

Geometric k shortest paths

Citation for published version (APA):

Eriksson-Bique, S., Hershberger, J., Polishchuk, V., Speckmann, B., Suri, S., Talvitie, T., ... Yildiz, H. (2015). Geometric k shortest paths. In Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'15, San Diego CA, USA, January 4-6, 2015) (pp. 1616-1625). Philadelphia: Society for Industrial and Applied Mathematics (SIAM). DOI: 10.1137/1.9781611973730.107

Document license:

Unspecified

DOI:

[10.1137/1.9781611973730.107](https://doi.org/10.1137/1.9781611973730.107)

Document status and date:

Published: 01/01/2015

Document Version:

Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Geometric k Shortest Paths*

Sylvester Eriksson-Bique[†]

John Hershberger[‡]

Valentin Polishchuk[§]

Bettina Speckmann[¶]

Subhash Suri^{||}

Topi Talvitie[§]

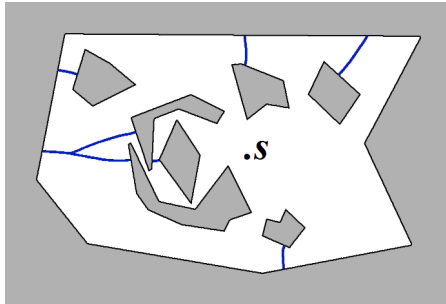
Kevin Verbeek^{||}

Hakan Yıldız^{||}

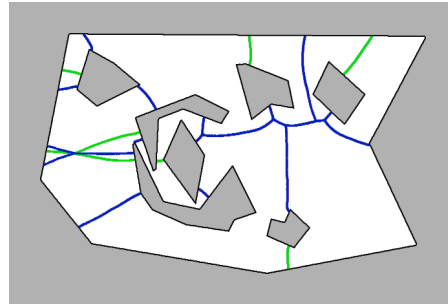
Abstract

We consider the problem of computing k shortest paths in a two-dimensional environment with polygonal obstacles, where the j th path, for $1 \leq j \leq k$, is the shortest path in the free space that is also *homotopically* distinct from each of the first $j - 1$ paths. In fact, we consider a more general problem: given a source point s , construct a partition of the free space, called the *k th shortest path map* (k -SPM), in which the homotopy of the k th shortest path in a region has the same structure. Our main combinatorial result establishes a tight bound of $\Theta(k^2h + kn)$ on the worst-case complexity of this map. We also describe an $O((k^3h + k^2n) \log(kn))$ time algorithm for constructing the map. In fact, the algorithm constructs the j th map for every $j \leq k$. Finally, we present a simple visibility-based algorithm for computing the k shortest paths between two fixed points. This algorithm runs in $O(m \log n + k)$ time and uses $O(m + k)$ space, where m is the size of the visibility graph. This latter algorithm can be extended to compute k shortest *simple* (non-self-intersecting) paths, taking $O(k^2m(m + kn) \log(kn))$ time.

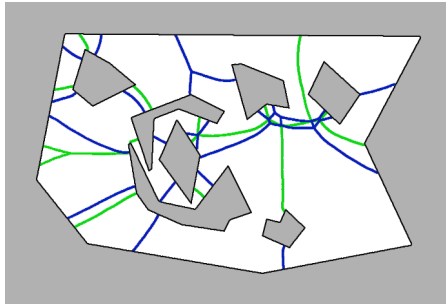
walls of 1-SPM:



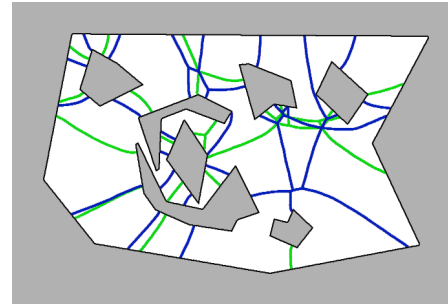
walls of 2-SPM:



walls of 3-SPM:



walls of 4-SPM:



We invite the reader to play with our applet demonstrating k -SPMs at

http://www.cs.helsinki.fi/group/compeom/kpath_slides/visualize/.

*B. Speckmann and K. Verbeek were supported by the Netherlands' Organisation for Scientific Research (NWO) under project no. 639.022.707. Sylvester E-B was supported as a Graduate Student Fellow by the National Science Foundation grant no. DGE-1342536.

[†]Courant Institute, NYU. ebs@cims.nyu.edu

[‡]Mentor Graphics Corporation. john.hershberger@mentor.com

[§]Communications and Transport Systems, ITN, Linköping University. firstname.lastname@liu.se

[¶]Dept. of Mathematics and Computer Science, TU Eindhoven. b.speckmann@tue.nl

^{||}Computer Science, University of California Santa Barbara. [suri|kverbeek|hakan]@cs.ucsb.edu

1 Introduction

In many applications of mathematical optimization, several “good” solutions are more desirable than a single optimum. This happens because a mathematical model is an imperfect formulation of complex reality, and its various constraints and objectives are only an approximation of the desired goal. Optimization problems are also typically part of a larger system with many interacting parts, where optimal solutions of different parts may be incompatible. In these settings, the system designer must find sub-optimal but high-quality solutions for each part to construct the overall solution. Motivated by these considerations, there is a long history of research on finding k best solutions for discrete optimization problems, including spanning trees and shortest paths in graphs [8, 12, 15, 22].

In this paper, we investigate the fundamental problem of computing k distinct shortest paths among polygonal obstacles in the plane. Because geometric shortest paths live in a continuous (free) space, we need a topological condition on paths to ensure that different paths are non-trivially distinct: otherwise, we can create many nearly identical shortest paths by adding infinitesimal “bumps” to the primary shortest path. The most natural condition is to require paths to have different *homotopy*, where two paths are said to be homotopically equivalent if they can be deformed into each other within the free space of obstacles. Intuitively, two paths are homotopically distinct if they lie on different sides of some obstacle. Multiple shortest paths of distinct homotopies naturally capture the high-level design criteria in geometric environments: e.g., in VLSI design or printed circuit board routing, where obstacles are electronic components, in robot path planning, where obstacles are physical obstructions, in air traffic management, where obstacles model hazardous weather or no-fly zones, etc.

We consider a more general form of the problem: given a source point s , construct a map of the entire free space, partitioning it into equivalence class regions such that the k th shortest path from s to any point in a single region has the same structure. We call this map the k th shortest path map (or k -SPM for short). With the k -SPM, one can compute the k th shortest path to any target easily. The following paragraph describes the key results of our paper.

Our Results We prove that the edges of the k -SPM are $O(k^2h + kn)$ linear or hyperbolic arcs, and give a construction showing that this bound is tight in the worst case (Section 4). We present an $O((k^3h + k^2n) \log(kn))$ time algorithm (Section 5), using the continuous Dijkstra paradigm, for constructing the map. The algorithm computes the j th shortest path map for all $1 \leq j \leq k$. Taking this into account, the algorithm is output sensitive: its running time is $O(\log(kn))$ times the total complexity of the first k shortest path maps. By preprocessing the k -SPM for point-location queries, we can answer k th shortest path queries in $O(\log(kn))$ time; if we want to report (in implicit form) all k shortest paths, the preprocessing time remains the same, but the storage and query time both increase by a factor of k . (If the paths are to be reported explicitly, the complexity of the paths must be added to the query time.) In Section 6, we also present a simpler, albeit asymptotically worse, algorithm for computing the k th shortest path between two fixed points based on the visibility graph. This algorithm runs in $O(m \log n + k)$ time and uses $O(m + k)$ space, where m is the size of the visibility graph. One advantage of this latter algorithm is that it can also be extended to find the k th *simple* (non-self-intersecting) path, taking $O(k^2m(m + kn) \log kn)$ time.

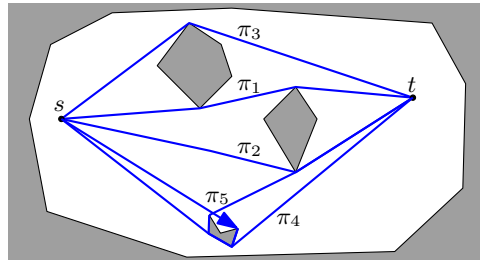


Figure 1: $|\pi_1| < |\pi_2| = |\pi_3| < |\pi_4| < |\pi_5|$. π_1 is the shortest path to t (a 1-path; cf. Def. 2.2), each of π_2 and π_3 is a 2-path, π_4 is a 4-path, π_5 is a 5-path (π_5 is nonsimple—it has a loop going clockwise around the hole).

60 **Related work** Finding shortest paths is also a central problem in the study of graph algorithms. Apart from
 61 finding the shortest path itself, considerable attention has been paid to computing its various alternatives
 62 including the second, third, and in general k th shortest path between two nodes in a graph; see, e.g., [8, 12]
 63 and references therein. On the other hand, *geometric k th shortest paths* have not been explored before.
 64 (One problem for which both the graph and the geometric versions were considered is finding the k smallest
 65 spanning trees [6, 7].)

66 In [19] Mitchell surveys many variations of the geometric shortest path problem; for some recent work
 67 see [3, 4]. In addition to computing one shortest path to a single target point, a lot of attention in the litera-
 68 ture has been devoted to building shortest path *maps*—structures supporting efficient shortest-path queries.
 69 A shortest path map can be viewed as the Voronoi diagram of vertices of the domain, where each vertex is
 70 (additively) weighted by the shortest-path distance from the source s [13]. Our study of “ k th shortest path
 71 maps” benefits from notions introduced by Lee [16] for *higher-order* Voronoi diagrams: when bounding the
 72 complexity of the maps in Section 4, we employ Lee’s ideas to define “old” and “new” features of the map
 73 and to derive relationships between them. Higher-order Voronoi diagrams have been recently reexamined in
 74 [1, 17, 18, 20]; in particular, [17] considered geodesic diagrams in polygonal domains. Perhaps unsurpris-
 75 ingly, the complexity of our k th shortest path map differs from that of an order- k geodesic Voronoi diagram;
 76 the major difference is that homotopies are irrelevant for Voronoi diagrams, but are central in our work.

77 2 Preliminaries

78 We are given a closed polygonal domain P with n vertices and h holes; the holes are also called “obstacles”
 79 and the domain is called the “free space.” We assume that no three vertices of P are collinear and make other
 80 general position assumptions below, as needed. We are also given a source point $s \in P$; unless otherwise
 81 stated, all paths will have s as an endpoint. For a point $p \in P$, two paths to p are *homotopically equivalent*
 82 if one can be continuously deformed to the other while staying within P . Homotopically equivalent paths
 83 form an equivalence class (the *homotopy class*) in the set of s - p paths. A path is *locally shortest* if its length
 84 cannot be reduced by an infinitesimal perturbation of the path (i.e., a pulled-taut path).

85 **Lemma 2.1** ([11]). *A locally shortest path is the unique shortest path of its homotopy class. Furthermore,*
 86 *all bends of a locally shortest path are at vertices of P and turn toward the corresponding obstacles.*

87 Let $d(p)$ denote the shortest-path (geodesic) distance from s to p . A vertex v of P is a *predecessor* of
 88 p if segment \overline{vp} is in free space and $d(p) = d(v) + |\overline{vp}|$. The *shortest path map* of P (or SPM for short) is
 89 the partitioning of P such that all points within the same cell of the SPM have the same unique predecessor.
 90 The edges of the partition are called *bisectors*; points on bisectors have more than one predecessor. We
 91 distinguish between two types of bisectors: *walls* and *windows*. A bisector is a wall if, for a point p on the
 92 bisector, there exist two homotopically different paths to p with length $d(p)$. If there is a unique shortest
 93 path to a point p on a bisector, then this bisector is a window; any point p on a window has two predecessors
 94 that are collinear with p . We assume that there is a unique shortest path to each vertex of P , and that there
 95 are at most three homotopically different shortest paths to each point in P . The former assumption implies
 96 that walls are 1-dimensional curves. The endpoints of a wall are either at an obstacle or at a *triple point*,
 97 where three walls meet. Windows start at vertices of P and extend until an obstacle or wall is hit. Intuitively,
 98 windows can mostly be ignored as far as homotopy types are concerned; walls, by contrast, are central to
 99 our study. Fig. “1-SPM” on the title page shows an example of walls in the SPM. By using standard point
 100 location structures on the SPM of P , one can query the shortest path length to any point in P in $O(\log n)$
 101 time and, in addition, report the path in linear output sensitive time [13]. Our goal is to compute a similar
 102 structure for k th shortest paths.

103 We now introduce the subject of our study. For a point $p \in P$, let $H(p)$ denote the set of locally shortest
 104 paths from s to p of all possible homotopy types.

105 **Definition 2.2.** A path $\pi \in H(p)$ is a k th shortest path (or is a k -path) to p if there are exactly $k - 1$ shorter
 106 paths in $H(p)$.

107 Figure 1 illustrates the definition. We denote the length of the k -path(s) to p by $d_k(p)$. Notice that, under
 108 these definitions, the term 1-path is synonymous with “shortest path” and $d(p) = d_1(p)$.

109 In order to extend the map concept to k -paths, we first generalize the definition of a predecessor. Let
 110 v be an obstacle vertex and i be an integer between 1 and k . For a point p in the plane, the pair (v, i) is a
 111 k -predecessor of p if the segment \overline{vp} is in free space and $d_k(p) = d_i(v) + |\overline{vp}|$. This implies that a k -path
 112 to p can be obtained by concatenating the segment \overline{vp} with the i -path to v . As with the SPM, we assume
 113 that each obstacle vertex has a unique i -path for any i , and that there are at most three i -paths in $H(p)$ for
 114 each point $p \in P$. Interestingly, for $i > 1$, the former assumption does not follow from a general position
 115 assumption. We discuss this issue in Appendix A. For the sake of simplicity, we will ignore the issue in the
 116 main body of the paper and stick to the assumption above.

117 Observe that, given the k -predecessors of all points in the plane and the i -predecessors of all obstacle
 118 vertices for $1 \leq i \leq k$, one can construct the k -path to any given point p . The k th shortest path map (or
 119 k -SPM for short) of P is a subdivision of P into cells such that all points within the same cell have the
 120 same unique k -predecessor. In order to construct k -paths from the k -SPM, we also assume that it stores
 121 the i -predecessors of all vertices, for all $1 \leq i \leq k$. As with the SPM, one can use standard point location
 122 structures to report the k -path length of a query point in $O(\log C_k)$ time, where C_k is the complexity of the
 123 k -SPM.

124 To distinguish the different types of bisectors that form the boundaries of the k -SPM, we generalize the
 125 definitions of walls and windows as follows:

126 **Definition 2.3.** A point p is on a k -wall if $H(p)$ contains at least two k -paths.

127 **Definition 2.4.** A point p is on a k -window if $H(p)$ contains exactly one k -path and p has two k -predecessors.

128 Note that the two predecessors of a point p on a k -window must be collinear with p . Furthermore, by
 129 the definition of k -paths, a point cannot be on a k -wall and a $(k + 1)$ -wall at the same time (if a point has
 130 two k -paths, then it has no $(k + 1)$ -path). Similarly to walls in the SPM, k -walls have their endpoints either
 131 on obstacles or at triple points, where three k -walls meet. In Section 3, we show that edges of the k -SPM
 132 are $(k - 1)$ -walls, k -walls and k -windows. We also show that our assumption that a k -predecessor is of the
 133 form (v, i) with $1 \leq i \leq k$ is indeed correct.

134 3 Structural results

135 Consider a path π from s to some target $t \in P$. This path crosses several walls (1-walls, 2-walls, etc.) in
 136 P . We define the *crossing sequence* of π as the sequence of positive integers that represents all the i -walls
 137 crossed by this path going back from t to s . That is, if π crosses an i -wall, we add i to the sequence. Although
 138 it is not strictly necessary, we generally assume an upper bound on the sequence values (the maximum wall
 139 class), so that the sequence is finite. We call a sequence a k -sequence if it adheres to the following inductive
 140 definition:

- 141 • A 1-sequence does not contain 1.
- 142 • A k -sequence contains $(k - 1)$, the first $(k - 1)$ occurs before the first k , and the tail of the sequence
 143 after the first $(k - 1)$ is a $(k - 1)$ -sequence.

144 We need the following property of k -sequences, whose proof appears with other omitted proofs in Ap-
 145 pendix E.

146 **Lemma 3.1.** *A sequence σ cannot be both a k -sequence and an ℓ -sequence if $k \neq \ell$.*

147 The relation between k -sequences and k -paths is summarized in the following lemma.

148 **Lemma 3.2.** *A locally shortest path π is a k -path if and only if its crossing sequence is a k -sequence.*

149 *Proof.* We first show that the crossing sequence of a k -path π is a k -sequence. Let us assume that distances
 150 have been scaled so that the length of π is 1. Define $p(x)$ for $0 \leq x \leq 1$ as the point on π such that the
 151 distance from t to $p(x)$ along π is x . Let $\gamma(x)$ be the subpath of π from $p(x)$ to t . For any $i \geq 1$, let π_i denote
 152 the i -path to t ($\pi = \pi_k$). (We assume that t is not on an i -wall, for any $1 \leq i \leq k$.) The concatenation of
 153 π_i and $\gamma(x)$ is a path from s to $p(x)$, via t ; let $\pi'_i(x)$ denote the shortest path of this homotopy class (Fig. 2,
 154 left). All paths $\pi'_i(x)$ must have different homotopy classes for different i .

155 Let $l_i(x)$ be the length of $\pi'_i(x)$; clearly l_i is continuous. By the definition of k -paths, $l_i(0) \leq l_j(0)$ for
 156 $i < j$. On the other hand, $l_k(1) = 0$ and $l_i(1) > 0$ for $i \neq k$. Note that as x grows from 0 to 1, $l_k(x)$
 157 decreases not slower than any other $l_i(x)$, $i \neq k$. Thus, the graph of $l_k(x)$ crosses the graphs of all $l_i(x)$ for
 158 $i < k$ exactly once, but no other graphs (Fig. 2, right).

159 The proof proceeds by induction. A point $p(x)$ is on a j -wall if $l_k(x)$ crosses some other graph at x ,
 160 and there are exactly $j - 1$ graphs that pass below this intersection. Clearly, if $k = 1$, the path π_k cannot
 161 cross a 1-wall, since $l_k(x)$ cannot intersect anything. For $k > 1$, the first intersection of $l_k(x)$ must be with a
 162 graph $l_i(x)$ with $i < k$, as described above. This means that $p(x)$ must cross a $(k - 1)$ -wall before crossing
 163 a k -wall. After the $(k - 1)$ -wall at $x = x^*$, the path $\pi'_k(x^*)$ is the $(k - 1)$ -path to $p(x)$. By induction, the
 164 remainder of the crossing sequence must be a $(k - 1)$ -sequence.

165 Finally note that a locally shortest path π must be an i -path for some $i \geq 1$. If the crossing sequence of π
 166 is a k -sequence, then it cannot be an i -sequence for $i \neq k$ by Lemma 3.1. Thus $i = k$, and π is a k -path. \square

167 Lemma 3.2 implies that a k -path from s to t crosses walls “in order”: it crosses a 1-wall, then a 2-wall,
 168 etc., until it crosses a $(k - 1)$ -wall, after which it reaches t . Also, any prefix of the k -path is an i -path if it
 169 crosses the $(i - 1)$ -wall and not the i -wall. This property of k -paths inspires the construction of a “parking
 170 garage” obtained by “stacking” k copies (or *floors*) of P on top of each other and gluing them along i -walls,
 171 for $1 \leq i \leq k$. To be precise, the k -garage is inductively defined as follows:

172 The 1-garage is simply P . The $(k + 1)$ -garage can be obtained by adding a copy of P (the
 173 $(k + 1)$ -*floor*) on top of the k -garage. We cut both the k -floor of the k -garage and the $(k + 1)$ -
 174 floor along k -walls. We then glue one side of a k -wall on the k -floor to the opposite side of the
 175 same k -wall on the $(k + 1)$ -floor, and vice versa, to obtain the $(k + 1)$ -garage.

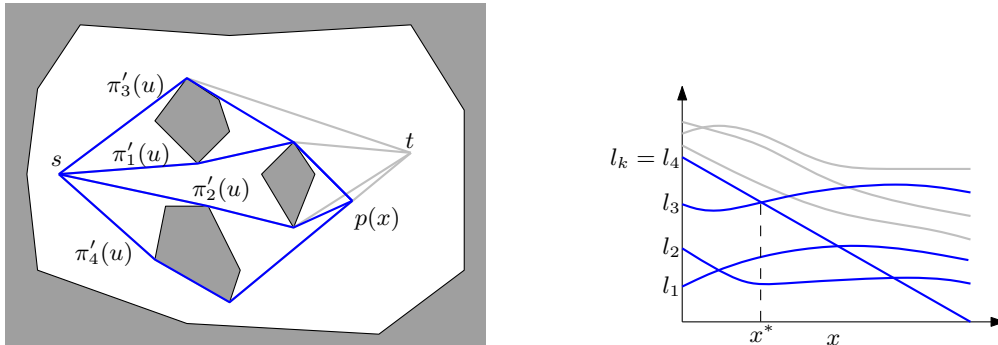


Figure 2: $k = 4$. Left: $\pi'_i(x)$ is the shortest path to $p(x)$, homotopically equivalent to s - π_i - t - $p(x)$. Right: l_k is k th smallest at $x = 0$ and decreases faster than any other l_i .

176 The k -garage resembles a covering space of P . However, due to the triple points formed by the i -walls
 177 ($i < k$), the k -garage is technically not a covering space, but something that is known as a ramified cover.
 178 Nonetheless, each path π in the garage can be projected down to a unique path π^\downarrow in P . The next lemma
 179 relates the k -SPM of P to the SPM of the k -garage.

180 **Lemma 3.3.** *If π is the shortest path in the k -garage from s on the 1-floor to some t on the k -floor, then π^\downarrow
 181 is a k -path to t .*

182 *Proof.* We show that the crossing sequence of π^\downarrow is a k -sequence. Then, by Lemma 3.2, π^\downarrow is a k -path. We
 183 use the property that every tail of a k -sequence is an i -sequence for some $i \leq k$. If, going back from t to s ,
 184 π only goes “down” in the k -garage, then it is easy to see that the crossing sequence of π^\downarrow is a k -sequence.
 185 (Because regions on the i -floor are bounded by $(i - 1)$ - and i -walls, π enters the i -floor by crossing an i -wall
 186 and does not cross any i -wall before it exits the i -floor by crossing an $(i - 1)$ -wall. Thus the tail of π 's
 187 crossing sequence that starts from any point on the i -floor is an i -sequence.) For the sake of contradiction,
 188 assume that π also goes up in the k -garage. Then there must be a point where π goes up to some i -floor,
 189 and then goes monotonically down to the 1-floor. The crossing sequence of the corresponding subpath of
 190 π^\downarrow must be of the form $\sigma = (i - 1, \sigma_i)$, where σ_i is an i -sequence. If σ is a j -sequence for $j \neq i$, then
 191 σ_i must be a j -sequence, which is not possible by Lemma 3.1. If σ is an i -sequence, then σ_i must be an
 192 $(i - 1)$ -sequence, which again is not possible by Lemma 3.1. Finally note that σ must be a j -sequence for
 193 some j , since π^\downarrow is locally shortest. Thus, π only goes down in the k -garage, and the crossing sequence of
 194 π^\downarrow must be a k -sequence. \square

195 Lemma 3.3 directly implies that the SPM on the k -floor of the k -garage is exactly the k -SPM of P . Thus,
 196 as claimed before, the edges of the k -SPM consist of $(k - 1)$ -walls, k -walls, and k -windows. Furthermore,
 197 the k -predecessor of a point $p \in P$ must be (v, i) for some i between 1 and k .

198 4 The complexity of the k -SPM

199 To obtain an upper bound on the complexity of the k -SPM, we consider a sparser partitioning of P . We
 200 define the $(\leq k)$ -SPM of P as the partitioning induced by only the k -walls of P . Let $H_k(p)$ be the set of
 201 the k shortest homotopy classes to $p \in P$. We refer to $H_k(p)$ as the k -homotopy set of p . We would like
 202 to claim that the set $H_k(p)$ is constant within each cell of the $(\leq k)$ -SPM. Unfortunately we cannot claim
 203 this, since the homotopy classes of paths with different endpoints cannot be compared. To overcome this
 204 technicality, we define $H_k(p) \oplus \pi$ as the set of homotopy classes obtained by concatenating each path in
 205 $H_k(p)$ with π . If π is a path between p and p' , then we can directly compare $H_k(p) \oplus \pi$ and $H_k(p')$.

206 **Lemma 4.1.** *If p and p' lie in the same cell of the $(\leq k)$ -SPM, and π is a path between p and p' that does
 207 not cross a k -wall, then $H_k(p) \oplus \pi = H_k(p')$.*

208 To keep the notation simple, we simply compare $H_k(p)$ and $H_k(p')$ directly, in which case we really
 209 mean that we compare $H_k(p) \oplus \pi$ and $H_k(p')$, where π is the shortest path in P between p and p' . Note that
 210 π can cross a k -wall. We need the following property of the $(\leq k)$ -SPM.

211 **Lemma 4.2.** *The cells of the $(\leq k)$ -SPM are simply connected.*

212 We now count the number of k -walls, starting with the case $k = 1$. Let F_1, V_1 , and B_1 be the number
 213 of faces, triple points, and 1-walls of the (≤ 1) -SPM, respectively. It is easy to see that the (≤ 1) -SPM is
 214 simply connected, hence $F_1 = 1$. Now consider the graph G in which each node corresponds to either a
 215 hole (including the outer polygon) or a triple point, and there is an edge between two nodes if there is a
 216 1-wall between the corresponding holes/triple points. Since the (≤ 1) -SPM is simply connected, G must be

217 a tree. Hence $B_1 = h + V_1$. (The number of polygons bounding P is $h + 1$.) Furthermore note that the
 218 degree of a triple point in G is three, and every node in G has degree at least one. So, by double counting,
 219 $2B_1 \geq 3V_1 + h + 1$ or $V_1 \leq h - 1$. To summarize, $F_1 = 1$, $V_1 \leq h - 1$, and $B_1 = h + V_1$.

220 To bound the complexity of the $(\leq k)$ -SPM for $k > 1$, we relate its features to those of the $(\leq k - 1)$ -
 221 SPM. We consider an in-place transformation of the $(\leq k - 1)$ -SPM into the $(\leq k)$ -SPM. We use lower-case
 222 letters a, b, c, \dots to denote the members of $H_k(p)$. Each k -wall of the $(\leq k)$ -SPM locally separates regions
 223 of P that differ in exactly one of their k shortest path homotopy classes. Note that a k -wall e of the $(\leq k)$ -
 224 SPM is not present in the $(\leq k + 1)$ -SPM: if the k -homotopy sets belonging to the two sides of e are $H \cup a$
 225 and $H \cup b$, with $a \neq b$, then the $(k + 1)$ -homotopy set of points in the neighborhood of e is uniformly
 226 $H \cup \{a, b\}$.

227 The triple points of the $(\leq k)$ -SPM fall into two classes, which we call *new* and *old*
 228 (borrowing the terms from [16]). If the three k -homotopy sets in the vicinity of a triple
 229 point p are $H \cup a$, $H \cup b$, and $H \cup c$, with a, b , and c all distinct, then p is a new triple
 230 point. On the other hand, if the three k -homotopy sets are $H \cup \{a, b\}$, $H \cup \{b, c\}$, and
 231 $H \cup \{a, c\}$, with a, b , and c all distinct, then p is an old triple point. These names highlight
 232 the difference between what happens in the vicinity of p in the $(\leq k + 1)$ -SPM. If p is a
 233 new triple point in the $(\leq k)$ -SPM, then it becomes an old triple point in the $(\leq k + 1)$ -
 234 SPM. The three $(k + 1)$ -walls incident to p in the $(\leq k + 1)$ -SPM separate points with
 235 $(k + 1)$ -homotopy sets $(H \cup a) \cup b$ from $(H \cup a) \cup c$, $(H \cup b) \cup a$ from $(H \cup b) \cup c$,
 236 and $(H \cup c) \cup a$ from $(H \cup c) \cup b$. If p is an old triple point in the $(\leq k)$ -SPM, then the
 237 $(k + 1)$ -homotopy set of points in the neighborhood of e is uniformly $H \cup \{a, b, c\}$, and
 238 hence p is in the interior of a face of the $(\leq k + 1)$ -SPM. See Fig. 3.

239 To transform the $(\leq k)$ -SPM to the $(\leq k + 1)$ -SPM, we consider shortest distances to points in each face
 240 f of the $(\leq k)$ -SPM from its k -walls. The distances from a particular k -wall e are measured according to the
 241 homotopy class belonging to the face on the opposite side of e from f . More concretely, let $p \in f$ be a point
 242 close to e , and let p' be on the other side of f . Then the shortest paths measured from e use the homotopy
 243 class $h_f(e) = H_k(p') \setminus H_k(p)$. For every point $q \in f$, we identify the k -wall e whose homotopy class $h_f(e)$
 244 gives the shortest path to q . Hence $H_{k+1}(q) = H_k(q) \cup h_f(e)$, and this partitions the face f into subfaces,
 245 one for each k -wall e , separated by $(k + 1)$ -walls. To finish the construction of the $(\leq k + 1)$ -SPM, we erase
 246 the k -walls on the boundary of f (recall that their neighborhoods have uniform $(k + 1)$ -homotopy sets),
 247 delete any old triple points whose neighborhoods have uniform $(k + 1)$ -homotopy sets, and erase any newly
 248 added $(k + 1)$ -walls incident to deleted old triple points on the boundary of f . (These “walls” are actually
 249 just windows generated by the triple points; they separate regions with equal $(k + 1)$ -homotopy sets.)

250 If a face f of the $(\leq k)$ -SPM is bounded by B k -walls, it is initially partitioned into B subfaces. Every
 251 pair of subfaces incident to a common old triple point will be merged, so the final number of subfaces is
 252 $F' = B - W$, where W is the number of old triple points of the $(\leq k)$ -SPM on the boundary of f . Since
 253 f is simply connected by Lemma 4.2, and every subface corresponding to a k -wall e must be adjacent to e ,
 254 the dual graph of the subfaces inside f must be an outerplanar graph. The number of triple points V' added
 255 inside f (all of them new) corresponds to the number of (triangular) faces of this outerplanar graph, and
 256 hence $0 \leq V' \leq \max(F' - 2, 0)$. By Euler’s formula, the number of $(k + 1)$ -walls created inside f (duals
 257 to the edges of the outerplanar graph) is $B' = F' - 1 + V'$.

258 During the iterative construction of the $(\leq k)$ -SPM, we count the features at each step. The description
 259 above considers what happens within a single face of the $(\leq k)$ -SPM during the transformation to the $(\leq k + 1)$ -
 260 SPM. To account for what happens in all the faces simultaneously, we note that each i -wall is shared
 261 between two faces, and each triple point is shared between three faces. Let F_i and B_i be the number of faces
 262 and i walls in the $(\leq i)$ -SPM. To distinguish between new and old triple points, let V_i and W_i be the number
 263 of new and old triple points of the $(\leq i)$ -SPM, respectively. (Note that $W_1 = 0$.) If we count just the features
 264 added inside faces of $(\leq i)$ -SPM, using primed notation, we have

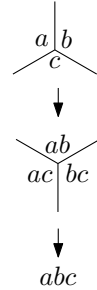


Figure 3: Life cycle of a triple point.

$$\begin{aligned}
F'_{i+1} &= 2B_i - 3W_i \\
B'_{i+1} &= 2B_i - 3W_i - F_i + V'_{i+1} \\
V'_{i+1} &\leq 2B_i - 3W_i - 2F_i \\
W'_{i+1} &= 0
\end{aligned}$$

265

266 Now let us take into account the deletion of previous i -walls and triple points. All the i -walls and old triple
267 points are deleted between one phase and the next. All new triple points turn into old ones. All subfaces
268 incident to an old triple point merge into one. Thus we obtain the following recurrence relations, whose
269 solution is given by Lemma 4.3.

$$\begin{aligned}
F_{i+1} &= F'_{i+1} - B_i + W_i = B_i - 2W_i & F_1 &= 1 \\
B_{i+1} &= B'_{i+1} = 2B_i - 3W_i - F_i + V_{i+1} & V_1 &\leq h - 1 \\
V_{i+1} &= V'_{i+1} \leq 2B_i - 3W_i - 2F_i & B_1 &= h + V_1 \\
W_{i+1} &= V_i & W_1 &= 0
\end{aligned}$$

270

271 **Lemma 4.3.** *The number of faces, walls, and triple points of the $(\leq k)$ -SPM is $O(k^2h)$.*

272 We now return to the complexity of the k -SPM. The number of k -walls and $(k-1)$ -walls can be bounded
273 by Lemma 4.3. Each k -wall consists of one or more hyperbolic arcs. Note that the number of hyperbolic
274 arcs for a single k -wall is exactly one more than the number of k -windows that end on the k -wall (and a
275 k -window can end on only one k -wall). Hence it is sufficient to count the number of k -windows. Each
276 k -window is an extension of the edge between a vertex v of P and its i -predecessor for $i \leq k$. Thus there
277 can be at most $O(kn)$ k -windows.

278 **Theorem 4.4.** *The k -SPM of a polygonal domain with n vertices and h holes has complexity $O(k^2h + kn)$.*

279 **Lower Bound.** The bound of Theorem 4.4 is in fact tight. Here we describe an example that has $\Omega(k^2h)$
280 k -walls and $\Omega(kn)$ k -windows. See Appendix B for the full details.

281 Consider the example in Fig. 4, which is constructed so that the
282 shortest paths from s to the vertices p_1 , p_2 , and p_3 have the same
283 length. Let q be the unique point equidistant from p_1, p_2, p_3 . Fur-
284 thermore, let π_{ij} ($i \in \{1, 2, 3\}$ and $1 \leq j \leq k$) be the j -path from
285 s to p_i , and let l_{ij} be the length of π_{ij} . If the obstacle ω_i is small
286 enough, then π_{ij} simply loops around ω_i zero or more times in a
287 clockwise or counterclockwise direction. Hence, for any $\epsilon > 0$, we
288 can ensure that $|l_{ik} - l_{i1}| \leq \epsilon$ for $i \in \{1, 2, 3\}$ by making the obsta-
289 cles ω_i small enough. Now define q_{abc} as the unique point such that
290 $|q_{abc} - p_1| + l_{1a} = |q_{abc} - p_2| + l_{2b} = |q_{abc} - p_3| + l_{3c}$. This point must
291 exist, since it is the vertex of an additively weighted Voronoi diagram
292 of p_1, p_2 , and p_3 . If ϵ is chosen small enough, then q_{abc} must lie in the
293 circle in Fig. 4 for $a, b, c \leq k$.

294 By construction there are three paths with equal length from s to q_{abc} , and there are exactly $a + b + c - 3$
295 shorter paths from s to q_{abc} . This means that q_{abc} is a triple point of the $(a + b + c - 2)$ -SPM. Thus, the
296 number of triple points of the k -SPM is exactly the number of triples (a, b, c) with $1 \leq a, b, c \leq k$ for which
297 $a + b + c - 2 = k$. It is easy to see that there are $\Omega(k^2)$ triples that satisfy these conditions. Note that
298 the gadget has $O(1)$ holes. By connecting $\Theta(h)$ copies of the basic gadget, we get a domain with h holes
299 and $\Omega(k^2h)$ k -SPM vertices. We can also replace p_3 in one copy by a convex chain of $n' = \Theta(n)$ vertices
300 v_1, \dots, v'_n , such that the line through v_i and v_{i+1} is very close to q for $1 \leq i < n'$. This way each vertex v_i
301 contributes k k -windows to the k -SPM (see Appendix B for details).

302 **Theorem 4.5.** *The k -SPM of a polygonal domain with n vertices and h holes can have $\Omega(k^2h)$ k -walls and
303 $\Omega(kn)$ k -windows.*

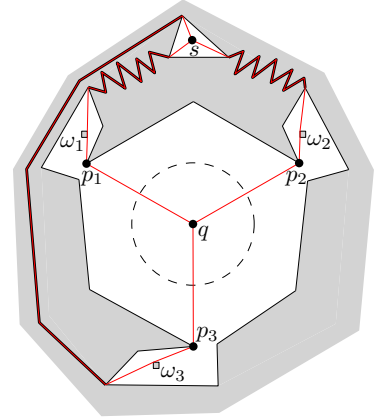


Figure 4: Lower bound gadget.

5 Computing the k -SPM

We now describe how to compute the k -SPM in $O((k^3h + k^2n) \log(kn))$ time. Inspired by the structure of the k -garage and Lemma 3.3, our algorithm iteratively computes the k -SPM for increasing values of k , starting from $k = 1$. Essentially we compute the SPM on the k -garage, one floor at a time. To compute the k -SPM at each iteration, we apply the “continuous Dijkstra” method, which Hershberger and Suri [13] used to compute the shortest path map among polygonal obstacles. We adopt most of the details of the Hershberger–Suri algorithm unchanged, but make a few modifications to support k -SPM computation.

The main idea of the continuous Dijkstra method is to simulate the progress of a wavefront that emerges from the source and expands through the free space with unit speed. If the wavefront reaches a point p at time t , then the shortest path distance between p and the source is t . At any time, the wavefront consists of circular arc *wavelets*, each expanding from a weighted obstacle vertex called a *generator* (see Fig. 5a). A generator γ is represented as a pair (v, w) , where v is an obstacle vertex and w is the shortest path distance from the source to v . For a generator $\gamma = (v, w)$ and a point p such that the segment \overline{vp} is contained in free space, the (weighted) distance between γ and p , denoted $d(p, \gamma)$, is defined as $w + |\overline{vp}|$; it is the length of the shortest path from the source to p that passes through v .

Points in the wavelet corresponding to a generator γ at time t satisfy the equation $d(p, \gamma) = t$. We say that a point p is *claimed* by γ if γ is the generator whose wavelet first reaches p ; this implies that the shortest path to p passes through v and has length $d(p, \gamma)$. The points where adjacent wavelets on the wavefront meet trace out the bisectors that form the walls and the windows of the shortest path map. Each bisector separates two cells of the shortest path map, each of which consists of points claimed by a particular generator. The bisector curve separating the regions claimed by two generators γ and γ' satisfies the equation $d(p, \gamma) = d(p, \gamma')$. Because $|vp| - |v'p| = w' - w$, the curve is a hyperbolic arc.

Using the continuous Dijkstra approach, the Hershberger–Suri algorithm computes shortest paths from a single source. It also works for shortest paths from multiple sources with delays. This is summarized in the following lemma, which was proved in [13].

Lemma 5.1 ([13]). *Given a set of polygonal obstacles with n vertices and a set of $O(n)$ sources with delays, one can compute the corresponding shortest path map in $O(n \log n)$ time.*

To compute the k -SPM, we apply the continuous Dijkstra framework on each floor of the k -garage. Imagine that we start a wavefront expansion from the source. When a wavelet collides with another wavelet during propagation (and thus forms a 1-wall), the portion of the wavelet that is claimed by the other wavelet continues to expand on the 2-floor (see Fig. 5b). Since this portion of the wavelet has passed through a 1-wall, it represents a set of 2-paths, by Lemma 3.3. Any bisectors formed by adjacent wavelets on the 2-floor belong to the 2-SPM. Similarly to the 1-floor, when two wavelets collide on the 2-floor, they form a 2-wall and continue to expand on the 3-floor. We continue to push the colliding wavelets up to higher floors

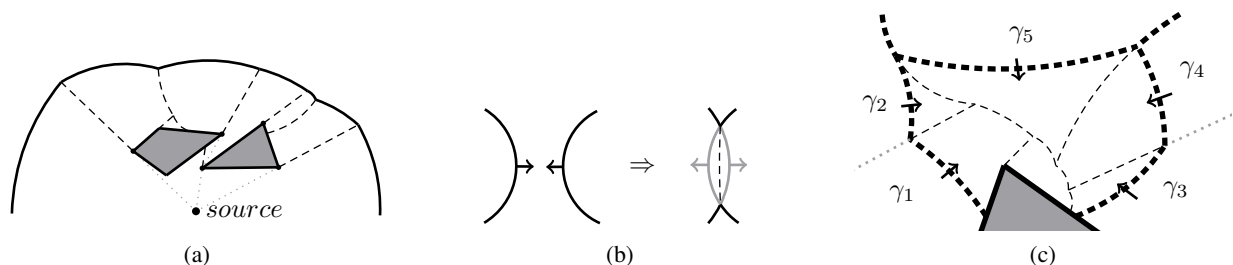


Figure 5: (a) An expanding wavefront. (b) Two colliding wavelets. After the collision, a wall is formed and both wavelets continue to grow on the next floor. (c) A shortest path map is computed by propagating outside generators into the region R .

338 until they reach the k -floor, which will correspond to the k -SPM.

339 Notice that the wavefront expansion on a single floor is not affected by the expansion on other floors,
340 with the exception of wavelet collisions on the previous floor. We now describe a method that exploits this
341 fact to compute the k -SPM once the $(k - 1)$ -SPM has been computed. Thus we can construct the k -SPM by
342 first running the Hershberger–Suri algorithm to compute the 1-SPM and then iteratively applying this step
343 to compute higher floor SPMs.

344 We compute the k -SPM from the $(k - 1)$ -SPM as follows. The boundaries of the $(k - 1)$ -SPM are
345 formed by $(k - 1)$ -windows, $(k - 1)$ -walls and $(k - 2)$ -walls. The $(k - 1)$ -windows and $(k - 2)$ -walls do
346 not appear in the k -SPM, so we simply remove them from the map. The $(k - 1)$ -walls remain in the map
347 and they subdivide the free space into simply connected regions (by Lemma 4.2). To complete the k -SPM,
348 in each such region we compute a special shortest path map whose walls and windows form the k -windows
349 and k -walls of the k -SPM.

350 The shortest path map computed in each region R is drawn with respect to multiple “restricted” sources
351 with delays, which are determined as follows. Consider a $(k - 1)$ -wall W bounding R in the $(k - 1)$ -SPM
352 and let $\gamma = (v, w)$ be the generator that claims the region outside R in the vicinity of W . (It is possible that
353 both sides of W are contained in R . In this case, our description applies to the generators claiming both
354 sides.) Note that W is formed by the collision of the wavelet of γ with another wavelet, and the wavelet of
355 γ is pushed up to the k -floor inside R . Conceptually, we want to continue expanding the wavelet of γ inside
356 R . To do this, we introduce γ as a source at v with delay w and impose the additional restriction that all
357 paths from γ to the interior of R pass through W .¹ In other words, we do not allow any paths from v that
358 do not pass through W . We create sources in this manner for each $(k - 1)$ -wall bounding R and draw the
359 shortest path map with respect to these sources (see Fig. 5c).

360 We can compute the shortest path map inside each region by running a single instance of the Hershberger–
361 Suri algorithm for delayed sources. Our restrictions necessitate some modifications, described in Ap-
362 pendix C, but with these modifications the algorithm computes the shortest path map in each region bounded
363 by $(k - 1)$ -walls. Since the paths used to compute the map in each region are k -paths by Lemma 3.3, the
364 walls and windows of the map form the k -walls and k -windows of the k -SPM. This completes the construc-
365 tion of the k -SPM.

366 **Theorem 5.2.** *Given a source point in a polygonal domain with n vertices and h holes, the corresponding*
367 *k -SPM can be computed in $O((k^3h + k^2n) \log(kn))$ time. If the total complexity of all i -SPMs for $1 \leq i \leq k$*
368 *is M , then the running time is $O(M \log(kn))$.*

369 6 Visibility-based algorithms

370 The k -SPM provides an efficient data structure for querying k -paths from a fixed source s . If we are simply
371 interested in the k -path between two fixed points s and t , then it may be inefficient to construct the k -SPM
372 for large values of k . In this section we present a simple visibility-based algorithm to compute the k -path
373 between s and t . For large k , this algorithm is faster than the k -SPM approach. Moreover, this algorithm is
374 relatively easy to implement and may therefore be of more practical interest.

375 We first compute the visibility graph (VG) of P in $O(n \log n + m)$ time [9, 21], where $m = O(n^2)$ is the
376 size of VG. We also include visibility edges to s and t . The graph contains every locally shortest path from
377 s to t and hence also the k -path to t . However, we cannot simply compute the k th shortest path in VG, since
378 different paths in the graph may be homotopic. We therefore modify VG so that locally shortest paths are in
379 one-to-one correspondence with paths in the modified graph—this ensures that different paths in the graph
380 belong to different homotopy classes by Lemma 2.1. (The same graph is defined in [11] and is called the

¹We also require that the subpath between v and W is a straight line.

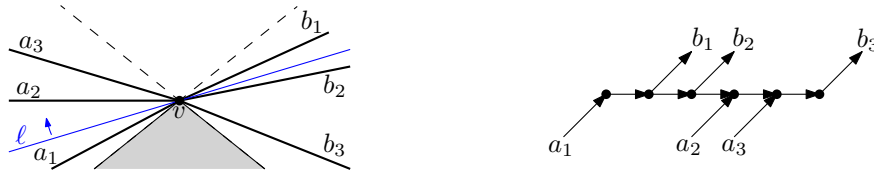


Figure 6: Vertex expansion for the taut graph.

381 *canonical graph*. Here we include its construction to argue the running time of computing this graph.) First,
 382 we make the graph directed by doubling each edge. Then we expand each vertex v as illustrated in Fig. 6:
 383 Draw the two lines supporting the two obstacle edges incident to v ; the lines partition the relevant visibility
 384 edges at v into two sets A and B (the visibility edges between the lines opposite the obstacle are irrelevant,
 385 because they cannot be used by shortest paths). Radially sweep a line through v , initially aligned with one
 386 of the obstacle edges, until it is aligned with the other obstacle edge. For each visibility edge e encountered,
 387 create a node with an incoming edge if $e \in A$, and an outgoing edge if $e \in B$. Connect all created nodes
 388 with a directed path. Also make a copy of this construction with all edges reversed. The expansion of v
 389 is connected with other expansions in the obvious way, as dictated by the visibility graph. Finally, remove
 390 edges directed toward s and away from t . The constructed graph—which we call the *taut graph* $\vec{G}(P)$ —has
 391 $O(m)$ vertices and $O(m)$ edges and can be built in $O(m)$ time. Note that, by construction, every path in
 392 $\vec{G}(P)$ must be locally shortest and every locally shortest path from s to t exists in $\vec{G}(P)$.

393 We can now use the algorithm by Eppstein [8] to compute the k th shortest path from s to t in $\vec{G}(P)$,
 394 which corresponds to the k -path from s to t in P .

395 **Theorem 6.1.** *The k -path between s and t in P can be computed in $O(m \log n + k)$ time, where m is the*
 396 *size of the visibility graph of P .*

397 Interestingly, this approach can be extended to compute the k th shortest *simple path* (*simple k -path*)
 398 between s and t in polynomial time. Here we define a *simple path* as a path that does not cross itself, although
 399 repeated vertices and segments are allowed. To compute simple k -paths, we adapt Yen’s algorithm [22] for
 400 computing simple k -paths in directed graphs (here “simple” means free of repeated nodes). The details are
 401 non-trivial and can be found in Appendix D. We obtain the following result.

402 **Theorem 6.2.** *The simple k -path between s and t can be computed in $O(k^2 m(m + kn) \log kn)$ time, where*
 403 *m is the number of edges of the visibility graph of P .*

404 7 Concluding remarks

405 We have introduced the k -SPM, a data structure that can efficiently answer k -path queries. We provided a
 406 tight bound for the complexity of the k -SPM, and presented an algorithm to compute the k -SPM efficiently.
 407 Our algorithm simultaneously computes all the i -SPMs for $i \leq k$. Whether there is a more direct algorithm
 408 to compute the k -SPM is an interesting open problem. We also provided a simple visibility-based algorithm
 409 to compute k -paths, which may be of practical interest, and is more efficient for large values of k . This latter
 410 approach can be extended to compute simple k -paths. Unfortunately, we do not know how to extend the
 411 k -SPM to simple k -paths. It seems that simple k -paths lack the useful property that a subpath of a simple
 412 k -path is a simple i -path for $i \leq k$. This makes finding a more efficient algorithm to compute simple k -paths
 413 a challenging open problem.

References

- [1] C. Bohler, P. Cheilaris, R. Klein, C.-H. Liu, E. Papadopoulou, and M. Zavershynskiy. On the complexity of higher order abstract Voronoi diagrams. In *ICALP (1)*, volume 7965 of *Lecture Notes in Computer Science*, pages 208–219. Springer, 2013.
- [2] S. Cabello, Y. Liu, A. Mantler, and J. Snoeyink. Testing homotopy for paths in the plane. *Discrete & Computational Geometry*, 31:61–81, 2004.
- [3] D. Z. Chen, J. Hershberger, and H. Wang. Computing shortest paths amid convex pseudodisks. *SIAM J. Comput.*, 42(3):1158–1184, 2013.
- [4] D. Z. Chen and H. Wang. L_1 shortest path queries among polygonal obstacles in the plane. In *STACS*, volume 20 of *LIPICs*, pages 293–304. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.
- [5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2001.
- [6] D. Eppstein. Finding the k smallest spanning trees. *BIT*, 32(2):237–248, 1992.
- [7] D. Eppstein. Tree-weighted neighbors and geometric k smallest spanning trees. *Int. J. Comput. Geometry Appl.*, 4(2):229–238, 1994.
- [8] D. Eppstein. Finding the k shortest paths. *SIAM J. Comput.*, 28(2):652–673, 1999.
- [9] S. K. Ghosh and D. M. Mount. An output-sensitive algorithm for computing visibility graphs. *SIAM Journal on Computing*, 20(5):888–910, 1991.
- [10] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics*. Addison-Wesley, Reading, Massachusetts, second edition, 1994.
- [11] D. Grigoriev and A. Slissenko. Polytime algorithm for the shortest path in a homotopy class amidst semi-algebraic obstacles in the plane. In *Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation*, ISSAC '98, pages 17–24. ACM, 1998.
- [12] J. Hershberger, M. Maxel, and S. Suri. Finding the k shortest simple paths: A new algorithm and its implementation. *ACM Trans. Algorithms*, 3(4):45, 2007.
- [13] J. Hershberger and S. Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM J. Comput.*, 28(6):2215–2256, 1999.
- [14] J. Hershberger, S. Suri, and H. Yıldız. A near-optimal algorithm for shortest paths among curved obstacles in the plane. In *Proceedings of the Twenty-Ninth Annual Symposium on Computational Geometry*, SoCG '13, pages 359–368. ACM, 2013.
- [15] E. L. Lawler. A procedure for computing the K best solutions to discrete optimization problems and its application to the shortest path problem. *Management Science*, 18:401–405, 1972.
- [16] D.-T. Lee. On k -nearest neighbor Voronoi diagrams in the plane. *IEEE Trans. Computers*, 31(6):478–487, 1982.
- [17] C.-H. Liu and D. T. Lee. Higher-order geodesic Voronoi diagrams in a polygonal domain with holes. In *SODA*, pages 1633–1645. SIAM, 2013.

- 450 [18] C.-H. Liu, E. Papadopoulou, and D. T. Lee. The k -nearest-neighbor Voronoi diagram revisited. *Algo-*
451 *rithmica*, 2014. To appear.
- 452 [19] J. S. B. Mitchell. Geometric shortest paths and network optimization. In J.-R. Sack and J. Urrutia,
453 editors, *Handbook of Computational Geometry*, pages 633–701. Elsevier Science B.V. North-Holland,
454 Amsterdam, 2000.
- 455 [20] E. Papadopoulou and M. Zavershynskiy. On higher order Voronoi diagrams of line segments. In
456 *ISAAC*, volume 7676 of *Lecture Notes in Computer Science*, pages 177–186. Springer, 2012.
- 457 [21] M. Pocchiola and G. Vegter. Topologically sweeping visibility complexes via pseudotriangulations.
458 *Discrete & Computational Geometry*, 16(4):419–453, 1996.
- 459 [22] J. Y. Yen. Finding the K shortest loopless paths in a network. *Management Science*, 17:712–716,
460 1971.

461 A Handling Degeneracies and Tie-Breaking

462 For simplicity of analysis we assumed that P satisfies the following conditions:

- 463 1. No three of the vertices of P , including the source s , are collinear.
- 464 2. There are at most three homotopically different i -paths to a single point in P , for $1 \leq i \leq k$. Equiva-
465 lently, no four i -walls meet at a single point.
- 466 3. There is a unique i -path to each vertex of P , for $1 \leq i \leq k$. Equivalently, no i -wall goes through a
467 vertex of P .

468 With these assumptions all walls are one-dimensional curves that meet only at triple points.

469 We now describe briefly how to adapt our analysis if these assumptions are false. If we are dealing with
470 first shortest paths only, then we can simply apply the standard technique of (symbolic) perturbation to the
471 input (i.e., perturb the positions of the vertices) so that the input is in general position and satisfies all of the
472 assumptions. However, for k -paths with $k \geq 2$, we need more than perturbation to enforce all assumptions.
473 In particular, Assumption 3 cannot be enforced by perturbation because it can be violated even when the
474 input is non-degenerate. For an example see Fig. 7: The 1-path from s to v is a straight line. There are two
475 2-paths from s to v , labeled π_1 and π_2 . The paths π_1 and π_2 are homotopically different; they pass through
476 v first and then loop around the same obstacle in different directions to return to v . Both π_1 and π_2 have the
477 same length, and thus v is on the 2-wall. This implies that v and all of the points to its left below ray r have
478 two distinct 2-paths and thus belong to a 2-wall; the 2-wall is thus a region, not a curve.

479 In order to avoid this issue, we introduce a tie-breaking mechanism between the paths so that all paths
480 to an obstacle vertex are strictly ordered by length and thus each obstacle vertex has a unique i -path. In
481 particular, suppose that π_1 and π_2 are two i -paths from s to a vertex v with the same length. We break the tie
482 between π_1 and π_2 by arbitrarily assuming that one of the two paths is infinitesimally shorter than the other.
483 Conceptually, this mechanism perturbs the i -wall by moving it slightly to one side. As a result, the i -wall
484 does not go through v and Assumption 3 is satisfied. Once the tie is broken, we assume that all paths that
485 are obtained by extending π_1 and π_2 with the same subpath preserve this order, maintaining consistency.²

486 By applying (symbolic) perturbation and enforcing a strict virtual order between the paths via tie-
487 breaking, we guarantee all our assumptions.

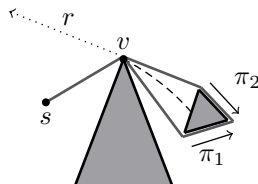


Figure 7: The equal-length paths π_1 and π_2 are both 2-paths to v . The 2-wall is shown with a dashed line.

²This still applies even if there are other tie-breakings in the extending subpath.

488 B Lower Bound

489 The construction of the lower bound example has already been explained in the main body of the paper.
 490 Recall that a point q_{abc} (for $1 \leq a, b, c \leq k$) is the unique point that satisfies $|q_{abc} - p_1| + l_{1a} = |q_{abc} -$
 491 $p_2| + l_{2b} = |q_{abc} - p_3| + l_{3c}$, and $l_{ik} - l_{i1} \leq \epsilon$, for any $1 \leq i \leq 3$. We first need to show that all points q_{abc}
 492 lie in the circle shown in Fig. 4, if ϵ is chosen small enough.

493 **Lemma B.1.** *If $\epsilon < |q - p_i|$ for $i \in \{1, 2, 3\}$, then $|q_{abc} - q| < \epsilon$, for $a, b, c \leq k$.*

Proof. Points p_1, p_2 , and p_3 are the vertices of an equilateral triangle, with q at its center. Define $L = |q - p_1|$.
 By assumption, $L > \epsilon$. Since $0 \leq l_{ij} - l_{i1} \leq \epsilon$, for $i \in \{1, 2, 3\}$ and any $1 \leq j \leq k$, and

$$|q_{abc} - p_1| + l_{1a} = |q_{abc} - p_2| + l_{2b} = |q_{abc} - p_3| + l_{3c},$$

494 we have $|q_{abc} - p_i| \leq |q_{abc} - p_j| + \epsilon$ for any $1 \leq i, j \leq 3$. The locus of points satisfying these inequalities
 495 is bounded by six hyperbolic arcs, as shown in Fig. 8. Each arc bulges toward the center, so putting q_{abc}
 496 at a vertex of the region maximizes $|q_{abc} - q|$. There are two classes of vertices of the region. They are
 497 defined by intersections of hyperbolae arranged in three pairs along the three angle bisectors at p_1, p_2 , and
 498 p_3 . By symmetry we can solve for points lying on an angle bisector satisfying the difference relations shown
 499 in Fig. 8. We apply the law of cosines to find minimum and maximum values of d , the distance from any
 500 of the p_i to the intersections of hyperbolae on the angle bisector at p_i . Solving for the lower bound on d
 501 (Fig. 8(left)), we have

$$\begin{aligned} d^2 + 3L^2 - 2d\sqrt{3}L \cos \frac{\pi}{6} &= (d + \epsilon)^2 \\ 3L^2 - 3dL &= 2d\epsilon + \epsilon^2 \\ d &= \frac{3L^2 - \epsilon^2}{3L + 2\epsilon} = L - \frac{2}{3}\epsilon + \frac{\epsilon^2}{3(3L + 2\epsilon)} \\ &> L - \frac{2}{3}\epsilon. \end{aligned}$$

502 Solving for the upper bound (Fig. 8(right)), we have

$$\begin{aligned} d^2 + 3L^2 - 2d\sqrt{3}L \cos \frac{\pi}{6} &= (d - \epsilon)^2 \\ 3L^2 - 3dL &= -2d\epsilon + \epsilon^2 \\ d &= \frac{3L^2 - \epsilon^2}{3L - 2\epsilon} = L + \frac{2}{3}\epsilon + \frac{\epsilon^2}{3(3L - 2\epsilon)} \\ &< L + \epsilon \end{aligned}$$

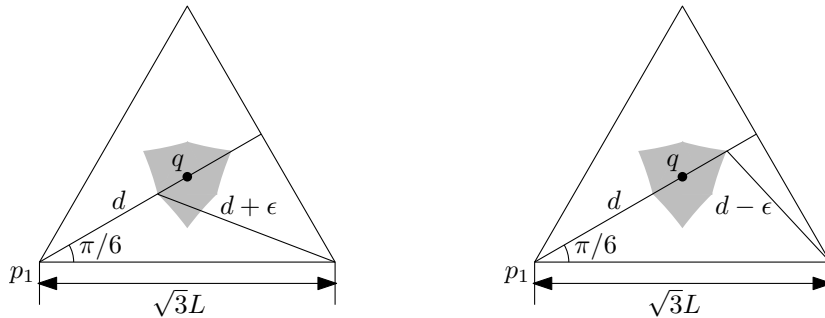


Figure 8: Extreme locations of q_{abc} .

503 since $L > \epsilon$. Because q_{abc} is constrained to lie in this hyperbolically bounded region, and the maximum
 504 distance from q to the boundary of the region is less than ϵ , we have $|q_{abc} - q| < \epsilon$. \square

505 As argued in the main body of the paper, a point q_{abc} is a triple point in the k -SPM if $a + b + c - 2 = k$,
 506 and there are $\Omega(k^2)$ such triples. The gadget in Fig. 4 has a constant number of holes. To obtain a lower
 507 bound of $\Omega(k^2h)$ k -walls, we need to connect h copies of the gadget together. We can do this as follows.
 508 First construct a thin polygon in the shape of a star graph with h leaves. Then connect a copy of the gadget to
 509 each of the leaves by opening up the gadget at the region that contains the source of the gadget (and scaling
 510 the gadgets so that they do not overlap). Finally we place the source s at the center of the star. This results
 511 in a polygonal domain with $\Theta(h)$ holes for which the k -SPM contains $\Omega(k^2h)$ triple points. Since triple
 512 points are adjacent to three k -walls, this directly implies that there must also be $\Omega(k^2h)$ k -walls.

513 In order to extend the construction to have $\Omega(kn)$ k -windows as well, we replace the vertex p_3 in one of
 514 the gadgets by a convex chain of $\Theta(n)$ vertices, as explained in the main body of the paper. We then obtain
 515 the following result.

516 **Theorem 4.5.** *The k -SPM of a polygonal domain with n vertices and h holes can have $\Omega(k^2h)$ k -walls and*
 517 *$\Omega(kn)$ k -windows.*

518 *Proof.* We use the construction described above. This means that the number of triple points is $\Omega(k^2h)$.
 519 However, the points q_{abc} might coincide for different values of a , b , and c . To argue that this is not the case,
 520 we rewrite the equations for q_{abc} as follows:

$$\begin{aligned} |q_{abc} - p_1| - |q_{abc} - p_2| &= l_{2b} - l_{1a} \\ |q_{abc} - p_1| - |q_{abc} - p_3| &= l_{3c} - l_{1a} \end{aligned}$$

521 A single one of these equations describes a hyperbolic arc. Also, if $l_{2b} - l_{1a}$ differs for different values of
 522 a and b , then the corresponding hyperbolic arcs are disjoint. The same holds for the second equation. That
 523 means that $q_{abc} = q_{a'b'c'}$ if and only if $l_{2b} - l_{1a} = l_{2b'} - l_{1a'}$ and $l_{3c} - l_{1a} = l_{3c'} - l_{1a'}$. We assume that all
 524 obstacles in the gadget have the same size, so that l_{ij} depends only on j . Hence, the location of q_{abc} depends
 525 only on the differences among a , b , and c . Finally note that for triples a, b, c with $a + b + c - 2 = k$, the
 526 differences among a , b , and c are unique. Thus, all triple points q_{abc} on the k -SPM are unique. (It does not
 527 matter that $q_{abc} = q_{(a+1)(b+1)(c+1)}$, since they are not part of the same map.)

528 Next we need to show that the k -SPM can have $\Omega(kn)$ k -windows. Since the number of vertices in the
 529 convex chain at p_3 is $\Theta(n)$, it is sufficient to show that each vertex in the chain (except the first) contributes
 530 k k -windows to the k -SPM. Let e_j be the edge formed by extending the edge between v_j and v_{j+1} toward q
 531 until it hits the boundary of P . We claim that, for every $i \leq k$, there must be a point $t \in e_j$ such that the path
 532 π consisting of the i -path to v_j followed by the segment $\overline{v_j t}$ is the k -path from s to t . In other words, t is on
 533 a k -window. If t is at v_j , then π is an i -path by definition. If t is the other endpoint of e_j and e_j is sufficiently
 534 close to q , then π must be an ℓ -path for $\ell > k$. Lemma 3.2 now implies that there must be a $t \in e_j$ such that
 535 π is the k -path from s to t . Thus, each vertex in the convex chain (except the first) contributes k k -windows,
 536 and the k -SPM has $\Omega(kn)$ k -windows. \square

537 C Implementing the Continuous Dijkstra Algorithm

538 The Hershberger–Suri algorithm [13] for finding shortest paths among polygonal obstacles simulates the
 539 wavefront expansion on a “conforming subdivision” of the free space. Each internal (free-space) edge e
 540 of this subdivision is contained in a set of cells whose union is called the “well-covering region” of e and
 541 denoted by $\mathcal{U}(e)$. (See Fig. 9a.) Briefly, the wavefront simulation computes the wavefront passing through
 542 each internal subdivision edge. The wavefront for a subdivision edge e is computed by propagating and
 543 combining the already computed wavefronts on the edges bounding $\mathcal{U}(e)$.³ Once the wavefronts for all edges
 544 have been computed, the shortest path map in each subdivision cell is constructed locally by computing a
 545 weighted Voronoi diagram for the generators that claim the boundaries of the cell or are inside the cell.
 546 These cell-wide maps are then easily combined into a global shortest path map.

547 As sketched in Section 5, we apply the Hershberger–Suri algorithm within regions of free space bounded
 548 by $(k - 1)$ -walls to compute the k -SPM. This requires two extensions to the algorithm:

549 First, in order to divide the free space into the separate regions of interest, we treat the $(k - 1)$ -walls as
 550 obstacles. The algorithm from [13] that builds the conforming subdivision of the free space assumes that the
 551 obstacles have straight boundaries, which may not hold for the $(k - 1)$ -walls. (Each $(k - 1)$ -wall consists
 552 of hyperbolic arcs.) We overcome this issue by using a slightly modified algorithm that creates conforming
 553 subdivisions for “curved” obstacles (within the same complexity bounds). This modified algorithm was
 554 described in [14], where it was used to compute shortest paths among curved obstacles; we omit its details.
 555 Note that even though we are using a subdivision that may have curved edges, we still apply the wavefront
 556 propagation algorithm for polygons on this subdivision, because each curved edge resides on a $(k - 1)$ -
 557 wall whose claiming generator is already known. Thus, the curved edges do not take part in the wavefront
 558 propagation or yield additional generators, as they do in [14].

559 Our second modification to the shortest path algorithm is the initialization of wavefront propagation
 560 in the subdivision. The original algorithm of Hershberger and Suri starts the propagation by passing the
 561 wavefront directly from each source point s to all edges e whose well covering region $\mathcal{U}(e)$ contains s . The
 562 sources in our setting are generators to be propagated into regions, each through its own $(k - 1)$ -wall, and
 563 thus we need a different way to initialize the wavefront. To meet our requirements, we initiate the wavefront
 564 propagation in the vicinity of the $(k - 1)$ -walls rather than the generators. In particular, the wavefront for a
 565 single generator γ is directly propagated to

- 566 (1) All edges e that bound a cell into which γ is to be propagated through a $(k - 1)$ -wall (see Fig. 9b).
- 567 (2) All edges e such that e contains an edge from (1) in its well-covering region $\mathcal{U}(e)$.

568 Note that propagating a generator’s wavefront to an edge does not mean that the wavefront claims the edge,
 569 because some or all of the wavefront may be eliminated by other propagated wavefronts when they are
 570 merged to compute the final wavefront.

³Well covering regions have special properties ensuring an acyclic propagation order between the edges of the subdivision.

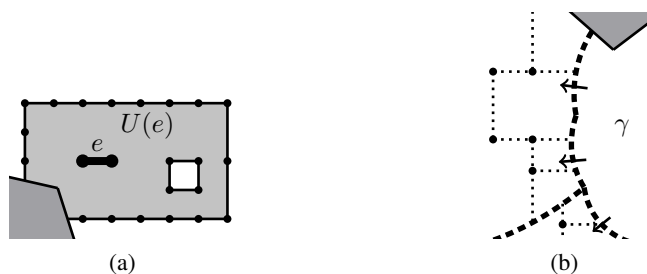


Figure 9: (a) The well-covering region $\mathcal{U}(e)$ (light gray) for an edge e in the conforming subdivision. (b) The set of subdivision edges in the vicinity of the $(k - 1)$ -walls through which a generator γ is propagated.

571 These two modifications enable the algorithm to compute the wavefront passing through every edge in
 572 the conforming subdivision, and hence to find the SPM in each region bounded by $(k - 1)$ -walls. The union
 573 of these shortest path maps is the k -SPM.

574 **Theorem 5.2.** *Given a source point in a polygonal domain with n vertices and h holes, the corresponding*
 575 *k -SPM can be computed in $O((k^3h + k^2n) \log(kn))$ time. If the total complexity of all i -SPMs for $1 \leq i \leq k$*
 576 *is M , then the running time is $O(M \log(kn))$.*

577 *Proof.* We construct the k -SPM iteratively for increasing values of k as described. We argue that at each
 578 iteration, the time spent to construct the k -SPM from a given $(k - 1)$ -SPM is $O((k^2h + kn) \log(kn))$. This
 579 implies the total time spent is $O((k^3h + k^2n) \log(kn))$.

580 Define the complexity of the i -SPM to be M_i . By Theorem 4.4, $M_{k-1} = O(k^2h + kn)$. We construct
 581 the k -SPM by running the modified Hershberger–Suri algorithm as described above. The algorithm is run
 582 on a set of obstacles with $O(M_{k-1})$ vertices (including the original obstacle vertices and the endpoints of
 583 the hyperbolic arcs forming the $(k - 1)$ -walls) with $O(M_{k-1})$ delayed sources (at most two sources per
 584 hyperbolic arc). By Lemma 5.1 (which applies also to our modified algorithm), the algorithm completes
 585 in $O(M_{k-1} \log M_{k-1}) = O(M_{k-1} \log(kn))$. The total complexity of all i -SPMs, for $1 \leq i \leq k$, is
 586 $\sum_{i=1}^k M_i = M$, and so the total running time is $O(M \log(kn))$. This completes the proof. \square

587 D Simple k -paths

588 Our definition of k -paths allows a path to be self-crossing. This may be undesirable for many applications.
 589 In this section we show how to compute the k th shortest *simple* path (*simple k -path*) in polynomial time,
 590 albeit slower than when we allow self-crossing paths. Here we define a *simple path* as a path that does
 591 not cross itself, although repeated vertices and segments are allowed. Note that we cannot use one of our
 592 previous methods to solve this problem: the simple 3-path may be a k -path for arbitrarily high k .

593 To compute the simple k -path between two fixed points s and t in P , we need to treat s and t as point
 594 obstacles (otherwise pulling a path taut may introduce self-crossings), but this trivializes the problem (the
 595 path may wind around s or t for free) unless special restrictions are added; therefore, for ease of presentation,
 596 we limit our attention to the case in which s and t are located on the boundaries of obstacles.

597 We again use the taut graph $\vec{G}(P)$ to reduce the problem to a graph problem. The taut graph ensures
 598 that every path between s and t is locally shortest, but it still allows crossings. To avoid crossings, we
 599 adapt Yen’s algorithm [22] for simple k -paths in directed graphs (in graphs, “simple” means free of repeated
 600 nodes). Yen’s algorithm first computes the shortest path, which must be simple; the same is true in our
 601 geometric setting. Next, the algorithm “expands” the shortest path π in the following way: It considers
 602 every possible prefix of π and chooses a next edge e that is different from the next edge in π . It then finds
 603 the shortest path starting from the endpoint of e that avoids the prefix including e ; this ensures that the
 604 resulting path is simple and different from π . Such paths are computed for every possible prefix and edge
 605 e , and stored in a heap; the shortest such path in this heap is the simple 2-path. The algorithm continues by
 606 expanding the simple 2-path and repeats this process, selecting the shortest of all the expanded paths in the
 607 heap, until the simple k -path is found.

608 Note that we cannot use Yen’s algorithm directly on $\vec{G}(P)$, since a simple path in $\vec{G}(P)$ is not neces-
 609 sarily simple in the geometric sense. To make this algorithm work in our setting, we need to make a small
 610 modification. Before we compute the shortest path with a given prefix π_p (including e), we add π_p as an
 611 obstacle to P , obtaining a new polygon P' . We then work with the taut graph $\vec{G}(P')$ of the new polygon
 612 (we separate each vertex of π_p and the corresponding obstacle vertex by an infinitesimal amount to allow
 613 paths that abut π_p but do not cross it). We need to show that the locally shortest path with a given prefix,
 614 i.e., the shortest path in $\vec{G}(P')$ starting after e , is simple. Clearly π_p is simple, and the suffix cannot cross
 615 π_p , but it is not clear that the suffix itself is simple, especially given the geometric nature of our paths. In
 616 order to prove this, we need some additional results.

617 Let π_{pq} denote the subpath of a path π between two points $p, q \in \pi$. We can apply a *shortcut* to a
 618 path π by replacing π_{pq} by the straight segment \overline{pq} , so long as \overline{pq} lies in free space. A shortcut is *valid* if
 619 it does not change the homotopy class of the path. We assume that a valid shortcut \overline{pq} does not cross π_{pq} ,
 620 for otherwise we can cut up the shortcut into multiple smaller shortcuts. A shortcut is valid if and only if
 621 the cycle formed by π_{pq} and \overline{pq} does not contain an obstacle. Note that a locally shortest path has no valid
 622 shortcuts. Furthermore, we can make a path locally shortest by repeatedly applying valid shortcuts until no
 623 more valid shortcuts exist.

624 A path π is *x -monotone* if every vertical line crosses π only once. Given a path π in P , we can obtain
 625 π' by repeatedly applying valid vertical shortcuts to π until no more valid vertical shortcuts exist. We call
 626 π' the *vertical reduction* of π . We can then find the smallest set S of vertices of P along π' such that the
 627 subpath of π' between two adjacent (along π') vertices in S is x -monotone. We call the vertices in S the
 628 *extremal vertices* of π' .

629 Now consider two homotopic paths π_1 and π_2 and their vertical reductions π'_1 and π'_2 . As was shown
 630 in [2, Lemmas 1 and 7], the set of extremal vertices of π'_1 and π'_2 must be the same. Hence the set of
 631 extremal vertices depends only on the homotopy class of π_1 , and we can also speak of the extremal vertices
 632 of π_1 . Finally note that a locally shortest path is its own vertical reduction. Thus the locally shortest path
 633 homotopic to a path π must pass through the extremal vertices of π .

634 We can now prove the following result.

635 **Lemma D.1.** *The shortest path in $\vec{G}(P')$ that starts with a fixed (simple) prefix π_p must be simple in P .*

636 *Proof.* For the sake of contradiction, assume that the shortest path π with fixed prefix π_p crosses itself at the
637 point $x \in \pi$ on edge e^* , where e^* is the first crossing edge after π_p . (See Fig. 10a.) Assume w.l.o.g. that the
638 bend at the vertex v before e^* makes a right turn. We can rotate the polygonal domain so that the direction
639 of e^* is infinitesimally clockwise from vertically up. As a result, v is an extremal vertex of π .

640 We will show that there is a locally shortest path π' that is shorter than π and also makes a right turn
641 at v . Since a locally shortest path must turn toward obstacles, it is sufficient to show that π' is shorter and
642 passes through v . We first construct a path π'' that is not longer than π , and then let π' be the locally shortest
643 path homotopic to π'' , which is shorter than π .

644 The path π (from s to t) crosses e^* either (i) from left to right (as in Fig. 10a) or (ii) from right to left
645 (as in Fig. 10c). Let π^* be the subpath of π between the two occurrences of the crossing. In case (i) π'' is
646 obtained by eliminating π^* . (See Fig. 10b.) In case (ii) π'' is obtained by reversing π^* . (See Fig. 10d.) In
647 case (i) π'' is clearly shorter than π . In case (ii) π'' has the same length as π , but note that π' must then be
648 shorter.

649 In both cases π'' makes a right turn at x . Now note that every vertical shortcut of π'' must also exist in
650 π . To see that, notice that the only shortcuts of π' we need to consider are those that span π^* in case (i) or
651 span or touch π^* in case (ii); any other shortcut also exists in π . A vertical shortcut that connects any point
652 before π^* to a point on or after π^* is blocked by v (i.e., the shortcut is not valid). A shortcut of π' within
653 π^* must also exist in π . A shortcut from a point on π^* to point after π^* (in case (ii)) is blocked by the first
654 extremal vertex after π^* . Since every vertical shortcut of π'' exists in π and π is locally shortest (i.e. has no
655 valid shortcuts), π'' must be its own vertical reduction. Thus, v is an extremal vertex of π'' , and π' must pass
656 through v .

657 Finally we need to show that π' is actually a path in $\vec{G}(P')$. Note that $\vec{G}(P')$ contains all locally shortest
658 paths in P that do not cross the fixed prefix π_p . So it is sufficient to show that π' does not cross π_p . Since π
659 did not cross π_p , the same is true for π'' . We can obtain π' from π'' by repeatedly applying valid shortcuts.
660 It is now sufficient to show that any valid shortcut \overline{pq} between $p, q \in \pi''$ cannot cross π_p . For the sake
661 of contradiction, assume that \overline{pq} crosses π_p . That means that some part of π_p must go inside the cycle C
662 formed by \overline{pq} and π''_{pq} . Note that s is outside C since we assumed that s belongs to an obstacle. If π_p ends
663 inside C , then there must be an obstacle inside C , which means that the shortcut was not valid. Otherwise,
664 π_p must also leave C . It cannot leave through π''_{pq} , since π'' did not cross π_p . If it leaves C through \overline{pq} , then
665 there must be a bend inside C . But this again means that there is an obstacle inside C , which contradicts the
666 validity of the shortcut.

667 Thus, the path π' contains π_p , it exists in $\vec{G}(P')$, and it is shorter than π . This contradicts the choice of
668 π . □

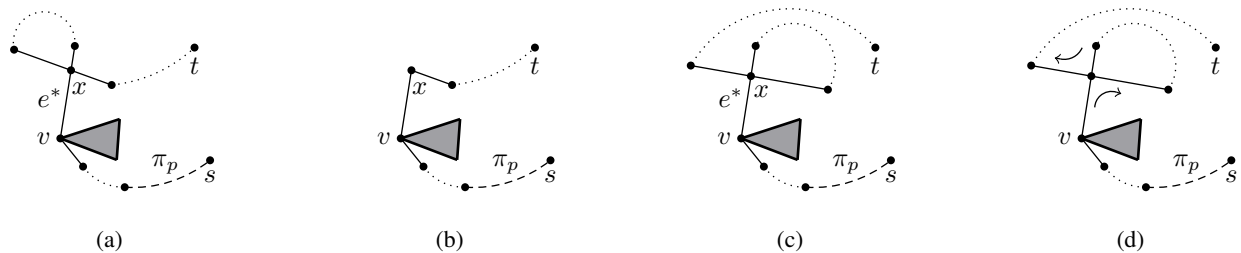


Figure 10: (a) π crosses e^* from left to right. (b) π'' is obtained by eliminating π^* . (c) π crosses e^* from right to left. (d) π'' is obtained by reversing π^* .

669 Thus, if we compute $\vec{G}(P')$ before every shortest path computation, every path obtained by our adapta-
670 tion of Yen's algorithm must be simple. We finally obtain the following result.

671 **Theorem 6.2.** *The simple k -path between s and t can be computed in $O(k^2m(m + kn) \log kn)$ time, where
672 m is the number of edges of the visibility graph of P .*

673 *Proof.* The simple k -path has at most kn edges since each vertex of P can be visited at most k times. This
674 means that a simple k -path can have at most $O(km)$ prefixes (including e). To compute $\vec{G}(P')$, note that
675 every visibility edge of P' is also a visibility edge of P , although some edges may occur multiple times in
676 P' (edges of P in the prefix are duplicated). Hence, to compute P' , we need to understand which visibility
677 edges of P still exist in P' . By considering the prefixes in order of increasing length (one edge at a time),
678 we only need to check which visibility edges of P cross the last edge of the prefix, which can be computed
679 in $O(m)$ time per prefix. Since the prefix can have at most kn edges, the visibility graph of P' can have at
680 most $O(m + kn)$ edges. We can then compute $\vec{G}(P')$ in $O(m + kn)$ time. Finally, we can use Dijkstra's
681 algorithm [5] to compute the shortest path in $\vec{G}(P')$ after the prefix in $O((m + kn) \log kn)$ time. To obtain
682 the simple k -path, we need to expand $k - 1$ paths. Each path may have $O(km)$ prefixes, and the shortest
683 path for each prefix can be computed in $O((m + kn) \log kn)$ time. Thus, we can compute the simple k -path
684 in $O(k^2m(m + kn) \log kn)$ time. \square

685 E Omitted Proofs

686 **Lemma 3.1.** *A sequence σ cannot be both a k -sequence and an ℓ -sequence if $k \neq \ell$.*

687 *Proof.* Assume without loss of generality that $\ell < k$. The definition of a k -sequence directly implies the
 688 following properties: (i) A k -sequence contains all integers in $\{1, \dots, k-1\}$, and (ii) every tail of a k -
 689 sequence is an i -sequence for some $i \leq k$.

690 Let k be the smallest number for which the lemma does not hold; clearly $k > 1$. If $\ell = 1$, then σ does not
 691 contain 1 while a k -sequence must contain 1 (property (i)); so assume $\ell > 1$. Since $k > \ell$, σ must contain
 692 ℓ (property (i) again). By definition, the tail of σ after one of the occurrences of ℓ is an ℓ -sequence. Since σ
 693 is also an ℓ -sequence, it must contain $(\ell-1)$ before ℓ , and the tail of σ after $(\ell-1)$ is an $(\ell-1)$ -sequence.
 694 In particular, the tail of σ after the occurrence of ℓ mentioned above must also be an i -sequence for some
 695 $i \leq \ell-1$ (property (ii)). But then the lemma does not hold for $k = \ell, \ell = i$, contradicting our choice of
 696 k . \square

697 **Lemma 4.1.** *If p and p' lie in the same cell of the $(\leq k)$ -SPM, and π is a path between p and p' that does
 698 not cross a k -wall, then $H_k(p) \oplus \pi = H_k(p')$.*

699 *Proof.* We reuse ideas from the proof of Lemma 3.2. Let us assume that distances have been scaled so that
 700 the length of π is 1. Define $p(x)$ ($0 \leq x \leq 1$) as the point on π such that the distance from p to $p(x)$ along
 701 π is x . Let $\gamma(x)$ be the subpath of π from p to $p(x)$. Furthermore, let π_i be the i -path to p , and let $\pi'_i(x)$ be
 702 the locally shortest path homotopic to the concatenation of π_i and $\gamma(x)$. The length of $\pi'_i(x)$ is denoted by
 703 $l_i(x)$ for $0 \leq x \leq 1$. Note that $l_i(0) < l_j(0)$ for $i < j$. If $l_i(x) \neq l_j(x)$ for all $0 \leq x \leq 1$ and $i \leq k < j$,
 704 then it is clear that $H_k(p) \oplus \pi = H_k(p')$. For the sake of contradiction, let x^* be the smallest x such that
 705 $l_i(x^*) = l_j(x^*)$ for some $i \leq k < j$. Let r be the number of graphs that pass below this intersection. If
 706 $r = k-1$, then $p(x^*)$ is on a k -wall, which is a contradiction. If $r < k-1$, then there must be an $m \leq k$
 707 such that $l_m(x^*) > l_j(x^*)$. But that means that $l_m(x) = l_j(x)$ for some $x < x^*$, contradicting the choice of
 708 x^* . Similarly, if $r > k-1$, then there must be an $m > k$ such that $l_m(x^*) < l_i(x^*)$. But that means that
 709 $l_m(x) = l_i(x)$ for some $x < x^*$, again contradicting the choice of x^* . \square

710 **Lemma 4.2.** *The cells of the $(\leq k)$ -SPM are simply connected.*

711 *Proof.* For the sake of contradiction, assume there is a cell of the $(\leq k)$ -SPM that is not simply connected.
 712 Let C be a cycle in this cell that is not contractible. If C contains only k -walls in its interior, then there
 713 must be a triple point with an angle larger than 180 degrees, which is not possible (a triple point is a Voronoi
 714 vertex of an additively weighted Voronoi diagram). Hence there must be an obstacle ω inside C . Let $p \in C$
 715 and let the largest winding number of any path in $H_k(p)$ with respect to ω be r . By Lemma 4.1 we have
 716 $H_k(p) \oplus C = H_k(p)$, where C is followed in counterclockwise direction. However, $H_k(p) \oplus C$ must contain
 717 a path with winding number $r+1$. This is a contradiction. \square

718 **Lemma 4.3.** *The number of faces, walls, and triple points of the $(\leq k)$ -SPM is $O(k^2h)$.*

Proof. We express the recurrence relations and the initial values using generating functions, which are
 formal power series with the sequence values as coefficients [10]. In general, for a sequence of values g_i ,
 the generating function $g(z)$ is

$$g(z) = \sum_{i \geq 0} g_i z^i.$$

719 For our sequences, we have

$$\begin{aligned}
 F(z) &= zB(z) - 2zW(z) + z \\
 B(z) &= z(2B(z) - 3W(z) - F(z)) + V(z) + zh \\
 V(z) &\leq z(2B(z) - 3W(z) - 2F(z)) + z(h-1) \\
 W(z) &= zV(z)
 \end{aligned}$$

720 Note that the constant term is zero, because we assume $F_0 = V_0 = B_0 = W_0 = 0$.

721 For convenience we will leave the “ z ” argument of the functions implicit during our manipulations. We
 722 can immediately eliminate the function $W = zV$:

$$\begin{aligned}
 F &= zB - 2z^2V + z \\
 B &= z(2B - 3zV - F) + V + zh \\
 V &\leq z(2B - 3zV - 2F) + z(h-1)
 \end{aligned}$$

723 Next we substitute $F = zB - 2z^2V + z$ into the last two relations to obtain

$$\begin{aligned}
 B &= z(2B - 3zV - (zB - 2z^2V + z)) + V + zh \\
 V &\leq z(2B - 3zV - 2(zB - 2z^2V + z)) + z(h-1)
 \end{aligned}$$

724 or, combining terms,

$$\begin{aligned}
 (1 - 2z + z^2)B &= (1 - 3z^2 + 2z^3)V + z(h - z) \\
 (1 + 3z^2 - 4z^3)V &\leq (2z - 2z^2)B - 2z^2 + z(h - 1)
 \end{aligned}$$

Substituting

$$B = V \frac{(1 - 3z^2 + 2z^3)}{(1 - z)^2} + \frac{z(h - z)}{(1 - z)^2}$$

725 into the inequality for V , we obtain

$$\begin{aligned}
 (1 + 3z^2 - 4z^3)V &\leq V \frac{2z(1 - z)(1 - 3z^2 + 2z^3)}{(1 - z)^2} \\
 &\quad + \frac{2z^2(1 - z)(h - z)}{(1 - z)^2} - 2z^2 + z(h - 1) \\
 &= 2z(1 + z - 2z^2)V + \frac{2z^2(h - z)}{1 - z} - 2z^2 + z(h - 1)
 \end{aligned}$$

Rearranging terms and simplifying, we obtain

$$V \leq \frac{z(1 + z)(h - 1)}{(1 - z)^3}.$$

726 Recall that $(1 - z)^{-3} = \sum_{i \geq 0} \binom{i+2}{2} z^i$, and hence

$$\begin{aligned}
 V &\leq \frac{z(1 + z)(h - 1)}{(1 - z)^3} \\
 &= \sum_{i \geq 1} z^i (h - 1) \left[\binom{i+1}{2} + \binom{i}{2} \right] \\
 &= \sum_{i \geq 0} z^i (h - 1) i^2.
 \end{aligned}$$

Returning from the domain of generating functions to our original recurrence relations, we have

$$V_i \leq (h - 1)i^2,$$

which immediately implies

$$W_i = V_{i-1} \leq (h - 1)(i - 1)^2.$$

Solving for $B(z)$ instead of $V(z)$ gives

$$B_i \leq (h - 1)(3i^2 - 3i + 2) + 1.$$

Finally, using $F_i = B_{i-1} - 2W_{i-1} \leq B_{i-1}$, we get

$$F_i \leq (h - 1)(3i^2 - 9i + 8) + 1.$$

727

□

728 **Theorem 4.4.** *The k -SPM of a polygonal domain with n vertices and h holes has complexity $O(k^2h + kn)$.*

729 *Proof.* We have already argued in the main body of the paper that the k -SPM has $O(k^2h)$ k -walls (and
 730 $(k - 1)$ -walls) and $O(kn)$ k -windows. For the sake of completeness, we finally need to argue that k -walls,
 731 $(k - 1)$ -walls, and k -windows cannot cross. As mentioned before, there is no k -path to a point that is on
 732 a $(k - 1)$ -wall, and hence $(k - 1)$ -walls and k -walls cannot cross. Furthermore, the k -path to a point on
 733 a k -window is unique and follows the k -window in some direction. As a result, k -windows behave like
 734 k -paths, and a k -window turns into a $(k + 1)$ -window as it crosses a k -wall. Thus, a k -window cannot cross
 735 a k -wall. Similarly, a k -window cannot cross a $(k - 1)$ -wall, since the window would be a $(k - 1)$ -window
 736 on the other side of the crossing. Hence, the complexity of the k -SPM is $O(k^2h + kn)$. □