

On the complexity of minimum-link path problems*

Irina Kostitsyna¹, Maarten Löffler², Valentin Polishchuk³, and Frank Staals⁴

- 1 Dept. of Mathematics and Computer Science, TU Eindhoven, The Netherlands, i.kostitsyna@tue.nl
- 2 Dept. of Computing and Information Sciences, Utrecht University, The Netherlands, m.loffler@uu.nl
- 3 Communications and Transport Systems, ITN, Linköping University, Sweden, valentin.polishchuk@liu.se
- 4 MADALGO, Aarhus University, Denmark, f.staals@cs.au.dk

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 min-link path's vertices or edges can be 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 2D, and provide first results in dimensions 3 and higher for several versions 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 [1] despite a large body of work on the topic. We also resolve the open problem from [2] mentioned in the handbook [3] (see Chapter 27.5, Open problem 3) and The Open Problems Project [4] (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.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases minimum-link path, diffuse reflection, terrain, bit complexity, NP-hardness

Digital Object Identifier 10.4230/LIPIcs.SoCG.2016.49

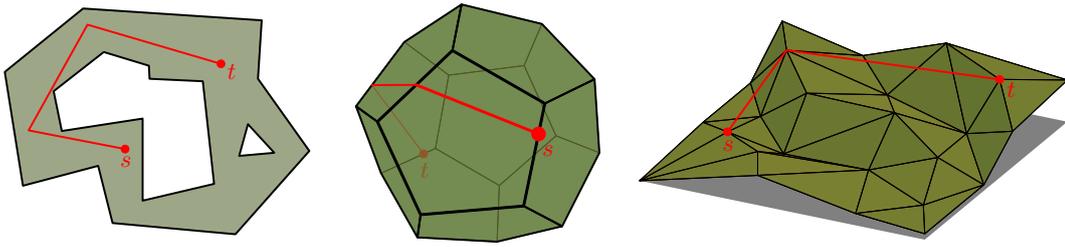
1 Introduction

The minimum-link path problem is fundamental in computational geometry [2, 5–11]. It concerns the following question: given a polyhedral domain D and two points s and t in D , find a 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 min-link paths. Second, we present and apply a novel generic NP-hardness construction. Third, we extend a simple polynomial-time approximation scheme.

* A preliminary version of this paper was presented at the 23rd Spanish Meeting on Computational Geometry in July 2015.





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

Concretely, our results resolve several open problems. We prove that computing a min-link *diffuse reflection path* in a polygon with holes [1] is NP-hard, and show that the min-link path problem in 3-space [3] (Chapter 27.5, Open problem 3) is NP-hard as well (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.

1.1 Problem Statement, Domains and Constraints

Due to their diverse applications, many different variants of min-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 consider several common domains, ranging from a simple polygon in 2D to complex scenes in full 3D or even in higher dimensions. Secondly, the links of the paths can be *constrained* to lie on the boundary of the domain, or bends may be restricted to vertices, edges, or higher-dimensional subcomplexes of the domain.

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

► **Definition 1.** We define $\text{MinLinkPath}_{a,b}(D, s, t)$, for $0 \leq a \leq b \leq d$ and $1 \leq b$, to be the problem of finding a min-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 $D|b$. Fig. 1 illustrates several instances of the problem.

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 [12]; 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 [13, 14]. For min-link paths, $O(n)$ -time algorithms are known for simple polygons [5–7], but for polygonal domains with holes the fastest known algorithm runs in nearly quadratic time [2], which may be close to optimal due to 3SUM-hardness of the problem [9].

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. Min-link paths are closely related to visibility problems, which have been studied extensively on terrains [15–17]. 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 [18–20]. To our knowledge, min-link paths in higher dimensions have not been studied before (with the exception of [21] that considered rectilinear paths).

Constraints In path planning on polyhedral surfaces or terrains, it is standard to restrict paths to the terrain. Min-link paths, on the other hand, have various geographic applications, ranging from feature simplification [11] to visibility in terrains [15]. 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 efficiency concerns, it is common to constrain bends even further to vertices of the domain (or, even more severely, the vertices of the terrain itself may be restricted to a grid, as in the *digital elevation map* (DEM) model).

In a vanilla min-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 [1, 10, 22, 23] 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 [24, 25]. 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 (and even in 2D, the complexity of the problem was open before).

1.2 Representation and Computation

In computational geometry, the standard model of computation is the *real RAM*, which represents the memory as an infinite sequence of storage cells, each of which can store any real number or integer. The real RAM is preferred for its elegance, but may not always be the best representation of physical computers. In contrast, the *word RAM* stores a sequence of w -bit words, where $w \geq \log n$ (and n is the problem size). The word RAM is much closer to reality, but complicates the analysis of geometric problems.

This difference is often insignificant, as the real numbers involved in solving many geometric problems are in fact algebraic numbers of low degree in a bounded domain, which can be described exactly with constantly many w -bit words. Path planning is notoriously different in this respect. Indeed, in the real RAM both the Euclidean shortest paths and the min-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 min-link paths, for which the models are *provably* different: Snoeyink and Kahan [8] observed that the region of points reachable by k -link paths may have vertices needing $\Omega(k \log n)$ bits to be represented. One of the results we present in this paper is the matching upper bound on the bit complexity of min-link paths in 2D.

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,

$\text{MinLinkPath}_{a,b}$	$b = 1$	$b = 2$	$b = 3$
$a = 0$	$O(n)$	$O(n^2)$	$O(n^2)$
$a = 1$	$O(n)$	Simple Polygon: $O(n^9)$ [10] Full 2D: NP-hard★ PTAS★	NP-hard★ (even in terrains) PTAS★
$a = 2$	N/A	Simple Polygon: $O(n)$ [5] Full 2D: $O(n^2 \alpha(n) \log^2 n)$ [2] PTAS★	NP-hard★ (even in terrains) PTAS★
$a = 3$	N/A	N/A	Terrains: $O(1)$ Full 3D: NP-hard★ PTAS★

■ **Table 1** Computational complexity of $\text{MinLinkPath}_{a,b}$ for $a \leq b \leq 3$. Results with citations are known, results marked with ★ are from this paper. Results without marks are trivial.

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 [26] (see also [27]).

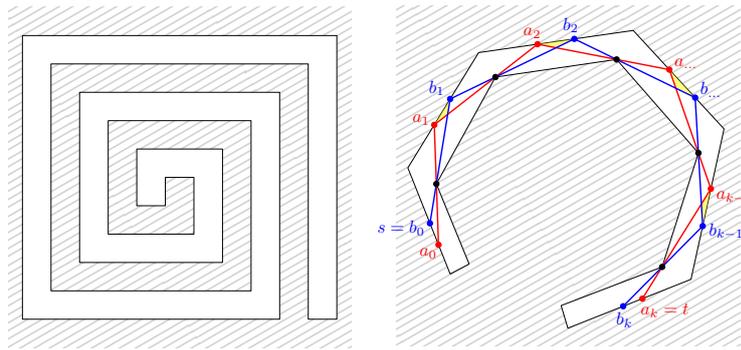
A common practical approach to avoiding bit complexity issues is to approximate the problem by restricting solutions to use only vertices of the input. In min-link paths, this corresponds to $\text{MinLinkPath}_{0,b}$. Although such paths can be computed efficiently, it can be shown with a simple example (refer to the full version [28]) that the number of links in such a setting may be a linear factor higher than when considering geometric versions.

1.3 Results

We give hardness results and approximation algorithms for various versions of the min-link path problem (see also Table 1). Specifically,

- In Section 2 we show a general lower bound on the bit complexity of min-link paths of $\Omega(n \log n)$ bits. (This was previously claimed, but not proven, by Snoeyink and Kahan [8].) We show that the bound is tight in 2D and we argue that this implies that $\text{MinLinkPath}_{a,2}$ is in NP. In Section 5, we argue that in 3D the boundary of the k -link reachable region can consist of $2k$ -th order algebraic curves, potentially leading to exponential bit complexity.
- In Section 3.1 we present a blueprint for showing NP-hardness of min-link path problems. We apply it to prove NP-hardness of the diffuse reflection path problem ($\text{MinLinkPath}_{1,2}$) in 2D polygonal domains with holes in Section 3.2. In Section 6, we use the same blueprint to prove that all non-trivial versions, defined above, of min-link path problems in 3D are weakly NP-hard. We also note that the min-link path problems have no FPTAS and no additive approximation (unless $P=NP$).
- In Section 4, we extend the 2-approximation algorithm from [3, Ch. 27.5], based on computing weak visibility between sets of potential locations of the path's bends, to provide a simple PTAS for $\text{MinLinkPath}_{2,2}$, which we also adapt to $\text{MinLinkPath}_{1,2}$. In Section 7, we give simple constant-factor approximation algorithms for higher-dimensional versions of min-link path problems, and use them to show that all versions admit PTASes.
- In Section 7.3, we focus on diffuse reflection ($\text{MinLinkPath}_{2,3}$) 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 [3, 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.

Omitted proofs can be found in the full version of the paper [28].



■ **Figure 2** (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.

2 Algebraic Complexity in \mathbb{R}^2

2.1 Lower bound on the Bit complexity

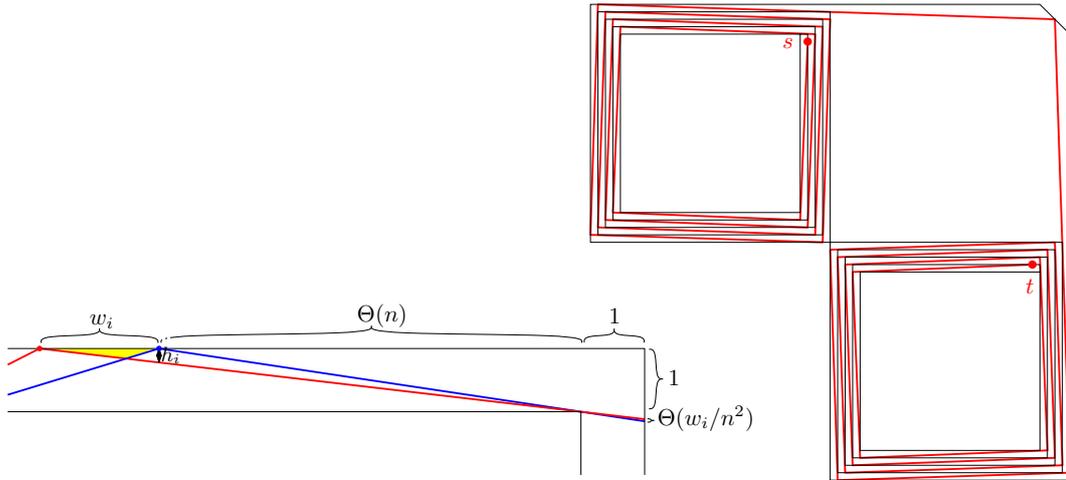
Snoeyink and Kahan [8] claim to “give a simple instance in which representing path vertices with rational coordinates requires $\Theta(n^2 \log n)$ bits”. In fact, they show that, there exists a simple polygon (whose vertices have integer coordinates encoded with $O(\log n)$ bits), such that the region reachable from one of its vertices s with k links has vertices whose coordinates have bit complexity $k \log n$. Note however, that this does not directly imply that a min-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 k -reachable region, one potentially could hope to avoid placing the internal vertices of an s - t min-link path on such high-complexity points). Below we present a construction where the intermediate vertices actually require $\Omega(k \log n)$ bits to be described, even if s and t can be represented using only $\log n$ bits each. We first prove this for the $\text{MinLinkPath}_{1,2}$ variant of the problem, and then extend our results to paths that may bend anywhere within the polygon, i.e., $\text{MinLinkPath}_{2,2}$.

► **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 min-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 min-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. 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 k -reachable point on e_k closest to b_k that has 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 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 Fig. 2 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



■ **Figure 3** Left: 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 . Right: An overview of our polygon P and the minimum-link path that has high-complexity coordinates.

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 one or close to k must contain low-complexity points. We now argue that we can build the construction such that $I_{k/2}$ 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 show that $I_{k/2}$ 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 Fig. 3 (left). Our overall construction is then shown in Fig. 3 (right).

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 point. 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. By choosing $J_{k/2}$ we have restricted the interval that must contain the $(k/2)^{\text{th}}$ bend. 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 of which much lie in the intervals J_i , $1 \leq i < k$. For example using the path that uses all points m_i . Thus, we have that t is reachable from s by a min-link path of k links, and we need $\Omega(k \log n)$ bits to describe the coordinates of some vertices in such a path. ◀

Next, we extend the construction from Lemma 2 to the case in which the bends may also lie in the interior of P .

► **Lemma 3.** *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 min-link path has vertices whose coordinates require $\Omega(k \log n)$ bits, where k is the length of a min-link path between s and t .*

2.2 Upper bound on the Bit complexity

In this section we show that the bound of Snoeyink and Kahan [8] on the complexity of k -link reachable regions is tight. Consider a polygon P and a point s in it. Let $\mathcal{R} = \{R_1, R_2, R_3, \dots\}$, where R_1 is the set of all points in P that see s , and R_i is the set of points in P that see some point in R_{i-1} for $i \geq 2$, i.e., region R_i consists of all the points in P that are illuminated by region R_{i-1} . Representing \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 matching lower and upper bounds on the bit complexity of a min-link path in \mathbb{R}^2 .

► **Theorem 4.** *Representing a vertex of region R_i requires $O(i \log n)$ bits. Representing the regions in \mathcal{R} as polygons with rational coordinates requires $O(n^2 \log n)$ bits each.*

► **Corollary 5.** *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.*

► **Theorem 6.** *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 5, 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 \mathbb{R}^2

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 $A = \{a_1, \dots, a_m\}$, find a subset $S \subseteq A$ whose sum is equal to half the sum of all the numbers. The main idea behind all the hardness reductions is as follows. Consider a 2D construction in Fig. 4 (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 paths from s to t that are only 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 $1 \leq i \leq m$ *important* (dashed lines in Fig. 4), and the set of horizontal lines $\ell'_i : y = 4i - 4$ for $2 \leq i \leq m$ *intermediate* (dash-dotted lines in Fig. 4). Each important line ℓ_i will “encode” the running sums of all

subsets of the first i integers $A_i = \{a_1, \dots, a_i\}$. 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 $2 \leq i \leq m$ *multiplying*, and the set of horizontal lines $f'_i : y = 4i - 3$ for $2 \leq i \leq m$ *reversing*. Each multiplying line f_i contains a fence with two zero-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 zero-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 min-link paths, i.e., into points on ℓ_i with x -coordinates equal to the 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 min-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 Fig. 4. The four points on line 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 of 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 $\{2a_1 + 2a_2 + a_3, a_1 + 2a_2 + a_3, 2a_1 + a_2 + a_3, a_1 + a_2 + a_3\}$. The reversing 0-slit of line f'_3 places the first four points into x -coordinates $\{0, a_1, a_2, a_1 + a_2\}$ on line ℓ_3 , and the reversing a_3 -slit of line f'_3 places the second four points into x -coordinates $\{a_3, a_1 + a_3, a_2 + a_3, a_1 + a_2 + a_3\}$ on ℓ_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 min-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 an 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 min-link path from s to t in our 2D construction is NP-hard.

► **Remark.** Instead of 0-width slits, we could use slits of positive width $w = o(\frac{1}{4m})$; since the width of the light beam grows by $2w$ between two consecutive important lines, the maximum shift of the path on ℓ_m due to the positive width of slits will be at most $(2m - 1) \times 2w < 1$.

3.2 Hardness of MinLinkPath_{1,2}

To show the hardness of the diffuse reflection problem in 2D, we turn our construction from Section 3.1 into a “zigzag” polygon (Fig. 4 (right)); 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 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 7.** *MinLinkPath_{1,2} in a 2D polygonal domain with holes is NP-hard.*

Overall our reduction bears resemblance to the classical *path encoding* scheme [29] used to prove hardness of 3D shortest path and other path planning problems, as we also repeatedly

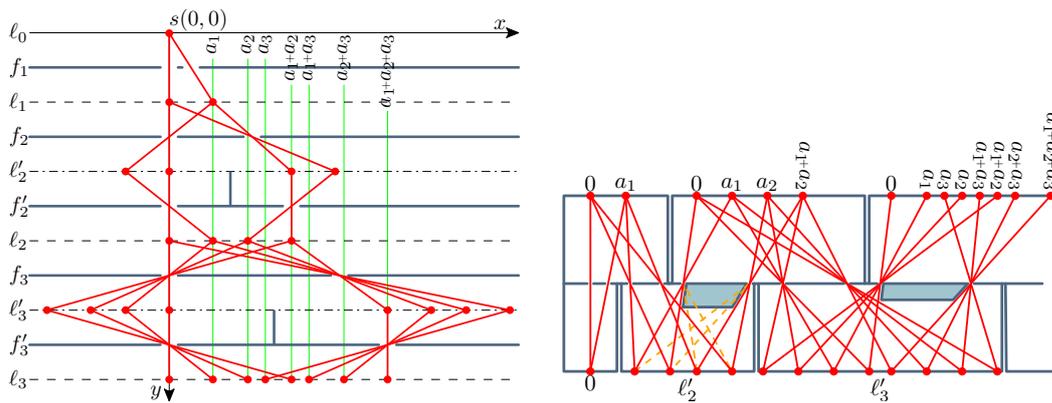


Figure 4 Left: The first few lines of a 2D construction depicting the general idea behind the hardness proofs: important lines l_0 – l_3 , intermediate lines l'_1 – l'_3 , multiplying lines f_1 – f_3 , and reversing lines f'_1 – f'_3 . The slits in the fences on multiplying and reversing lines are placed 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 $\{a_1, \dots, a_i\}$. Right: There exists an s - t diffuse reflection path with $2m - 1$ links iff 2-Partition instance is feasible.

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 & No additive approximation Obviously, problems with a discrete answer (in which a second-best solution is separated by at least 1 from the optimum) have no FPTAS. 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 $\text{poly}(mK) = \text{poly}(m)$ time, would also be able to solve 2-Partition in the same time.

4 Algorithmic Results in \mathbb{R}^2

4.1 Constant-factor Approximation

MinLinkPath $_{2,2}$ in 2D can be solved exactly [2]. For MinLinkPath $_{1,2}$, [1] gives a 3-approximation.

4.2 PTAS

We describe a $(1 + \varepsilon)$ -approximation scheme for 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 (i.e., $\bigcup F = D^{[1]}$). 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 $p \in f$ and $p' \in f'$; more generally, we say f and f' are k -link weakly visible if there is a k -link path from p to p' (with 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 $\pi^k = \{f_0, f_1, \dots, f_\ell\}$, 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 . We describe how to transform π^k into a solution π_*^k for the $\text{MinLinkPath}_{1,2}$ problem. Embed edges of π^k into D as k -link paths. This does not yet connect s to t since the endpoint of edge $f_{i-1}f_i$ inside feature f_i does not necessarily coincide with the endpoint of 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 8.** *The number of links in π_*^k is at most $(1 + 1/k)\text{opt}$.*

We now argue that the weak k -link visibility between features can be determined in polynomial time using “staged illumination”—the predominant technique for finding min-link paths (see Chapters 26.4 and 27.3 in the handbook [3]): 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 with k links iff $W^k(f) \cap f' \neq \emptyset$. For constant k , building $W^k(f)$ takes time polynomial in n , although possibly exponential in k (in fact, explicit bounds on the complexity of $W^k(f)$ for diffuse reflection problem were obtained in [10, 23, 30]). 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.

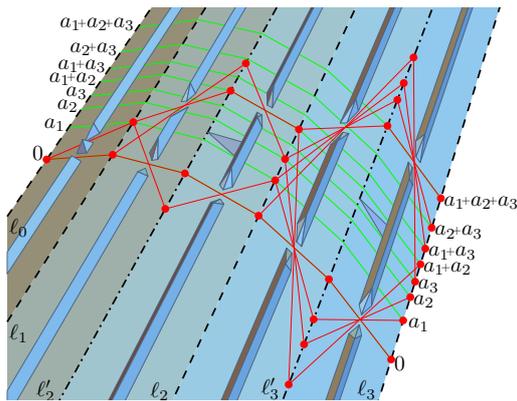
► **Theorem 9.** *For a constant k the path π_*^k , having at most $(1 + 1/k)\text{opt}$ links, can be constructed in polynomial time.*

5 Algebraic Complexity in \mathbb{R}^3

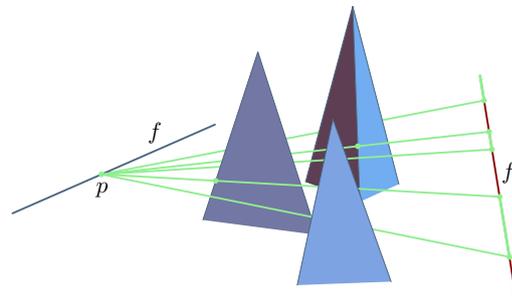
Consider the bit complexity of $\text{MinLinkPath}_{a,b}$ for a polyhedral domain D in \mathbb{R}^3 . The lower bounds on the bit complexity for $a \leq b \leq 2$ in \mathbb{R}^2 obviously extend to \mathbb{R}^3 . In this section we focus on the upper bound; we characterize region R_i reachable from a given point s with i links and discuss the upper bound on its bit complexity.

Order of the boundary curves Assume that 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., region R_i consists of all the 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 will speak of the boundaries (in the plural form of the word) of R_i , that we will denote as ∂R_i , we will 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 a union of all lines intersecting three given lines in 3D forms a hyperboloid, and therefore, a straight-line segment on the boundaries of R_{i-1}



■ **Figure 5** 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.



■ **Figure 6** 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.

forces the corresponding part of ∂R_i to be an intersection of a hyperboloid and a plane, i.e., a hyperbola. Moreover, the order of the curves of ∂R_i will grow with i , but at most linearly.

► **Theorem 10.** *The boundaries of region R_i are curves of order at most $2i + 1$ for $i \geq 2$, and at most 2 for $i = 1$.*

Bit complexity 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 allowed us to obtain only an exponential upper bound for the space required to store the coordinates of vertices of R_i .

► **Lemma 11.** *The coordinates of a vertex of R_i can be stored in $O(9^i)$ space.*

6 Computational Complexity in \mathbb{R}^3

We will show now how to use our blueprint from Section 3.1 to build a terrain for the $\text{MinLinkPath}_{1,2}$ problem such that a path from s to t with $2n - 1$ 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 $A = \{a_1, \dots, a_m\}$. Take the 2D construction and bend it along all the lines ℓ_i and ℓ'_i , except ℓ_0 and ℓ_m (refer to Fig. 5). Let the angles between consecutive faces be $\pi - \delta$ for some small angle $\delta < \pi/4m$ (so that the sum of the bends between the first face (between ℓ_0 and ℓ_1) and the last face (between ℓ'_m and ℓ_m) is less than π). Build a fence of height $\tan(\delta/4)$ on each face 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 through the slits in the fences. Finding a min-link path on such a terrain is equivalent to finding a min-link path (with bends restricted to ℓ_i and ℓ'_i) in the 2D construction. Therefore,

► **Theorem 12.** *$\text{MinLinkPath}_{1,2}$ on a terrain is NP-hard.*

Observe that bending in the interior of a face cannot reduce the link distance between s and t . Hence, our reduction also shows that $\text{MinLinkPath}_{2,2}$ is NP-hard. Furthermore,

lifting the links from the terrain surface into \mathbb{R}^3 also does not reduce link distance; we can make sure that the fences are low in height, so that fences situated on different faces do not see each other. Therefore, jumping onto the fences is useless. Hence, $\text{MinLinkPath}_{1,3}$ and $\text{MinLinkPath}_{2,3}$ are also NP-hard.

MinLinkPath_{a,b} in general polyhedra Since a terrain is a special case of a 3D polyhedra, it follows that $\text{MinLinkPath}_{1,2}$, $\text{MinLinkPath}_{2,2}$, $\text{MinLinkPath}_{1,3}$, and $\text{MinLinkPath}_{2,3}$ are also NP-hard for an arbitrary polyhedral domain in \mathbb{R}^3 . Our construction does not immediately imply that $\text{MinLinkPath}_{3,3}$ 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, $\text{MinLinkPath}_{3,3}$ is also NP-hard.

► **Corollary 13.** *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.*

7 Algorithmic Results in \mathbb{R}^3

7.1 Constant-factor Approximation

Our approximations refine and extend the 2-approximation for min-link paths in higher dimensions suggested in Chapter 26.5 (section Other Metrics) of the handbook [3] (see also Ch. 6 in [31]); 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 $D|^\alpha$, the set of all points where the path is allowed to bend, decompose $D|^\alpha$ into a set F of small-complexity convex pieces, and call each piece a *feature*. Similar to Section 4.2, we say that 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 $f' \cap W(f) \neq \emptyset$ (in terms of illumination, $W(f)$ is the set of points that get illuminated when light sources are placed at every point of f). See Fig. 6 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 a set $B(e) = \{(p, p') \in f \times f' : pp' \cap e \neq \emptyset\}$ 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 $a = 1$ [32]. 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 14.** *$M(f, f')$ can be built in $O(n^{\max(2, 4a-3+\varepsilon)})$ time, for an arbitrary $\varepsilon > 0$.*

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 $\pi = \{f_0, f_1, \dots, f_\ell\}$, with $f_0 = s$ and $f_\ell = t$ be a shortest s - t path in G ; ℓ is the length of π . Embed the edges of π into the geometric domain, putting endpoints of the edges arbitrarily into the corresponding features. This does not yet connect s to t since the endpoint of edge $f_{i-1}f_i$ inside feature f_i does not necessarily coincide with the endpoint of 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 min-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 $\ell \leq \text{opt}$. 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 $2\ell - 1 < 2\text{opt}$.

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 15.** (cf. [3, Ch. 27.5].) *A 2-approximation to $\text{MinLinkPath}_{a,b}$ 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 15 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 a 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 16.** *For a constant k the path π_*^k , having at most $(1 + 1/k)\text{opt}$ links, can be constructed in polynomial time.*

7.3 The global visibility map of a terrain

Using the result from Theorem 15 for $\text{MinLinkPath}_{2,3}$ on terrains, we get a 2-approximate min-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 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.

Specifically, the following results are proved in the full version of the paper [28]:

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

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

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

► **Theorem 20.** *The complexity of the 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 path problems in 3D, showing that most versions are NP-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. A related question is exploring bit complexity of min-link paths in 3D (note that finding a min-link path with integer vertices is already weakly NP-hard for the case of simple polygons in 2D [33]).

Acknowledgments We thank Joe Mitchell and Jean Cardinal for fruitful discussions on this work and the anonymous reviewers for their helpful comments. M.L. and I.K. are supported by the Netherlands Organisation for Scientific Research (NWO) under grants 639.021.123 and 639.023.208 respectively. 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] S. K. Ghosh, P. P. Goswami, A. Maheshwari, S. C. Nandy, S. P. Pal, and S. Sarvatomananda. “Algorithms for computing diffuse reflection paths in polygons.” In: *The Visual Computer* 28.12 (2012).
- [2] J. S. B. Mitchell, G. Rote, and G. Woeginger. “Minimum-link paths among obstacles in the plane.” In: *Algorithmica* 8.1 (1992).
- [3] J. E. Goodman and J. O’Rourke. *Handbook of Discrete and Computational Geometry*. Chapman & Hall/CRC, 2004.
- [4] E. D. Demaine, J. S. B. Mitchell, and J. O’Rourke. *The Open Problems Project*. <http://maven.smith.edu/~orourke/TOPP/>.
- [5] S. Suri. “A linear time algorithm with minimum link paths inside a simple polygon.” In: *Computer Vision, Graphics and Image Processing* 35.1 (1986).
- [6] S. K. Ghosh. “Computing the Visibility Polygon from a Convex Set and Related Problems.” In: *Journal of Algorithms* 12.1 (1991).
- [7] J. Hershberger and J. Snoeyink. “Computing Minimum Length Paths of a Given Homotopy Class.” In: *Computational Geometry: Theory and Applications* 4 (1994).
- [8] 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).
- [9] J. S. B. Mitchell, V. Polishchuk, and M. Sysikaski. “Minimum-link paths revisited.” In: *Computational Geometry: Theory and Applications* 47.6 (2014).
- [10] B. Aronov, A. R. Davis, J. Iacono, and A. S. C. Yu. “The Complexity of Diffuse Reflections in a Simple Polygon.” In: *Proc. 7th Latin American Symposium on Theoretical Informatics*. 2006.
- [11] L. J. Guibas, J. Hershberger, J. S. B. Mitchell, and J. Snoeyink. “Approximating Polygons and Subdivisions with Minimum Link Paths.” In: *Proceedings of the 2nd International Symposium on Algorithms*. 1991.
- [12] J. Adegest, M. H. Overmars, and J. Snoeyink. “Minimum-link C -oriented paths: Single-source queries.” In: *Int. J. of Computational Geometry and Applications* 4.1 (1994).

- [13] 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).
- [14] J. Hershberger and J. Snoeyink. “Computing minimum length paths of a given homotopy class.” In: *Computational Geometry: Theory and Applications* 4 (1994).
- [15] 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).
- [16] A. J. Stewart. “Hierarchical Visibility in Terrains.” In: *Eurographics Rendering Workshop*. 1997.
- [17] F. Hurtado, M. Löffler, I. Matos, V. Sacristán, M. Saumell, R. I. Silveira, and F. Staals. “Terrain visibility with multiple viewpoints.” In: *Proc. 24th International Symposium on Algorithms and Computation*. 2013.
- [18] E. Moet. “Computation and complexity of visibility in geometric environments.” PhD thesis. Utrecht University, 2008.
- [19] 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: Theory and Applications* 38.3 (2007).
- [20] G. Viglietta. “Face-Guarding Polyhedra.” In: *CCCG 2011*.
- [21] M. de Berg, M. J. van Kreveld, B. J. Nilsson, and M. H. Overmars. “Shortest path queries in rectilinear worlds.” In: *Int. J. of Computational Geometry and Applications* 3.2 (1992).
- [22] D. Prasad, S. P. Pal, and T. Dey. “Visibility with multiple diffuse reflections.” In: *Computational Geometry: Theory and Applications* 10 (1998).
- [23] 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).
- [24] 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.
- [25] M. de Berg. “Generalized hidden surface removal.” In: *Computational Geometry* 5.5 (1996).
- [26] D. El-Khechen, M. Dulieu, J. Iacono, and N. van Omme. “Packing 2×2 unit squares into grid polygons is NP-complete.” In: *CCCG 2009*.
- [27] M. Schaefer, E. Sedgwick, and D. Stefankovic. “Recognizing string graphs in NP.” In: *J. Comput. Syst. Sci.* 67.2 (2003).
- [28] I. Kostitsyna, M. Löffler, V. Polishchuk, and F. Staals. *On the complexity of minimum-link path problems*. <http://arxiv.org/abs/1603.06972>. 2016.
- [29] J. Canny and J. H. Reif. “New lower bound techniques for robot motion planning problems.” In: *Proc. 28th Annual Symposium on Foundations of Computer Science*. 1987.
- [30] 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).
- [31] C. Piatko. “Geometric bicriteria optimal path problems.” PhD thesis. Cornell University, 1993.
- [32] P. K. Agarwal and M. Sharir. “Arrangements and their applications.” In: *Handbook of computational geometry* (2000).
- [33] W. Ding. “On Computing Integral Minimum Link Paths in Simple Polygons.” In: *EuroCG*. 2008.