Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Food Engineering 70 (2005) 293–298
JOURNAL OF FOOD ENGINEERING
www.elsevier.com/locate/jfoodeng

Computing shortest paths for transportation of hazardous materials in continuous spaces

J.M. Díaz-Báñez [a,*], F. Gómez [b], G.T. Toussaint [c]

[a] Departamento de Matemática Aplicada II, Universidad de Sevilla, Virgen de Africa 7, Sevilla 410l1, Spain
[b] Departamento de Matemática Aplicada, Universidad Politécnica de Madrid, Madrid, Spain
[c] Schools of Computer Science, McGill University, Montreal, Canada

Received 15 October 2003; received in revised form 2 April 2004; accepted 13 May 2004
Available online 21 November 2004

Abstract

This paper is concerned about the problem of locating a path for a shipment of dangerous goods between a pre-specified origin–destination pair on the plane. Minimization of risks during the transportation and cost of the path have been taken into account. The formulation of the main problem considered in this paper is the following: Given a source point a, a destination point b, a set S of demand sites (points in the plane) and a positive value l, we want to compute a path connecting a and b with length at most l such that the minimum distance to the points in S is maximized. We propose an approximate algorithm based on the bisection method to solve this problem. Our technique reduces the optimization problem to a *decision problem*, where one needs to compute the shortest path such that the minimum distance to the demand points is not smaller that a certain amount r. To solve the decision task, we transform the problem to the computation of the shortest path avoiding obstacles. This approach provides efficient algorithms to compute shortest obnoxious paths under several kinds of distances.

Keywords: Hazardous material transportation; Extensive facility location; Analysis of algorithms: Computational complexity

1. Introduction

It has long been recognized that the management of hazardous material (hazmat for short) is an extremely complex issue involving a multitude of environmental, engineering, economic, social and political concerns. There is general agreement that the shipment of these type of materials is sizable and growing. Thus, the development of planning criteria for the minimization of industrial risk requires the application of efficient techniques. In all processes that transform raw material into final products by-products are unfailingly generated. Some of those by-products are dangerous and have to be taken to specialized places where they are processed. An immediate problem one can think of is that of finding routes as far as possible from the surrounding cities. Also, in food industries, the transportation of dangerous products arises, for instance, in the management of chemical substances necessary in the production.

Minimization of risks for such problems has been extensively studied when the underlying space is a network (Erkut & Verter, 1995, 1998; Verter & Erkut, 1996). However, little progress has been reported in continuous spaces. Because accidents involving hazardous materials may occur during transportation, the continuous context must be take into account. In some situations, a risk reduction could pass through a redefinition of the transport system and new better paths should be constructed.

* Corresponding author. Tel.: +34 95 4552853; fax: +34 95 4282777.
E-mail addresses: dbanez@us.es (J.M. Díaz-Báñez), fmartin@eui.upm.es (F. Gómez), godfried@cs.mcgill.ca (G.T. Toussaint).

0260-8774/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jfoodeng.2004.05.076

This paper address the design of an efficient algorithm for locating a path for a shipment of dangerous goods between a pre-specified origin–destination pair on the plane. In order to minimize the risks, the minimum distance between the existing installations and the path must be maximized. This task can be considered as an obnoxious facility location problem of dimensional structures in the sense that we are interested in the placement of an undesirable facility modelled by a path amidst existing facilities. See Díaz-Báñez, Mesa, & Shöbel (2004) for a recent survey on the current state-of-art of these problems. It is clear that the location of a such trajectory must be constrained, as otherwise the route may be simply removed to infinity. Drezner & Wesolowsky (1989) provide an approximate algorithm for calculating an obnoxious route within a given region. We also consider an additional constraint on the length of the path (clearly relevant to minimizing the costs).

This class of operations research problems has also been considered in the computational geometry community. In the papers by Follert, Schömer, Sellen, Smid, & Thiel (1997), Barcia, Díaz-Báñez, Lozano, & Ventura (2003), Díaz-Báñez, Hurtado, Meijer, Rappaport, & Sellarès (2003), among others, some geometric techniques have been proven successful to efficiently compute a solution to obnoxious location problems.

The rest of the paper is organized as follows. In Section 2, we present the formal definition of the problem. In Section 3, a general approach and the key ideas are given. The main problem is reduced to some particular geometric problems. Section 4 is devoted to put all pieces together to solve the problem with an elegant and efficient algorithm that yields an approximate solution. Finally, Section 5 contains some concluding remarks.

2. The obnoxious short path problem

The *obnoxious short path problem* is formulated as follows. We are given two points, a and b, to be called the source point and the destination point, respectively; along with them, we have a set of distinct sites $S = \{s_1, \ldots, s_n\}$ $(a, b, \notin S)$. Let Q be any path that joins a and b (but not necessarily a polygonal one) and denote its length by $L(Q)$. Then, we want to find a path Q that holds two conditions: (1) the minimum distance from Q to S is maximized and; (2) the length $L(Q)$ is at most a given bound l.

As expected, here the distance is the usual Euclidean distance and $d(Q, S)$ means $\min\{d(q, s_i) | q \in Q, i = 1, \ldots, n\}$. The role of l is that of a constraint on the length of the path (it could be interpreted as a budget constraint, for example). For obvious reasons, we will require l to be greatest than $d(a, b)$. In the Fig. 1 we can see all the elements that form part of the problem.

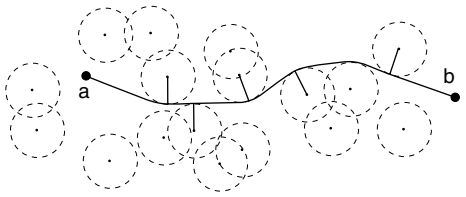

Fig. 1. Posing the problem.

Note that if we removed the constraint of the length, then the solution would be trivial: the path would go to infinity, since there is where the minimum distance is maximized. Of course, in that case, there would be infinitely many solutions. But even when the constraint on the length is imposed, the path Q may not be unique. For example, if line segment ab and $CH(S)$ are disjoint, then we do not have uniqueness. Note that in this case the sites have to do very little with the obnoxious path, since the distance function only depends on the distance to a or b. In order to avoid those cases that make little sense, we will require that line segment ab intersect the interior of $CH(S)$ so the sites have influence on the distance function. Once this condition is satisfied, then Q is unique. To see this, place circles of radius r centered at sites s_i, $i = 1, \ldots, n$ and consider all paths joining a and b not intersecting the set of circles. Among all those paths, only the shortest path is a candidate for solution; all the others are not because the radius r can be infinitesimally increased without reached the bound l for the length and, therefore, they are not optimum. This remark shows that the shortest path among the circles with length equal to l is the solution.

However, for this problem we are not interested in finding the exact solution, but an approximate solution, a situation that often appears in practice. As a matter of fact, there are many senses in which we could define the term "approximate solution" and each would give place to a new problem. We here define the approximation in terms of the length. More precisely, a path P joining a and b is said to be a tight path if it reaches the maximum of the minimum distance for all the paths whose length is $L(P)$. Given two tight paths P, P_1, we say that P_1 is an ε-approximation of P if $|L(P) - L(P_1)| < \varepsilon$. Note that with this definition of approximation we can have two very close paths whose topology is quite different. In the Fig. 2, we can see how a little increment on the radius of the circle can change the topology of the path.

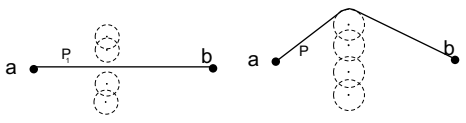

Fig. 2. Two close paths with different topologies.

3. A general method

The idea of computing the solution with shortest length lead us to the following algorithm (for the moment, it is a conceptual algorithm; we will later give the technical details). Place a circle of radius r at each site s_i, $i = 1, \ldots, n$. The desired path must go among the union of those circles if r is to be equal to the maximum of the minimum distance. Next, compute the union of the arrangements of circles and, from it, compute the contour of the union. Such union will be formed by many connected components (the number of components ranges from 1 to n). Find the one that contains a and b. Finally, obtain the shortest path Q between a and b among the circular obstacles. If $L(Q)$ is equal to the bound l, then we have computed a solution of the problem where r is the maximum achieved.

Several conclusions can be drawn from the description of this algorithm. One can see that the solution is composed of line segments and arcs of circles. Another question is what happens if a and b lie on different connected components of the contour of the union. Then no path between a and b can be constructed (refer to Fig. 3). This, in turn, means that the radius r cannot be the maximum of the minimum distance; it is too big. So therefore, we should reduce r. Here we are inspired by bisection algorithms for *rootfinding* used in Numerical Analysis. We take $r/2$ as the new radius for the circles and re-compute the contour of the union. If a and b are not in the same component, we consider $r/4$. This process is carried on until a and b can be connected. It always exists a radius r_1 such that a and b are path-connected. For a zero radius the statement is obvious as for the number of components is one. This number does not change at least until three circles intersect each other and hence there exists a whole interval for the radius that guarantees the existence of r_1. Once we know that a and b are path-connected, we can continue to apply the bisection algorithm in order to find an ε-approximation of the solution Q.

In order to formalize the above ideas, we introduce the function $F(r)$. Let δ be $\min\{d(a,S), d(b,S)\}$. For each $r \in [0,\delta]$, let P_r be the shortest path from a to b among the obstacles formed by the circles of radius r centered at sites s_i, $i = 1, \ldots, n$. We then define $F(r)$ as $L(P_r) - l$. Function $F(r)$ has the following properties:

- $F(r)$ is a strictly increasing function.
- $F(r)$ is a piecewise left continuous function. The number of discontinuity points depends on the topology of the arrangement of circles.
- $F(0) = d(a,b) - l < 0$.

If $F(\delta) < 0$, then l is too big and the problem, as stated, makes little sense. For the sake of reasonableness, from now on, we will assume that $F(\delta) > 0$. The first property of $F(r)$ is proved in the following Lemma.

Lemma 1. *$F(r)$ is a strictly increasing function.*

Proof. Let r_1, r_2 be two radius with $r_1 < r_2$. We will proceed by contradiction. Suppose that $L(P_{r_1}) > L(P_{r_2})$ and consider P_{r_2}. Given that $r_1 < r_2$, path P_{r_2} cannot touch any ball of radius r_1 centered at the sites. Therefore, its length is smaller than $L(P_{r_1})$ and that contradicts the fact that P_{r_1} is the shortest path for radius r_1. □

Note that we cannot strictly apply a bisection algorithm for finding r' such that $F(r') = 0$, that is, $L(P_{r'}) = l$. The reason is that $F(r)$ is not continuous. Let us carry out a finer analysis of the situation. In the Fig. 4 we see a typical function $F(r)$.

If $F(r)$ intersects the line $r = 0$, then we know we can apply the bisection algorithm (case (1)), since there exists a continuous piece that allows such an application. Otherwise, line $r = 0$ is at a discontinuity point and the best we can do is choosing the values of the function at that point (case (2)). We will later discuss how to deal with the latter case. For now, we focus ourselves on the former case.

If line $r = 0$ crosses a piece of $F(r)$, then the bisection algorithm can be applied to that piece. Let $[r_0, r_1]$ the interval defining that piece. We define the following sequences:

$$s_m = \begin{cases} s_0 = r_0 & \\ s_m = s_{m-1} & \text{if } F\left(\frac{s_{m-1}+t_{m-1}}{2}\right) > 0, \\ s_m = \frac{s_{m-1}+t_{m-1}}{2} & \text{if } F\left(\frac{s_{m-1}+t_{m-1}}{2}\right) < 0 \end{cases}$$

$$t_m = \begin{cases} t_0 = r_1 & \\ t_m = t_{m-1} & \text{if } F\left(\frac{s_{m-1}+t_{m-1}}{2}\right) < 0 \\ t_m = \frac{s_{m-1}+t_{m-1}}{2} & \text{if } F\left(\frac{s_{m-1}+t_{m-1}}{2}\right) > 0 \end{cases}$$

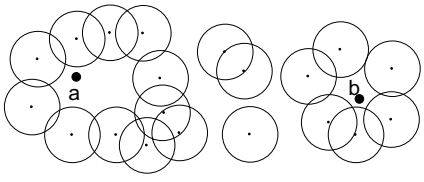

Fig. 3. a and b are in different connected components.

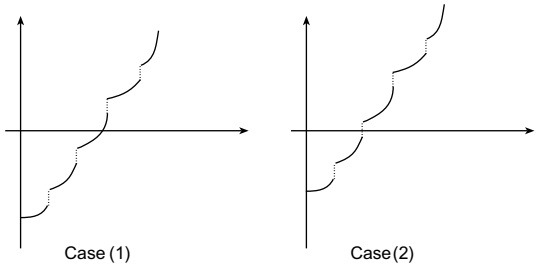

Fig. 4. Function $F(r)$.

Sequences s_m, t_m have the following properties:

- s_m and t_m are converging sequences with a common limit.
- The common limit is the unique zero r' of $F(r)$ in the interval $[r_0, r_1]$, hence, s_m, $t_m \to r'$ as m tends to infinity.
- Given $\varepsilon > 0$, the number m of steps to achieve an error less than ε is given by the following expression:

$$m \geqslant \left\lceil \log_2\left(\frac{r_1 - r_0}{\varepsilon}\right)\right\rceil.$$

Although the conceptual algorithm seems simple and straightforward, there are many technical details to be worked out; particularly, it should be taken care over the presence of circular arcs. Note that the problem of finding the shortest path between a and b avoiding polygonal obstacles is a well-known problem in Computational Geometry. Usually, the input of such algorithms is a set of non-intersecting obstacles. Prior to applying a shortest path algorithm we have to compute the contour of the union of the circles. For polygons, there exists a nice plane-sweep algorithm by Nievergelt & Preparata (1982); unfortunately, it cannot be easily generalized to circles. We then have to take over a new idea that allows us to handle circles in a elegant way. That idea consists of using Voronoi diagrams with Laguerre geometry.

3.1. Voronoi diagrams with Laguerre geometry

The natural way of working with circles is turning to Laguerre geometry. The distance between a point p and a circle C is defined as the length of the line segment pp_i, where p_i is such that pp_i is tangent to C. Imai, Iri, & Murota (1985) have solved the following problems concerning circles: (1) point location problem: given a set of circles in the plane, determine whether a given point is contained in their union or not; (2) partition the set of circles into connected components (that is, give the graph of the intersection of the circles); (3) compute the contour of the union of the circles. In particular, problem (3) was solved in $\Theta(n \log n)$, where n is the number of circles. Once the contour of the union is given, the question of whether a and b are in the same connected component can be answered in $O(\log n)$ time.

Their approach consists of computing the Voronoi diagram and, from it, solving all the problems mentioned above. The construction of the Voronoi diagram is based on the divide-and-conquer technique in a way similar to that of Shamos & Hoey (1975). Of course, there are important differences in the algorithm, mainly in how the merging process of the subdiagrams is carried out. We refer the reader to Imai et al. (1985) and Okabe, Boots, & Sugihara (1992) for finding all the technical details.

3.2. Shortest paths avoiding obstacles

Finding a shortest path among polygonal obstacles is a classical problem in Computational Geometry (Mitchell, 2000). A straightforward algorithm for computing a shortest path in the plane is to construct a graph containing one vertex for each obstacle vertex and one edge for each pair of obstacle vertices that are mutually visible and then searching this graph using Dijkstra's algorithm. In our case, the boundary is formed by circular arcs, but still we can extend the idea of the visibility graph to obtain a new graph, the so-called tangent graph. This graph has the circular arcs of the boundary and common tangents of circular arcs as edges. Again, searching on this graph we are able of finding the shortest path. Unfortunately, the time complexity turns to be $O(n^2 \log n)$ as for the graph possesses $O(n^2)$ edges. We will show how to lower the time complexity of this part of the problem.

Storer & Reif (1994) have an algorithm that allows us to compute the shortest path among circular obstacles in $O(kn)$, where n is the total number of vertices and k is the number of obstacles (islands, as they call them, or connected components). For the sake of completeness, we review it briefly here as well. We should note that their algorithm requires to have the Voronoi diagram of the obstacle space in advance to achieve that time complexity (computing such a diagram takes $O(n \log n)$ time). The way the algorithm works is as follows. First, they solve the *shortest path problem with no islands* and for polygonal obstacles by defining a *shorcut* operation, which is based on the following remark. Suppose we are given a convex vertex y whose neighboring vertices are x and z; if neither the destination point nor the source point are inside the triangle (x, y, z), then we can replace (x, y, z) by (x, z) (here it is the shorcut operation). An immediate problem arising is that line segment xz can intersect the set of obstacles. In this case, the best it can be done is bending around the intruding walls as closely as possible. By taking some care over the definition of this shortcut operation, it can be given an algorithm that outputs the shortest path with no islands. More details can be found in the Algorithm 3.2, p. 997, in Storer & Reif (1994).

After that, they permit the presence of islands and reduce this case to the previous one by applying an island merging algorithm. The idea behind this island merging algorithm is to successively link the islands of the obstacle space together with "safe" paths. Safe paths means paths connecting two obstacles so that they do not cross the shortest path between the source point and the destination point. In order to merge the islands some extra points must be added to the obstacle space. Storer and Reif prove that not more than $O(k)$ points are needed.

Finally, they consider the problem of finding the minimal movement of a disc. Their approach consists of

building a new obstacle spaces in which movement of a point is equivalent to movement of a disc. Given a polygonal obstacle space and a disc of radius r, by "padding" each obstacle a distance r, they obtain a new obstacle space, some of whose edges are circular arcs. A shortest path among those circular obstacles gives the minimal movement of a disc of radius r.

After the Storer and Reif's algorithm is executed, a pseudo-triangulation is produced with the following properties:

1. All internal faces are rounded triangles that consist of two line segments and one arc segment.
2. Associated with each vertex of the pseudo-triangulation are two pointers: (a) dist(v): is the length of the shortest path from v to a; (b) adj(v): is a vertex on the perimeter of the face containing v such that va is a shortest path from v to a.
3. Finally, associated with each pseudo-triangle T, there is an "exit" vertex $v(T)$ such that for every point x contained in T, the shortest path from x to a can be obtained in O(1) time by setting $d(x,a) = d(x,\text{adj}(v)) + \text{dist}(v)$ and then following the pointers adj( ) back to a.

Remarkably, Storer and Reif's algorithm works out when obstacles are merely points and, therefore, our set of circles can be viewed as padded points. Thus, in O(kn) time, having pre-computed the Voronoi diagram of the obstacle space, we are able of finding the shortest path among circular obstacles. In this discussion, we have assumed that the model of computation is the real RAM augmented with the square root operation, as Storer and Reif do in their work.

4. Application to the HazMat transportation problem

Let us put all the pieces together so the obnoxious shortest path problem is solved. As pointed out in Section 3, the main idea is locate a continuous piece of $F(r)$ and then apply the bisection algorithm. If such piece does not exist, the best approximation is given by the greatest negative value of $F(r)$ (refer to Fig. 4, case (2) again).

The algorithm we present below consists of two parts; the first one is a pre-processing step and the second the algorithm itself.

In order to find the appropriate continuous piece, we first compute the Delaunay triangulation of the sites. By walking around its vertices, we can create, in O($n \log n$) time, a sorted list with all distances between edges (in increasing order). This will serve for determining how the connected components are fused together. We start by taking r, the radius of the circles, as half the distance between the points of the closest pair. With this radius, two circles are about to form a new connected component. Now, the Voronoi diagram of the circles and the contour of the union of the circles is computed. Finally, we apply the algorithm by Storer and Reif (see Fig. 5).

The pre-processing step is over. Let us describe the algorithm. Let P_r be the shortest path obtained at the pre-processing step. If $L(P_r)$ is greater than l, then we have located the continuous piece of $F(r)$ that crosses $r = 0$ (So far, we have only examined the first continuous piece of $F(r)$.). Therefore, the bisection algorithm can be applied to find the ε-approximation.

Otherwise, we take the next closest pair from the list and set the new radius r to half the distance of that closest pair. Now, it is necessary to re-compute the contour of the union of the circles and the Voronoi diagram of the obstacle space. As a consequence of the choosing of the new radius, two important facts should be observed: (1) only one change occurs in the Voronoi diagram as for exactly two connected components of the contour of the union are fused together; (2) except for the two connected components that are merged, the rest of the contour of the union does not change topologically. Furthermore, those two changes can be updated in linear time. Again, compute the shortest path P_r and check whether $L(P_r)$ is greater than l.

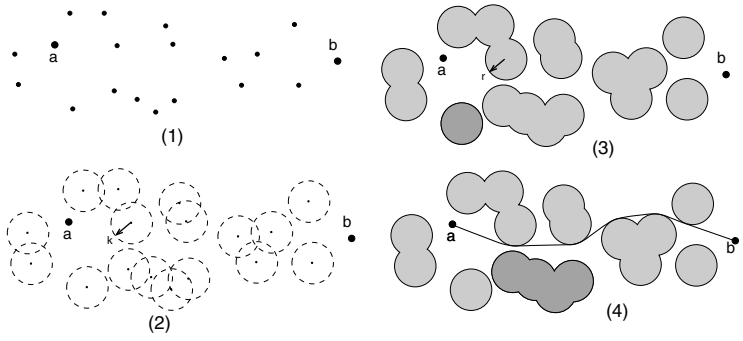

Fig. 5. Solving the obnoxious short path problem.

We repeat this procedure until obtaining the ε-approximation.

The complexity of this algorithm is the following:

- Computing the Delaunay triangulation and obtaining the list of edge distances take $O(n \log n)$ time.
- Updating the Voronoi diagram and re-computing the contours of the union of the circles can be done in linear time. This operation must be done $O(n)$ times.
- Once the continuous piece is located, computing $L(P_r)$ is $O(n) + O(kn)$. Therefore, the total complexity of the bisection method is $O(\log(\varepsilon^{-1}))[O(n) + O(kn)]$.

The total complexity is:

$$O(n \log n) + O(n)[O(n) + O(kn)] + O(\log(\varepsilon^{-1}))[O(n) + O(kn)]$$

Therefore, we have proved the following theorem.

Theorem 1. *Given a set of n sites, a source point, a destination point and an error* ε, *an* ε*-approximation to the solution of the obnoxious short path problem can be computed in* $[O(n) + O(\log(\varepsilon^{-1}))][O(n) + O(kn)]$ *time, where k is the number of connected components of the obstacle space when computing the shortest path.*

5. Concluding remarks

We have solved the (approximate) obnoxious short path problem in $[O(n) + O(\log(\varepsilon^{-1}))][O(n) + O(kn)]$ time, where ε is the allowed error, by using several geometric techniques (Voronoi diagrams in the Laguerre geometry and shortest paths among obstacles). Those techniques have proven to be very useful for solving optimization problems. In fact, they can be generalized, for example, to sites that are formed by polygons. The algorithm by Storer and Reif still works out when polygons are padded to become rounded obstacles.

Finally, the above techniques can be adopted for other distances, for example, the polyhedral metrics case. In the particular case of the l_∞ or l_1 metrics, efficiently algorithm can be found. It is well known that the contour of an union of n isothetic rectangles can be constructed in optimal time $\Theta(n \log n)$ (Preparata & Shamos, 1985). Then by applying the Storer–Reif data structure, the obnoxious short path problem with l_∞ metric can be solved in $O(kn)$ time and linear space, when k is the number of islands. With l_1 metric, we can use a similar approach.

Acknowledgement

The first author is partially supported by Project MCYT BFM2000-1052-C02-01 and BFM2003-04062.

References

Barcia, J. A., Díaz-Báñez, J. M., Lozano, A., & Ventura, I. (2003). Computing an obnoxious anchored segment. *Operations Research Letters., 31*, 293–300.

Díaz-Báñez, J. M., Hurtado, F., Meijer, H., Rappaport, D., & Sellarès, J. A. (2003). The largest empty annulus problem. *International Journal of Computational Geometry and Applications., 13*(4), 317–325.

Díaz-Báñez, J. M., Mesa, J. A., & Shöbel, A. (2004). Continuous location of dimensional structures. *European Journal of Operational Research., 152*, 22–44.

Drezner, Z., & Wesolowsky, G. O. (1989). Location of an obnoxious route. *Journal of Operational Research, 40*, 1011–1018.

Erkut, E., & Verter, V. (1995). Hazardous materials logistics. In Z. Drezner (Ed.), *Facility location: a survey of applications and methods* (pp. 467–506). Springer-Verlag.

Erkut, E., & Verter, V. (1998). Modelling of transport risk for hazardous materials. *Operations Research, 46*(5), 625–642.

Follert, F., Schömer, E., Sellen, J., Smid, M., & Thiel, C. (1997). Computing a largest empty anchored cylinder and related problems. *International Journal of Computational Geometry and Applications, 7*, 563–580.

Imai, H., Iri, M., & Murota, K. (1985). Voronoi diagrams in the Laguerre geometry and its applications. *SIAM Journal of Computing, 14*(1), 93–105.

Mitchell, J. S. B. (2000). Geometric shortest paths and network optimization. In J. R. Sack & J. Urrutia (Eds.), *Handbook of computational geometry* (pp. 633–701). Elsevier Science.

Nievergelt, J., & Preparata, F. P. (1982). Plane-sweep algorithms for intersecting geometric figures. *Communications of the ACM, 25*(10), 739–747.

Okabe, A., Boots, B., & Sugihara, K. (1992). *Spatial tesselations: concepts and applications of Voronoi diagrams*. Wiley.

Preparata, F. P., & Shamos, M. I. (1985). *Computational geometry: an introduction*. New York: Springer-Verlag.

Shamos, M. I., Hoey, D. (1975). Closest-point problems, In Proceedings of the 16th IEEE symposium on foundations of computer science, Berkeley, California (pp. 151–162).

Storer, J. A., & Reif, J. H. (1994). Shortest paths in the plane with polygonal obstacles. *Journal of ACM, 41*(5), 982–1012.

Verter, V., & Erkut, E. (1996). Hazardous materials logistics: an annotated bibliography. In C. Carraro & A. Haurie (Eds.), *Operations research and environmental management* (pp. 221–267). Kluwer Academic Publishers.