The Role of Graph Theory in Solving Euclidean Shortest Path Problems in 2D and 3D*

Phan Thanh An[1,2], Nguyen Ngoc Hai[3], and Tran Van Hoai[4]

[1] Institute of Mathematics, Hanoi, Vietnam
[2] CEMAT, Instituto Superior Tecnico, Portugal
[3] International University, Vietnam National University, Ho Chi Minh City, Vietnam
[4] Faculty of Computer Science and Engineering, HCMC University of Technology, Ho Chi Minh City, Vietnam
thanhan@math.ist.utl.pt, nnhai@hcmiu.edu.vn, hoai@cse.hcmut.edu.vn

Abstract. Determining Euclidean shortest paths between two points in a domain is a fundamental problem in computing geometry and has many applications in GIS, robotics, computer graphics, CAD, etc. To date, solving Euclidean shortest path problems inside simple polygons has usually relied on triangulation of the entire polygons and graph theory. The question: "Can one devise a simple O(*n*) time algorithm for computing the shortest path between two points in a simple polygon (with *n* vertices), without resorting to a (complicated) linear-time triangulation algorithm?" raised by J. S.B. Mitchell in Handbook of Computational Geometry (J. Sack and J. Urrutia, eds., Elsevier Science B.V., 2000), is still open. The aim of this paper is to show that in 2D, convexity contributes to the design of an efficient algorithm for finding the approximate shortest path between two points inside a simple polygon without triangulation of the entire polygons or graph theory. Conversely, in 3D, we show that graph tools (e.g., Dijkstra's algorithm for solving shortest path problems on graphs) are crucial to find an Euclidean shortest path between two points on the surface of a convex polytope.

Keywords: Approximate algorithm, convex hull, discrete geometry, Dijkstra's algorithm, extreme point, Euclidean shortest path, graph theory, shortest path on graph.

1 Introduction

Determining Euclidean shortest paths between two points in a domain is a fundamental problem in computing geometry and has many applications in GIS,

* This work received financial support from Portuguese National Funds through FCT (Fundação para a Ciência e a Tecnologia) under the scope of project Pest-OE/MAT/UI0822/2011 and the National Foundation for Science and Technology Development, Vietnam (NAFOSTED) under grant number 101.02-2011.45.

H.-Y. Jeong et al. (eds.), *Advances in Computer Science and Its Applications*, Lecture Notes in Electrical Engineering 279,
DOI: 10.1007/978-3-642-41674-3_27,

robotics, computer graphics, CAD, etc. To date, all methods for solving this problem in case the domain is a simple polygon, as presented in [9], [11] etc, rely on starting with a rather complicated, but linear-time triangulation of a simple polygon. This leads to the open question below raised by J. S. B. Mitchell [11] "Can one devise a simple O(n) time algorithm for computing the shortest path between two points in a simple polygon (with n vertices), without resorting to a (complicated) linear-time triangulation algorithm?" In 1987, the Steiner's problem of finding the inpolygon of a given convex polygon with minimal circumference was solved completely by the method of orienting curves [13]. Efficient algorithms for determining convex ropes in robotics were introduced in [4] and [12]. These problems are variations of the shortest path problem in 2D and thus can be solved without resorting to a linear-time triangulation algorithm or graph theory. In the first part of this paper we recall Lee and Preparata's algorithm that relies on triangulation of the entire polygons and graph theory. Then we present an algorithm for finding the approximate shortest path between two points inside a simple polygon. We show that convexity contributes to the design of this algorithm without triangulation of the entire polygons and graph theory. In the second part of this paper, we investigate Li and Klette's algorithm [10] and show that graph tools (e.g., Dijkstra's algorithm for solving shortest path problems on graphs) are crucial to find the Euclidean approximate shortest path between two points on the surface of a convex polytope. Numerical experiences are presented.

2 Lee and Preparata's Algorithm

Lee and Preparata's algorithm [9] for finding the shortest path SP(a,b) between two points a and b in a simple polygon consists of three steps:

Step 1: Triangulate the simple polygon then get a corresponding dual tree.

Step 2: Find shortest path on this tree (by Dijkstra's Algorithm [8]/Thorup's Algorithm [15]) and get corresponding sleeve domain.

Step 3: Using a so-called funnel algorithm to find shortest path on the sleeve.

Thus, graph theory is crucial in Lee and Preparata's algorithm.

3 Convexity and Approximate Shortest Paths in a Simple Polygon

We introduce an approximation algorithm for solving the shortest path problem without triangulation and graph theory in [2]. The idea of the direct multiple shooting method [7] is used for the discretization of the shortest path problem based on cut segments parallel to y-coordinate $0y$. Assume that a and b respectively are the first and the final of P and Q. The simple polygon $D=PQ$ is divided by cut segments $\xi_i = [u_i, v_i]$, $u_i \in P$, $v_i \in Q$ such that

$\xi_i = [u_i, v_i] \subseteq D$ and $]u_i, v_i[\subseteq \text{int}D_i$

$[u_i, v_i]$ strictly separates a and b

T_i is bounded by P, Q, cut segments ξ_i and ξ_{i+1}

$PQ = \bigcup_{i=0}^{k} T_i$, $\text{int}T_i \cap \text{int}T_j = \phi$ for $i \neq j$

with $\xi_0 := a$, $\xi_{k+1} := b$. Assume that

$$\text{SP}(u_i, u_{i+1}) \cap \text{SP}(v_i, v_{i+1}) \neq \phi \text{ for all } i = 1,...,k-1.$$

The shortest path SP(a,b) between two points a and b in a simple polygon is due to An and Hoai in [6] without triangulation and theory graph. (To avoid triangulation and theory graph, we do not use Lee and Preparata's algorithm in Sect. 2). Convexity is crucial in An and Hoai's algorithm (see [6]).

Given: vertices a and b of a simple polygon PQ, where P (Q, respectively) is the polyline formed by vertices of the polygon from a to b (from b to a , respectively) in counterclockwise order.

Find: the shortest path SP(a,b) between a and b inside PQ.

1. Divide the polygon PQ into suitable subpolygons T_i by cut segments $\xi_1,..., \xi_k$ satisfying (1)-(2). Set $j=0$ and choose initial shooting points $a^j_i \in \xi_i$, $i=1,...,k$.

2. Find the shortest path $Z^j_i := \text{SP}(a^j_i, a^j_{i+1})$ in T_i. Check if all Z^j_i $(i=1,...,k)$ satisfy simultaneously the following
(a) If either $(\pi - \alpha\_u_i)(\pi - \alpha\_v_i) < 0$ and $\alpha\_a^j_i = \pi$, or $\alpha\_u_i \geq \pi$ (or $\alpha\_v_i \leq \pi$) then

$\text{SP}(a,b) := \bigcup^{i=k}_{i=1} Z^j_i$. STOP.
(b) Else, refine shootings $a^j_i \in \xi_i$ to ensure that the condition $\alpha\_a^j_i = \pi$ holds true. Set $j=j+1$, go to step 2.

Here, α_u_i (α_v_i , respectively) is the measure of the angle between two polylines $\text{SP}(u_i, u_{i-1})$ and $\text{SP}(u_i, u_{i+1})$ ($\text{SP}(v_i, v_{i-1})$ and $\text{SP}(v_i, v_{i+1})$, respectively), $\alpha\_a^j_i$ is the measure of the angle between two polylines $\text{SP}(a^j_i, a^j_{i-1})$ and $\text{SP}(a^j_i, a^j_{i+1})$, and the conditions (a) and (b) are obtained by convexity of the shortest paths. The number k of cut segments ξ_i, $i=1,...,k$, is specified by the user. Thus, we can compute the shortest path between two points inside a simple polygon without triangulation of the entire polygon and graph theory.

4 Shortest Paths on Graphs and Shortest Paths on Polytopes

In this section we describe the use of the ideas of the cut slices parallel to x0y and the direct multiple shooting method [7] for solving the shortest path problem in 3D. These cut slices are due to Li and Klette's algorithm [10].

We consider the following shortest path problem:

> Find the Euclidean shortest path Z between two vertices a and b on the surface of a convex polytope D.

We split the convex polytope D into sub convex polytopes T_i by cut slices $\xi_1,\ldots,\xi_k$ (i.e., a convex polytope shooting grid) as follows:

ξ_i consists of simple polygons $\xi^j_i = u^j_i\, u^j_1 \ldots u^j_{n-1} \subseteq \mathrm{bd}D \cap \xi_i$

ξ_i is parallel to $0xy$, ξ_i strictly separates a and b, T_i is bounded by D, cut slices ξ_i and ξ_{i+1},

$\cup_{i=0}^{k} T_i \subseteq D$, $\mathrm{int}T_i \cap \mathrm{int}T_j = \phi$ for $i \neq j$

with $\xi_0 := a$, $\xi_{k+1} := b$.

In the same manner with the method of the algorithm in Sect. 3, instead of cut segments, cut slices parallel to $0xy$ are used.

These cut slices are constructed via the shortest path on a graph constructed by vertices and edges of the polytope ([10]), where the path is determined by Dijkstra's Algorithm [8]/Thorup's Algorithm [15]. Thus, the number k of cut slices ξ_i, $i=1,\ldots,k$, is not specified by the user.

5 Numerical Experiences

Both the running time of the algorithm given in Sect. 3 and the peak memory usage of the system significantly reduce as cut segments are used (see Table 1 ([3]) below).

Table 1. The peak memory usage of the system significantly reduce as cut segments are used

Number of cut segments	Time *(in sec.)*	Peak Memory *(in MB) (freed on nodes)*	Peak Memory *(in MB) (freed on nodes)*
0	20858.53	0.80	4839.91
5	3641.00	0.13	2040.96
10	1988.84	0.07	610.29
100	298.82	0.00	7.85
1000	283.01	0.00	0.09

The running time of the algorithm given in Sect. 3 decreases significantly from 399.125 seconds to 1.94997 seconds as the number $\alpha\_a^j_i = \pi$ of cut segments increases from 0 to 2500. Here, $k=0$ means the shortest path between a and b inside the polygon PQ is determined by [5] without cut segments (see Figure 1 [2] below).

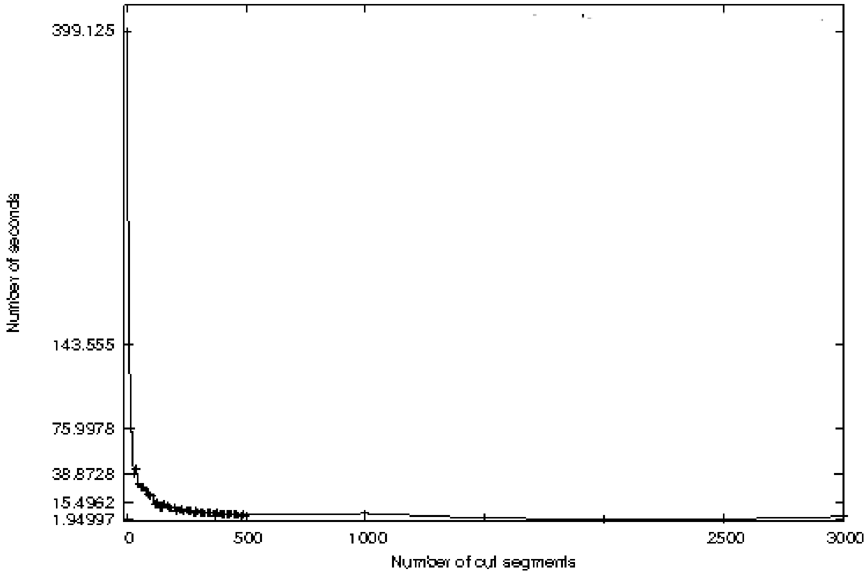

Fig. 1. The running time of the algorithm given in Sect. 3 decreases significantly as the number k of cut segments increases

6 Conclusion

We have shown that in 2D, convexity contributes to the design of an efficient algorithm for finding the approximate shortest path between two points inside a simple polygon without triangulation of the entire polygons or graph theory. Conversely, in 3D, graph tools are crucial to find an Euclidean shortest path between two points on the surface of a convex polytope.

References

1. An, P.T., Hai, N.N., Hoai, T.V.: The role of convexity for solving some shortest path problems in plane without triangulation. In: International Conference on Mathematical Sciences and Statistics, Kuala Lump, Malaysia, February 5-7. AIP Conference Proceedings, American Institute of Physics, NY (2013)
2. An, P.T., Hai, N.N., Hoai, T.V.: Direct multiple shooting method for solving approximate shortest path problems. Journal of Computational and Applied Mathematics 244, 67–76 (2013)
3. An, P.T., Hai, N.N., Hoai, T.V., Trang, L.H.: On the performance of trangulation-based multiple shooting method for 2D shortest path problems. In: International Conference on Advanced Computing and Applications (ACOMP 2013), Ho Chi Minh City, Vietnam, October 23-25 (2013)
4. An, P.T.: Reachable grasps on a polygon of a robot arm: finding convex ropes without triangulation. Journal of Robotics and Automation 25(4), 304–310 (2010)

5. An, P.T.: Method of orienting curves for determining the convex hull of a finite set of points in the plane. Optimization 59(2), 175–179 (2010)
6. An, P.T., Hoai, T.V.: Incremental convex hull as an orientation to solving the shortest path problem. In: IEEE Proc. 3rd Int. Conf. Comp. & Auto. Eng., Chongqing, China, January 21-23 (2011)
7. Bock, H.G., Plitt, K.J.: A multiple shooting algorithm for direct solution optimal control problems. In: Proceedings of the 9th IFAC World Congress, Budapest, pp. 243–247. Pergamon Press (1984)
8. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271 (1959)
9. Lee, D.T., Preparata, F.P.: Euclidean shortest paths in the presence of rectilinear barriers. Networks 14, 393–410 (1984)
10. Li, F., Klette, R.: Euclidean Shortest Paths: Exact or Approximate Algorithms. Springer (2011)
11. Mitchell, J.S.B.: Geometric shortest paths and network optimization. In: Sack, J.R., Urrutia, J. (eds.) Handbook of Computational Geometry, pp. 633–701. Elsevier Science B. V. (2000)
12. Peshkin, M.A., Sanderson, A.C.: Reachable grasps on a polygon: the convex rope algorithm. IEEE Journal on Robotics and Automation 2(1), 53–58 (1986)
13. Phu, H.X.: Ein konstruktives Loesungsverfahren fuer das Problem des Inpolygons kleinsten Umfangs von J. Steiner. Optimization 18, 349–359 (1987)
14. O'Rourke, J.: Computational Geometry in C, 2nd edn. Cambridge University Press (1998)
15. Thorup, M.: Undirected single-source shortest paths with positive integer weights in linear time. J. ACM 46(3), 362–394 (1999)