31 SHORTEST PATHS AND NETWORKS

Joseph S.B. Mitchell

INTRODUCTION

Computing an optimal path in a geometric domain is a fundamental problem in computational geometry, with applications in robotics, geographic information systems (GIS), wire routing, etc.

A taxonomy of shortest-path problems arises from several parameters that define the problem:

1. Objective function: the length of the path may be measured according to the Euclidean metric, an L_p metric, the number of links, a combination of criteria, etc.

2. Constraints on the path: the path may have to get from s to t while visiting a specified set of points or regions along the way.

3. Input geometry: the map of the geometric domain also specifies constraints on the path, requiring it to avoid various types of obstacles.

4. Type of moving object: the object to be moved along the path may be a single point or may be a robot of some specified geometry.

5. Dimension of the problem: often the problem is in 2 or 3 dimensions, but higher dimensions arise in some applications.

6. Single shot vs. repetitive mode queries.

7. Static vs. dynamic environments: in some cases, obstacles may be inserted or deleted or may be moving in time.

8. Exact vs. approximate algorithms.

9. Known vs. unknown map: the on-line version of the problem requires that the moving robot sense and discover the shape of the environment along its way. The map may also be known with some degree of uncertainty, leading to stochastic models of path planning.

We survey various forms of the problem, primarily in two and three dimensions, for motion of a single point, since most results have focused on these cases. We discuss shortest paths in a simple polygon (Section 31.1), shortest paths among obstacles (Section 31.2), and other metrics for length (Section 31.3). We also survey other related geometric network optimization problems (Section 31.4). Higher dimensions are discussed in Section 31.5.

Preliminary version (July 17, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

GLOSSARY

Polygonal s-t path: A path from point s to point t consisting of a finite number of line segments (***edges***, or ***links***) joining a sequence of points (***vertices***).

Length of a path: A nonnegative number associated with a path, measuring its total cost according to some prescribed metric. Unless otherwise specified, the length will be the Euclidean length of the path.

Shortest/optimal/geodesic path: A path of minimum length among all paths that are feasible (satisfying all imposed constraints).

Shortest-path distance: The metric induced by a shortest-path problem. The shortest-path distance between s and t is the length of a shortest s-t path; in many geometric contexts, it is also referred to as ***geodesic distance***.

Locally shortest/optimal path: A path that cannot be improved by making a small change to it that preserves its ***combinatorial structure*** (e.g., the ordered sequence of triangles visited, for some triangulation of a polygonal domain P); also known as a ***taut-string*** path in the case of a shortest obstacle-avoiding path.

Simple polygon P of n vertices: A closed, simply-connected region whose boundary is a union of n (straight) line segments (edges), whose endpoints are the vertices of P.

Polygonal domain of n vertices and h holes: A closed, multiply-connected region whose boundary is a union of n line segments, forming $h+1$ closed (polygonal) cycles. A simple polygon is a polygonal domain with $h=0$.

Triangulation of a simple polygon P: A decomposition of P into triangles such that any two triangles intersect in either a common vertex, a common edge, or not at all. A triangulation of P can be computed in $O(n)$ time. See Section 29.2.

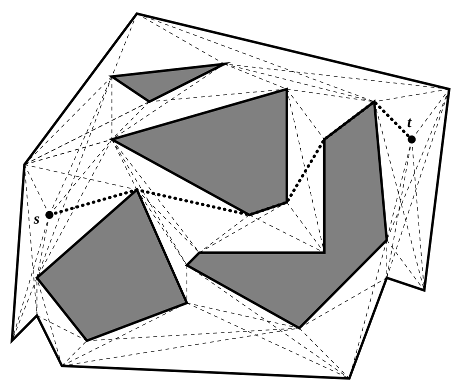

FIGURE 31.0.1
The visibility graph $VG(P)$. Edges of $VG(P)$ are of two types: (1) the heavy dark boundary edges of P, and (2) the edges that intersect the interior of P, shown with thin dashed segments. A shortest s-t path is highlighted.

Obstacle: A region of space whose interior is forbidden to paths. The complement of the set of obstacles is the ***free space***. If the free space is a polygonal domain P, the obstacles are the $h+1$ connected components (h ***holes***, plus the ***face at infinity***) of the complement of P.

Visibility graph $\mathrm{VG}(P)$: A graph whose nodes are the vertices of P and whose edges join pairs of nodes for which the corresponding segment lies inside P. See Chapter 33. An example is shown in Figure 31.0.1.

Preliminary version (July 17, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

Single-source query: A query that specifies a goal point t, and requests the length of a shortest path from a *fixed* source point s to t. The query may also require that a shortest s-t path be reported; in general, this can be done in additional time $O(k)$, where k is the number of edges in the output path. (Throughout this survey, when we report query times we omit the "$+O(k)$" that generally allows one to report a path after spending the initial query time to determine the length of a path.)

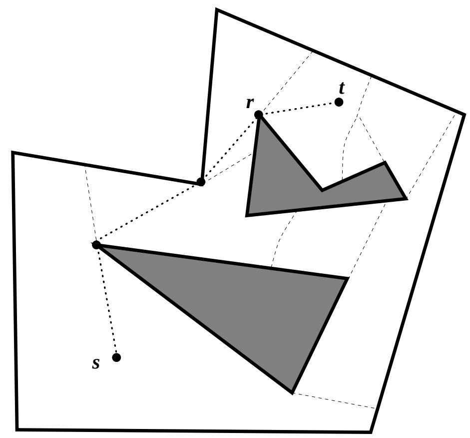

FIGURE 31.0.2
A shortest path map with respect to source point s within a polygonal domain. The dotted path indicates the shortest s-t path, which reaches t via the root r of its cell.

Shortest path map,* SPM(s)*: A decomposition of free space into regions (***cells***) according to the "combinatorial structure" of shortest paths from a fixed source point s to points in the regions. Specifically, for shortest paths in a polygonal domain, SPM(s) is a decomposition of P into cells such that for all points t interior to a cell, the sequence of obstacle vertices along a shortest s-t path is fixed. In particular, the *last* obstacle vertex along a shortest s-t path is the ***root*** of the cell containing t. Each cell is ***star-shaped*** with respect to its root, which lies on the boundary of the cell. See Figure 31.0.2, where the root of the cell containing t is labeled r. If SPM(s) is preprocessed for point location (see Chapter 38), then single-source queries can be answered efficiently by locating the query point t within the decomposition.

Two-point query: A query that specifies two points, s and t, and requests the length of a shortest path between them. It may also request that a path be reported.

Geodesic Voronoi diagram* (VD)*: A Voronoi diagram for a set of ***sites***, in which the underlying metric is the geodesic distance. See Chapters 27 and 29.

Geodesic center of P: A point within P that minimizes the maximum of the shortest-path lengths to any other point in P.

Geodesic diameter of P: The length of a longest shortest path between a pair of points $s, t \in P$; s and t are vertices for any longest s-t shortest path.

31.1 PATHS IN A SIMPLE POLYGON

The most basic geometric shortest-path problem is to find a shortest path inside a ***simple*** polygon P (having no holes), connecting two points, s and t. The complement of P serves as an obstacle through which the path is not allowed to travel. In this case, there is a unique taut-string path from s to t, since there is only one way to "thread" a string through a simply-connected region.

Algorithms for computing a shortest s-t path begin with a triangulation of P ($O(n)$ time; Section 29.2), whose dual graph is a tree. The ***sleeve*** is comprised of the triangles that correspond to the (unique) path in the dual that joins the triangle containing s to that containing t. By considering the effect of adding the triangles in order along the sleeve, it is not hard to obtain an $O(n)$ time algorithm for collapsing the sleeve into a shortest path. At a generic step of the algorithm, the sleeve has been collapsed to a structure called a ***funnel*** (with *base* ab and *root* r) consisting of the shortest path from s to a vertex r, and two (concave) shortest paths joining r to the endpoints of the segment ab that bounds the triangle abc processed next (see Figure 31.1.1). In adding triangle abc, we "split" the funnel in two according to the taut-string path from r to c, which will, in general, include a segment uc joining c to some (vertex) point of tangency u, along one of the two concave chains of the funnel. After the split, we keep that funnel (with base ac or bc) that contains the s-t taut-string path. The work needed to search for u can easily be charged off to those vertices that are discarded from further consideration. Thus, a shortest s-t path is found in time $O(n)$, which is worst-case optimal. See [GH89, GHL$^+$87, LP84] for further details about computing shortest paths in simple polygons.

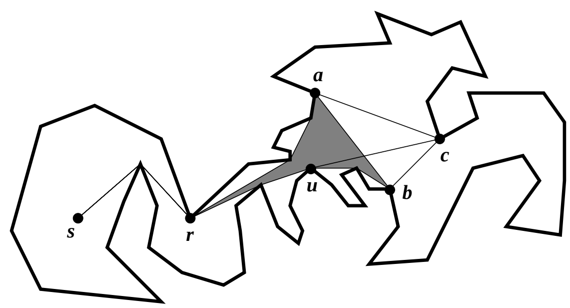

FIGURE 31.1.1
Splitting a funnel.

SHORTEST PATH MAPS

The shortest path map SPM(s) for a simple polygon has a particularly simple structure, since the boundaries between cells in the map are (line segment) chords of P obtained by extending appropriate edges of the visibility graph VG(P). It can be computed in time $O(n)$ by using somewhat more sophisticated data structures to do funnel splitting efficiently; in this case, we cannot discard one side of each split funnel. Single-source queries can be answered in $O(\log n)$ time, after storing the SPM(s) in an appropriate $O(n)$-size point location data structure (see Chapter 38). SPM(s) includes a tree of shortest paths from s to every vertex of P. For further details and proofs involving shortest path maps, see [GH89, GHL$^+$87, LP84, Mit91].

TWO-POINT QUERIES

A simple polygon can be preprocessed in time $O(n)$, into a data structure of size $O(n)$, to support shortest-path queries between any two points $s, t \in P$. In time

$O(\log n)$ the length of the shortest path can be reported, and in additional time $O(k)$, the shortest path can be reported, where k is the number of vertices in the output path [GH89, Her91].

DYNAMIC VERSION

In the dynamic version of the problem, one allows the polygon P to change with addition and deletion of edges and vertices. If the changes are always made in such a way that the set of all edges yields a ***connected planar subdivision*** of the plane into simple polygons (i.e., no "islands" are created), then one can maintain a data structure of size $O(n)$ that supports two-point query time of $O(\log^2 n)$ (plus $O(k)$ if the path is to be reported), and update time of $O(\log^2 n)$ for each addition/deletion of an edge/vertex [GT97].

TABLE 31.1.1 Shortest paths and geodesic distance in simple polygons.

PROBLEM VERSION	COMPLEXITY	NOTES	SOURCE
Shortest s-t path	$O(n)$	space $O(n)$	[LP84]
	$O(\frac{n^2}{m} + n \log m \log^4(\frac{n}{m}))$ exp.	space m	[Har15]
	$O(n^2/m)$ expected	space $m = O(\frac{n}{\log^2 n})$	[Har15]
Single-source query	$O(\log n)$ query	builds SPM(s)	[GHL+87]
	$O(n)$ preproc/space		
Two-point query	$O(\log n)$ query		[GH89]
	$O(n)$ preproc/space		
Two-polygon query	$O(\log k + \log n)$ query	between convex k-gons	[CT97]
	$O(n)$ space	in simple n-gon	
Dynamic two-point query	$O(\log^2 n)$ update/query		[GT97]
	$O(n)$ space		
Dynamic two-polygon	$O(\log k + \log^2 n)$ query	between convex k-gons	[CT97]
	$O(\log^2 n)$ update	in simple n-gon	
	$O(n)$ space		
Parallel algorithm	$O(\log n)$ time	in triangulated polygon	[Her95]
(CREW PRAM)	$O(n/\log n)$ processors	also builds SPM(s)	
Geodesic VD	$O((n+k)\log(n+k))$	k point sites	[PL98]
All nearest neighbors	$O(n)$	for set of vertices	[HS97]
Geodesic farthest-site VD	$O((n+k)\log(n+k))$ time	k point sites	[AFW93]
	$O(n+k)$ space		
Geodesic farthest-site VD	$O((n+k)\log\log n)$ time	k sites on ∂P	[OBA16]
All farthest neighbors	$O(n)$	for set of vertices	[HS97]
Geodesic diameter	$O(n)$		[HS97]
Geodesic center	$O(n)$		[ABB+16]

OTHER RESULTS

Several other problems studied with respect to geodesic distances induced by a simple polygon are summarized in Table 31.1.1. See also Table 29.4.1.

Shortest paths within simple polygons yield a wealth of structural information about the polygon. In particular, they have been used to give an output-sensitive algorithm for constructing the visibility graph of a simple polygon ([Her89]) and can be used for constructing a ***geodesic triangulation*** of a simple polygon, which allows for efficient ray-shooting (see [CEG+94]). They also form a crucial step in solving ***link distance*** problems, as we will discuss later. Simplification of a simple polygon to preserve geodesic distances leads to algorithms whose running times can be written in terms of the number of reflex vertices (having internal angle greater than π), instead of the total number, n, of vertices; see [AHK+14]. An output-sensitive algorithm for computing geodesic disks within a simple polygon is given in [BKL11], along with a clustering algorithm in the geodesic metric within a simple polygon.

OPEN PROBLEMS

1. Can one devise a simple $O(n)$ time algorithm for computing the shortest path between two points in a simple polygon, *without* resorting to a (complicated) linear-time triangulation algorithm?
2. What are the best possible query/update times possible for the dynamic versions of the shortest path problem?
3. Can the geodesic Voronoi diagram for k sites within P be computed in time $O(n + k \log k)$?

31.2 PATHS IN A POLYGONAL DOMAIN

While in a simple polygon there is a unique taut-string path between two points, in a general polygonal domain P, there can be an exponential number of taut-string simple paths between two points.

The homotopy type of a path can be expressed as a sequence (with repetitions) of triangles visited, for some triangulation of P. For any given homotopy type, expressed with N triangles, a shortest path of that type can be computed in $O(N)$ time [HS94]. Efficient algorithms for computing a set of homotopic shortest paths among obstacles, for many pairs of start and goal points, are known [Bes03b, EKL06]. One can also efficiently test, in time $O(n \log n)$, if two simple paths are of the same homotopy type in a polygonal domain; here, n is the total number of vertices of the input paths and the polygonal domain [CLMS02]. For more details on shortest curves of a specified topology, see Chapter 23.

SEARCHING THE VISIBILITY GRAPH

Without loss of generality, we can assume that s and t are vertices of P (since we can make "point" holes in P at s and t). It is easy to show that any locally optimal s-t path must lie on the visibility graph VG(P) (Figure 31.0.1). We can construct VG(P) in output-sensitive time $O(E_{\rm VG} + n \log n)$, where $E_{\rm VG}$ denotes the number of edges of VG(P) [GM91], even if we allow only $O(n)$ working space [PV96a]. In fact, a recent result [CW15b] shows that, after triangulation of P, VG(P) can be computed in time $O(E_{\mathrm{VG}} + n + h \log h)$. Given the graph VG($P$), whose edges are weighted by their Euclidean lengths, we can use Dijkstra's algorithm to construct a tree of shortest paths from s to all vertices of P, in time $O(E_{\mathrm{VG}} + n \log n)$ [FT87]. Thus, Euclidean shortest paths among obstacles in the plane can be computed in time $O(E_{\mathrm{VG}} + n \log n)$. This bound is worst-case quadratic in n, since $E_{\mathrm{VG}} \leq \binom{n}{2}$; note too that domains exist with $E_{\mathrm{VG}} = \Omega(n^2)$.

Given the tree of shortest paths from s, we can compute SPM(s) in time $O(n \log n)$, by computing an additively weight Voronoi diagram (see Chapter 27) of the vertices, with each vertex weighted by its distance from s.

CONTINUOUS DIJKSTRA METHOD

Instead of searching the visibility graph (which may have quadratic size), an alternative paradigm for shortest-path problems is to construct the (linear-size) shortest path map directly. The ***continuous Dijkstra*** method was developed for this purpose.

Building on the success of the method in solving (in nearly linear time) the shortest-path problem for the L_1 metric, Mitchell [Nit96] developed a version of the continuous Dijkstra method applicable to the Euclidean shortest-path problem, obtaining the first subquadratic ($O(n^{1.5+\varepsilon})$) time bound. Subsequently, this result was improved by Hershberger and Suri [HS99], who achieve a nearly optimal algorithm based also on the continuous Dijkstra method. They give an $O(n \log n)$ time and $O(n \log n)$ space algorithm, coming close to the lower bounds of $\Omega(n + h \log h)$ time and $O(n)$ space.

The continuous Dijkstra paradigm involves simulating the effect of a wavefront propagating out from the source point, s. The ***wavefront*** at distance δ from s is the set of all points of P that are at geodesic distance δ from s. It consists of a set of curve pieces, called ***wavelets***, which are arcs of circles centered at obstacle vertices that have already been reached. At certain critical "events," the structure of the wavefront changes due to one of the following possibilities:

(1) a wavelet disappears (due to the closure of a cell of the SPM);

(2) a wavelet collides with an obstacle vertex;

(3) a wavelet collides with another wavelet; or

(4) a wavelet collides with an obstacle edge at a point interior to that edge.

It is not difficult to see from the fact that SPM(s) has linear size, that the total number of such events is $O(n)$. The challenge in applying this propagation scheme is devising an efficient method to know *what* events are going to occur and in being able to *process* each event as it occurs (updating the combinatorial structure of the wavefront).

One approach, used in [Nit96], is to track a "pseudo-wavefront," which is allowed to run over itself, and to "clip" only when a wavelet collides with a vertex that has already been labeled due to an earlier event. Detection of when a wavelet collides with a vertex is accomplished with range-searching techniques. An alternative approach, used in [HS99], simplifies the problem by first decomposing the domain P using a *conforming subdivision*, which allows one to propagate an approximate wavefront on a cell-by-cell basis. A key property of a conforming subdivision is that any edge of length L of the subdivision has only a constant number of (constant-sized) cells within geodesic distance L.

APPROXIMATION ALGORITHMS

One can compute approximate Euclidean shortest paths using standard methods of discretizing the set of directions. Clarkson [Cla87] gives an algorithm that uses $O((n \log n)/\varepsilon)$ time to build a data structure of size $O(n/\varepsilon)$, after which a $(1+\varepsilon)$-approximate shortest path query can be answered in time $O(n \log n + n/\varepsilon)$. (These bounds rely also on an observation in [Che95].) Using a related approach, based on approximating Euclidean distance with fixed orientation distances, Mitchell [Mit92] computes a $(1+\varepsilon)$-approximate shortest path in time $O((n \log n)/\sqrt{\varepsilon})$ using $O(n/\sqrt{\varepsilon})$ space. Chen, Das, and Smid [CDM01] have shown an $\Omega(n \log n)$ lower bound, in the algebraic computation tree model, on the time required to compute any approximation to the shortest path.

TWO-POINT QUERIES

Two-point queries in a polygonal domain are much more challenging than in the case of simple polygons, where optimal algorithms are known. One natural approach (observed by Chen et al. [CDK01]) is to store the shortest path map, SPM(v), rooted at each vertex v; this requires $O(n^2)$ space. Then, for a query pair (s,t), we compute the set of k_s vertices visible to s and k_t vertices visible to t, in time $O(\min\{k_s, k_t\} \log n)$, using the visibility complex of Pocchiola and Vegter [PV96b]. Then, assuming that $k_s \leq k_t$, we simply locate t in each of the k_s SPM's rooted at the vertices visible from s. This permits two-point queries to be answered in time $O(\min\{k_s, k_t\} \log n)$, which is worst-case $\Omega(n \log n)$, making it no better than computing a shortest path from scratch, in the worst case.

Methods for exact two-point queries that are efficient in the worst case utilize an ***equivalence decomposition*** of the domain P, for which all points z within a cell of the decomposition have topologically equivalent shortest path maps. Given query points s and t, one locates s within the decomposition, and then uses the resulting SPM, along with a parametric point location data structure, to locate t within the SPM with respect to s. The complexity of the decomposition can be quite high; there can be $\Omega(n^4)$ topologically distinct shortest path maps with respect to points within P. Chiang and Mitchell [CM99] have utilized this approach to obtain various tradeoffs between space and query time; see Table 31.2.1. Unfortunately, the space bounds are all impractically high. Quadratic space is possible with a query time of $O(h \log n)$ [GMS08]. If the query points are restricted to lie on the boundary, ∂P, of the domain, then $O(\log n)$ query can be achieved with preprocessing time/space of $\tilde{O}(n^5)$ [BO12], where $\tilde{O}(\cdot)$ indicates that polylogarithmic factors are ignored.

More efficient methods allow one to approximately answer two-point queries. As observed in [Che95], the method of Clarkson [Cla87] can be used to construct a data structure of size $O(n^2 + n/\varepsilon)$ in $O(n^2 \log n + (n/\varepsilon) \log n)$ time, so that two-point $(1+\varepsilon)$-optimal queries can be answered in time $O((\log n)/\varepsilon)$, for any fixed $\varepsilon > 0$. Chen [Che95] was the first to obtain nearly *linear*-space data structures for approximate shortest path queries; these were obtained, though, at the cost of a higher approximation factor. He obtains a $(6+\varepsilon)$-approximation, using $O(n^{3/2}/\log^{1/2} n)$

time to build a data structure of size $O(n \log n)$, after which queries can be answered in time $O(\log n)$. Arikati et al. [ACC$^+$96] give a spectrum of results based on planar t-spanners (see Section 32.3), with tradeoffs among the approximation factor and the preprocessing time, storage space, and query time. One such result gives a $(3\sqrt{2}+\varepsilon)$-approximation in query time $O(\log n)$, after using $O(n^{3/2}/\log^{1/2} n)$ time to build a data structure of size $O(n \log n)$.

In the special case that the polygonal domain is "t-rounded," meaning that the shortest path distance between any two vertices is at most some constant t times the Euclidean distance between them, Gudmundsson et al. [GLNS08] show that in query time $O(\log n)$, one can give a $(1+\varepsilon)$-approximate answer to a two-point shortest path query while using only $O(n \log n)$ space and preprocessing time. Their result utilizes approximate distance oracles in t-spanner graphs, giving $O(1)$-time approximate distance queries between pairs of vertices; see Section 32.3.

OTHER RESULTS

The geodesic Voronoi diagram of k sites inside P can be constructed in time $O((n+k)\log(n+k))$, using the continuous Dijkstra method, simply starting with multiple source points. While the geodesic center/diameter problem has been solved in linear time for the case of simple polygons, in polygonal domains, the problem becomes much harder and the time bounds are much worse. For computing the geodesic diameter, [BKO13] show time bounds of $O(n^{7.73})$ or $O(n^7(\log n+h))$; one complicating fact is that the endpoints of a shortest path achieving the geodesic diameter can be interior to the domain. The geodesic center can be computed in time $O(n^{12+\varepsilon})$ [BKO15], which has been recently improved to $O(n^{11}\log n)$ [Wan16].

In the case of a planar domain with h *curved* obstacles, specified as a set of *splinegons* (polygons in which edges are replaced by convex curved arcs), having a total complexity of n, recent results have generalized the methods that were developed for polygonal domains. In particular, [HSY13] employ the continuous Dijkstra paradigm to obtain shortest paths in time $O(n \log n)$, under certain assumptions on the curved arcs, and $(1+\varepsilon)$-approximate shortest paths (and a shortest path map) in time $O(n \log n + n \log(1/\varepsilon))$, under mild assumptions on the curved arcs. Further, [CW15a] provide an algorithm with running time $O(n+k+h\log^{1+\varepsilon} h)$, where $k=O(h^2)$ is the number of free common tangents among curved obstacles (related to the size of the relevant visibility graph); while the running time is worst-case quadratic in the number of obstacles, it is *linear* in n. Shortest paths for a point moving among curved obstacles arises in optimal path planning for a robot (e.g., a circular disk) among polygonal obstacles.

In [AEK$^+$16], the notion of a shortest path map is generalized to consider geodesic length queries from a source point s to a query line segment within P, or to a query visibility polygon; this allows one to rapidly compute the geodesic distance from s to a point that sees the query point t. Another generalization of the notion of a shortest path map is the *kth shortest path map*, k-SPM, in which the domain is decomposed according to the combinatorial type of the kth shortest homotopically distinct (different "threading") path from source s to destination t. The combinatorial complexity of the k-SPM is $\Theta(k^2h+kn)$, and it can be computed in time $O((k^3h+k^2n)\log(kn))$ [EHP$^+$15].

Table 31.2.1 summarizes various results.

TABLE 31.2.1 Shortest paths among planar obstacles, in a polygonal domain.

PROBLEM	COMPLEXITY	NOTES	SOURCE
Shortest s-t path	$O(n \log n)$	$O(n \log n)$ space	[HS99]
	$O(n + h^2 \log n)$	$O(n)$ space	[KMM97]
	$O(n^{1.5+\varepsilon})$	$O(n)$ space	[Nit96]
Approx shortest s-t path	$O((n \log n)/\sqrt{\varepsilon})$	$O(n/\sqrt{\varepsilon})$ space	[Mit92]
SPM(s)/geodesic VD	$O(n \log n)$	$O(n \log n)$ space	[HS99]
	$O(n^{1.5+\varepsilon})$	$O(n)$ space	[Nit96]
Two-point query	$O(\log n)$ query $O(n^{11})$ preproc/space	exact	[CM99]
Two-point query	$O(\log^2 n)$ query $O(n^{10} \log n)$ preproc/space	exact	[CM99]
Two-point query	$O(n^{1-\delta} \log n)$ query $O(n^{5+10\delta+\varepsilon})$ preproc/space	exact $0 < \delta \leq 1$	[CM99]
Two-point query	$O(\log n + h)$ query $O(n^5)$ preproc/space	exact	[CM99]
Two-point query	$O(h \log n)$ query $O(n + h^5)$ preproc/space	exact	[CM99]
Two-point query	$O(h \log n)$ query $O(n^2)$ space	exact	[GMS08]
Two-point query	$O(\log n)$ query $\tilde{O}(n^5)$ space	exact queries on ∂P	[BO12]
Approx two-point query	$O(\log n + \rho)$ query $O(n^2/\rho)$ space $O(n^2/\rho)$ preproc	$(1+\varepsilon)$-approx any integer ρ $1 \leq \rho \leq \sqrt{n}$	[Che13]
Approx two-point query	$O(n)$ query $O(n)$ space $O(n \log n)$ preproc	$(1+\varepsilon)$-approx	[Che13]
Approx two-point query	$O(\log n)$ query $O(n^{3/2})$ space $O(n^{3/2})$ preproc	$(2+\varepsilon)$-approx	[Che13]
Approx two-point query	$O(\log n)$ query $O(n \log n)$ space $O(n \log n + q^{3/2}/\sqrt{\log q})$ preproc	$(3+\varepsilon)$-approx q is cover number $1 \leq q \leq n$	[Che13]
Approx two-point query	$O(\log n)$ query $O(n \log n)$ space $O(n \log n)$ preproc	$(1+\varepsilon)$-approx t-rounded domain	[GLNS08] [GLNS08]
Geodesic diameter	$O(n^{7.73})$ or $O(n^7(\log n + h))$		[BKO13]
Geodesic center	$O(n^{12+\varepsilon})$, $O(n^{11} \log n)$		[BKO15, Wan16]

OPEN PROBLEMS

1. Can the Euclidean shortest-path problem be solved in $O(n + h \log h)$ time and $O(n)$ space? (The L_1 shortest path problem can be solved in time $O(n + h \log h)$ in a triangulated domain [CW13].)
2. How efficiently, and using what size data structure, can one preprocess a polygonal domain for exact two-point queries? Can one obtain sublinear queries using a reasonable amount of space (say, subquadratic)?
3. How efficiently can one compute a geodesic center/diameter for a polygonal domain? Current polynomial-time bounds are likely far from optimal.

Preliminary version (July 17, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

31.3 OTHER METRICS FOR LENGTH

In the problems considered so far, the Euclidean metric has been used to measure the length of a path. We consider now several other possible objective functions for measuring path length. Tables 31.3.1 and 31.3.2 summarize results.

GLOSSARY

L_p metric: The L_p distance between $q = (q_x, q_y)$ and $r = (r_x, r_y)$ is given by $d_p(q,r) = [|q_x - r_x|^p + |q_y - r_y|^p]^{1/p}$. The L_p length of a polygonal path is the sum of the L_p lengths of each edge of the path. Special cases of the L_p metric include the L_1 metric (***Manhattan metric***) and the L_∞ metric ($d_\infty(q,r) = \max\{|q_x - r_x|, |q_y - r_y|\}$).

Rectilinear path: A polygonal path with each edge parallel to a coordinate axis; also known as an ***isothetic*** path.

C-oriented path: A polygonal path with each edge parallel to one of a finite set C of $c = |C|$ ***fixed orientations***.

Link distance: The minimum number of edges in a polygonal path from s to t within a polygonal domain P. If the paths are restricted to be rectilinear or C-oriented, then we obtain the ***rectilinear link distance*** or ***C-oriented link distance***.

Min-link s-t path: A polygonal path from s to t that achieves the link distance.

Weighted region problem: Given a piecewise-constant function $f : \mathbb{R}^2 \to \mathbb{R}$ that is defined by assigning a nonnegative ***weight*** to each face of a given triangulation in the plane. The ***weighted length*** of an s-t path π is the path integral, $\int_\pi f(x,y)d\sigma$, of the weight function along π. The ***weighted region metric*** associated with f defines the distance $d_f(s,t)$ to be the infimum over all s-t paths π of the weighted length of π. The ***weighted region problem*** (WRP) asks for an s-t path of minimum weighted length.

Anisotropic path problem: Compute a minimum-cost path, where the cost of motion is *direction-dependent*. Additionally, the cost of motion may depend on a weight function f, as in the WRP.

Sailor's problem: Compute a minimum-cost path, where the cost of motion is *direction-dependent*, and there is a cost L per turn (in a polygonal path).

Bounded curvature shortest-path problem: Compute a shortest obstacle-avoiding smooth (C^1) path joining point s, with prescribed velocity orientation, to point t, with prescribed velocity orientation, such that at each point of the path the radius of curvature is at least 1.

Maximum concealment path: A path within polygonal domain P that minimizes the length during which the robot is exposed to a given set of "enemy" observers. This problem is a special case of the weighted region problem, in which weights are 0 (for travel in concealed free space), 1 (for travel in exposed free space), or ∞ (for travel through obstacles).

Total turn for an s-t path: The sum of the absolute values of all turn angles for a polygonal s-t path.

Preliminary version (July 17, 2017). To appear in the Handbook of Discrete and Computational Geometry,
J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

Minimum-time path problem: Find a path to minimize the total time required to move from an initial position, at an initial velocity, to a goal position and velocity, subject to bounds on the allowed acceleration and velocity along the path. This problem is also known as the ***kinodynamic motion planning problem***.

LINK DISTANCE

In the min-link path problem, our goal is to minimize the number of links (and hence the number of turns) in a path connecting s and t. In many problems, the link distance provides a more natural measure of path complexity than the Euclidean length, as well as having applications to curve simplification.

In a simple polygon P, a min-link path can be computed in time $O(n)$; see also [LSD00] for a survey on link distance. In fact, in time $O(n)$ a ***window partition*** of P with respect to a point s can be computed, after which a min-link path from s to t can be reported in time proportional to the link distance. The algorithm, due to Suri [Sur90], computes the partition via "staged illumination," essentially a form of the continuous Dijkstra method under the link distance metric.

In a polygonal domain with holes, min-link paths can also be computed using a staged illumination method, but the algorithm is not simple: it relies on efficient methods for computing a single face in an arrangement of line segments (see Chapter 28). A min-link s-t path can be computed in time $O(E_{\mathrm{VG}}\alpha^2(n)\log n)$, where $\alpha(n)$ is the inverse Ackermann function; see Section 28.10. Computing link distance in a polygonal domain in significantly subquadratic time may not be possible; deciding if the link distance between two points is 3 is 3SUM-hard [MPS14]. This implies that obtaining a $4/3-\varepsilon$ factor approximation is also 3SUM-hard; in fact, obtaining an $O(1)$ additive approximation or a factor $(2-\varepsilon)$ approximation is also 3SUM-hard, even in rectilinear domains [MPS14]. If we consider C-oriented and rectilinear link distance, in which edges of the polygonal path must be from among a given set of C directions (axis-parallel in the rectilinear case), then significantly better time/space bounds are possible, and some of these apply also to combined metrics, in which there is a cost for length as well as links.

Refer to Table 31.3.1 for many related results on link distance, including rectilinear link distance, and on two-point queries.

L_1 METRIC

Instead of measuring path length according to the L_2 (Euclidean) metric, consider the problem of computing shortest paths in a polygonal domain P that are short according to the L_1 metric.

A method based on visibility graph principles allows one to construct a sparse graph (with $O(n\log n)$ nodes and edges) that is ***path-preserving*** in that it is guaranteed to contain a shortest path between any two vertices. Applying Dijkstra's algorithm then gives an $O(n\log^{1.5} n)$ time ($O(n\log n)$ space) algorithm for L_1-shortest paths.

A method based on the continuous Dijkstra paradigm allows the SPM(s) to be constructed in time $O(n\log n)$, using $O(n)$ space [Mit92], and, most recently in time $O(n+h\log h)$ for a triangulated domain with h holes [CW13]. The special

TABLE 31.3.1 Link distance shortest-path problems.

PROBLEM	COMPLEXITY	NOTES	SOURCE
Min-link path	$O(E_{\rm VG}\alpha^2(n)\log n)$	polygonal domain	[MRW92]
Min-link path	3SUM-hard	polygonal domain	[MPS14]
Min-link path	$O(\sqrt{h})$-approx	polygonal domain, h holes	[MPS14]
Min-link path	$O(n)$	simple polygon	[Sur86, Sur90]
Rectilinear link path	$O(n)$	rectilinear simple polygon	[Ber91, HS94]
Rectilinear link path	$O(n+h\log h)$ time	rectilinear domain	[MPSW15]
	$O(n)$ space, $O(\log n)$ query	h holes, triangulated	
C-oriented link path	$O(C^2 n\log n)$ time	polygonal domain	[MPS14]
	$O(Cn)$ space, $O(C\log n)$ query		
C-oriented link path	$O(Cn\log n)$ time	polygonal domain	[MPS14]
	$O(n)$ space, $O(\log n)$ query	2-approx	[MPS14]
Two-point link query	$O(\log n)$ query	simple polygon	[AMS95]
	$O(n^3)$ space, preproc		
Two-point rectilinear	$O(\log n)$ query	rectilinear simple polygon	[Sch96]
link query	$O(n\log n)$ preproc, $O(n)$ space	also is L_1-opt	
Shortest k-link path	$O(n^3k^3\log(Nk/\varepsilon^{1/k}))$	simple polygon	[MPA92]

property of the L_1 metric that is exploited in this algorithm is the fact that the wavefront in this case is piecewise-linear, with wavelets that are line segments of slope ± 1, so that the first vertex hit by a wavelet can be determined by rectangular range searching techniques (see Chapter 40).

Methods for finding L_1-shortest paths generalize to the case of C-oriented paths, in which $c = |C|$ fixed directions are given. Shortest C-oriented paths can be computed in time $O(cn\log n)$. Since the Euclidean metric is approximated to within accuracy $O(1/c^2)$ if we use c equally-spaced orientations, this results in an algorithm that computes, in time $O((n/\sqrt{\varepsilon})\log n)$, a path guaranteed to have length within a factor $(1+\varepsilon)$ of the Euclidean shortest path length.

WEIGHTED REGION METRIC

The weighted region problem (WRP) seeks an optimal s-t path according to the weighted region metric d_f induced by a given weight function f, often specified by a piecewise-constant (or piecewise linear) function on a given triangulated domain in two (or more) dimensions. This problem is a natural generalization of the shortest-path problem in a polygonal domain: consider a weight function that assigns weight 1 to P and weight ∞ (or a sufficiently large constant) to the obstacles (the complement of P). The WRP models the minimum-time path problem for a point robot moving in a terrain of varied types (e.g., grassland, brushland, blacktop, bodies of water, etc.), where each type of terrain has an assigned weight equal to the reciprocal of the maximum speed of traversal for the robot.

A standard formulation of the WRP assumes a piecewise-constant weight function f, specified by a triangulation in the plane having n vertices, with each face assigned an integer weight $\alpha \in \{0, 1, \ldots, W, +\infty\}$. (We can allow each edge of the triangulation to have a weight that is possibly distinct from that of the triangular facets on either side of it; in this way, linear features such as roads can be mod-

TABLE 31.3.2 Shortest paths in other metrics.

PROBLEM	COMPLEXITY	NOTES	SOURCE
L_1-shortest path, SPM(s)	$O(n \log n)$	polygonal domain	[Mit92, Mit89]
L_1 geodesic diameter	$O(n)$	simple polygon	[BKOW15]
L_1 geodesic center	$O(n)$	simple polygon	[BKOW15]
L_1 geodesic diameter	$O(n^2 + h^4)$	polygon with h holes	[BAE+17]
L_1 geodesic center	$O((n^4 + n^2h^4)\alpha(n))$	polygon with h holes	[BAE+17]
L_1 two-point query	$O(\log^2 n)$ query $O(n^2 \log n)$ space $O(n^2 \log^2 n)$ preproc	polygonal domain	[CKT00]
L_1 two-point query	$O(\log n + k)$ query $O(n + h^{2+\varepsilon})$ space, preproc	polygonal domain	[CIW14]
L_1 two-point query	$O(\log n)$ query $O(n^2)$ space, preproc	rectangle obstacles	[AC91, AC93] [EM94]
L_1 two-point query	$O(\sqrt{n})$ query $O(n^{1.5})$ space, preproc	rectangle obstacles	[EM94]
L_1 two-point query	$O(\log n)$ query $O(n \log n)$ space $O(n \log^2 n)$ preproc	3-approx rectangle obstacles	[CK96]
C-oriented shortest path two-point query	$O(cn \log n)$ $O(c^2 \log^2 n)$ query	 $O(c^2 n^2 \log^2 n)$ preproc	[Mit92] [CDK01]
Weighted region problem	$O(ES)$, or $O(n^8 L)$ $L = O(\log \frac{nNW}{\varepsilon})$	(1+ε)-approx	[MP91]
Weighted region problem	$O(\frac{kn+k^4 \log(k/\varepsilon)}{\varepsilon} \log^2 \frac{Wn}{\varepsilon})$ parameter $3 \leq k \leq n$	(1+ε)-approx weights $[1, W] \cup \{\infty\}$	[CJV15] [CJV15]
Weighted region problem	$O(\frac{n}{\sqrt{\varepsilon}} \log \frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$	(1+ε)-approx geometric parameters	[AMS05]
Weighted region problem	$O((W \log W) \frac{n^3}{\varepsilon} \log \frac{Wn}{\varepsilon})$	(1+ε)-approx indep of geometry also anisotropic cost	[CNVW08]
Weighted region problem	$O(n^2)$	weights 0, 1, ∞	[GMMN90]
L_1 weighted region prob	$O(n \log^{3/2} n)$ preproc $O(\log n)$ query $O(n \log n)$ space	rectilinear regions single-source queries	[CKT00]
L_1 WRP, two-point query	$O(\log^2 n)$ query $O(n^2 \log^2 n)$ space, preproc	rectilinear regions	[CKT00]
L_1 WRP, two-point query	$O(\log n)$ query $O(n^{2+\varepsilon})$ space, preproc	rectilinear regions	[CIW14]
Bounded curvature path	$O(n^4 \log n)$	moderate obstacles	[BL96]
Bounded curvature path	$O(n^2 \log n)$	within convex polygon	[ABL+02]
Anisotropic path problem	$O(\frac{\rho^2 n^3}{\varepsilon^2} (\log \frac{\rho n}{\varepsilon})^2)$ $O(\frac{\rho^2 n^3}{\varepsilon^2} \log \frac{\rho n}{\varepsilon})$ space	parameter $\rho \geq 1$ $O(\log \frac{\rho n}{\varepsilon})$ query	[CNVW10]
Sailor's problem ($L = 0$)	$O(n^2)$	polygonal domain	[Sel95]
Sailor's problem ($L > 0$)	poly(n, ε)	ε-approx	[Sel95]
Max concealment v viewpoints	$O(v^2(v + n)^2)$ $O(v^4 n^4)$	simple polygon polygonal domain	[GMMN90] [GMMN90]
Min total turn	$O(E_{\mathrm{VG}} \log n)$	polygonal domain	[AMP91]

Preliminary version (July 17, 2017). To appear in the Handbook of Discrete and Computational Geometry,
J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

eled.) The local optimality condition, which follows from basic calculus, is that an optimal path must be polygonal (for piecewise-constant f), bending according to *Snell's Law of Refraction* when crossing a region boundary, and, if it utilizes a portion of an edge of the triangulation, it must turn to enter/leave the edge at the "critical angle" of refraction determined by the weight of the edge and the weight of the adjacent face.

Exact solution of the general WRP in the plane seems to be very difficult for algebraic reasons; the problem cannot be solved in the "algebraic computational model over the rational numbers" [DCG$^+$14]. Algorithms are, therefore, focused on approximation and generally fall into two categories: (1) those based on the continuous Dijkstra paradigm, propagating "intervals of optimality," which partition edges according to the combinatorial type of paths that optimally reach the edge from either side, while utilizing the local optimality condition during a breadth-first propagation; and, (2) those based on placing discrete sample points along edges or interior to faces, and searching for a shortest path in a corresponding network of edges interconnecting the sample points.

An algorithm of type (1) can be viewed as "exact" in the sense that it would give exactly optimal paths if the underlying predicates could be performed exactly. The predicates require determining a refraction path through a specified edge sequence in order to reach a specified vertex, or to find a "bisector" point b along an edge where the refraction paths to b through two specified edge sequences have the same lengths. The first provable result for the WRP was of type (1) [MP91]; it computes a $(1+\varepsilon)$-approximate optimal path, for any fixed $\varepsilon > 0$, in time $O(E \cdot S)$, where E is the number of "events" in the continuous Dijkstra algorithm, and S is the complexity of performing a numerical search to solve approximately the refraction-path predicates. It is shown that $O(n^4)$ is an upper bound on E, and that this bound is tight in the worst case, since there are instances in which the total number of intervals of optimality is $\Omega(n^4)$. The numerical search can be accomplished in time $S = O(k^2 \log(nNW/\varepsilon))$, on k-edge sequences, and it is shown that the maximum length k of an edge sequence is $O(n^2)$; thus, the overall time bound is at most $O(n^8 \log(nNW/\varepsilon)$ [MP91]. (A new variant utilizing a "discretized wavefront" approach, saving a factor of $O(n^3)$, has recently been announced [IK15].)

Algorithms of type (2) carefully place Steiner points on the edges (or, possibly interior to faces) of the input subdivision. Using a logarithmic discretization (as in [Pap85]), with care in how Steiner points are placed near vertices, provable approximation guarantees are obtained. A time bound with near-linear dependence on n, specifically $O((n/\sqrt{\varepsilon}) \log(n/\varepsilon) \log(1/\varepsilon))$, is possible [AMS05]; however, this bound has a hidden constant (in the big-O) that depends on the geometry (smallest angles) of the triangulation in such a way that a single tiny angle can cause the bounds to go to infinity. Using a different discretization method, [CNVW08] give an algorithm with running time $O(((W \log W)/\varepsilon)n^3 \log(Wn/\varepsilon))$, independent of the angles of the triangulation; in fact, this algorithm solves also the *anisotropic* generalization, in which a (possibly asymmetric) convex distance function, specified for each face of the triangulation, is utilized. Single-source approximate optimal path queries can be answered efficiently, even in anisotropic weighted regions, using a type of shortest path map data structure [CNVW10]. In [CJV15], new bounds that depend on k, the smallest integer so that the sum of the k smallest angles in the triangular faces is at least π; specifically, they obtain a $(1+\varepsilon)$-approximation in time $O((kn + k^4 \log(k/\varepsilon))(1/\varepsilon) \log^2(Wn/\varepsilon))$. If the triangulation is "nice," one expects k to be a small constant, and the running time is near-linear in n. It should be noted that in algorithms of type (2) the dependence on $1/\varepsilon$ is polynomial (versus logarithmic in algorithms of type (1)), and the methods are not "exact" in that no matter how accurately predicates are evaluated, the algorithms do not guarantee to find the correct combinatorial type of an optimal path.

There are special cases of the weighted region problem that admit faster and simpler algorithms. For example, if the weighted subdivision is rectilinear, and path length is measured according to weighted L_1 length, then efficient algorithms for single-source and two-point queries can be based on searching a path-preserving graph [CKT00]. Similarly, if the region weights are restricted to $\{0, 1, \infty\}$ (while edges may have arbitrary nonnegative weights), then an $O(n^2)$ time algorithm can be based on a path-preserving graph similar to a visibility graph [GMMN90]. This also leads to an efficient method for performing ***lexicographic*** optimization, in which one prioritizes various types of regions according to which is most important for path length minimization.

The anisotropic path problem includes a generalization to the case in which each face of a given polygonal subdivision may have a different cost function, and the cost may depend on the direction of movement. In the model of Cheng et al. [CNVW10], distance in face f is measured according to a (possibly asymmetric) convex distance function whose unit disk D_f is contained within a (concentric) Euclidean unit disk and contains a (concentric) Euclidean disk of radius $1/\rho$, for a real parameter $\rho \geq 1$ that quantifies the degree of directionality of asymmetry in the cost function. Their algorithm uses time $O(\frac{\rho^2 n^3}{\varepsilon^2}(\log \frac{\rho n}{\varepsilon})^2)$ to compute a data structure of size $O(\frac{\rho^2 n^3}{\varepsilon^2} \log \frac{\rho n}{\varepsilon})$ that enables $(1+\varepsilon)$-approximate optimal path queries in time $O(\log \frac{\rho n}{\varepsilon})$.

The weighted region model applies also to the problem of computing "high-quality" paths among obstacles. In particular, it is natural to consider the cost of motion that is very close to an obstacle to be more costly than motion that has high clearance from obstacles. Letting the cost of motion at clearance δ from the nearest obstacle be weighted by $1/\delta$, and the cost of a path to be the weighted length (path integral of the cost function), one can obtain a fully polynomial time approximation scheme to compute a path with cost at most $(1+\varepsilon)$ times optimal in time $O((n^2/\varepsilon^2)\log(n/\varepsilon))$ in a planar polygonal environment with n vertices [AFS16].

MINIMUM-TIME PATHS

The *kinodynamic motion planning problem* (also known as the *minimum-time path problem*) is a nonholonomic motion planning problem in which the objective is to compute a *trajectory* (a time-parameterized path, $(x(t), y(t))$) within a domain P that minimizes the total time necessary to move from an initial configuration (position and initial velocity) to a goal configuration (position and velocity), subject to bounds on the allowed acceleration and velocity along the path. (Algorithmic motion planning is discussed in detail in Chapter 50.) The minimum-time path problem is a difficult optimal control problem; optimal paths will be complicated curves given by solutions to differential equations.

The bounds on acceleration and velocity are most often given by upper bounds on the L_∞ norm (the "decoupled case") or the L_2 norm (the "coupled case").

If there is an upper bound on the L_∞ norm of the velocity and acceleration vectors, one can obtain an *exact*, exponential-time, polynomial-space algorithm, based

Preliminary version (July 17, 2017). To appear in the Handbook of Discrete and Computational Geometry,
J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

on characterizing a set of "canonical solutions" (related to "bang-bang" controls) that are guaranteed to include an optimal solution path. This leads to an expression in the first-order theory of the reals, which can be solved exactly; see Chapter 37. However, it remains an open question whether or not a polynomial-time algorithm exists.

Donald et al. [DXCR93, DX95, RW00] developed approximation methods, including a polynomial-time algorithm that produces a trajectory requiring time at most $(1 + \varepsilon)$ times optimal, for the decoupled case. Their approach is to discretize (uniformly) the four-dimensional phase space that represents position and velocity, with special care to ensure that the size of the grid is bounded by a polynomial in $1/\varepsilon$ and n. Approximation algorithms for the coupled case are also known [DX95, RT94].

Optimal paths for a "car-like" robot (a "Dubins car") leads to the closely related shortest-path problem, the ***bounded curvature shortest-path problem***, in which we require that no point of the path have a radius of curvature less than 1. For this problem, $(1+\varepsilon)$-approximation algorithms are known, with polynomial ($O(\frac{n^2}{\varepsilon^2}\log n)$) running time [AW01]. The problem is known to be NP-hard in a polygonal domain [KKP11, RW98]; further, deciding if there exists a simple curvature-constrained path is NP-hard [KKP11]. For the special case in which the obstacles are "moderate" (have differentiable boundary curves, with radius of curvature at least 1), both an approximation algorithm and an exact $O(n^4 \log n)$ algorithm have been found [BL96]. Within a convex polygon, one can determine if a curvature constrained path exists and, if so, compute one in time $O(n^2 \log n)$ [ABL$^+$02]. See also [KO13] for an analysis of how bounded curvature impacts path length.

MULTIPLE CRITERION OPTIMAL PATHS

The standard shortest-path problem asks for paths that minimize some *one* objective (length) function. Frequently, however, an application requires us to find paths to minimize *two or more* objectives; the resulting problem is a ***bicriterion*** (or ***multi-criterion***) shortest-path problem. A path is called ***efficient*** or ***Pareto optimal*** if no other path has a better value for one criterion without having a worse value for the other criterion.

Multi-criterion optimization problems tend to be hard. Even the bicriterion path problem in a graph is NP-hard: Does there exist a path from s to t whose length is less than L and whose weight is less than W? Pseudo-polynomial-time algorithms are known, and many heuristics have been devised.

In geometric problems, various optimality criteria are of interest, including any pair from the following list: Euclidean (L_2) length, rectilinear (L_1) length, other L_p metrics, link distance, total turn, and so on. NP-hardness is known for several versions [AMP91]. One problem of particular interest is to compute a Euclidean shortest path within a polygonal domain, constrained to have at most k links. No exact solution is currently known for this problem. Part of the difficulty is that a minimum-link path will not, in general, lie on the visibility graph (or on any simple discrete graph). Furthermore, the computation of the turn points of such an optimal path appears to require the solution to high-degree polynomials. A $(1 + \varepsilon)$-approximation to the shortest k-link path in a simple polygon P can be found in time $O(n^3k^3 \log (Nk/\varepsilon^{1/k}))$, where N is the largest integer coordinate of any vertex of P [MPA92]. In a *simple* polygon, one can always find an s-t path that simultaneously is within a factor 2 of optimal in link distance and within a factor $\sqrt{2}$ of optimal in Euclidean length; a corresponding result is not possible for polygons with holes. However, in $O(kE_{\rm VG}^2)$ time, one can compute a path in a polygonal domain having at most $2k$ links and length at most that of a shortest k-link path.

In a rectilinear polygonal domain, efficient algorithms are known for the bicriterion path problem that combines *rectilinear* link distance and L_1 length [LYW96]. For example, efficient algorithms are known in two or more dimensions for computing optimal paths according to a *combined metric*, defined to be a linear combination of rectilinear link distance and L_1 path length [BKNO92]. (Note that this is not the same as computing the Pareto-optimal solutions.) Chen et al. [CDK01] give efficient algorithms for computing a shortest k-link rectilinear path, a minimum-link shortest rectilinear path, or any combined objective that uses a monotonic function of rectilinear link length and L_1 length in a rectilinear polygonal domain. Single-source queries can be answered in time $O(\log n)$, after $O(n \log^{3/2} n)$ preprocessing time to construct a data structure of size $O(n \log n)$; two-point queries can be answered in time $O(\log^2 n)$, using $O(n^2 \log^2 n)$ preprocessing time and space [CDK01].

OPEN PROBLEMS

1. Can one approximate link distance in a polygonal domain with a factor better than $O(\sqrt{h})$ in significantly subquadratic time? (Obtaining approximation factor $(2-\varepsilon)$ is 3SUM-hard [MPS14].)
2. What is the smallest size data structure for a simple polygon P that allows logarithmic-time two-point link distance queries?
3. For a polygonal domain (with holes), what is the complexity of computing a shortest k-link path between two given points?
4. What is the complexity of the ladder problem for a polygonal domain, in which the cost of motion is the total work involved in translation/rotation?
5. Is it NP-hard to minimize the d_1-distance of a ladder endpoint?
6. What is the complexity of the bounded curvature shortest-path problem in a simple polygon?

31.4 GEOMETRIC NETWORK OPTIMIZATION

All of the problems considered so far involved computing a shortest path from one point to another (or from one point to all other points). We consider now some other network optimization problems, in which the objective is to compute a shortest path, cycle, tree, or other graphs, subject to various constraints. A summary of results is given in Table 31.4.1.

Preliminary version (July 17, 2017). To appear in the Handbook of Discrete and Computational Geometry,
J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

GLOSSARY

Minimum spanning tree* (MST) *of S: A tree of minimum total length whose nodes are a given set S of n points, and whose edges are line segments joining pairs of points.

Minimum Steiner spanning tree (Steiner tree) of S: A tree of minimum total length whose nodes are a superset of a given set S of n points, and whose edges are line segments joining pairs of points. Those nodes that are not points of S are called ***Steiner points***.

Minimum Steiner forest of n ***point pairs:*** A forest of minimum total length such that each of the given point pairs lies within the same tree of the forest. The forest is allowed to utilize (Steiner) points that are not among the input points.

k-Minimum spanning tree* (*k*-MST)*: A minimum-length tree that spans some subset of $k \leq n$ points of S.

Traveling salesman problem* (TSP)*: Find a shortest cycle that visits every point of a set S of n points.

MAX* TSP*: Find a *longest* cycle that visits every point of a set S of n points.

Minimum latency tour problem: Find a tour on S that minimizes the sum of the "latencies," where the latency of $p \in S$ is the length of the tour from the given depot to p. Also known as the ***deliveryman problem***, the ***school-bus driver problem***, or the ***traveling repairman problem***.

k***-Traveling repairman problem:*** Find k tours covering S for k repairmen, minimizing the total latency. The repairmen may originate at a single common depot or at multiple depots.

Min/max-area* TSP*: Find a cycle on a given set S of points such that the cycle defines a simple polygon of minimum/maximum area.

TSP *with neighborhoods:* Find a shortest cycle that visits at least one point in each of a set of neighborhoods (e.g., polygons), $\{P_1, P_2, \ldots, P_k\}$.

Touring polygons problem: Find a shortest path/cycle that visits ***in order*** at least one point of each polygon in a sequence $(P_1, P_2, \ldots, P_k)$.

Watchman route (path) problem: Find a shortest cycle (path) within a polygonal domain P such that every point of P is visible from some point of the cycle.

Lawnmowing problem: Find a shortest cycle (path) for the center of a disk (a "lawnmower" or "cutter") such that every point of a given (possibly disconnected) region is covered by the disk at some position along the cycle (path).

Milling problem: Similar to the lawnmowing problem, but with the constraint that the cutter must at all times remain inside the given region (the "pocket" to be milled). When milling a polygonal region with a circular cutter, the portion that must be covered is the union of all disks within the polygonal region; the cutter cannot reach into convex corners of the polygon.

Zookeeper's problem: Find a shortest cycle in a simple polygon P (the ***zoo***) through a given vertex v such that the cycle visits every one of a set of k disjoint convex polygons (***cages***), each sharing an edge with P.

Aquarium-keeper's problem: Find a shortest cycle in a simple polygon P (the ***aquarium***) such that the cycle touches every edge of P.

Preliminary version (July 17, 2017). To appear in the Handbook of Discrete and Computational Geometry,
J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

Safari route problem: Find a shortest tour visiting a set of convex polygonal cages attached to the inside wall of a simple polygon P.

Relative convex hull of point set S within simple polygon P***:*** The shortest cycle within P that surrounds S. The relative convex hull is necessarily a simple polygon, with vertices among the points of S and the vertices of P.

Monotone path problem: Find a shortest monotone path (if any) from s to t in a polygonal domain P. A polygonal path is ***monotone*** if there exists a direction vector d such that every directed edge of the path has a nonnegative inner product with d.

Doubling dimension (ddim): A metric space $\mathcal{X}$ is said to have *doubling constant* c_d if any ball of radius r can be covered by c_d balls of radius $r/2$; the logarithm of c_d is the *doubling dimension (ddim)* of $\mathcal{X}$. Euclidean d-space, $\mathbb{R}^d$, has ddim $O(d)$.

MINIMUM SPANNING TREES

The (Euclidean) minimum spanning tree problem can be solved to optimality in the plane in time $O(n \log n)$ by appealing to the fact that the MST is a subgraph of the Delaunay triangulation; see Chapters 27 and 29. Efficient approximations in $\mathbb{R}^d$ are based on spanners (Section 32.3).

The Steiner tree and k-MST problems, however, are NP-hard; both have polynomial-time approximation schemes [Aro98, Mit99]. (In comparison, in graphs a 2-approximation is known for k-MST, as well as k-TSP [Gar05].) A PTAS for Steiner tree that takes time $O(n \log n)$ in any fixed dimension has been devised based on the concept of *banyans*, a generalization of the notion of t-spanners (Section 32.3), in combination with the PTAS techniques developed for the TSP and related problems [Aro98, Mit99, RS98]. A "t-banyan" approximates to within factor t the interconnection cost (allowing Steiner points) for subsets of sites of *any* cardinality (not just 2 sites, as in the case of t-spanners); [RS98] show that for any fixed $\varepsilon > 0$ and $d \geq 1$, there exists a $(1+\varepsilon)$-banyan having $O(n)$ vertices and $O(n)$ edges, computable in $O(n \log n)$ time.

TRAVELING SALESMAN PROBLEM

The traveling salesman problem is a classical problem in combinatorial optimization, and has been studied extensively in its geometric forms. The problem is NP-hard, but has a simple 2-approximation algorithm based on "doubling" the minimum spanning tree. The Christofides heuristic augments a minimum spanning tree with a minimum-length matching on the odd-degree nodes of the tree, thereby obtaining an Eulerian graph from which a tour can be extracted; this yields a 1.5-approximation algorithm. (For the s-t path TSP, of computing a shortest Hamiltonian path between specified endpoints s and t, there is a $(1+\sqrt{5})/2$-approximation in metric spaces [AKS15].) For the *graphic TSP*, in which distances are given by shortest path lengths in an unweighted graph, there is a $(13/9)$-approximation [Muc14].

Geometry helps in obtaining improved approximations: There are polynomial-time approximation schemes for geometric versions of the TSP, allowing one, for

Preliminary version (July 17, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

TABLE 31.4.1 Other optimal path/cycle/network problems.

PROBLEM	COMPLEXITY	NOTES	SOURCE
Min spanning tree	$O(n \log n)$	exact, in $\mathbb{R}^2$	[PS85]
(MST) in $\mathbb{R}^d$	$O(n \log n)$	$(1+\varepsilon)$-approx, fixed d	[CK95]
Steiner tree in $\mathbb{R}^d$	$O(n \log n)$	$(1+\varepsilon)$-approx, fixed d	[RS98]
Steiner forest in $\mathbb{R}^2$	$O(n \log^c n)$	$(1+\varepsilon)$-approx	[BKM15]
k-MST in $\mathbb{R}^d$	$O(n \log n)$	$(1+\varepsilon)$-approx, fixed d	[RS98]
Min bicon. subgraph	$(1+\varepsilon)$-approx	$O(n \log n)$	[CL00]
Traveling salesman prob	$O(n \log n)$	$(1+\varepsilon)$-approx, fixed d	[Aro98, Mit99, RS98]
(TSP) in $\mathbb{R}^d$	$O(n)$	$(1+\varepsilon)$-approx, fixed d	[BG13]
	(randomized)	real RAM, atomic floor	[BG13]
TSP in low-dimensional	$2^{(k/\varepsilon)^{O(k^2)}} n+$	$(1+\varepsilon)$-approx	[BGK12, Got15]
metric space	$+(k/\varepsilon)^{O(k)} n \log^2 n$	ddim k, randomized	
MAX TSP	NP-hard in $\mathbb{R}^3$	$(1+\varepsilon)$-approx	[BFJ03]
	$O(n)$	L_1, L_∞ in $\mathbb{R}^2$	
	$O(n^{f-2} \log n)$	f-facet polyhedral norm	
Min-area TSP	NP-complete		[Fek00]
Max-area TSP	NP-complete	$(1/2)$-approx	[Fek00]
TSP w/neighborhoods	NP-hard	$O(\log n)$-approx	[MM95, GL99]
	no $O(1)$-approx	disconnected regions, $\mathbb{R}^2$	[SS06]
	APX-hard	connected regions, $\mathbb{R}^2$	[SS06]
	$O(1)$-approx	disjoint regions, $\mathbb{R}^2$	[Mit10]
	$(1+\varepsilon)$-approx	disjoint fat regions, $\mathbb{R}^2$	[Mit07]
	$(1+\varepsilon)$-approx	fat, weakly disjoint, ddim k	[CJ16]
Touring polygons prob	NP-hard	$(1+\varepsilon)$-approx	[DELM03, AMZ14]
	$O(nk^2 \log n)$	convex polygons	[DELM03]
	$O(nk \log(n/k))$	disjoint convex polygons	[DELM03]
Minimum latency prob	3.59-approx	metric space	[CGRT03, GK97]
	$(1+\varepsilon)$-approx	polytime in $\mathbb{R}^2$	[Sit14]
k-Traveling repairman	8.497-approx	single depot	[CGRT03, FHR07]
k-Traveling repairman	$(12+\varepsilon)$-approx	multidepot	[CK04]
Watchman route	$O(n^4 \log n)$	simple polygon	[DELM03]
(fixed source)	$O(n^3 \log n)$	simple polygon	[DELM03]
	$O(n)$	rectilinear simple polygon	[CN91]
	NP-hard	polygonal domain	[CN88]
	$O(\log^2 n)$-approx	polygonal domain	[Mit13]
Min-link watchman	NP-hard	$O(\log n)$-approx	[AMP03]
	NP-hard	simple polygon	[AL93]
	$O(1)$-approx	simple polygon	[AL95]
Lawnmowing problem	NP-hard	$O(1)$-approx, PTAS	[AFM00, FMS12]
Milling problem	$O(1)$-approx, PTAS	simple polygon	[AFM00, FMS12]
	NP-hard, $O(1)$-approx	polygonal domain	[AFM00, AFI+09]
Simple Hamilton path	$O(n^2 m^2)$	m points in simple n-gon	[CCS00]
	NP-complete	polygonal domain	[CCS00]
Aquarium-keeper's prob	$O(n)$	simple polygon	[CEE+91]
Zookeeper's problem	$O(n \log n)$	simple polygon	[Bes03a]
Relative convex hull	$\Theta(n + k \log kn)$	k points in simple n-gon	[GH89]
Monotone path prob	$O(n^3 \log n)$		[ACM89]

any fixed $\varepsilon > 0$, to get within a factor $(1+\varepsilon)$ of optimality [Aro98, Mit99]. Using the key idea of using t-spanners, the running time was improved to $O(n \log n)$ in any fixed dimension [RS98]. In fact, in the real RAM model (with atomic floor or mod function), a randomized linear-time PTAS is known in fixed dimension [BG13]. More generally, the TSP in metric spaces of bounded doubling dimension (ddim) also has a PTAS; specifically, for ddim k there is a PTAS running in time $2^{(k/\varepsilon)^{O(k^2)}} n + (k/\varepsilon)^{O(k)} n \log^2 n$, based on computing light spanners [BGK12, Got15]. The geometric TSP has also been studied with the objective of minimizing the sum of the direction changes at the points along the tour; this angular metric TSP is known to be NP-hard and to have a polynomial-time $O(\log n)$-approximation [ACK+00].

The ***TSP-with-neighborhoods*** (TSPN) problem arises when we require the tour/path to visit a set of regions, rather than a set of points. An $O(\log n)$-approximation algorithm is known for general connected regions in the plane [MM95, GL99]. It is NP-hard to approximate TSPN to within factor $2 - \varepsilon$ for connected regions that are allowed to overlap in the plane [SS06]. TSPN is APX-hard even for regions that are (intersecting) line segments, all of about the same length [EFS06]. Constant-factor approximation algorithms are known for some special cases [AH94, BGK+05, DM03], as well as for the general case of disjoint connected regions in the plane [Mit10]. Polynomial-time approximation schemes are known for disjoint (or "weakly disjoint") fat regions in the plane [Mit07], and, more generally, in metric spaces of bounded doubling dimension [CJ16]. For general disconnected regions in the plane, no constant factor approximation is possible (unless P=NP) [SS06]. For regions consisting each of k discrete points, a $(3/2)k$-approximation holds (even in general metric spaces) [Sla97]; for instances in the Euclidean plane, there is a lower bound of $\Omega(\sqrt{k})$, for $k > 4$, on the approximation factor [SS06], and the problem is APX-hard for $k = 2$ [DO08]. Results in higher dimensions include $O(1)$-approximation for neighborhoods that are planes or unit disks and $O(\log^3 n)$-approximation for lines [DT16]. It is NP-hard to approximate within any constant factor, even for connected, disjoint regions in 3-space [SS06].

LAWNMOWER AND WATCHMAN ROUTE PROBLEM

The lawnmowing problem is a TSP variant that seeks an optimal path for a lawnmower, modeled as, say, a circular cutter that must sweep out a region that covers a given domain of "grass." The milling problem requires that the cutter remains within the given domain. These problems are NP-hard in general, but constant-factor approximation algorithms are known [AFM00, AFI+09], and some variants have a PTAS [FMS12].

The watchman route problem seeks a shortest-path/tour so that every point of a domain is seen from some point along the path/tour; i.e., the path/tour must visit the visibility region associated with each point of the domain. In the case of a simple polygonal domain, the watchman route problem has an $O(n^4 \log n)$ time algorithm to compute an exact solution and $O(n^3 \log n)$ is possible if we are given a point through which the tour must pass [DELM03]. In the case of a polygonal domain with holes, the problem is easily seen to be NP-hard (from Euclidean TSP); an $O(\log^2 n)$-approximation algorithm is given by [Mit13], as well as a lower bound of $\Omega(\log n)$ on the approximation factor (assuming P$\neq$NP).

Preliminary version (July 17, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

OPEN PROBLEMS

1. Is the MAX TSP NP-hard in the Euclidean plane? What if the tour is required to be noncrossing?

2. Is there a PTAS for the minimum latency problem and for the k-traveling repairman problem for points in any fixed dimension? (In $\mathbb{R}^2$, the minimum latency problem has a PTAS [Sit14].)

3. Can one obtain a PTAS for the TSP with neighborhoods (TSPN) problem in the plane if the regions are disjoint and each is a connected set? (Without disjointness, the problem is APX-hard even for regions that are line segments all of very nearly the same length [EFS06].) For (disconnected) regions each consisting of a discrete set of k points in the plane, what approximation factor (as a function of k) can be achieved? (The known $(3/2)k$-approximation does not exploit geometry [Sla97]; the lower bound $\Omega(\sqrt{k})$ applies to instances in the plane [SS06].) Is there a polynomial-time exact algorithm for TSPN in 3-space for neighborhoods that are planes?

4. Is the milling problem in simple polygons NP-hard?

5. Can the (Euclidean) watchman route in a simple polygon be computed in (near) linear time? (The fixed-source version is currently solved in $O(n^3 \log n)$ time [DELM03] in general, but in time $O(n)$ in rectilinear polygons [CN91].)

31.5 HIGHER DIMENSIONS

GLOSSARY

Polyhedral domain: A set $P \subset \mathbb{R}^3$ whose interior is connected and whose boundary consists of a union of a finite number of triangles. (The definition is readily extended to d dimensions, where the boundary must consist of a union of $(d-1)$-simplices.) The complement of P consists of connected (polyhedral) components, which are the ***obstacles***.

Orthohedral domain: A polyhedral domain having each boundary facet orthogonal to one of the coordinate axes.

Polyhedral surface: A connected union of triangles, with any two triangles intersecting in a common edge, a common vertex, or not at all, and such that every point in the relative interior of the surface has a neighborhood homeomorphic to a disk.

Polyhedral terrain surface: A polyhedral surface given by an altitude function of position (x, y); a vertical line meets the surface in at most one point.

Edge sequence: The ordered list of obstacle edges that are intersected by a path.

COMPLEXITY

In three or more dimensions, most shortest-path problems become very difficult. In particular, there are two sources of complexity, even in the most basic Euclidean shortest-path problem in a polyhedral domain P.

One difficulty arises from algebraic considerations. In general, the shortest path in a polyhedral domain need not lie on any kind of discrete graph. Shortest paths in a polyhedral domain will be polygonal, with bend points that generally lie *interior* to obstacle edges, obeying a simple "unfolding" property: The path must enter and leave at the same angle to the edge. It follows that any locally optimal subpath joining two consecutive obstacle vertices can be "unfolded" at each edge along its edge sequence, thereby obtaining a straight segment. Given an edge sequence, this local optimality property uniquely identifies a shortest path through that edge sequence. However, to compare the lengths of two paths, each one shortest with respect to two (different) edge sequences, requires exponentially many bits, since the algebraic numbers that describe the optimal path lengths may have exponential degree.

A second difficulty arises from combinatorial considerations. The number of combinatorially distinct (i.e., having distinct edge sequences) shortest paths between two points may be exponential. This fact leads to a proof of the NP-hardness of the shortest-path problem [CR87]. In fact, the problem is NP-hard even for the case of obstacles that are disjoint axis-aligned boxes, even if the obstacles are all "stacked" axis-aligned rectangles (i.e., horizontal rectangles, orthogonal to the z-axis, with edges parallel to the x- and y-axes), and even if the rectangles are quadrants, each of which is unbounded to the northeast or to the southwest [MS04].

Thus, it is natural to consider approximation algorithms for the general case, or to consider special cases for which polynomial bounds are achievable.

SPECIAL CASES

If the polyhedral domain P has only a small number k of convex obstacles, a shortest path can be found in $n^{O(k)}$ time [Sha87]. If the obstacles are known to be vertical "buildings" (prisms) having only k different heights, then shortest paths can be found in time $O(n^{6k-1})$ [GHT89], but it is not known if this version of the problem is NP-hard if k is allowed to be large.

If we require paths to stay on a polyhedral surface (i.e., the domain P is essentially 2D), then the unfolding property of optimal paths can be exploited to yield polynomial-time algorithms. The continuous Dijkstra paradigm leads to an algorithm requiring $O(n^2 \log n)$ time to construct a shortest path map (or a geodesic Voronoi diagram), where n is the number of vertices of the surface [MMP87]. The worst-case running time has been improved to $O(n^2)$ by Chen and Han [CH96]. For the case of shortest paths on a convex polyhedral surface (or avoiding a single convex polytope obstacle in 3-space), Schreiber and Sharir [SS08] have given an optimal time $O(n \log n)$ algorithm based on the continuous Dijkstra paradigm; Schreiber [Sch07] has extended the methods to yield an $O(n \log n)$ time algorithm on "realistic" polyhedral surfaces. Kapoor [Kap99, O'R99] has announced an $O(n \log^2 n)$ time algorithm for general polyhedral surfaces, also based on the continuous Dijkstra paradigm. Since shortest paths on polyhedral surfaces are critical in many applications in computer graphics, practical experimental studies of shortest path algorithms have been conducted; see, e.g., [SSK$^+$05]. There has been considerable study of shortest paths and cycles on surfaces of complex topology; see Chapter 23.

Several facts are known about the set of edge sequences corresponding to shortest paths on the surface of a *convex* polytope P in $\mathbb{R}^3$. In particular, the worst-case number of distinct edge sequences that correspond to a shortest path between some pair of points is $\Theta(n^4)$, and the exact set of such sequences can be computed in time $O(n^6\beta(n)\log n)$, where $\beta(n) = o(\log^* n)$ [AAOS97]. (A simpler $O(n^6)$ algorithm can compute a small superset of the sequences.) The number of ***maximal*** edge sequences for shortest paths is $\Theta(n^3)$. Some of these results depend on a careful study of the ***star unfolding*** with respect to a point p on the boundary, ∂P, of P. The star unfolding is the (nonoverlapping) cell complex obtained by subtracting from ∂P the shortest paths from p to the vertices of P, and then flattening the resulting boundary.

Results on exact algorithms for special cases are summarized in Table 31.5.1.

TABLE 31.5.1 Shortest paths in 3-space, d-space: exact algorithms.

OBSTACLES/DOMAIN	COMPLEXITY	NOTES	SOURCE
Polyhedral domain	NP-hard	convex obstacles	[CR87]
Axis-parallel stacked rect.	NP-hard	L_2 metric	[MS04]
One convex 3D polytope	$O(n\log n)$ time	$O(n\log n)$ space	[SS08]
k convex polytopes	$n^{O(k)}$	fixed k	[Sha87]
Vertical buildings	$O(n^{6k-1})$	k different heights	[GHT89]
Axis-parallel boxes	$O(n^2\log^3 n)$	L_1 metric	[CKV87]
Axis-parallel	$O(n^2\log n)$	L_1 metric	[CY95]
(disjoint)		path monotonicity, $\mathbb{R}^d$	[CY96]
Axis-parallel boxes, $\mathbb{R}^d$	$O(n^d\log n)$ preproc	combined L_1, link dist	[BKNO92]
	$O(\log^{d-1} n)$ query	single-source queries	
	$O((n\log n)^{d-1})$ space		
Above a terrain	$O(n^3\log n)$ time	L_1 metric	[MS04]
Polyhedral surface	$O(n^2)$ time	builds SPM(s)	[CH96, MMP87]
		geodesic Voronoi	
Polyhedral surface	$O(n^2\log^4 n)$ time	L_1, L_∞ metric	[CJ14b]
Two-point query	$O((\sqrt{n}/m^{1/4})\log n)$ query	convex polytope	[AAOS97]
	$O(n^6 m^{1+\delta})$ space, preproc	$1\le m\le n^2$, $\delta>0$	
Geodesic diameter	$O(n^8\log n)$	convex polytope	[AAOS97]

APPROXIMATION ALGORITHMS

Papadimitriou [Pap85] was the first to study the general problem from the point of view of approximations, giving a fully polynomial approximation scheme that produces a path guaranteed to be no longer than $(1+\varepsilon)$ times the length of a shortest path, in time $O(n^3(L+\log(n/\varepsilon))^2/\varepsilon)$, where L is the number of bits necessary to represent the value of an integer coordinate of a vertex of P. An alternate bound of Clarkson [Cla87] improves the running time in the case that $n\varepsilon^3$ is large. Choi, Sellen, and Yap [CSY95, CSY97] introduce the notion of "precision-sensitivity," writing the complexity in terms of a parameter, δ, that measures the implicit precision of the input instance, while drawing attention to the distinction between bit complexity and algebraic complexity.

Har-Peled [Har99b] shows how to compute an ***approximate shortest path map*** in polyhedral domains, computing, for fixed source s and $0 < \varepsilon < 1$, a subdivision of size $O(n^2/\varepsilon^{4+\delta})$ in time roughly $O(n^4/\varepsilon^6)$, so that for any point $t \in \mathbb{R}^3$ a $(1+\varepsilon)$-approximation of the length of a shortest s-t path can be reported in time $O(\log(n/\varepsilon))$.

Considerable effort has been devoted to approximation algorithms for shortest paths on polyhedral surfaces. Given a convex polytope obstacle, Agarwal et al. [AHPSV97] show how to surround the polytope with a constant-size ($O(\varepsilon^{-3/2})$, now improved to $O(\varepsilon^{-5/4})$ [CLM05]) convex polytope having the property that shortest paths are approximately preserved (within factor $(1+\varepsilon)$) on the outer polytope. This results in an approximation algorithm of time complexity $O(n\log(1/\varepsilon)+f(\varepsilon^{-5/4}))$, where $f(m)$ denotes the time complexity of solving exactly a shortest-path problem on an m-vertex convex surface (e.g., $f(m) = O(m^2)$ using [CH96]). Har-Peled [Har99a] gives an $O(n)$-time algorithm to preprocess a convex polytope so that a two-point query can be answered in time $O((\log n)/\varepsilon^{3/2} + 1/\varepsilon^3)$, yielding the $(1+\varepsilon)$-approximate shortest path distance, as well as a path having $O(1/\varepsilon^{3/2})$ segments that avoids the interior of the input polytope.

Varadarajan and Agarwal [VA99] obtained the first subquadratic-time algorithms for approximating shortest paths on general (nonconvex) polyhedral surfaces, computing a $(7+\varepsilon)$-approximation in $O(n^{5/3}\log^{5/3} n)$ time, or a $(15+\varepsilon)$-approximation in $O(n^{8/5}\log^{8/5} n)$ time. Their method is based on a partitioning of the surface into $O(n/r)$ patches, each having at most r faces, using a planar separator theorem. (The parameter r is chosen to be $n^{1/3}\log^{1/3} n$ or $n^{2/5}\log^{2/5} n$.) Then, on the boundary of each patch, a carefully selected set of points ("portals") is chosen, and these are interconnected with a graph that approximates shortest paths within each patch.

For a polyhedral terrain surface, one often wants to model the cost of motion that is anisotropic, depending on the steepness (gradient) of the ascent/descent. Cheng and Jin [CJ14b] give a $(1+\varepsilon)$-approximation, taking time $O(\frac{1}{\sqrt{\varepsilon}}n^2\log n + n^2\log^4 n)$, in this model in which the cost of movement is a linear combination of path length and total ascent, and there are constraints on the steepness of the path. For the problem of computing a shortest path on a polyhedral terrain surface, subject to the constraint that the path be *descending* in altitude along the path, Cheng and Jin [CJ14a] have given a $(1+\varepsilon)$-approximation algorithm with running time $O(n^4\log(n/\varepsilon))$. Exact solution of the shortest descending path problem on a polyhedral terrain seems to be quite challenging; see, e.g., [AL09, AL11, RDN07].

Practical approximation algorithms are based on searching a discrete graph (an "edge subdivision graph," or a "pathnet")[LMS01, MM97] by placing Steiner points judiciously on the edges (or, possibly interior to faces) of the input surface. This approach applies also to the case of weighted surfaces and weighted convex decompositions of $\mathbb{R}^3$; see the earlier discussion of the weighted region problem. One can obtain provable results on the approximation factor; see Table 31.5.2. It is worth noting, however, that these complexity bounds are under the assumption that certain geometric parameters are "constants"; these parameters may be unbounded in terms of ε and the combinatorial input size n.

TABLE 31.5.2 Shortest paths in 3-space: approximation algorithms.

OBSTACLES/DOMAIN	COMPLEXITY	NOTES	SOURCE
Polyhedral domain	$O(n^4(L+\log(\frac{n}{\varepsilon}))^2/\varepsilon^2)$	$(1+\varepsilon)$-approx	[Pap85]
	$O(n^2 \text{polylog } n/\varepsilon^4)$	$(1+\varepsilon)$-approx	[Cla87]
Polyhedral domain	$O(\frac{n^2}{\varepsilon^3} \log \frac{1}{\varepsilon} \log n)$	$(1+\varepsilon)$-approx	[AMS00]
		geometric parameters	
Polyhedral domain	NP-hard in 3-space	PTAS	[KLPS16]
Orthohedral poly. dom.	$O(\frac{n^2}{\varepsilon^3} \log \frac{1}{\varepsilon} \log n)$	$(1+\varepsilon)$-approx	[AMS00]
Orthohedral poly. dom.	$O(n^2 \log^2 n)$	3D, rectilinear link dist.	[PS11]
Poly. terrain surface	$O(\frac{1}{\sqrt{\varepsilon}} n^2 \log n + n^2 \log^4 n)$	$(1+\varepsilon)$-approx	[CJ14b]
	cost(length,ascent)	gradient constraints	
Weighted poly. dom.	$O(K \frac{n}{\varepsilon^3} \log \frac{1}{\varepsilon}(\frac{1}{\sqrt{\varepsilon}} + \log n))$	$(1+\varepsilon)$-approx	[ADMS13]
	n tetrahedra	geometric parameter K	
Weighted poly. dom.	$O(2^{2^{O(\kappa)}} \frac{n}{\varepsilon^7} \log^2 \frac{W}{\varepsilon} \log^2 \frac{NW}{\varepsilon})$	$(1+\varepsilon)$-approx	[CCJV15]
	n tetrahedra	weights $\{1,\dots,W\}$	
	coordinates $\{1,\dots,N\}$	κ skinny tetra per comp	
One convex obstacle	$O(\varepsilon^{-5/4}\sqrt{n})$ expected	$(1+\varepsilon)$-approx	[CLM05]
k convex polytopes	$O(n)$	$2k$-approx	[HS98]
Convex poly. surface	$O(n \log \frac{1}{\varepsilon} + \frac{1}{\varepsilon^3})$	$(1+\varepsilon)$-approx	[AHPSV97]
Convex poly. surface	$O(\frac{1}{\varepsilon^{1.5}} \log n + \frac{1}{\varepsilon^3})$ query	$(1+\varepsilon)$-approx	[Har99a]
	$O(n)$ preproc (pp.)	two-point query	
Convex poly. surface	$O(\log \frac{n}{\varepsilon})$ query	single-source queries	[Har99b]
	$O(\frac{n}{\varepsilon^3} \log \frac{1}{\varepsilon} + \frac{n}{\varepsilon^{1.5}} \log \frac{1}{\varepsilon} \log n)$ pp.	$O(\frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$ size SPM	
Nonconv. poly. surface	$O(\log \frac{n}{\varepsilon})$ query	single-source queries	[Har99b]
	$O(n^2 \log n + \frac{n}{\varepsilon} \log \frac{1}{\varepsilon} \log \frac{n}{\varepsilon})$ pp.	$O(\frac{n}{\varepsilon} \log \frac{1}{\varepsilon})$ size SPM	
Convex poly. surface	$O(n + \frac{1}{\varepsilon^6})$	$(1-\varepsilon)$-approx diameter	[Har99a]
Nonconv. poly. surface	$O(n^{5/3} \log^{5/3} n)$	$(7+\varepsilon)$-approx	[VA99]
	$O(n^{8/5} \log^{8/5} n)$	$(15+\varepsilon)$-approx	[VA99]
Nonconv. poly. surface	$O(\frac{n}{\varepsilon} \log \frac{1}{\varepsilon} \log n)$	$(1+\varepsilon)$-approx	[AMS00]
		geometric parameters	
Vertical buildings	$O(n^2)$	1.1-approx	[GHT89]

OTHER METRICS

Link distance in a polyhedral domain in $\mathbb{R}^d$ can be approximated (within factor 2) in polynomial time by searching a weak visibility graph whose nodes correspond to simplices in a simplicial decomposition of the domain. Computing the exact link distance is NP-hard in $\mathbb{R}^3$, even on terrains; however, a PTAS is known [KLPS16]. If the domain is orthohedral, rectilinear link distance can be computed efficiently, in time $O(n^2 \log^2 n)$ [PS11].

For the case of orthohedral domains and rectilinear (L_1) shortest paths, the shortest-path problem in $\mathbb{R}^d$ becomes relatively easy to solve in polynomial time, since the grid graph induced by the facets of the domain serves as a path-preserving graph that we can search for an optimal path. In $\mathbb{R}^3$, we can do better than to use the $O(n^3)$ grid graph induced by $O(n)$ facets; an $O(n^2 \log^2 n)$ size subgraph suffices, which allows a shortest path to be found using Dijkstra's algorithm in time $O(n^2 \log^3 n)$ [CKV87]. More generally, in $\mathbb{R}^d$ one can compute a data structure of size $O((n \log n)^{d-1})$, in $O(n^d \log n)$ preprocessing time, that supports fixed-source

Preliminary version (July 17, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

link distance queries in $O(\log^{d-1} n)$ time [BKNO92]. In fact, this last result can be extended, within the same complexities, to the case of a combined metric, in which path cost is measured as a linear combination of L_1 length and rectilinear link distance.

For the special case of disjoint rectilinear box obstacles and rectilinear (L_1) shortest paths, a structural result may help in devising very efficient algorithms: There always exists a coordinate direction such that *every* shortest path from s to t is monotone in this direction [CY96]. In fact, this result has led to an $O(n^2 \log n)$ algorithm for the case $d = 3$.

OPEN PROBLEMS

1. Can one compute shortest paths on a polyhedral surface in $\mathbb{R}^3$ in $O(n \log n)$ time using $O(n)$ space?
2. Can one compute a shortest path map for a polyhedral domain in output-sensitive time?
3. Is it 3SUM-hard to compute minimum-link rectilinear paths in 3D, or can one obtain a subquadratic-time algorithm? A nearly-quadratic-time algorithm is known [PS11].
4. What is the complexity of the shortest-path problem in 3-space among disjoint unit disk obstacles or disjoint axis-aligned unit cubes?
5. Can two-point queries be solved efficiently for Euclidean (or L_1) shortest paths among obstacles in 3-space?

31.6 SOURCES AND RELATED MATERIAL

SURVEYS

Some other related surveys offer additional material and references:

[BMSW11]: A survey of shortest paths on 3D surfaces. [Har11]: A book on geometric approximation algorithms.

[Mit00]: Another survey on geometric shortest paths and network optimization.

[Mit15]: A survey on approximation schemes for geometric network optimization.

RELATED CHAPTERS

Chapter 10: Geometric graph theory
Chapter 23: Computational topology of graphs on surfaces
Chapter 29: Triangulations and mesh generation
Chapter 30: Polygons

Preliminary version (July 17, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

Chapter 32: Proximity algorithms
Chapter 33: Visibility
Chapter 50: Algorithmic motion planning

REFERENCES

[AAOS97] P.K. Agarwal, B. Aronov, J. O'Rourke, and C.A. Schevon. Star unfolding of a polytope with applications. *SIAM J. Comput.*, 26:1689–1713, 1997.

[ABB+16] H.-K. Ahn, L. Barba, P. Bose, J.-L. De Carufel, M. Korman, and E. Oh. A linear-time algorithm for the geodesic center of a simple. *Discrete Comput. Geom.*, 56:836–859, 2016.

[ABL+02] P.K. Agarwal, T. Biedl, S. Lazard, S. Robbins, S. Suri, and S. Whitesides. Curvature-constrained shortest paths in a convex polygon. *SIAM J. Comput.*, 31:1814–1851, 2002.

[AC91] M.J. Atallah and D.Z. Chen. Parallel rectilinear shortest paths with rectangular obstacles. *Comput. Geom.*, 1:79–113, 1991.

[AC93] M.J. Atallah and D.Z. Chen. On parallel rectilinear obstacle-avoiding paths. *Comput. Geom.*, 3:307–313, 1993.

[ACC+96] S.R. Arikati, D.Z. Chen, L.P. Chew, G. Das, M. Smid, and C.D. Zaroliagis. Planar spanners and approximate shortest path queries among obstacles in the plane. In *Proc. 4th Eur. Sympos. Algorithms*, vol. 1136 of *LNCS*, pages 514–528, Springer, Berlin, 1996.

[ACM89] E.M. Arkin, R. Connelly, and J.S.B. Mitchell. On monotone paths among obstacles, with applications to planning assemblies. In *Proc. 5th Sympos. Comput. Geom.*, pages 334–343, ACM Press, 1989.

[ACK+00] A. Aggarwal, D. Coppersmith, S. Khanna, R. Motwani, and B. Schieber. The angular-metric traveling salesman problem. *SIAM J. Comput.*, 29:697–711, 2000.

[ADMS13] L. Aleksandrov, H. Djidjev, A. Maheshwari, and J.-R. Sack. An approximation algorithm for computing shortest paths in weighted 3-d domains. *Discrete Comput. Geom.*, 50:124–184, 2013.

[AEK+16] E.M. Arkin, A. Efrat, C. Knauer, J.S.B. Mitchell, V. Polishchuk, G. Rote, L. Schlipf, and T. Talvitie. Shortest path to a segment and quickest visibility queries. *J. Comput. Geom.*, 7:77–100, 2016.

[AFI+09] E.M. Arkin, S.P. Fekete, K. Islam, H. Meijer, J.S.B. Mitchell, Y. Núñez-Rodríguez, V. Polishchuk, D. Rappaport, and H. Xiao. Not being (super)thin or solid is hard: A study of grid Hamiltonicity. *Comput. Geom.*, 42:582–605, 2009.

[AFM00] E.M. Arkin, S.P. Fekete, and J.S.B. Mitchell. Approximation algorithms for lawn mowing and milling. *Comput. Geom.*, 17:25–50, 2000.

[AFW93] B. Aronov, S.J. Fortune, and G. Wilfong. Furthest-site geodesic Voronoi diagram. *Discrete Comput. Geom.*, 9:217–255, 1993.

[AFS16] P.K. Agarwal, K. Fox, and O. Salzman. An efficient algorithm for computing high-quality paths amid polygonal obstacles. In *Proc. 27th ACM-SIAM Sympos. Discrete Algorithms*, pages 1179–1192, 2016.

[AH94] E.M. Arkin and R. Hassin. Approximation algorithms for the geometric covering salesman problem. *Discrete Appl. Math.*, 55:197–218, 1994.

[AHK+14] O. Aichholzer, T. Hackl, M. Korman, A. Pilz, and B. Vogtenhuber. Geodesic-preserving polygon simplification. *Internat. J. Comput. Geom. Appl.*, 24:307–323, 2014.

[AHPSV97] P.K. Agarwal, S. Har-Peled, M. Sharir, and K.R. Varadarajan. Approximate shortest paths on a convex polytope in three dimensions. *J. ACM*, 44:567–584, 1997.

[AKS15] H.-C. An, R. Kleinberg, and D.B. Shmoys. Improving Christofides' algorithm for the s-t path TSP. *J. ACM*, 62:34, 2015.

[AL93] M.H. Alsuwaiyel and D.T. Lee. Minimal link visibility paths inside a simple polygon. *Comput. Geom.*, 3:1–25, 1993.

[AL95] M.H. Alsuwaiyel and D.T. Lee. Finding an approximate minimum-link visibility path inside a simple polygon. *Inform. Process. Lett.*, 55:75–79, 1995.

[AL09] M. Ahmed and A. Lubiw. Shortest descending paths through given faces. *Comput. Geom.*, 42:464–470, 2009.

[AL11] M. Ahmed and A. Lubiw. Shortest descending paths: Towards an exact algorithm. *Internat. J. Comput. Geom. Appl.*, 21:431–466, 2011.

[AMP03] E.M. Arkin, J.S.B. Mitchell, and C.D. Piatko. Minimum-link watchman tours. *Inform. Process. Lett.*, 86:203–207, 2003.

[AMP91] E.M. Arkin, J.S.B. Mitchell, and C.D. Piatko. Bicriteria shortest path problems in the plane. In *Proc. 3rd Canad. Conf. Comput. Geom.*, pages 153–156, 1991.

[AMS95] E.M. Arkin, J.S.B. Mitchell, and S. Suri. Logarithmic-time link path queries in a simple polygon. *Internat. J. Comput. Geom. Appl.*, 5:369–395, 1995.

[AMS00] L. Aleksandrov, A. Maheshwari, and J.-R. Sack. Approximation algorithms for geometric shortest path problems. In *Proc. 32nd Sympos. Theory Comput.*, pages 286–295, ACM Press, 2000.

[AMS05] L. Aleksandrov, A. Maheshwari, and J.-R. Sack. Determining approximate shortest paths on weighted polyhedral surfaces. *J. ACM*, 52:25–53, 2005.

[AMZ14] A. Ahadi, A. Mozafari, and A. Zarei. Touring a sequence of disjoint polygons: Complexity and extension. *Theoret. Comput. Sci.*, 556:45–54, 2014.

[Aro98] S. Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *J. ACM*, 45:753–782, 1998.

[AW01] P.K. Agarwal and H. Wang. Approximation algorithms for curvature-constrained shortest paths. *SIAM J. Comput.*, 30:1739–1772, 2001.

[BAE+17] S.W. Bae, M. Korman, J.S.B. Mitchell, Y. Okamoto, V. Polishchuk, and H. Wang. Computing the L_1 geodesic diameter and center of a polygonal domain. *Discrete Comput. Geom.*, 57:674–701, 2017.

[Ber91] M. de Berg. On rectilinear link distance. *Comput. Geom.*, 1:13–34, 1991.

[Bes03a] S. Bespamyatnikh. An $O(n \log n)$ algorithm for the zoo-keeper's problem. *Comput. Geom.*, 24:63–74, 2002.

[Bes03b] S. Bespamyatnikh. Computing homotopic shortest paths in the plane. In *Proc. 14th ACM-SIAM Sympos. Discrete Algorithms*, pages 609–617, 2003.

[BFJ03] A. Barvinok, S.P. Fekete, D.S. Johnson, A. Tamir, G.J. Woeginger, and R. Woodroofe. The geometric maximum traveling salesman problem. *J. ACM*, 50:641–664, 2003.

[BG13] Y. Bartal and L.-A. Gottlieb. A linear time approximation scheme for Euclidean TSP. In *Proc. 54th Sympso. Found. Comp. Sci.*, pages 698–706, 2013.

[BGK12] Y. Bartal, L. Gottlieb, and R. Krauthgamer. The traveling salesman problem: low-dimensionality implies a polynomial time approximation scheme. In *Proc. 44th ACM Sympos. Theory Comput.*, pages 663–672, 2012.

[BGK+05] M. de Berg, J. Gudmundsson, M.J. Katz, C. Levcopoulos, M.H. Overmars, and A.F. van der Stappen. TSP with neighborhoods of varying size. *J. Algorithms*, 57:22—36, 2005.

[BKL11] M.G. Borgelt, M. van Kreveld, and J. Luo. Geodesic disks and clustering in a simple polygon. *Internat. J. Comput. Geom. Appl.*, 21:595–608, 2011.

[BKM15] G. Borradaile, P.N. Klein, and C. Mathieu. A polynomial-time approximation scheme for euclidean Steiner forest. *ACM Trans. Algorithms*, 11:19, 2015.

[BKNO92] M. de Berg, M. van Kreveld, B.J. Nilsson, and M.H. Overmars. Shortest path queries in rectilinear worlds. *Internat. J. Comput. Geom. Appl.*, 2:287–309, 1992.

[BKO13] S.W. Bae, M. Korman, and Y. Okamoto. The geodesic diameter of polygonal domains. *Discrete Comput. Geom.*, 50:306–329, 2013.

[BKO15] S.W. Bae, M. Korman, and Y. Okamoto. Computing the geodesic centers of a polygonal domain. *Comput. Geom.*, to appear, 2015.

[BKOW15] S.W. Bae, M. Korman, Y. Okamoto, and H. Wang. Computing the L_1 geodesic diameter and center of a simple polygon in linear time. *Comput. Geom.*, 48:495–505, 2015.

[BL96] J.-D. Boissonnat and S. Lazard. A polynomial-time algorithm for computing a shortest path of bounded curvature amidst moderate obstacles. In *Proc. 12th Sympos. Comput. Geom.*, pages 242–251, ACM Press, 1996.

[BMSW11] P. Bose, A. Maheshwari, C. Shu, and S. Wuhrer. A survey of geodesic paths on 3d surfaces. *Comput. Geom.*, 44:486–498, 2011.

[BO12] S.W. Bae and Y. Okamoto. Querying two boundary points for shortest paths in a polygonal domain. *Comput. Geom.*, 45:284–293, 2012.

[CCS00] Q. Cheng, M. Chrobak, and G. Sundaram. Computing simple paths among obstacles. *Comput. Geom.*, 16:223–233, 2000.

[CCJV15] S.-W. Cheng, M.-K. Chiu, J. Jin, and A. Vigneron. Navigating weighted regions with scattered skinny tetrahedra. In *Proc. 26th Int. Sympos. Algorithms and Computation*, vol. 9472 of *LNCS*, pages 35–45, Springer, Berlin, 2015.

[CDK01] D.Z. Chen, O. Daescu, and K.S. Klenk. On geometric path query problems. *Internat. J. Comput. Geom. Appl.*, 11:617–645, 2001.

[CDM01] D.Z. Chen, G. Das, and M. Smid. Lower bounds for computing geometric spanners and approximate shortest paths. *Discrete Appl. Math.*, 110:151–167, 2001.

[CEE+91] J. Czyzowicz, P. Egyed, H. Everett, D. Rappaport, T. Shermer, D. Souvaine, G. Toussaint, and J. Urrutia. The aquarium keeper's problem. In *Proc. 2nd ACM-SIAM Sympos. Discrete Algorithms*, pages 459–464, 1991.

[CEG+94] B. Chazelle, H. Edelsbrunner, M. Grigni, L.J. Guibas, J. Hershberger, M. Sharir, and J. Snoeyink. Ray shooting in polygons using geodesic triangulations. *Algorithmica*, 12:54–68, 1994.

[CGRT03] K. Chaudhuri, B. Godfrey, S. Rao, and K. Talwar. Paths, trees, and minimum latency tours. In *Proc. 44th IEEE Sympos. Found. Comp. Sci.*, pages 36–45, 2003.

[CH96] J. Chen and Y. Han. Shortest paths on a polyhedron. *Internat. J. Comput. Geom. Appl.*, 6:127–144, 1996.

[Che95] D.Z. Chen. On the all-pairs Euclidean short path problem. In *Proc. 6th ACM-SIAM Sympos. Discrete Algorithms*, pages 292–301, 1995.

[Che13] D.Z. Chen. Efficient algorithms for geometric shortest path query problems. In P.M. Pardalos, D.-Z. Du, and R.L. Graham, editors, *Handbook of Combinatorial Optimization*, pages 1125–1154. Springer, New York, 2013.

[CIW14] D.Z. Chen, R. Inkulu, and H. Wang. Two-point L_1 shortest path queries in the plane. In *Proc. 30th Sympos. Comput. Geom.*, page 406, ACM Press, 2014.

[CJ14a] S.-W. Cheng and J. Jin. Approximate shortest descending paths. *SIAM J. Comput.*, 43:410–428, 2014.

[CJ14b] S.-W. Cheng and J. Jin. Shortest paths on polyhedral surfaces and terrains. In *Proc. 46th ACM Sympos. Theory Comput.*, pages 373–382, 2014.

[CJ16] T.H. Chan and S.H. Jiang. Reducing curse of dimensionality: Improved PTAS for TSP (with neighborhoods) in doubling metrics. In *Proc. 27th ACM-SIAM Sympos. Discrete Algorithms*, pages 754–765, 2016.

[CJV15] S.-W. Cheng, J. Jin, and A. Vigneron. Triangulation refinement and approximate shortest paths in weighted regions. In *Proc. 26th ACM-SIAM Sympos. Discrete Algorithms*, pages 1626–1640, 2015.

[CK95] P.B. Callahan and S.R. Kosaraju. A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. *J. ACM*, 42:67–90, 1995.

[CK96] D.Z. Chen and K.S. Klenk. Rectilinear short path queries among rectangular obstacles. *Inform. Process. Lett.*, 57:313–319, 1996.

[CK04] C. Chekuri and A. Kumar. Maximum coverage problem with group budget constraints and applications. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, vol. 3122 of *LNCS*, pages 72–83, Springer, Berlin, 2004.

[CKT00] D.Z. Chen, K.S. Klenk, and H.-Y.T. Tu. Shortest path queries among weighted obstacles in the rectilinear plane. *SIAM J. Comput.*, 29:1223–1246, 2000.

[CKV87] K.L. Clarkson, S. Kapoor, and P.M. Vaidya. Rectilinear shortest paths through polygonal obstacles in $O(n(\log n)^2)$ time. In *Proc. 3rd Sympos. Comput. Geom.*, pages 251–257, ACM Press, 1987.

[CL00] A. Czumaj and A. Lingas. Fast approximation schemes for Euclidean multi-connectivity problems. In *Proc. 27th Int. Conf. on Automata, Languages and Prog.*, vol. 1853 of *LNCS*, pages 856–868, Springer, Berlin, 2000.

[Cla87] K.L. Clarkson. Approximation algorithms for shortest path motion planning. In *Proc. 19th Sympos. Theory Comput.*, pages 56–65, ACM Press, 1987.

[CLM05] B. Chazelle, D. Liu, and A. Magen. Sublinear geometric algorithms. *SIAM J. Comput.*, 35:627–646, 2005.

[CLMS02] S. Cabello, Y. Liu, A. Mantler, and J. Snoeyink. Testing homotopy for paths in the plane. In *Proc. 18th Sympos. Comput. Geom.*, pages 160–169, ACM Press, 2002.

[CM99] Y.-J. Chiang and J.S.B. Mitchell. Two-point Euclidean shortest path queries in the plane. In *Proc. 10th ACM-SIAM Sympos. Discrete Algorithms*, pages 215–224, 1999.

[CN88] W.-P. Chin and S. Ntafos. Optimum watchman routes. *Inform. Process. Lett.*, 28:39–44, 1988.

[CN91] W.-P. Chin and S. Ntafos. Shortest watchman routes in simple polygons. *Discrete Comput. Geom.*, 6:9–31, 1991.

[CNVW08] S.-W. Cheng, H.-S. Na, A. Vigneron, and Y. Wang. Approximate shortest paths in anisotropic regions. *SIAM J. Comput.*, 38:802–824, 2008.

[CNVW10] S.-W. Cheng, H.-S. Na, A. Vigneron, and Y. Wang. Querying approximate shortest paths in anisotropic regions. *SIAM J. Comput.*, 39:1888–1918, 2010.

[CR87] J. Canny and J.H. Reif. New lower bound techniques for robot motion planning problems. In *Proc. 28th IEEE Sympos. Found. Comput. Sci.*, pages 49–60, 1987.

[CSY95] J. Choi, J. Sellen, and C.K. Yap. Precision-sensitive Euclidean shortest path in 3-space. In *Proc. 11th Sympos. Comput. Geom.*, pages 350–359, ACM Press, 1995.

[CSY97] J. Choi, J. Sellen, and C.K. Yap. Approximate Euclidean shortest paths in 3-space. *Internat. J. Comput. Geom. Appl.*, 7:271–295, 1997.

[CT97] Y.-J. Chiang and R. Tamassia. Optimal shortest path and minimum-link path queries between two convex polygons inside a simple polygonal obstacle. *Internat. J. Comput. Geom. Appl.*, 7:85–121, 1997.

[CW13] D.Z. Chen and H. Wang. L_1 shortest paths among polygonal obstacles in the plane. In *Proc. 30th Sympos. Theoret. Aspects Comp. Sci.*, pages 293–304, Schloss Dagstuhl, 2013.

[CW15a] D.Z. Chen and H. Wang. Computing shortest paths among curved obstacles in the plane. *ACM Trans. Algorithms*, 11:26, 2015.

[CW15b] D.Z. Chen and H. Wang. A new algorithm for computing visibility graphs of polygonal obstacles in the plane. *J. Comput. Geom.*, 6:316–345, 2015.

[CY95] J. Choi and C.K. Yap. Rectilinear geodesics in 3-space. In *Proc. 11th Sympos. Comput. Geom.*, pages 380–389, ACM Press, 1995.

[CY96] J. Choi and C.K. Yap. Monotonicity of rectilinear geodesics in d-space. In *Proc. 12th Sympos. Comput. Geom.*, pages 339–348, ACM Press, 1996.

[DCG$^+$14] J.-L. De Carufel, C. Grimm, A. Maheshwari, M. Owen, and M. Smid. A note on the unsolvability of the weighted region shortest path problem. *Comput. Geom.*, 47:724–727, 2014.

[DELM03] M. Dror, A. Efrat, A. Lubiw, and J.S.B. Mitchell. Touring a sequence of polygons. In *Proc. 35th ACM Sympos. Theory Comput.*, pages 473–482, 2003.

[DM03] A. Dumitrescu and J.S.B. Mitchell. Approximation algorithms for TSP with neighborhoods in the plane. *J. Algorithms*, 48:135–159, 2003.

[DO08] M. Dror and J.B. Orlin. Combinatorial optimization with explicit delineation of the ground set by a collection of subsets. *SIAM J. Discrete Math.*, 21:1019–1034, 2008.

[DT16] A. Dumitrescu and C.D. Tóth. The traveling salesman problem for lines, balls, and planes. *ACM Trans. Algorithms*, 12:43, 2016.

[DX95] B.R. Donald and P.G. Xavier. Provably good approximation algorithms for optimal kinodynamic planning for Cartesian robots and open chain manipulators. *Algorithmica*, 14:480–530, 1995.

[DXCR93] B. Donald, P. Xavier, J. Canny, and J. Reif. Kinodynamic motion planning. *J. ACM*, 40:1048–1066, 1993.

[EFS06] K. Elbassioni, A.V. Fishkin, and R. Sitters. On approximating the TSP with intersecting neighborhoods. In *Proc. Int. Sympos. Algorithms Comput.*, vol. 4288 of *LNCS*, pages 213–222, Springer, Berlin, 2006.

[EHP$^+$15] S. Eriksson-Bique, J. Hershberger, V. Polishchuk, B. Speckmann, S. Suri, T. Talvitie, K. Verbeek, and H. Yıldız. Geometric k shortest paths. In *Proc. 26th ACM-SIAM Sympos. Discrete Algorithms*, pages 1616–1625, 2015.

[EKL06] A. Efrat, S.G. Kobourov, and A. Lubiw. Computing homotopic shortest paths efficiently. *Comput. Geom.*, 35:162–172, 2006.

[EM94] H. ElGindy and P. Mitra. Orthogonal shortest route queries among axis parallel rectangular obstacles. *Internat. J. Comput. Geom. Appl.*, 4:3–24, 1994.

[Fek00] S.P. Fekete. On simple polygonalizations with optimal area. *Discrete Comput. Geom.*, 23:73–110, 2000.

[FHR07] J. Fakcharoenphol, C. Harrelson, and S. Rao. The k-traveling repairmen problem. *ACM Trans. Algorithms*, 3:40, 2007.

[FMS12] S.P. Fekete, J.S.B. Mitchell, and C. Schmidt. Minimum covering with travel cost. *J. Comb. Optim.*, 24:32–51, 2012.

[FT87] M.L. Fredman and R.E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, 34:596–615, 1987.

[Gar05] N. Garg. Saving an epsilon: a 2-approximation for the k-mst problem in graphs. In *Proc. 37th ACM Sympos. Theory Comput.*, pages 396–402, 2005.

[GH89] L.J. Guibas and J. Hershberger. Optimal shortest path queries in a simple polygon. *J. Comput. Syst. Sci.*, 39:126–152, 1989.

[GHL+87] L.J. Guibas, J. Hershberger, D. Leven, M. Sharir, and R.E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2:209–233, 1987.

[GHT89] L. Gewali, S. Ntafos, and I.G. Tollis. Path planning in the presence of vertical obstacles. Technical report, Computer Science, University of Texas at Dallas, 1989.

[GK97] M.X. Goemans and J. Kleinberg. An improved approximation ratio for the minimum latency problem. *Math. Prog.*, 82:111–124, 1999.

[GL99] J. Gudmundsson and C. Levcopoulos. A fast approximation algorithm for TSP with neighborhoods. *Nord. J. Comput.*, 6:469, 1999.

[GLNS08] J. Gudmundsson, C. Levcopoulos, G. Narasimhan, and M. Smid. Approximate distance oracles for geometric spanners. *ACM Trans. Algorithms*, 4:10, 2008.

[GM91] S.K. Ghosh and D.M. Mount. An output-sensitive algorithm for computing visibility graphs. *SIAM J. Comput.*, 20:888–910, 1991.

[GMMN90] L. Gewali, A. Meng, J.S.B. Mitchell, and S. Ntafos. Path planning in $0/1/\infty$ weighted regions with applications. *ORSA J. Comput.*, 2:253–272, 1990.

[GMS08] H. Guo, A. Maheshwari, and J.-R. Sack. Shortest path queries in polygonal domains. In *Proc. 4th Int. Conf. Algorithmic Aspects Inform. Management*, vol. 5034 of *LNCS*, pages 200–211, Springer, Berlin, 2008.

[Got15] L.-A. Gottlieb. A light metric spanner. In *Proc. 56th Sympos. Found. Comp. Sci.*, pages 759–772, 2015.

[GT97] M.T. Goodrich and R. Tamassia. Dynamic ray shooting and shortest paths in planar subdivisions via balanced geodesic triangulations. *J. Algorithms*, 23:51–73, 1997.

[Har99a] S. Har-Peled. Approximate shortest paths and geodesic diameters on convex polytopes in three dimensions. *Discrete Comput. Geom.*, 21:216–231, 1999.

[Har99b] S. Har-Peled. Constructing approximate shortest path maps in three dimensions. *SIAM J. Comput.*, 28:1182–1197, 1999.

[Har11] S. Har-Peled. *Geometric Approximation Algorithms.* Vol. 173 of *Math. Surveys Monogr.*, AMS, Providence, 2011.

[Har15] S. Har-Peled. Shortest path in a polygon using sublinear space. *J. Comput. Geom.*, 7:19–45, 2015.

[Her89] J. Hershberger. An optimal visibility graph algorithm for triangulated simple polygons. *Algorithmica*, 4:141–155, 1989.

[Her91] J. Hershberger. A new data structure for shortest path queries in a simple polygon. *Inform. Process. Lett.*, 38:231–235, 1991.

[Her95] J. Hershberger. Optimal parallel algorithms for triangulated simple polygons. *Internat. J. Comput. Geom. Appl.*, 5:145–170, 1995.

[HS94] J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. *Comput. Geom.*, 4:63–98, 1994.

[HS97] J. Hershberger and S. Suri. Matrix searching with the shortest path metric. *SIAM J. Comput.*, 26:1612–1634, 1997.

[HS98] J. Hershberger and S. Suri. Practical methods for approximating shortest paths on a convex polytope in $\mathbb{R}^3$. *Comput. Geom.*, 10:31–46, 1998.

[HS99] J. Hershberger and S. Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM J. Comput.*, 28:2215–2256, 1999.

[HSY13] J. Hershberger, S. Suri, and H. Yıldız. A near-optimal algorithm for shortest paths among curved obstacles in the plane. In *Proc. 29th Sympos. Comput. Geom.*, pages 359–368, ACM Press, 2013.

[IK15] R. Inkulu and S. Kapoor. A polynomial time algorithm for finding an approximate shortest path amid weighted regions. Preprint, `arXiv:1501.00340`, 2015.

[Kap99] S. Kapoor. Efficient computation of geodesic shortest paths. In *Proc. 31st ACM Sympos. Theory Comput.*, pages 770–779, 1999.

[KMM97] S. Kapoor, S.N. Maheshwari, and J.S.B. Mitchell. An efficient algorithm for Euclidean shortest paths among polygonal obstacles in the plane. *Discrete Comput. Geom.*, 18:377–383, 1997.

[KKP11] D.G. Kirkpatrick, I. Kostitsyna, and V. Polishchuk. Hardness results for two-dimensional curvature-constrained motion planning. In *Proc. 23rd Canad. Conf. Comput. Geom.*, 2011.

[KLPS16] I. Kostitsyna, M. Löffler, V. Polishchuk, and F. Staals. On the complexity of minimum-link path problems. In *Proc. 32nd Sympos. Comput. Geom.*, vol. 51 of *LIPIcs*, article 49, Schloss Dagstuhl, 2016.

[KO13] H.-S. Kim and O. Cheong. The cost of bounded curvature. *Comput. Geom.*, 46:648–672, 2013.

[LMS01] M. Lanthier, A. Maheshwari, and J.-R. Sack. Approximating shortest paths on weighted polyhedral surfaces. *Algorithmica*, 30:527–562, 2001.

[LP84] D.T. Lee and F.P. Preparata. Euclidean shortest paths in the presence of rectilinear barriers. *Networks*, 14:393–410, 1984.

[LSD00] A. Maheshwari, J.-R. Sack, and H.N. Djidjev. Link distance problems. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 519–558, Elsevier, Amsterdam, 2000.

[LYW96] D.T. Lee, C.D. Yang, and C.K. Wong. Rectilinear paths among rectilinear obstacles. *Discrete Appl. Math.*, 70:185–215, 1996.

[Mit13] J.S.B. Mitchell. Approximating watchman routes. In *Proc. 24th ACM-SIAM Sympos. Discrete Algorithms*, pages 844–855, 2013.

[Mit89] J.S.B. Mitchell. An optimal algorithm for shortest rectilinear paths among obstacles. In *Proc. 1st Canad. Conf. Comput. Geom.*, page 22, 1989.

[Mit91] J.S.B. Mitchell. A new algorithm for shortest paths among obstacles in the plane. *Ann. Math. Artif. Intell.*, 3:83–106, 1991.

[Mit92] J.S.B. Mitchell. L_1 shortest paths among polygonal obstacles in the plane. *Algorithmica*, 8:55–88, 1992.

[Nit96] J.S.B. Mitchell. Shortest paths among obstacles in the plane. *Internat. J. Comput. Geom. Appl.*, 6:309–332, 1996.

[Mit99] J.S.B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. *SIAM J. Comput.*, 28:1298–1309, 1999.

[Mit00] J.S.B. Mitchell. Geometric shortest paths and network optimization. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 633–701, Elsevier, Amsterdam, 2000.

[Mit07] J.S.B. Mitchell. A PTAS for TSP with neighborhoods among fat regions in the plane. In *Proc. 18th ACM-SIAM Sympos. Discrete Algorithms*, pages 11–18, 2007.

[Mit10] J.S.B. Mitchell. A constant-factor approximation algorithm for TSP with pairwise-disjoint connected neighborhoods in the plane. In *Proc. 26th Sympos. Comput. Geom.*, pages 183–191, ACM Press, 2010.

[Mit15] J.S.B. Mitchell. Approximation schemes for geometric network optimization problems. M.-Y. Kao, editor, *Encyclopedia of Algorithms*, Springer, Berlin, 2015.

[MM95] C. Mata and J.S.B. Mitchell. Approximation algorithms for geometric tour and network design problems. In *Proc. 11th Sympos. Comput. Geom.*, pages 360–369, ACM Press, 1995.

[MM97] C. Mata and J.S.B. Mitchell. A new algorithm for computing shortest paths in weighted planar subdivisions. In *Proc. 13th Sympos. Comput. Geom.*, pages 264–273, ACM Press, 1997.

[MMP87] J.S.B. Mitchell, D.M. Mount, and C.H. Papadimitriou. The discrete geodesic problem. *SIAM J. Comput.*, 16:647–668, 1987.

[MP91] J.S.B. Mitchell and C.H. Papadimitriou. The weighted region problem: finding shortest paths through a weighted planar subdivision. *J. ACM*, 38:18–73, 1991.

[MPA92] J.S.B. Mitchell, C. Piatko, and E.M. Arkin. Computing a shortest k-link path in a polygon. In *Proc. 33rd IEEE Sympos. Found. Comput. Sci.*, pages 573–582, 1992.

[MPS14] J.S.B. Mitchell, V. Polishchuk, and M. Sysikaski. Minimum-link paths revisited. *Comput. Geom.*, 47:651–667, 2014.

[MPSW15] J.S.B. Mitchell, V. Polishchuk, M. Sysikaski, and H. Wang. An optimal algorithm for minimum-link rectilinear paths in triangulated rectilinear domains. In *Proc. 42nd Int. Conf. Automata, Languages, Progr.*, Part I, vol. 9134 of *LNCS*, pages 947–959, Springer, Berlin, 2015.

[MRW92] J.S.B. Mitchell, G. Rote, and G. Woeginger. Minimum-link paths among obstacles in the plane. *Algorithmica*, 8:431–459, 1992.

[MS04] J.S.B. Mitchell and M. Sharir. New results on shortest paths in three dimensions. In *Proc. 20th Sympos. Comput. Geom.*, pages 124–133, ACM Press, 2004.

[Muc14] M. Mucha. $\frac{13}{9}$-approximation for graphic TSP. *Theory Comput. Syst.*, 55:640–657, 2014.

[OBA16] E. Oh, L. Barba, and H.-K. Ahn. The farthest-point geodesic Voronoi diagram of points on the boundary of a simple polygon. In *Proc. 32nd Sympos. Comput. Geom.*, vol. 51 of *LIPIcs*, article 56, Schloss Dagstuhl, 2016.

[O'R99] J. O'Rourke. Computational geometry column 35. *Internat. J. Comput. Geom. Appl.*, 9:513–516, 1999.

[Pap85] C.H. Papadimitriou. An algorithm for shortest-path motion in three dimensions. *Inform. Process. Lett.*, 20:259–263, 1985.

[PL98] E. Papadopoulou and D.T. Lee. A new approach for the geodesic Voronoi diagram of points in a simple polygon and other restricted polygonal domains. *Algorithmica*, 20:319–352, 1998.

[PV96a] M. Pocchiola and G. Vegter. Topologically sweeping visibility complexes via pseudo-triangulations. *Discrete Comput. Geom.*, 16:419–453, 1996.

[PV96b] M. Pocchiola and G. Vegter. The visibility complex. *Internat. J. Comput. Geom. Appl.*, 6:279–308, 1996.

[PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York, 1985.

[PS11] V. Polishchuk and M. Sysikaski. Faster algorithms for minimum-link paths with restricted orientations. In *Proc. 12th Sympos. Algorithms Data Structures*, vol. 6844 of *LNCS*, pages 655–666, Springer, Berlin, 2011.

[RS98] S.B. Rao and W.D. Smith. Approximating geometrical graphs via "spanners" and "banyans." In *Proc. 30th ACM Sympos. Theory Comput.*, pages 540–550, 1998.

[RDN07] S. Roy, S. Das, and S.C. Nandy. Shortest monotone descent path problem in polyhedral terrain. *Comput. Geom.*, 37:115–133, 2007.

[RT94] J.H. Reif and S.R. Tate. Approximate kinodynamic planning using L_2-norm dynamic bounds. *Comput. Math. Appl.*, 27:29–44, 1994.

[RW98] J. Reif and H. Wang. The complexity of the two dimensional curvature-constrained shortest-path problem. In *Proc. 3rd Workshop Algorithmic Found. Robot.*, pages 49–57, A.K. Peters, Natick, 1998.

[RW00] J. Reif and H. Wang. Non-uniform discretization for kinodynamic motion planning and its applications. *SIAM J. Comput.*, 30:161–190, 2000.

[Sch96] S. Schuierer. An optimal data structure for shortest rectilinear path queries in a simple rectilinear polygon. *Internat. J. Comput. Geom. Appl.*, 6:205–226, 1996.

[Sch07] Y. Schreiber. Shortest paths on realistic polyhedra. In *Proc. 23rd Sympos. Comput. Geom.*, pages 74–83, ACM Press, 2007.

[Sel95] J. Sellen. Direction weighted shortest path planning. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 1970–1975, 1995.

[Sha87] M. Sharir. On shortest paths amidst convex polyhedra. *SIAM J. Comput.*, 16:561–572, 1987.

[Sit14] R. Sitters. Polynomial time approximation schemes for the traveling repairman and other minimum latency problems. In *Proc. 25th ACM-SIAM Sympos. Discrete Algorithms*, pages 604–616, 2014.

[Sla97] P. Slavík. The errand scheduling problem. Technical report 97-2, Department of Computer Science, SUNY, Buffalo, 1997.

[SS06] S. Safra and O. Schwartz. On the complexity of approximating TSP with neighborhoods and related problems. *Comput. Complexity*, 14:281–307, 2006.

[SS08] Y. Schreiber and M. Sharir. An optimal-time algorithm for shortest paths on a convex polytope in three dimensions. *Discrete Comput. Geom.*, 1:500–579, 2008.

[SSK+05] V. Surazhsky, T. Surazhsky, D. Kirsanov, S.J. Gortler, and H. Hoppe. Fast exact and approximate geodesics on meshes. *ACM Trans. Graph.*, 24:553–560, 2005.

[Sur86] S. Suri. A linear time algorithm for minimum link paths inside a simple polygon. *Comput. Vision Graph. Image Process.*, 35:99–110, 1986.

[Sur90] S. Suri. On some link distance problems in a simple polygon. *IEEE Trans. Robot. Autom.*, 6:108–113, 1990.

[VA99] K.R. Varadarajan and P.K. Agarwal. Approximating shortest paths on a nonconvex polyhedron. *SIAM J. Comput.*, 30:1321–1340, 1999.

[Wan16] H. Wang. On the geodesic centers of polygonal domains. In *Proc. 24th Eur. Sympos. Algorithms*, vol. 57 of *LIPIcs*, article 77, Schloss Dagstuhl, 2016.

Preliminary version (July 17, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.