$ were obtained [5–7]). This can be seen by induction: Partition the set $W^{i-1}(f)$ into the polynomial number of constant-complexity pieces. For each piece p, each element e of the boundary of the domain and each feature f' compute the part of f' shadowed by e from the light sources on p—this can be done in constant time analogously to determining weak visibility between two features above (by considering the part of $p \times f'$ carved out by the occluder e). The part of f' weakly seen from $W^{i-1}(f)$ is the union, over all parts p, of the complements of the sets occluded by all elements e; since there is a polynomial number of parts, elements and features, it follows that $W^i(f)$ can be constructed in polynomial time.

Theorem 8. *For a constant* k *the path* π^k_**, having at most* $(1 + 1/k)\mathsf{opt}$ *links, can be constructed in polynomial time.*

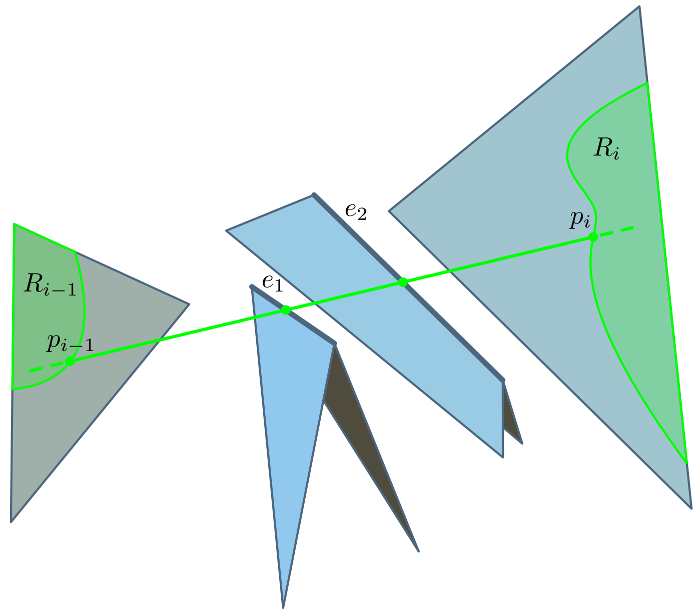

Figure 8: The order of the curves on the boundaries of R_i grows with i.

5 Algebraic Complexity in 3D

Order of the boundary curves. Assume the representations of the coordinates of any vertex of D and s require at most $c_0 \log n$ bits for some constant c_0. Analogous to Section 2, we define a sequence of regions $\mathcal{R} = \{R_1, R_2, R_3, \dots\}$, where R_1 is the set of all points in D that see s, and R_i is the region of points in D that see some point in R_{i-1} for $i \geq 2$, i.e., the region R_i consists of all points of D that are illuminated by region R_{i-1}. Note that R_i is a union of subsets of faces of D. Therefore, when we speak of the boundaries (in the plural form of the word) of R_i, which we denote as ∂R_i, we mean the illuminated sub-intervals of edges of D as well as the frontier curves interior to the faces of D.

Unlike in 2D, the boundaries of R_i interior to the faces of D do not necessarily consist of straight-line segments. Observe that the union of all lines intersecting three given lines in 3D is a hyperboloid, and therefore, illuminating a straight-line segment on the boundaries of R_{i-1} leads to the corresponding part of ∂R_i to be an intersection of a hyperboloid and a plane, i.e., a hyperbola. Moreover, consider some point $p_{i-1} \in \partial R_{i-1}$ interior to some face f_{i-1} of D, and two edges e_1 and e_2 of the domain D which p_{i-1} sees partially and which will cast a shadow on some face f_i of D (refer to Figure 8). We can express the coordinates of p_i as:

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \begin{pmatrix} \dfrac{A_1 x_{i-1}^2 + B_1 y_{i-1}^2 + C_1 x_{i-1} y_{i-1} + D_1 x_{i-1} + E_1 y_{i-1} + F_1}{A x_{i-1}^2 + B y_{i-1}^2 + C x_{i-1} y_{i-1} + D x_{i-1} + E y_{i-1} + F} \\ \dfrac{A_2 x_{i-1}^2 + B_2 y_{i-1}^2 + C_2 x_{i-1} y_{i-1} + D_2 x_{i-1} + E_2 y_{i-1} + F_2}{A x_{i-1}^2 + B y_{i-1}^2 + C x_{i-1} y_{i-1} + D x_{i-1} + E y_{i-1} + F} \\ U x_i + V y_i + W \end{pmatrix}, \qquad (1)$$

for some constants $A_1, A_2, A, B_1, \ldots, U, V, W$ that depend on the parameters of f_{i-1}, f_i, e_1, e_2. Denote a polynomial of degree d as $\mathsf{poly}^d(\cdot)$. We can rewrite the x- and the y-coordinates of p_i as

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} \frac{\mathsf{poly}^2_{x,i-1}(x_{i-1}, y_{i-1})}{\mathsf{poly}^2_{i-1}(x_{i-1}, y_{i-1})} \\ \frac{\mathsf{poly}^2_{y,i-1}(x_{i-1}, y_{i-1})}{\mathsf{poly}^2_{i-1}(x_{i-1}, y_{i-1})} \end{pmatrix} = \begin{pmatrix} \frac{\mathsf{poly}^4_{x,i-2}(x_{i-2}, y_{i-2})}{\mathsf{poly}^4_{i-2}(x_{i-2}, y_{i-2})} \\ \frac{\mathsf{poly}^4_{y,i-2}(x_{i-2}, y_{i-2})}{\mathsf{poly}^4_{i-2}(x_{i-2}, y_{i-2})} \end{pmatrix} = \begin{pmatrix} \frac{\mathsf{poly}^{2i}_{x,0}(x_0, y_0)}{\mathsf{poly}^{2i}_{0}(x_0, y_0)} \\ \frac{\mathsf{poly}^{2i}_{y,0}(x_0, y_0)}{\mathsf{poly}^{2i}_{0}(x_0, y_0)} \end{pmatrix},$$

where point $p_0(x_0, y_0, z_0)$ lies on some straight-line segment of ∂D, and we use different subscripts of the polynomials to distinguish between different expressions. Notice that the denominators of x_i and y_i expressed as functions of x_j and y_j (for all $j < i$) are always the same. If we slide p_0 along the line segment, and express its coordinates in terms of a parameter t, we get

$$x_i = \frac{\mathsf{poly}^{2i}_x(t)}{\mathsf{poly}^{2i}(t)}, \qquad y_i = \frac{\mathsf{poly}^{2i}_y(t)}{\mathsf{poly}^{2i}(t)}, \qquad z_i = \mathsf{poly}^1(x_i, y_i).$$

Thus, the curve that point p_i traces on f_i is an intersection of a plane in 3D (face f_i) and two surfaces of order $2i+1$ in 4D space (with coordinates x, y, z, and t). Therefore, the order of that curve is not greater than $2i+1$. In fact, as we have mentioned above, for $i = 1$, the curve that p_1 traces on face f_1 is a hyperbola, with order 2, and not $2i+1 = 3$. The fact that the denominators of the expressions of x_1 and y_1 are the same allows us to reduce the order of the expressions in the following way:

$$\begin{aligned} x_1 &= \frac{\mathsf{poly}^2_x(t)}{\mathsf{poly}^2(t)} = x'_1 + \frac{\mathsf{poly}^1_{x'}(t)}{\mathsf{poly}^2(t)}, \\ y_1 &= \frac{\mathsf{poly}^2_y(t)}{\mathsf{poly}^2(t)} = y'_1 + \frac{\mathsf{poly}^1_{y'}(t)}{\mathsf{poly}^2(t)}, \end{aligned} \tag{2}$$

Therefore,

$$\frac{x_1 - x'_1}{y_1 - y'_1} = \frac{\mathsf{poly}^1_{x'}(t)}{\mathsf{poly}^1_{y'}(t)}.$$

Solving this equation for t and substituting the resulting expression into Equations 2, we get that the actual order of the curve traced by p_1 is 2. For larger i, denominators of the expressions of x_i and y_i are also equal, however the explicit formula for the curve traced by p_i cannot be derived in a similar way. We summarize our findings:

Theorem 9. *The boundaries of region R_i are curves of order at most $2i+1$ for $i \geq 2$, and at most 2 for $i = 1$.*

The fact that the order of the curves on the boundaries of R_i grows linearly may give hope that the bit complexity of representation of R_i can be bounded from above similarly to Section 2.2. However, following similar calculations we will get that the space required to store the coordinates of p_i grows exponentially with i.

The parameters $A_1, A_2, A, B_1, \ldots, W$ of Equation 1 have bit complexity not greater than $c \log n$ for some constant c. Let x_{i-1} be represented as a rational number p_x/q_x, and y_{i-1} be represented as a rational number p_y/q_y, where p_x and q_x, and p_y and q_y are two pairs of relatively prime integers. Then the number of bits required to represent x_{i-1}, i.e., $sp(x_{i-1})$, is at least $\max\{\log p_x, \log q_x\}$. Therefore, the number of bits required to represent x_i

$$\begin{aligned} sp(x_i) \le& \log(A_1 p_x^2 q_y^2 + B_1 q_x^2 p_y^2 + C_1 p_x q_x p_y q_y + D_1 p_x q_x q_y^2 + E_1 q_x^2 p_y q_y + F_1 q_x^2 q_y^2) + \\ & \log(A p_x^2 q_y^2 + B q_x^2 p_y^2 + C p_x q_x p_y q_y + D p_x q_x q_y^2 + E q_x^2 p_y q_y + F q_x^2 q_y^2) \le \\ \le& 2\log(6Mr^4) = 2\log 6 + 2\log M + 8\log r \le \\ \le& 6 + 2c\log n + 8\max\{sp(x_{i-1}), sp(y_{i-1})\}, \end{aligned}$$

where $M = \max\{A_1, B_1, \ldots, E, F\}$ and $r = \max\{p_x, q_x, p_y, q_y\}$. Solving the above recurrence we get $sp(x_i) \le 9^i$, which implies an exponential upper bound of the space required to store x_i.

Lemma 10. *The coordinates of a vertex of R_i can be stored in $O(9^i)$ space.*

We conjecture that the lower bound for the bit complexity of the vertices of a minimum-link path in 3D is exponential as well. This would imply that $\mathsf{MinLinkPath}_{2,3}$ in 3D is not in NP.

Conjecture 1. *There exists a polyhedral domain D and two points s and t such that: (i) all the coordinates of the vertices of D and of s and t can be represented using $O(\log n)$ bits, and (ii) any s-t minimum-link path that bends only on the faces of D has vertices whose coordinates require $\Omega(c^k)$ bits, where c is some constant and k is the length of a minimum-link path between s and t.*

6 Computational Complexity in 3D

We will now show how to use our blueprint from Section 3.1 to build a terrain for the $\mathsf{MinLinkPath}_{1,2}$ problem such that a path from s to t with $2m-1$ links will exist if and only if there exists a subset $S \subseteq A$ whose sum is equal to half the sum of all integers $A = \{a_1, \ldots, a_m\}$. Take the 2D construction and bend it along all the lines ℓ_i and ℓ'_i, except ℓ_0 and ℓ_m (refer to Figure 9). Let the angles between consecutive faces be $\pi - \delta$ for some small angle $\delta < \pi/4m$ (so that the sum of bends between the first face (between the lines ℓ_0 and ℓ_1) and the last face (between the lines ℓ'_m and ℓ_m) is less than π). On each face build a fence of height $\tan(\delta/4)$ according to the 2D construction. The height of the fences is small enough so that no two points on consecutive fences see each other. Therefore, for two points s and t placed on ℓ_0 and ℓ_m as described above, an s-t path with $2m-1$ links must bend only on ℓ_i and ℓ'_i and pass in the slits in the fences. Finding a minimum-link path on such a terrain is equivalent to finding a minimum-link path (with bends restricted to ℓ_i and ℓ'_i) in the 2D construction. Therefore,

Theorem 11. *$\mathsf{MinLinkPath}_{1,2}$ on a terrain is NP-hard.*

Remark. *Instead of 0-width slits, we could use slits of positive width $w < \frac{1}{8m}$; since the width of the light beam grows by $2w$ between two consecutive creases, on any crease, the*

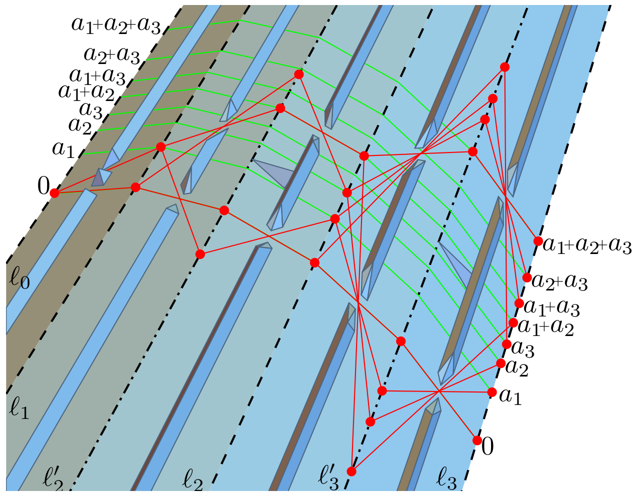

Figure 9: Right: The terrain obtained by bending the 2D construction along the important and intermediate lines. The height of the fences is low enough that no two points on consecutive fences can see each other.

maximum shift of the path due to the positive slits width will be at most $(2m-1) \times 2w < 1/2$. *Thus, the positive width cannot change the number of links, and the reduction works even in the case when all slits widths are positive.*

Observe that bending in the interior of a face cannot reduce the link distance between s and t. Hence, our reduction also shows that $\mathsf{MinLinkPath}_{2,2}$ is NP-hard. Furthermore, lifting the links from the terrain surface also does not reduce the link distance; we ensured that the fences are low in height, so that fences situated on different faces of the creased rectangle do not see each other. Therefore, jumping onto the fences is useless. Hence, $\mathsf{MinLinkPath}_{1,3}$ and $\mathsf{MinLinkPath}_{2,3}$ are also NP-hard.

$\mathsf{MinLinkPath}_{a,b}$ in general polyhedra. Since a terrain is a special case of a 3D polyhedra, it follows that $\mathsf{MinLinkPath}_{1,2}$, $\mathsf{MinLinkPath}_{2,2}$, $\mathsf{MinLinkPath}_{1,3}$, and $\mathsf{MinLinkPath}_{2,3}$ are also NP-hard for an arbitrary polyhedral domain in 3D. Our construction does not immediately imply that $\mathsf{MinLinkPath}_{3,3}$ is NP-hard. However, we can put a copy of the terrain slightly above the original terrain (so that the only free space is the thin layer between the terrains). When this layer is thin enough, the ability to take off from the terrain and bend in the free space does not help in decreasing the link distance from s to t. Thus, $\mathsf{MinLinkPath}_{3,3}$ is also NP-hard.

Corollary 12. *$\mathsf{MinLinkPath}_{a,b}$, with $a \geq 1$ and $b \geq 2$, in a 3D domain D is NP-hard. This holds even if D is just a terrain.*

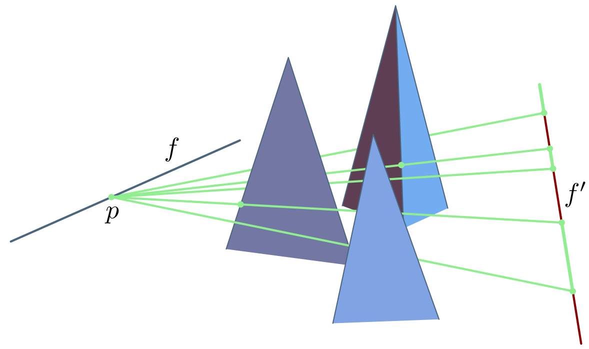

Figure 10: The weak visibility $W(f)$ restricted to edge f' is the union of all visible intervals (green) over all points $p \in f$. If this region is non-empty, f and f' are weakly visible.

7 Algorithmic Results in 3D

7.1 Constant-factor Approximation

Our approximations refine and extend the 2-approximation for minimum-link paths in higher dimensions suggested in Chapter 26.5 (section Other Metrics) of the handbook [29] (see also Ch. 6 in [44]); since the suggestion is only one sentence long, we fully quote it here:

> Link distance in a polyhedral domain in $\mathbb{R}^d$ can be approximated (within factor 2) in polynomial time by searching a weak visibility graph whose nodes correspond to simplices in a simplicial decomposition of the domain.

Indeed, consider $D|^a$, the set of all points where the path is allowed to bend, and decompose $D|^a$ into a set F of small-complexity convex pieces; call each piece a *feature*. Similar to Section 4.2, we say two features f and f' are *weakly visible* if there exist mutually visible points $p \in f$ and $p' \in f'$; more generally, the *weak visibility* region $W(f)$ is the set of points that see at least one point of f, so f' is weakly visible from f iff $f' \cap W(f) \neq \varnothing$ (in terms of *illumination* $W(f)$ is the set of points that get illuminated when a light source is put at every point of f). See Figure 10 for an illustration.

Weak visibility between two features f and f' can be determined straightforwardly by building the set of pairs of points (p, p') in the parameter space $f \times f'$ occluded by (each element of) the obstacles. To be precise, $f \times f'$ is a subset of $\mathbb{R}^{2a}$. Now, consider $D|^{d-1}$, which we also decompose into a set of constant-complexity *elements*. Each element e defines the set $B(e) = \{(p, p') \in f \times f' : pp' \cap e \neq \varnothing\}$ of pairs of points that it blocks; since e has constant complexity, the boundary of $B(e)$ consists of a constant number of curved surfaces, each described by a low degree polynomial. Since there are $O(n)$ elements, the union (and, in fact, the full arrangement) of the sets $B(e)$ for all e can be built in $O(n^{4a-3+\varepsilon})$ time, for an arbitrarily small $\varepsilon > 0$, or $O(n^2)$ time in case $a = 1$ [3]. We define the *visibility map* $M(f, f') \subseteq f \times f'$ to be the complement of the union of the blocking sets, i.e., the map is the set of mutually visible pairs of points from $f \times f'$. We have:

Lemma 13. *$M(f, f')$ can be built in $O(n^{\max(2,4a-3+\varepsilon)})$ time, for an arbitrarily small $\varepsilon > 0$.*

The features f and f' weakly see each other iff $M(f, f')$ is not empty. Let G be the graph on features whose edges connect weakly visible features; s and t are added as vertices of G, connected to features (weakly) seen from them. Let $\pi = \{f_0, f_1, \ldots, f_\ell\}$, with $f_0 = s$ and $f_\ell = t$ be a shortest s-t path in G; ℓ is the length of π. Embed edges of π into the geometric domain, putting endpoints of the edges arbitrarily into the corresponding features. This does not necessarily connect s to t since it could be that, inside a feature f_i, the endpoint of the edge $f_{i-1}f_i$ does not coincide with endpoint of the edge $f_i f_{i+1}$; to create a connected path, connect the two endpoints by an extra link within f_i (this is possible since the features are convex).

Bounding the approximation ratio of the above algorithm is straightforward: Let opt denote a minimum-link s-t path and, abusing notation, also the number of links in it. Consider the features to which consecutive bends of opt belong; the features are weakly visible and hence are adjacent in G. Thus $\ell \leq \mathsf{opt}$. Adding the extra links inside the features adds at most $\ell - 1$ links. Hence the total number of links in the produced path is at most $2\ell - 1 < 2\mathsf{opt}$.

Since G has $O(n)$ nodes and $O(n^2)$ edges, Dijkstra's algorithm will find the shortest path in it in $O(n^2)$ time.

Theorem 14. *(cf. [29, Ch. 27.5].) A 2-approximation to $\mathsf{MinLinkPath}_{a,b}$ can be found in $O(n^{2+\max(2,4a-3+\varepsilon)})$ time, where $\varepsilon > 0$ is an arbitrarily small constant.*

Interestingly, the running time in Theorem 14 depends only on a, and not on b or d, the dimension of D (of course, $a \leq d$, so the runtime is bounded by $O(n^{2+\max(2,4d-3+\varepsilon)})$ as well).

7.2 PTAS

To get a $(1 + 1/k)$-approximation algorithm for any constant $k \geq 1$, we expand the above handbook idea by searching for shortest s-t path π^k in the graph G^k whose edges connect features that are *k-link weakly visible*. Similarly to Section 4.2, we obtain the following.

Theorem 15. *For a constant k the path π^k_*, having at most $(1 + 1/k)\mathsf{opt}$ links, can be constructed in polynomial time.*

Proof. The approximation factor follows from the same argument as in Section 4.2. To show the polynomial running time, we argue that the weak k-link visibility between features can be determined in polynomial time using the staged illumination: starting from each feature f, find the set $W(f)$ of points on other features weakly visible from f, then find the set weakly visible from $W^2(f) = W(W(f))$, repeat k times to obtain the set $W^k(f)$ reachable from f with k links; feature f' can be reached from f in k links iff $W^k(f) \cap f' \neq \varnothing$. For constant k, building $W^k(f)$ takes time polynomial in n, although possibly exponential in k (in fact, for diffuse reflection explicit bounds on the complexity of $W^k(f)$ were obtained [5–7]). This can be seen by induction: Partition the set $W^{i-1}(f)$ into the polynomial number of constant-complexity pieces. For each piece p, each element e of the boundary of the domain and each feature f' compute the part of f' shadowed by e from the light sources on p—this can be done in constant time analogously to determining weak visibility between two features

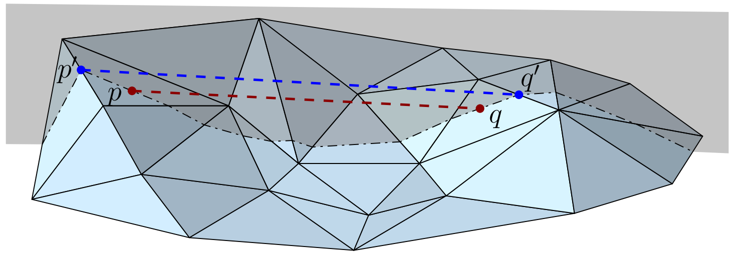

Figure 11: For every pair of points $p \in f_p$ and $q \in f_q$ that can see each other, there exist points p' and q' on the edges bounding f_p and f_q, respectively, that can also see each other.

above (by considering the part of $p \times f'$ carved out by the occluder e). The part of f' weakly seen from $W^{i-1}(f)$ is the union, over all parts p, of the complements of the sets occluded by all elements e; since there is a polynomial number of parts, elements and features, it follows that $W^i(f)$ can be constructed in polynomial time. □

7.3 The global visibility map of a terrain

Using the result from Theorem 14 for $\mathsf{MinLinkPath}_{2,3}$ on terrains, we get a 2-approximate minimum-link path in $O(n^{7+\varepsilon})$ time (since the path can bend anywhere on a triangle of the terrain, the features are the triangles and intrinsic dimension $d = 2$). In this section we show that a faster, $O(n^4)$-time 2-approximation algorithm is possible. We also consider encoding visibility between all points on a terrain (not just between features, as the visibility map from Section 7.1 does): we give an $O(n^4)$-size data structure for that, which we call the terrain's *global visibility map*, and provide an example showing that the size of the structure is worst-case optimal.

We start with connecting approximations of $\mathsf{MinLinkPath}_{2,3}$ and $\mathsf{MinLinkPath}_{1,3}$ on terrains. Let opt be an optimal solution in an instance of $\mathsf{MinLinkPath}_{2,3}$, let opt_e be the optimal solution to $\mathsf{MinLinkPath}_{1,3}$ in the same instance, and let apx_e be the 2-approximate path for the $\mathsf{MinLinkPath}_{1,3}$ version output by the algorithm in Section 7.1 (Theorem 14); abusing notation, let opt, opt_e and apx_e denote also the number of links in the paths. Clearly, $\mathsf{apx}_e \leq 2\mathsf{opt}_e$; what we show is that actually a stronger inequality holds (the inequality is stronger since $\mathsf{opt} \leq \mathsf{opt}_e$):

Lemma 16. $\mathsf{apx}_e \leq 2\mathsf{opt}$.

Proof. Consider some link pq on optimal path opt from s to t. Draw a vertical plane through p and q and denote as p' and q' the uppermost intersections of this plane with the boundaries of the triangles containing p and q (refer to Figure 11). Then p' and q' see each other, and they lie on edges of the terrain.

Replace every link pq of opt by $p'q'$, and interconnect the consecutive links by straight segments. Such interconnecting segments will belong to an edge of the terrain, or go through the interior of a triangle containing the corresponding vertex of the optimal path. The resulting chain of edges is a proper path from s to t whose bends lie only on edges of the

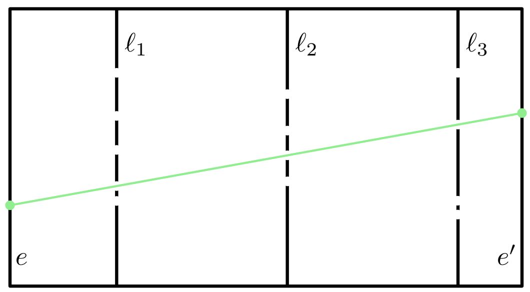

Figure 12: Start from an instance of the 3SUM-hard problem GeomBase [26]: Given a set S of points lying on 3 parallel lines ℓ_1, ℓ_2, ℓ_3, do there exist 3 points from S lying on a line $\ell \notin \{\ell_1, \ell_2, \ell_3\}$? Construct an instance of the weak visibility problem for edges e, e' in a polygonal domain: ℓ_1, ℓ_2, ℓ_3 become obstacles and each point $p \in S$ is a gap punched in the obstacle; the lines are in a box whose two opposite edges (parallel to the lines) are the edges e, e'. The edges are weakly visible iff there exist 3 collinear gaps $p_i, i = 1, 2, 3$, such that $p_i \in \ell_i$.

terrain. Thus, it has a corresponding path in graph G (refer to Theorem 14). The length of such a path is at most $2\mathsf{opt} - 1$, and it is not shorter than apx_e (the shortest path in G). Therefore, $\mathsf{apx}_e \leq 2\mathsf{opt}$. □

Lemma 16 allows us to use the 2-approximation for $\mathsf{MinLinkPath}_{1,3}$ as a 2-approximation for $\mathsf{MinLinkPath}_{2,3}$. The former can be found more efficiently: by Theorem 14, apx_e can be found in $O(n^4)$ time.

Theorem 17. *A 2-approximation for $\mathsf{MinLinkPath}_{2,3}$ in a terrain can be found in $O(n^4)$ time.*

The running time of the algorithm in Theorem 17 is dominated by determining weak visibility between all $\binom{n}{2}$ pairs of edges; the approach from Section 7.1 does it with brute force in $O(n^2)$ time per pair. An obvious question is whether this could be done faster for a single pair. We now show that this is hardly the case. We start from the analogous result for 2D polygonal domains:

Theorem 18. *Determining weak visibility between a pair of edges in a polygonal domain with holes is 3SUM-hard.*

Proof. The proof is by picture; see Figure 12. □

The domain in Figure 12 can be turned into a terrain by erecting the lines ℓ_1, ℓ_2, ℓ_3 into 3 vertical walls (the gaps in the lines become slits in the walls); similarly to the 2D case, the edges e, e' weakly see each other iff GeomBase is feasible:

Theorem 19. *Determining weak visibility between a pair of edges in a terrain is 3SUM-hard.*

The above 3SUM-hardness results are not the end of the story: the fact that determining weak visibility for a single pair of edges may require quadratic time does not imply that determining the visibility between all pairs of edges should require quatric time. In fact, the 3SUM-hardness of the 2D case (Theorem 18) does not preclude existence of an $O(n^2)$-time algorithm for finding *all pairs* of weakly visible edges in a polygonal domain with holes—such an algorithm is used, e.g., in Section 4 of [28]. Moreover, in [13] it is shown that a data structure of $O(n^2)$ size can be built in $O(n^2)$ time, encoding visibility between *all pairs of points* in a domain; the data structure, which can be called the *global* visibility map of the domain, is an extension of the standard visibility graph that encodes visibility only between the domain's vertices. An immediate question is whether such a data structure can be built for terrains; below is our answer.

The *global visibility map* that encodes all mutually visible pairs of points on a terrain (or in a full 3D domain) will live in four dimensions—this is because a line in a 3D space has four degrees of freedom, and our data structure will use the projective dual 4D space $\mathcal{S}_d$ to the primary 3D space $\mathcal{S}_p$ where the terrain is located. A line $\ell \in \mathcal{S}_p$ will correspond to a point $\ell^* \in \mathcal{S}_d$. To build the global visibility map, consider a 5D space $\mathcal{S}_5$ where $\mathcal{S}_p$ and $\mathcal{S}_d$ are subspaces, and a point O in $\mathcal{S}_5$ with coordinates $(0, 0, 0, 0, 1)$. The dual point $\ell^* \in \mathcal{S}_d$ for a line $\ell \in \mathcal{S}_p$ is constructed as follows: Draw a 4D hyperplane in $\mathcal{S}_5$ that goes through line ℓ and point O. A perpendicular line to such hyperplane that goes through O intersects $\mathcal{S}_d$ in a point. This point will be ℓ^*—the dual point to line ℓ.

Now, the visibility map is a partition of $\mathcal{S}_d$ into cells, such that each cell contains points whose duals have the same combinatorial structure, i.e., they intersect the same set of obstacles' faces in $\mathcal{S}_p$.

Lemma 20. *The global visibility map that encodes all pairs of mutually visible points on terrain T (or on a set of obstacles $\mathcal{O}$ in full 3D model) has complexity $O(n^4)$.*

Proof. Let $\mathcal{L}$ be a set of n lines in $\mathcal{S}_p$. $\mathcal{L}$ implies a subdivision W of space $\mathcal{S}_d$ into cells that correspond to lines that touch the same sets of lines in $\mathcal{L}$. W consists of 0-cells (vertices), 1-cells (edges), 2-cells, 3-cells, and 4-cells. The k-cells of W correspond to a set of lines that intersects exactly $4-k$ lines of $\mathcal{L}$. There are clearly $O(n^4)$ 0-cells, since there are n lines in $\mathcal{L}$. For each k-cell, the number of incident $(k+1)$-cells is $O(1)$, since they correspond to the sets of lines we get by dropping incidence to 1 of the $4-k$ lines (and $4-k$ is constant). Therefore, the number of k-cells is also bounded by $O(n^4)$ for all k. Hence, W has complexity $O(n^4)$.

Now, consider our terrain T (or a set of obstacles $\mathcal{O}$ in full 3D model) in $\mathcal{S}_p$. We are interested in the subdivision S of $\mathcal{S}_d$ into cells that correspond to line segments that are combinatorially equal (their end points are on the same features of T or $\mathcal{O}$). Then, W is a sub-subdivision of S (in the sense of subgraph, so something with fewer components). Hence, S also has complexity $O(n^4)$. □

Remark. *The first part of the above argument (the complexity of the configuration space of lines among lines in 3-space) is a natural question and it is well-studied. McKenna and O'Rourke [37] argue quartic bounds on the numbers of 0-faces, 1-faces and 4-faces (although many proofs in their paper are omitted). They also describe how to compute the complex consisting of all 0-faces and 1-faces in $O(n^4\alpha(n))$ time.*

We now argue that the bound in Lemma 20 is tight: the global visibility map may have

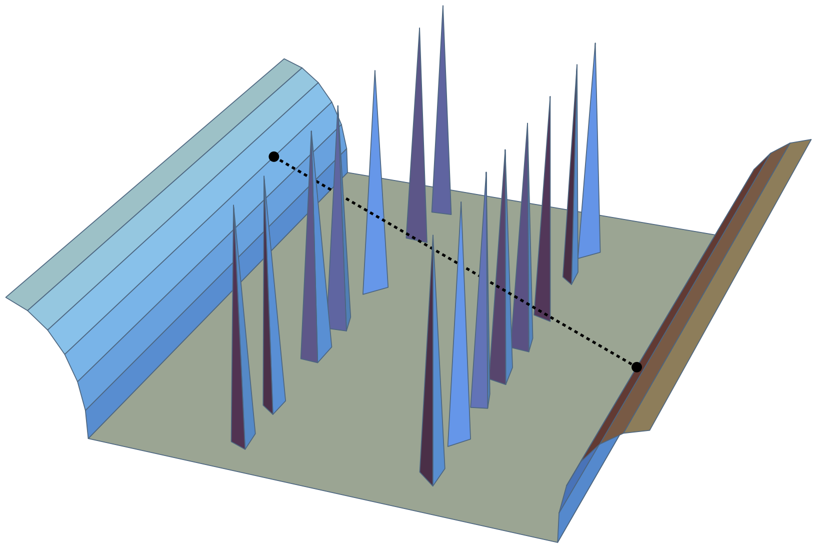

Figure 13: Every vertex (0-face) in the visibility map corresponds to a line that crosses 4 edges of the terrain. In this example, there is a line that connects any horizontal edge on the left-hand side with any horizontal segment on the right-hand side, and that also pins two spikes in the middle. Thus, there are $\Omega(n^4)$ 0-faces in the visibility map.

complexity $\Omega(n^4)$. Other then being an interesting result by itself, this implies that the running time of the algorithm in Theorem 17 may not be improved if one were to compute the weak visibility between all pairs of edges.

Lemma 21. *The global visibility map that encodes all pairs of mutually visible points on terrain T can have complexity $\Omega(n^4)$.*

Proof. See Figure 13. It is easy to see that this construction yields a visibility map of complexity $\Omega(n^4)$. □

Lemmas 20 and 21 give tight bounds on the complexity of the visibility map:

Theorem 22. *The complexity of global visibility map, encoding all pairs of mutually visible points on a terrain (or on a set of obstacles in 3D) of complexity n, is $\Theta(n^4)$.*

8 Conclusion

We considered minimum-link in 2D and 3D, giving bounds on the combinatorial complexity of the paths and algorithmic complexity of the problems of finding the paths. We showed that in 3D most of the versions of the problem are hard but admit PTASes; we also obtained similar results for the diffuse reflection problem in 2D polygonal domains with holes. The biggest remaining open problem is whether pseudopolynomial-time algorithms are possible for the problems: our reductions are from 2-PARTITION, and hence do not show strong hardness (we believe that our techniques can be extended to show strong hardness via more sophisticated path-encoding reductions). A related question is exploring bit complexity of the minimum-link paths in 3D (note that already in simple polygons in 2D finding a minimum-link path with integer vertices is weakly NP-hard [18]).

Acknowledgments

We thank Joe Mitchell and Jean Cardinal for fruitful discussions on this work and the anonymous reviewers for their helpful comments. M.L. is supported by the Netherlands Organisation for Scientific Research (NWO) under grant 639.021.123. The work of I.K. was supported in part by the Netherlands Organisation for Scientific Research (NWO) under grant 639.023.208 and by F.R.S.-FNRS. V.P. is supported by grant 2014-03476 from the Sweden's innovation agency VINNOVA. F.S. is supported by the Danish National Research Foundation under grant nr. DNRF84.

References

[1] J. Adegeest, M. H. Overmars, and J. Snoeyink. "Minimum-link C-oriented paths: Single-source queries". In: *International Journal of Computational Geometry & Applications* 4.1 (1994).

[2] P. K. Agarwal and J. Matoušek. "Ray shooting and parametric search". In: *SIAM Journal on Computing* 22.4 (1993).

[3] P. K. Agarwal and M. Sharir. "Arrangements and their applications". In: *Handbook of computational geometry* (2000).

[4] M. H. Alsuwaiyel and D. T. Lee. "Minimal link visibility paths inside a simple polygon". In: *Computational Geometry* 3.1 (1993).

[5] B. Aronov, A. R. Davis, J. Iacono, and A. S. C. Yu. "The Complexity of Diffuse Reflections in a Simple Polygon". In: *7th Latin American Symposium on Theoretical Informatics*. 2006.

[6] B. Aronov, A. R. Davis, T. K. Dey, S. P. Pal, and D. C. Prasad. "Visibility with Multiple Reflections". In: *Discrete & Computational Geometry* 20.1 (1998).

[7] B. Aronov, A. R. Davis, T. K. Dey, S. P. Pal, and D. C. Prasad. "Visibility with One Reflection". In: *Discrete & Computational Geometry* 19.4 (1998).

[8] B. Ben-Moshe, P. Carmi, and M. J. Katz. "Approximating the Visible Region of a Point on a Terrain". In: *GeoInformatica* 12.1 (2008).

[9] B. Ben-Moshe, M. J. Katz, J. S. B. Mitchell, and Y. Nir. "Visibility preserving terrain simplification—an experimental study". In: *Computational Geometry* 28.2-3 (2004).

[10] M. de Berg, M. J. van Kreveld, B. J. Nilsson, and M. H. Overmars. "Shortest path queries in rectilinear worlds". In: *International Journal of Computational Geometry & Applications* 3.2 (1992).

[11] M. de Berg. "Generalized hidden surface removal". In: *Computational Geometry* 5.5 (1996).

[12] A. Bishnu, S. K. Ghosh, P. P. Goswami, S. P. Pal, and S. Sarvattomananda. *An Algorithm for Computing Constrained Reflection Paths in Simple Polygon*. http://arxiv.org/abs/1304.4320. 2013.

[13] K. Buchin, I. Kostitsyna, M. Löffler, and R. Silveira. "Region-based approximation of probability distributions (for visibility between imprecise points among obstacles)". In: *17th Workshop on Algorithm Engineering and Experiments*. 2015.

[14] J. Canny and J. H. Reif. "New lower bound techniques for robot motion planning problems". In: *28th Annual Symposium on Foundations of Computer Science*. 1987.

[15] S. Carlsson, H. Jonsson, and B. J. Nilsson. "Finding the shortest watchman route in a simple polygon". In: *Discrete & Computational Geometry* 22.3 (1999).

[16] R. Cole and M. Sharir. "Visibility problems for polyhedral terrains". In: *Journal of Symbolic Computation* 7 (1989).

[17] E. D. Demaine, J. S. B. Mitchell, and J. O'Rourke. *The Open Problems Project*. http://maven.smith.edu/~orourke/TOPP/.

[18] W. Ding. "On Computing Integral Minimum Link Paths in Simple Polygons". In: *European Workshop on Computational Geometry*. 2008.

[19] M. Dror, A. Efrat, A. Lubiw, and J. S. B. Mitchell. "Touring a Sequence of Polygons". In: *35th Annual ACM Symposium on Theory of Computing*. 2003.

[20] A. Efrat, L. J. Guibas, O. A. Hall-Holt, and L. Zhang. "On incremental rendering of silhouette maps of a polyhedral scene". In: *Computational Geometry* 38.3 (2007).

[21] D. El-Khechen, M. Dulieu, J. Iacono, and N. van Omme. "Packing 2×2 unit squares into grid polygons is NP-complete". In: *Canadian Conference on Computational Geometry*. 2009.

[22] L. D. Floriani and P. Magillo. "Algorithms for visibility computation on terrains: a survey". In: *Environment and Planning B: Planning and Design* 30.5 (2003).

[23] J. D. Foley, R. L. Phillips, J. F. Hughes, A. v. Dam, and S. K. Feiner. *Introduction to Computer Graphics*. Addison-Wesley Longman Publishing Co., Inc., 1994.

[24] M. L. Fredman and D. E. Willard. "Trans-dichotomous algorithms for minimum spanning trees and shortest paths". In: *Journal of Computer and System Sciences* 48.3 (1994).

[25] M. Furst, J. B. Saxe, and M. Sipser. "Parity, circuits, and the polynomial-time hierarchy". In: *Mathematical Systems Theory* 17.1 (1984).

[26] A. Gajentaan and M. H. Overmars. "On a Class of $O(n^2)$ Problems in Computational Geometry". In: *Computational Geometry* 5 (1995).

[27] S. K. Ghosh. "Computing the Visibility Polygon from a Convex Set and Related Problems". In: *Journal of Algorithms* 12.1 (1991).

[28] S. K. Ghosh, P. P. Goswami, A. Maheshwari, S. C. Nandy, S. P. Pal, and S. Sarvattomananda. "Algorithms for computing diffuse reflection paths in polygons". In: *The Visual Computer* 28.12 (2012).

[29] J. E. Goodman and J. O'Rourke. *Handbook of Discrete and Computational Geometry*. Chapman & Hall/CRC, 2004.

[30] L. J. Guibas, J. Hershberger, J. S. B. Mitchell, and J. Snoeyink. "Approximating Polygons and Subdivisions with Minimum Link Paths". In: *2nd International Symposium on Algorithms*. 1991.

[31] L. J. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. E. Tarjan. "Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons". In: *Algorithmica* 2 (1987).

[32] J. Hershberger and J. Snoeyink. "Computing minimum length paths of a given homotopy class". In: *Computational Geometry* 4 (1994).

[33] J. Hershberger and J. Snoeyink. "Computing Minimum Length Paths of a Given Homotopy Class". In: *Computational Geometry: Theory and Applications* 4 (1994).

[34] F. Hurtado, M. Löffler, I. Matos, V. Sacristán, M. Saumell, R. I. Silveira, and F. Staals. "Terrain visibility with multiple viewpoints". In: *24th International Symposium on Algorithms and Computation*. 2013.

[35] S. Kahan and J. Snoeyink. "On the bit complexity of minimum link paths: Superquadratic algorithms for problems solvable in linear time". In: *Computational Geometry: Theory and Applications* 12.1-2 (1999).

[36] F. Kammer, M. Löffler, P. Mutser, and F. Staals. "Practical Approaches to Partially Guarding a Polyhedral Terrain". In: *Geographic Information Science*. 2014.

[37] M. McKenna and J. O'Rourke. "Arrangements of Lines in 3-space: A Data Structure with Applications". In: *4th Annual Symposium on Computational Geometry*. 1988.

[38] J. S. B. Mitchell, V. Polishchuk, and M. Sysikaski. "Minimum-link paths revisited". In: *Computational Geometry* 47.6 (2014).

[39] J. S. B. Mitchell and M. Sharir. "New Results on Shortest Paths in Three Dimensions". In: *20th Annual Symposium on Computational Geometry*. 2004.

[40] J. S. Mitchell. "Approximating watchman routes". In: *24th Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2013.

[41] J. Mitchell, G. Rote, and G. Woeginger. "Minimum-link paths among obstacles in the plane". In: *Algorithmica* 8.1 (1992).

[42] E. Moet. "Computation and complexity of visibility in geometric environments". PhD thesis. Utrecht University, 2008.

[43] M. H. Overmars and E. Welzl. "New Methods for Computing Visibility Graphs". In: *4th Annual Symposium on Computational Geometry*. 1988.

[44] C. Piatko. "Geometric bicriteria optimal path problems". PhD thesis. Cornell University, 1993.

[45] D. Prasad, S. P. Pal, and T. Dey. "Visibility with multiple diffuse reflections". In: *Computational Geometry* 10 (1998).

[46] M. Schaefer, E. Sedgwick, and D. Stefankovic. "Recognizing string graphs in NP". In: *Journal of Computer and System Sciences* 67.2 (2003).

[47] A. Schönhage. “On the power of random access machines”. In: *6th Colloquium on Automata, Languages and Programming*. 1979.

[48] A. J. Stewart. “Hierarchical Visibility in Terrains”. In: *Eurographics Rendering Workshop*. 1997.

[49] S. Suri. “A linear time algorithm with minimum link paths inside a simple polygon”. In: *Computer Vision, Graphics and Image Processing* 35.1 (1986).

[50] G. Viglietta. “Face-Guarding Polyhedra”. In: *Canadian Conference on Computational Geometry*. 2011.

[51] C. D. Yang, D. T. Lee, and C. K. Wong. “On bends and distances of paths among obstacles in 2-layer interconnection model”. In: *IEEE Transactions on Computing* 43.6 (1994).

[52] C. D. Yang, D. T. Lee, and C. K. Wong. “On bends and lengths of rectilinear paths: a graph theoretic approach”. In: *International Journal of Computational Geometry & Applications* 2.1 (1992).

[53] C. D. Yang, D. T. Lee, and C. K. Wong. “Rectilinear paths problems among rectilinear obstacles revisited”. In: *SIAM Journal on Computing* 24 (1995).