On the complexity of minimum-link path problems

Irina Kostitsyna*[1], Maarten Löffler†[2], Valentin Polishchuk‡[3], and Frank Staals§[2]

[1]Dept. of Mathematics and Computer Science, TU Eindhoven, Eindhoven, the Netherlands
[2]Dept. of Computing and Information Sciences, Utrecht University, Utrecht, the Netherlands
[3]Communications and Transport Systems, ITN, Linköping University, Sweden

Abstract

We study the problem of connecting two points on a polyhedral terrain with a minimum-link chain, whose vertices lie on the terrain and whose links go above the terrain. This kind of problem arises, e.g., in free-space optical communication technology, when a transmission channel has to be set up between two points that do not directly see each other. We give hardness results for several versions of the problem.

As a by-product, we resolve the open problem mentioned in the handbook [1] (see Chapter 27.5, Open problem 3) and The Open Problems Project [2] (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.

Last, but not least, we give similar results for *diffuse reflection paths* (paths that are allowed to bend only on the boundary of the domain) for two-dimensional polygonal domains with holes. Hardness of the problem has remained an open problem, despite a large body of work on the paths, motivated by ray tracing in graphics applications.

Introduction

Staged illumination. Link distance is intimately related to guarding and visibility: the set of points reachable with 1 link from a point s is exactly the set seen by s, and more generally, the region that can be seen from the set reachable from s with k links, is exactly the set reachable from s with $k+1$ links (for any k). This simple observation forms the basis of "staged illumination" paradigm—the predominant technique for finding minlink paths; see Chapters 26.4 and 27.3 in the handbook [1]. Visibility in terrains and general 3D scenes has been studied extensively both in theory and in practice [3–7], and it is well understood how to compute the area (volume) visible by one or even multiple viewpoints [8].

An intriguing open question has been to see whether the staged illumination works in higher dimensions: finding minlink path in 3D has been mentioned as an open problem in the handbook [1] (Chapter 27.5, Open problem 3) and in The Open Problems Project [2] (see Problem 22); the earliest reference to the problem is, perhaps, the 1993 thesis [9] (Chapter 6). In this paper we answer the question in the negative, showing NP-hardness of computing minimum-link paths in 3D (even when the obstacles are just a terrain).

Diffuse reflection. In a vanilla minlink 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 [10–14] 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 [15, 16]. 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. While there is evidence that in simple polygons the problem can be solved in polynomial time [12, 14], the complexity of the problem in polygonal domains with holes has been open [10]. In this paper we resolve the open question: we show that the problem is NP-hard.

1 Settings

The input to the minimum-link path problem consists of the environment and two points, s and t in it; the goal is to connect s to t by a path with minimum number of links. We use terms *links* and *bends* for edges and vertices of the path, saving the terms *edges* and *vertices* for those of the environment.

We consider various settings, which can be cate-

*Email: i.kostitsyna@tue.nl. Research supported by the Netherlands Organisation for Scientific Research (NWO) under grant 612.001.106

†Email: m.loffler@uu.nl. Research supported by the Netherlands Organisation for Scientific Research (NWO) under grant 639.021.123

‡Email: valentin.polishchuk@liu.se. Research supported by grant 2014-03476 from the Swedens innovation agency VINNOVA

§Email: f.staals@uu.nl. Research supported by the Netherlands Organisation for Scientific Research (NWO) under grant 612.001.022

	FreeFlight	Crawler	DiffRefl
Polygon	$O(n)$[17]	$O(n)$	$O(n^9)$[14]
Full 2D	$O^*(n^2)$[18]	$O(n)$	NP-hard
Terrain	$O(1)$	NP-hard	NP-hard
Full 3D	NP-hard	NP-hard	NP-hard

Table 1: Complexity of minimum-link path in the different environments and path types. Polynomial results are known or trivial; hardness results are from this work.

gorized by two aspects. Firstly, the *environment* in which we work can be, in order of increasing complexity, a simple polygon in 2D, a general 2D polygonal domain, a 3D terrain (xy-monotone surface), or a full 3D scene. Secondly, our minimum-link-path may be allowed to go anywhere in the domain (free flight), to stay on the boundary of the domain (crawler), or have its bends on the boundary while allowing the links to go anywhere in the domain (diffuse reflection). We now survey these settings in more detail.

Environments One classical distinction between working setups in 2D is *simple polygons* vs. *polygonal domains* (polygons with holes). For many geometric problems, more efficient algorithms are possible in simple polygons. Similarly, *terrains* are an important special case of *full* 3D scenes.

A common assumption in work on 3D scenes is that faces of the environment are triangles. For shortest paths planning and some other applications such an assumption can be made w.l.o.g., since all faces can be triangulated in linear time and adjacent coplanar triangles can be perturbed to break the coplanarity—this will not change the answer to the problem. On the contrary, for link distance computation (as well as, e.g., for visibility) coplanarity breaking may be fatal: we allow (and in some versions actually require) the links to stay on the surface of a face and span multiple coplanar triangles with a single link. Therefore in our problem formulation we will assume that faces are triangles, but allow degenerate inputs in which adjacent triangles may be coplanar. Similarly, for the diffuse reflection path problem in a 2D domain, we will assume that the domain is triangulated. Note that in the 2D setting, the restriction to bend on the domain boundary is equivalent to the restriction to bend on the domain edges (in diffuse reflection studies it is assumed that no reflection happens from the vertices [10]).

Path types We consider three different versions of minimum-link paths, depending on where the links and bends of the paths are allowed to be.

FreeFlight: The path may go anywhere through the free space in the environment. This model is standard for 2D path planning, as well as some applications in 3D.

Crawler: The path must stay on the boundary of the environment. This model is the standard for path planning on polyhedral surfaces or terrains. For full 3D scenes this model makes sense only if there is a single 3D obstacle, in which case it reduces to the case of a single polyhedral surface. In 2D scenes, the path is fully determined.

DiffuseReflection: The path's vertices (bends) must belong to the environment boundary, but the path's edges (links) may go anywhere through the free space. This model is motivated from diffuse reflection.

The different environments and path types can be combined into many different problems, some of which are trivial. Table 1 shows an overview of the possible combinations.

For minimum-link paths, $O(n)$-time algorithms are known for simple polygons in the realRAM model (standard in computational geometry) [17, 19], the bit complexity of the path's vertices may be quadratic [20]. For polygonal domains with holes the fastest known algorithm for minlink path runs in nearly quadratic time [18], which may be close to optimal due to 3SUM-hardness of the problem [21]. The Crawler model in 2D comes down to following the boundary of the domain, which is silly and can be done in linear time.

The state-of-the-art of link distance problems in 3D is much duller: complexity of no version has been known before. For FreeFlight paths on terrains, the problem is not interesting: any two points can be connected via a path with just a single bend. In this paper, we show that all other versions are NP-hard.

2 Hardness

We prove that all versions, defined above, of minlink problems in 3D are NP-hard. We worked out a single reduction idea that works for all the problems, and we believe the idea can work to give hardness results for other problems as well: as a demonstration, we apply it to prove the NP-hardness of the diffuse reflection path problem in 2D polygonal domains with holes.

We reduce from the 2-Partition problem: Given a set of integers $A = \{a_1, \ldots, a_m\}$, is there a subset $A' \subseteq A$ whose sum is equal to half the sum of all numbers. One feature of our hardness proof is that essentially the same reduction works for all our problems, including the 2D diffuse reflection path. For concreteness we present the reduction for Terrain-Crawler-e (path must stay on the terrain and bend only on edges); the modifications for the other problems are given next.

The terrain in the reduction consists of a *creased rectangle*, on which *fences*, with two *slits* in each fence, are installed between consecutive creases; the rectangle does not depend on the instance of 2-Partition, while the fences do (more precisely, the location of one of the slits in each fence is instance-dependent)—see Figure 1 for the overall construction. All creases

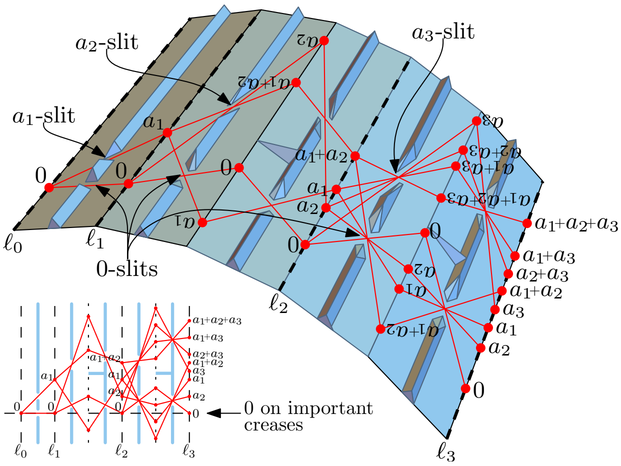

Figure 1: The first 6 creases; the important creases are dashed. The locations on ℓ_i that are reachable with $2i-1$ links correspond to sums formed by all possible subsets of $\{a_1, \dots, a_i\}$. The flipped-over numbers on intermediate creases are just labels (not coordinates of the points) showing the positions from which the corresponding locations on the following important creases are obtained by mirroring in the slits (the labels are flipped to emphasize that they are mirror images). Bottom left corner shows the view of our construction from the top when the creased rectangle is unfolded (i.e., creases are straightened). The important and intermediate creases are dashed and dotted resp. For the fences, only their top ridges are shown (blue).

are parallel to two sides, ℓ_0 and ℓ_m, of the creased rectangle; abusing the notation, ℓ_0 and ℓ_m are also called creases (they can be turned into real creases by attaching an extra rectangular piece to each of ℓ_0, ℓ_m). Overall there are $2m$ creases, and the even creases are named $\ell_1, \ell_2, \dots, \ell_m$; we call them (and the first crease, ℓ_0) the *important* creases. The other creases, $\ell'_1, \ell'_2, \dots, \ell'_{m-1}$, are called *intermediate*. We envision that each of the important creases is a real line, and that the 0s on all important creases are aligned; in particular, by *location* of point p on an important crease we will understand the (signed) distance from 0 to p (i.e., the coordinate of p on the crease). The points s and t are put on the creases ℓ_0 and ℓ_m at locations 0 and $\sum a_i/2$ resp.

Fences run parallel to the creases, exactly in the middle of the face between the consecutive creases. Any fence is just a wall of small height (the reason for making the height small is to make sure that the path would not benefit from jumping onto fences—this is important in other versions). The two slits on the fence between creases ℓ_{i-1} and ℓ'_{i-1} correspond to values 0 and a_i. And the two slits on the fence between creases ℓ'_{i-1} and ℓ_i restores the order of locations between ℓ'_{i-1} and ℓ_i (see Figure 1).

Starting from a point at any location a on an important crease ℓ_{i-1}, one can reach the same location a on the important crease ℓ_i in 2 links (straddling through the 0-slits in the two fences that separate the creases), or the location $a + a_i$ (going through the a_i-slits). Reaching any other point $b \neq a$ or $a + a_i$ on ℓ_i requires more than 2 links since any path must bend on every crease.

To avoid leakage from 0-slits to a_i-slits (or back) between consecutive important creases, additional fences, serving as curtains, are installed perpendicularly to the fences, between the slits, in faces between creases ℓ'_{i-1} and ℓ_i. This way, in order to reach ℓ_i from ℓ_{i-1}, any path must either use the two 0-slits in the fences between the creases (which brings the path to the same location and corresponds to not taking a_i into the subset) or use the two a_i-slits (which shifts the location by a_i and corresponds to picking a_i to the subset). Overall our construction possesses the following crucial property: the locations on the crease ℓ_i that can be reached from s with $2i-1$ links are exactly the sums of all possible subsets of the first i numbers in A. In particular, t, the point at location $\sum a_i/2$ on ℓ_m can be reached with $2m-1$ links iff the instance of 2-Partition is feasible.

Theorem 1 *Terrain-Crawler-e is NP-hard.*

Our reduction directly works to show hardness also of Terrain-Crawler (path must stay on the terrain but can bend anywhere), Terrain-DiffRefl (path can go anywhere through the free space but can bend only on the surface of the terrain) and Terrain-DiffRefl-e (path can go anywhere through the free space but can bend only on the edges of the terrain) because bending in the interior of faces and/or lifting the links from the terrain surface does not reduce link distance between s and t (to make jumping onto the fences useless, make the fences low in height, so that fences situated on different faces of the creased rectangle do not see each other). The 3D versions are more general than the terrain ones; to see this put a copy of the terrain, without the fences, slightly above the original terrain (so that the only free space is the thin layer between the terrains): if the layer is thin enough, the ability to take off from the terrain does not help decreasing the link distance from s to t.

Corollary 2 *All versions of the minlink problem in 3D settings defined in Section 1 are NP-hard.*

Diffuse reflection in 2D. We can "flatten" our construction and turn the faces of the creased rectangle into "corridors", while turning the creases into edges of a "zigzag" polygon; 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 edges that correspond to consecutive creases of the terrain (Figure 2). This retains the crucial property of our terrain construction: locations reachable with fewest links on the edges correspond to sums of numbers in the subsets of A.

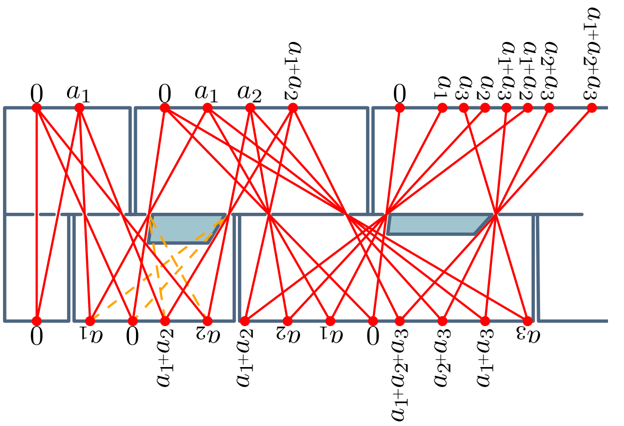

Figure 2: There exists an s-t diffuse reflection path with $2m-1$ links iff 2-Partition instance is feasible.

Theorem 3 *The Diffuse reflection path problem in a 2D polygonal domain with holes is NP-hard.*

Overall our reduction bears resemblance to the classical *path encoding* scheme [22] 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.

References

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

[2] E. D. Demaine, J. S. B. Mitchell, and J. O'Rourke, *The open problems project*, `http://maven.smith.edu/~orourke/TOPP/`.

[3] L. D. Floriani and P. Magillo, "Algorithms for visibility computation on terrains: A survey," *Environment and Planning B: Planning and Design*, vol. 30, no. 5, 2003.

[4] F. Kammer *et al.*, "Practical approaches to partially guarding a polyhedral terrain," in *Geographic Information Science*, 2014.

[5] B. Ben-Moshe, P. Carmi, and M. J. Katz, "Approximating the visible region of a point on a terrain," *GeoInformatica*, vol. 12, no. 1, 2008.

[6] A. Efrat *et al.*, "On incremental rendering of silhouette maps of a polyhedral scene," *Computational Geometry*, vol. 38, no. 3, 2007.

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

[8] F. Hurtado *et al.*, "Terrain visibility with multiple viewpoints," in *Proc. 24th International Symposium on Algorithms and Computation*, 2013.

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

[10] S. K. Ghosh *et al.*, "Algorithms for computing diffuse reflection paths in polygons," *The Visual Computer*, vol. 28, no. 12, 2012.

[11] D. Prasad, S. P. Pal, and T. Dey, "Visibility with multiple diffuse reflections," *Computational Geometry: Theory and Applications*, vol. 10, 1998.

[12] A. Bishnu *et al.*, *An algorithm for computing constrained reflection paths in simple polygon*, `http://arxiv.org/abs/1304.4320`, 2013.

[13] B. Aronov *et al.*, "Visibility with multiple reflections," *Discrete & Computational Geometry*, vol. 20, no. 1, 1998.

[14] B. Aronov *et al.*, "The complexity of diffuse reflections in a simple polygon," in *Proc. 7th Latin American Symposium on Theoretical Informatics*, 2006.

[15] J. D. Foley *et al.*, *Introduction to Computer Graphics*. Addison-Wesley Longman Publishing Co., Inc., 1994.

[16] M. de Berg, "Generalized hidden surface removal," *Computational Geometry*, vol. 5, no. 5, 1996.

[17] S. Suri, "A linear time algorithm with minimum link paths inside a simple polygon," *Computer Vision, Graphics and Image Processing*, vol. 35, no. 1, 1986.

[18] J. Mitchell, G. Rote, and G. Woeginger, "Minimum-link paths among obstacles in the plane," *Algorithmica*, vol. 8, no. 1, 1992.

[19] J. Hershberger and J. Snoeyink, "Computing minimum length paths of a given homotopy class," *Computational Geometry: Theory and Applications*, vol. 4, 1994.

[20] S. Kahan and J. Snoeyink, "On the bit complexity of minimum link paths: Superquadratic algorithms for problems solvable in linear time," *Computational Geometry: Theory and Applications*, vol. 12, no. 1-2, 1999.

[21] J. S. B. Mitchell, V. Polishchuk, and M. Sysikaski, "Minimum-link paths revisited," *Computational Geometry*, vol. 47, no. 6, 2014.

[22] 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.