Iranian Journal of Mathematical Sciences and Informatics
Vol. 17, No. 2 (2022), pp 235-241
DOI: 10.52547/ijmsi.17.2.235

Single-Point Visibility Constraint Minimum Link Paths in Simple Polygons

Mohammad Reza Zarrabi, Nasrollah Moghaddam Charkari*

Faculty of Electrical Engineering and Computer Science, Tarbiat Modares University, Tehran, Iran

E-mail: m.zarabi@modares.ac.ir
E-mail: charkari@modares.ac.ir

Abstract. We address the following problem: Given a simple polygon P with n vertices and two points s and t inside it, find a minimum link path between them such that a given target point q is visible from at least one point on the path. The method is based on partitioning a portion of P into a number of faces of equal link distance from a source point. This partitioning is essentially a shortest path map (SPM). In this paper, we present an optimal algorithm with $O(n)$ time bound, which is the same as the time complexity of the standard minimum link paths problem.

Keywords: Minimum link path, Shortest path map, Point location.

2000 Mathematics subject classification: 68U05, 52B55, 68W40, 68Q25.

1. Introduction

Link paths problems in computational geometry have received considerable attention in recent years due to not only their theoretical beauty, but also their wide range of applications in many areas of the real world. A minimum link path is a polygonal path between two points s and t inside a simple polygon P with n vertices that has the minimum number of links. Minimum link paths are fundamentally different from traditional Euclidean shortest path, which has

*Corresponding Author

Received 10 October 2018; Accepted 27 January 2019
©2022 Academic Center for Education, Culture and Research TMU

[Downloaded from ijmsi.ir on 2026-11-24]
[DOI: 10.52547/ijmsi.17.2.235]

the shortest length among all the polygonal paths without crossing edges of P. Minimum link paths have important applications in various areas like robotic, motion planning, wireless communications, geographic information systems, image processing, etc. These applications benefit from minimum link paths since turns are costly while straight line movements are inexpensive.

Many algorithms structured around the notion of link path have been devised to parallel those designed using the Euclidean path. In [11], Suri introduced a linear time algorithm for computing a minimum link path between two points inside a triangulated simple polygon P (Ghosh presented a simpler algorithm in [7]). After that, Suri developed the proposed solution to a query version by constructing a window partition in linear time for a fixed point inside P [12]. This window partition is essentially a shortest path map, because it divides the simple polygon into faces of equal link distance from a fixed source point. By contrast, the work of Arkin et al. [3] supports $O(\log n)$ time queries between any two points inside P after building n shortest path maps for all vertices of P, i.e., the total time complexity for this construction is $O(n^2)$ (an optimal algorithm for this case was presented by Chiang et al. in [4]). On the other hand, when there are holes in the polygon, Mitchell et al. [10] proposed an incremental algorithm with $O(E\alpha(n)\log^2 n))$ time bound, where n is the total number of edges of the obstacles, E is the size of the visibility graph, and $\alpha(n)$ denotes the extremely slowly growing inverse of Ackermann's function. An interesting survey of minimum link paths appears in [9].

Minimum link paths problems are usually more difficult to solve than equivalent Euclidean shortest path problems since optimal paths that are unique under the Euclidean metric need not be unique under the link distance. Another difficulty is that, Euclidean shortest paths only turn at reflex vertices of P while minimum link paths can turn anywhere.

The problem is studied with several variations. One of these variations is to constrain the path to view a point q from at least one point on the path. One possible application is resource collection in which a robot moving from a source point to a destination one has to collect some resources found in a certain region. Some other applications are visibility related constraints such as wireless communication or guarding systems, where direct visibility is crucial. In this paper, we study the constrained version of minimum link paths problem based on the shortest path map called SPM approach. The proposed algorithm runs in linear time.

Section 2 introduces the basic definitions and lemmas. Section 3 presents our algorithm for a visibility point q. We conclude in Section 4 with some open problems.

2. Preliminaries

We use the following notation throughout the paper:

[Downloaded from ijmsi.ir on 2026-11-24]

[DOI: 10.52547/ijmsi.17.2.235]

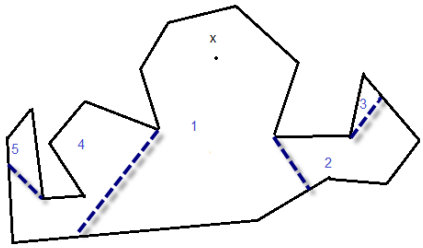

FIGURE 1. SPM(x) and its faces.

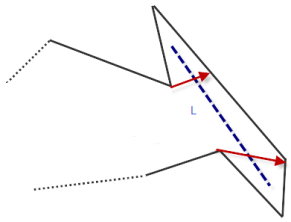

FIGURE 2. Dividing any line segment L to at most three parts.

(1) $V(x)$: the visibility polygon of a point $x \in P$
(2) $\pi_L(x, y)$: a minimum link path between two points x and y in P
(3) $D_L(x, y)$: the link distance (minimum number of links) of $\pi_L(x, y)$
(4) $D_\pi(x, y)$: the number of links between x and y on the path π
(5) $Pocket(x)$: the regions of P, invisible to x

Definition 2.1. q-visible path: A minimum link path, which has a non-empty intersection with $V(q)$ for any point $q \in P$.

Definition 2.2. SPM(x): Let x be a point in P. The edges of $V(x)$ either are (part of) edges of P or chords of P. The edges of the latter variety are called windows of $V(x)$. The set of points at link distance one from x is precisely $V(x)$. The points of link distance two are the points in $P - V(x)$ that are visible from some windows of $V(x)$. Repeatedly, applying this procedure partitions P into faces of constant link distance from x [12]. For the sake of consistency, we call this partitioning the shortest path map of P with respect to x (see Figure 1).

The following two lemmas are the fundamental properties of SPM used in our algorithm:

Lemma 2.3. *For any point $x \in P$, $SPM(x)$ is constructed in $O(n)$ time* [12].

Lemma 2.4. *Given a point $x \in P$, any line segment L intersects at most three faces of $SPM(x)$* [3].

Figure 2 illustrates the worst case intersection described in Lemma 2.4. Our goal is to compute a *q-visible* path inside a simple polygon P with n vertices.

[Downloaded from ijmsi.ir on 2026-11-24]

[DOI: 10.52547/ijmsi.17.2.235]

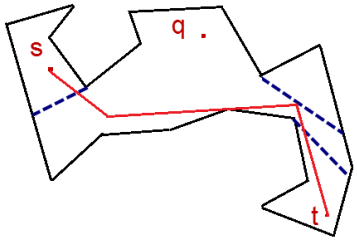

FIGURE 3. Step 4 of the algorithm.

3. The Algorithm

Suppose that three points s, t and q are given in P. We sketch the generic algorithm for computing a *q-visible* path between s and t as follows:

(1) If $D_L(s,q) = 1$ or $D_L(t,q) = 1$, report $\pi_L(s,t)$.
(2) Compute $V(q)$ and $Pocket(q)$.
(3) Use the point location algorithm for s and t to determine whether they are in the same region in $Pocket(q)$ or not.
(4) If s and t are in two different regions of $Pocket(q)$, report $\pi_L(s,t)$. Since this path crosses $V(q)$, it is a *q-visible* path (see Figure 3).
(5) Suppose that e is a chord of P that separates $V(q)$ and the region of $Pocket(q)$, which contains both s and t. Indeed, e is a window of $V(q)$ and divides P into two subpolygons, only one of which contains q. We define p as the subpolygon containing q. A *q-visible* path should have a non-empty intersection with p (note that $e \in p$).
(6) Compute both SPM(s) and SPM(t). Add two labels S_j and T_k to each *face* of SPM(s) and SPM(t) as the link distance from s and t to those faces, respectively ($1 \leq j \leq ||\text{SPM(s)}||$, $1 \leq k \leq ||\text{SPM(t)}||$ and $||.||$ denotes the number of faces of SPM).
(7) Use the map overlay technique to find the intersections of SPM(s), SPM(t) and p. Construct the planar subdivision of p with new faces (called *Cells*) by the *quad view* data structure [6]. Use a set Ce as a reference to these *Cells* (each element of this set points to the value or structure of a *Cell*).
(8) Assign the value of each *Cell*, i.e., $Ce(i) = s_i + t_i$, where $1 \leq i \leq$ the number of *Cells* and $s_i = S_j$, $t_i = T_k$. In other words, s_i and t_i are the link distances from any point $x \in Ce(i)$ to s and t, respectively.
(9) Find the minimum value of $Ce(i)$ for some i. It ensures that a *q-visible* path between s and t enters $Ce(i)$.
(10) Report $\pi_L(s,x)$ appended by $\pi_L(x,t)$, where x is a point in $Ce(i)$. The link distance of this new path would be $s_i + t_i$ (Lemma 3.1).

[Downloaded from ijmsi.ir on 2026-11-24]
[DOI: 10.52547/ijmsi.17.2.235]

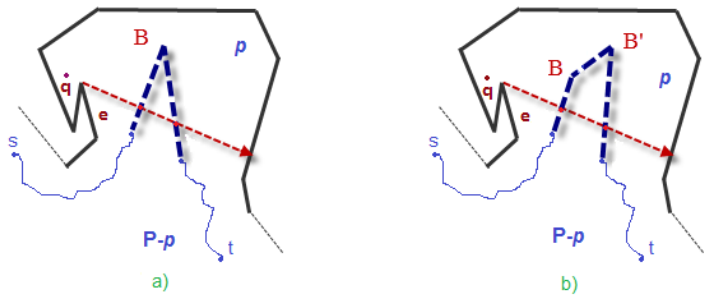

FIGURE 4. Two points s and t lie on the same side of e.

Lemma 3.1. *Assume s and t lie on the same side of e. Then, the link distance of a q-visible path π between s and t is: $s_i + t_i$ (i might not be unique).*

Proof. For any point x on any optimal path π, we have the following equalities ($|.|$ denotes the link distance):

$|\pi| = D_\pi(s,x) + D_\pi(x,t)$ *or* $|\pi| = D_\pi(s,x) + D_\pi(x,t) - 1$

The first equation occurs, if x is a bending point on π. In this case, $D_\pi(s,x) = D_L(s,x)$ and $D_\pi(x,t) = D_L(x,t)$ due to the local optimality principle. Similarly, the second equation holds, if the last bending point from s to x, x, and the first bending point from x to t are collinear.

Since two points s and t lie on the same side of e inside $P - p$, there is always a bending point B in p on π. Indeed, there are at most two points B and B', because if we have more than two bending points in p, the link distance of π (optimal path) can be shortened by at least one link due to the triangle inequality (see Figure 4).

We construct a path π' like this: for $x \in Ce(i)$, append $\pi_L(s,x)$ to $\pi_L(x,t)$. The following inequality clearly holds:

$s_i + t_i - 1 = D_L(s,x) + D_L(x,t) - 1 \leq |\pi'| \leq D_L(s,x) + D_L(x,t) = s_i + t_i$

Thus, we have: $|\pi| \leq |\pi'| \leq s_i + t_i$ for π. According to the above equalities, if we have one bending point B, then $|\pi| = D_\pi(s,B) + D_\pi(B,t)$. In this case, B must be in $Ce(i)$, i.e., $|\pi| = s_i + t_i$ (Figure 4(a)). On the other hand, there are three options for two bending points B and B' (Figure 4(b)):

(1) Only $B \in Ce(i)$ or $B' \in Ce(i)$. Again in this case, $|\pi| = s_i + t_i$. More precisely, if $B \in Ce(i)$, then $B' \in Ce(j)$, where $s_i + t_i = s_j + t_j$.
(2) (B and B') $\in Ce(i)$, then $|\pi| = D_\pi(s,B) + 1 + D_\pi(B',t) = s_i + t_i + 1$.
(3) Neither B nor B' belongs to *Cells* with minimum value (e.g., $Ce(i)$). Thus, $B \in Ce(j)$ and $B' \in Ce(k)$, where $s_j + t_j = s_k + t_k > s_i + t_i$.

For the last two options, π would not be an optimal path. Therefore, only the first case can occur. The above discussion indicates that $|\pi| = s_i + t_i - 1$ would not be possible. □

[Downloaded from ijmsi.ir on 2026-11-24]

[DOI: 10.52547/ijmsi.17.2.235]

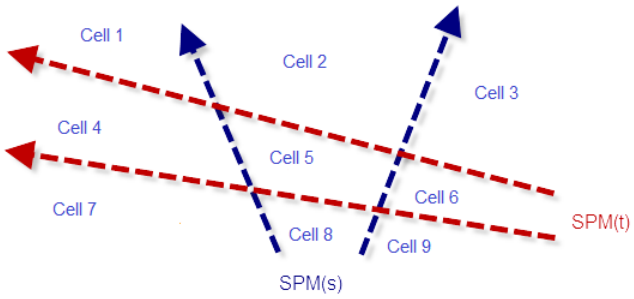

FIGURE 5. At most 4 intersections and 9 *Cells*.

To analyze the time complexity of the algorithm, observe that the check in Step 1 can be easily done in linear time [11]. The computations of $V(q)$ and $Pocket(q)$ in Step 2 can be done using the linear time algorithm in [8]. Steps $3, 4$ and 5 can be done in linear time [5, 11]. Based on Lemma 2.3, Step 6 can be done in linear time as well.

Note that, windows of a SPM never intersect each other. According to Lemma 2.4, each window of SPM(s) intersects at most two windows of SPM(t) and vice versa. Therefore, if we have k_1 and k_2 windows of SPM(s) and SPM(t) inside p, respectively, then there are at most k_1+k_2 intersection points inside p, where k_1, $k_2 = O(n)$. Thus, we have $O(n)$ intersection points inside p between the windows of SPM(s) and SPM(t).

To find the intersections, we make use of some known algorithms for subdivision overlay like [6], which solves the problem in optimal time $O(n + k)$. Here, k is the number of intersections, which is $O(n)$ in the worst case in our problem. Therefore, Step 7 can be done in linear time as well.

These intersections partition p to at most $9/4(k_1 + k_2) = O(n)$ *Cells* (see Figure 5). If either k_1 or k_2 does not exist, then we ignore the effect of that SPM to p. Furthermore, if neither k_1 nor k_2 exists, the whole p will be assumed as a *Cell*. Thus, Steps $8, 9$ can be done in linear time. According to [11], Step 10 is done in linear time. The following theorem is concluded:

Theorem 3.2. *For any three points s, t and q inside a simple polygon P with total n vertices, a q-visible path between s and t and its link distance can be computed in $O(n)$ time.*

4. Conclusion

We studied the problem of finding a minimum link path between two points with point visibility constraint in a simple polygon. The time bound for a *q-visible* path is similar to the bound for the standard minimum link paths problem (without constraint). So, this bound cannot further be improved.

[Downloaded from ijmsi.ir on 2026-11-24]

[DOI: 10.52547/ijmsi.17.2.235]

Possible extensions to this problem involve studying the same problem when the path is required to meet an arbitrary general region, not necessarily a visibility region or having a query point instead of a fixed point.

A modified version of the problem is to constrain the path to meet several target polygons in a fixed order. If we fix the order of meeting, the problem seems to be less complex. As Alsuwaiyel et al. [1] have shown, this problem is NP-hard even for any points on the boundary of a simple polygon. This type of problem is defined as minimum link watchman route with no fixed order.

Another extension is to improve the approximation factor of minimum link visibility path problem mentioned in [2], using the same approach (SPM).

Acknowledgements

The first author wishes to thank Dr Ali Gholami Rudi from Babol Noshirvani University of Technology for many pleasant discussions.

References

1. M. H. Alsuwaiyel, D. T. Lee, Minimal Link Visibility Paths Inside a Simple Polygon, *Computational Geometry Theory and Applications*, **3**(1), (1993), 1-25.
2. M. H. Alsuwaiyel, D. T. Lee, Finding an Approximate Minimum-Link Visibility Path Inside a Simple Polygon, *Information Processing Letters*, **55**(2), (1995), 75-79.
3. E. M. Arkin, J. S. B. Mitchell, S. Suri, Logarithmic-Time Link Path Queries in a Simple Ppolygon, *International Journal of Computational Geometry and Applications*, **5**(4), (1995), 369-395.
4. Y. J. Chiang, R. Tamassia, Optimal Shortest Path and Minimum-Link Path Queries Between Two Convex Polygons Inside a Simple Polygonal Obstacle, *International Journal of Computational Geometry and Applications*, **7**(01n02), (1997), 85-121.
5. H. Edelsbrunner, L. J. Guibas, J. Stolfi, Optimal Point Location in a Monotone Subdivision, *SIAM Journal on Computing*, **15**(2), (1986), 317-340.
6. U. Finke, K. H. Hinrichs, Overlaying Simply Connected Planar Subdivisions in Linear Time, *In Proceedings of the eleventh annual symposium on Computational Geometry*, **ACM**, (1995), 119-126.
7. S. K. Ghosh, Computing the Visibility Polygon from a Convex Set and Related Problems, *Journal of Algorithms*, **12**(1), (1991), 75-95.
8. B. Joe, R. B. Simpson, Corrections to Lee's Visibility Polygon Algorithm, *BIT Numerical Mathematics*, **27**(4), (1987), 458-473.
9. A. Maheshwari, J. R. Sack, H. N. Djidjev, Link Distance Problems, *In Handbook of Computational Geometry*, **(**2000), 519-558.
10. J. S. B. Mitchell, G. Rote, G. Woeginger, Minimum-Link Paths Among Obstacles in the Plane, *Algorithmica*, **8**(1-6), (1992), 431-459.
11. S. Suri, A Linear Time Algorithm for Minimum Link Path Inside a Simple Polygon, *Computer Vision, Graphics, and Image Processing*, **35**(1), (1986), 99-110.
12. S. Suri, On Some Link Distance Problems in a Simple Polygon, *IEEE transactions on Robotics and Automation*, **6**(1), (1990), 108-113.

[Downloaded from ijmsi.ir on 2026-11-24]

[DOI: 10.52547/ijmsi.17.2.235]