Journal of Convex Analysis
Volume 30 (2023), No. 1, 249–270

Finding Approximately Convex Ropes in the Plane

Le Hong Trang
Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology; and: Vietnam National University Ho Chi Minh City, Thu Duc City, Ho Chi Minh City, Vietnam

Nguyen Thi Le
Institute of Mathematics, Vietnam Academy of Science and Technology, Hanoi, Vietnam

Phan Thanh An*
Institute of Mathematical and Computational Sciences and Faculty of Applied Science, Ho Chi Minh City University of Technology; and: Vietnam National University Ho Chi Minh City, Thu Duc City, Ho Chi Minh City, Vietnam
thanhan@hcmut.edu.vn

Received: February 28, 2022
Accepted: July 22, 2022

The convex rope problem is to find a counterclockwise or clockwise convex rope starting at the vertex a and ending at the vertex b of a simple polygon $\mathcal{P}$, where a is a vertex of the convex hull of $\mathcal{P}$ and b is visible from infinity. The convex rope mentioned is the shortest path joining a and b that does not enter the interior of $\mathcal{P}$. In this paper, the problem is reconstructed as one of finding such shortest path in a simple polygon and solved by the method of multiple shooting. We then show that if the collinear condition of the method holds at all shooting points, then these shooting points form the shortest path. Otherwise, the sequence of paths obtained by the update of the method converges to the shortest path. The algorithm is implemented in C++ for numerical experiments.

Keywords: Convex hull, approximate algorithm, convex rope, shortest path, non-convex optimization, geodesic convexity.

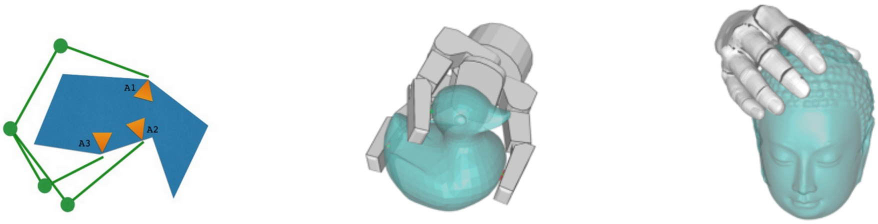
2010 Mathematics Subject Classification: 52A30, 52B55, 68Q25, 90C26, 65D18, 68R01.

1. Introduction

Solving automatic grasp planning problems is to evaluate must-touch regions, what forces should be applied to an object, and how those forces can be used by robotic hands [15, 18, 20]. A geometric construction, namely the convex rope posted by Peshkin and Sanderson in 1986 [17] gives valuable information for grasping automatically with robot hands. The problem of finding convex ropes plays a significant role in grasp planning a polygonal object with a simple robot hand. Besides the weight information of the object relevant to grasping and their geometric information are also necessary for this task. To simplify and decrease the number of possible hand configurations, objects are modeled as a set of shape primitives, such as polygons in 2D or polyhedrons, spheres, cylinders, and cones in 3D.

In 2D, determining convex ropes helps to find contact points which are points on a polygonal object corresponding to the robot's finger joint position (see Fig. 1.1(i)).

ISSN 0944-6532 / $ 2.50



(i) (ii) (iii)

Figure 1.1: Illustration of grasping with a simple robot hand; (i): grasping a polygonal object in 2D, images from *https://www.shadowrobot.com*; (ii) and (iii): grasping a polyhedral object in 3D, images from [18].

Figure 1.2: Contact points would not necessarily be in edges of the convex hull of the polygon.

For an object in 3D, the automatic grasp planning concentrates on finding must-touch regions on the object to get information about possible hand configurations (see Figs. 1.1(ii) and (iii)). Contact points usually belong to the trajectory of the convex rope joining two vertices of a simple polygon. These two vertices are not required to be located on the edges of the convex hull of the polygon (see Fig. 1.2). It turns out to consider that the endpoints of the convex rope can be visible from infinity (see Section 2 for the definition). In the non-convex optimization context, the problem can be stated as follows

$$\min_{(k,x_0,x_1,\ldots,x_{k+1})} \sum_{i=0}^{k} \|x_i - x_{i+1}\|$$

$$\text{subject to } \; x_i \in \mathcal{P}, \text{for } i = 1, 2, \ldots, k, \; x_0 = a, \;\; x_{k+1} = b$$
$$]x_i, x_{i+1}[\cap \text{int}(\mathcal{P}) = \emptyset, \;\text{ where }]x, y[:= [x, y] \setminus \{x, y\}.$$

Unfortunately, no optimization methods have been found for solving the problem. In this paper, we consider geometrical methods for solving the problem to get the global solution.

The non-convex optimization problem can be addressed via solving alternatively one of two fundamental problems which are convex hull and shortest path problems. In 1986 an algorithm based on evaluating cumulative angles was introduced [17], in which these angles are calculated for each side of the polygon. Another algorithm proposed by An in 2010 [1] was derived from Melkman convex hull algorithm. Using triangulation, a linear time algorithm in 1987 was presented in [8] to compute convex ropes as boundaries of the convex hull of a polyline. In 1990, Heffernan and Mitchell [11] also addressed a linear time algorithm by sorting to a partial triangulation of a polygon. All algorithms mentioned are exact. In 2006, Li and Klette [14] introduced an approximate algorithm that starts with a special trapezoidal segmentation of a polygon and their segmentation is simpler than the triangulation procedure.

* Corresponding author.

In this paper, we present an approximate algorithm using the geometric multiple shooting approach, which was proposed for solving the geometric shortest path problems in 2D and 3D [3, 6, 12]. In particular, the convex rope problem will be formulated into the form of finding shortest paths in a simple polygon since the closed relationship between convex sets and shortest paths. A set is convex if the line segment joining any two points of the set lies completely inside the set. Substituting a line segment joining two points by the shortest path joining these two points expands the notion of classical convexity to geodesic convexity [9, 19]. We then use successfully the method of multiple shooting (MMS for short) to find approximately these shortest paths. The three main factors of the method are given in detail. As an advantage of MMS, instead of solving the problem for the whole formed polygon, we deal with a set of smaller partitioned subpolygons. This is useful in the implementation of devices with limited memory.

In [3], the multiple shooting approach of An et al. is applied to simple polygons with a hypothesis of the so-called "oracle", not for general simple polygons. The "oracle" condition requests that between two consecutive cutting segments there exists at least one point in which every shortest path joining two corresponding shooting points passes through it, where notions of cutting segments and shooting points are introduced in Section 3. Besides, the relationship between the path satisfying the stop condition and the shortest path; the convergence of paths obtained by An et al.'s algorithm is not stated. In the following we state some open questions related to the algorithm in [3]

- Can MMS be used for simple polygons that are not "oracle"?
- If Z^* is a path obtained by the algorithm at an iteration which satisfies the stop condition, whether Z^* is the shortest path or not?
- Does the sequence of paths $\{Z^j\}_{j=0}^{+\infty}$ obtained by the algorithm converge to the shortest path?

In simple polygons, Hai and An in [9] showed the relationship between the convergence of sequences of paths with respect to the Hausdorff distance and the one with respect to the length. In this paper, we will apply this result to get the convergence to the shortest path of the sequence of paths obtained by the proposed algorithm. Particularly, we adapt MMS to the convex rope problem with slight modifications on determining the stop condition and updating new paths to answer these open questions. We construct a simple polygon in which our corresponding algorithm can apply even though the polygon does not satisfy the "oracle" condition (Section 3.1). We can also show that, at an iteration step, if a path satisfies the stop condition at all shooting points, then the path and the shortest path are identical. Otherwise, Theorems 4.11 and 4.12 state that the sequence of paths obtained by the algorithm converges to the shortest path, then the global solution is obtained.

The rest of the paper is organized as follows. Section 2 introduces the convex rope problem and three main factors of MMS. Section 3 presents the proposed algorithm in which MMS with three factors (f1)–(f3) can be applied. The correctness of the proposed algorithm and answers for the above open questions are stated in results in Section 4. The algorithm is implemented in C++ and numerical results are given and visualized in Section 5 to describe how our method works. Some advantages of MMS are established in Section 6.

2. Preliminaries

A vertex of a simple polygon $\mathcal{P}$ is *visible from infinity* if there exists a ray beginning at the vertex which does not intersect $\mathcal{P}$ anywhere else. In Fig. 3.1(i), b is visible from infinity with ray bb'. Clearly, a vertex of the convex hull of $\mathcal{P}$ is also visible from infinity. We say a path starting at the vertex a and ending at the vertex b of $\mathcal{P}$ is *counterclockwise* (*clockwise*, resp.) if the interior of $\mathcal{P}$ lies to the left (right, resp.) as we step along the path, starting at a.

Definition 2.1. The counterclockwise (clockwise, resp.) shortest path starting at a and ending at b of $\mathcal{P}$ that does not enter the interior of $\mathcal{P}$, where a is a vertex of the convex hull of $\mathcal{P}$ and b is visible from infinity, is called a *counterclockwise (clockwise, resp.) convex rope* starting at a and ending at b.

The convex rope problem is necessary to find a counterclockwise (clockwise, resp.) convex rope between a and b. For simplicity, we consider the counterclockwise convex rope case only. The clockwise convex rope case is considered similarly. The counterclockwise convex rope between a and b is illustrated as a dashed curve in Fig. 3.1(i). When both a and b are visible infinity, we obtain another version of the convex rope problem which can be also solved by the same way as the case that a is a vertex of the convex hull of $\mathcal{P}$ by our algorithm. For details see Section 3.1.

We now recall some basic concepts and properties. For any points x, y in the plane, we define

$$[x,y] := \{(1-\lambda)x + \lambda y : 0 \leq \lambda \leq 1\},$$
$$]x,y] := [x,y] \setminus \{x\},$$
and
$$]x,y[:= [x,y] \setminus \{x,y\}.$$

Let x and y be two points in a simple polygon $\mathcal{A}$. The *shortest path* joining x and y in $\mathcal{A}$, denoted by $\mathrm{SP}(x,y)$, is the path of minimal length joining x and y in $\mathcal{A}$. As shown in [7, 13], $\mathrm{SP}(x,y)$ is a polyline whose vertices except for x and y are reflex vertices of $\mathcal{A}$, i.e., the internal angles at these vertices are at least π. In addition, these vertices are also reflex vertices of $\mathrm{SP}(x,y)$. Furthermore, let $y' \in \mathcal{A}$, for any two line segments e of $\mathrm{SP}(x,y)$ and g of $\mathrm{SP}(x,y')$, we have either (i) $e = g$; or (ii) e and g are disjoint or share at most one endpoint. Fundamental concepts such as polylines, polygons, convex (reflex) vertices, shortest paths and their properties which will be used in this paper can be seen in [2, 10].

Consider a point $x \in \mathbb{R}^2$ and let A be a nonempty subset of $\mathbb{R}^2$. Then the distance from x to A, denoted by $d(x,A)$, is defined as $d(x,A) = \inf_{y\in A} \|x-y\|$, where $\|.\|$ is the Euclidean norm in $\mathbb{R}^2$. Let A and B be two nonempty subsets of $\mathbb{R}^2$. The *Hausdorff distance* between A and B, denoted by $d_{\mathcal{H}}(A,B)$ is defined as $d_{\mathcal{H}}(A,B) = \max\{\sup_{x\in A} d(x,B), \sup_{y\in B} d(y,A)\}$.

Regarding MMS for finding the shortest path joining two points p and q in a domain $\mathcal{D}$ [3, 6, 12], three following factors are included:

(f1) Partition of the domain $\mathcal{D}$ which is a polygon in 2D (polytope in 3D, resp.) into subpolygons (subpolytopes, resp.) is created by cutting segments (cutting slices, resp.). In each cutting segment (the boundary of cutting slice, resp.) we take a point that is called a shooting point.

(f2) Construct a path in $\mathcal{D}$ joining p and q, formed by the ordered set of shooting points. A stop condition called collinear condition (straightness condition, resp.) is established at shooting points.

(f3) The algorithm enforces (f2) at all shooting points to check the stop condition. Otherwise, an update of shooting points improves the paths joining p and q.

These factors which are established for the convex rope problem are described in the next section.

3. MMS-based algorithm for finding convex ropes

The proposed algorithm includes two following phases. We first construct a simple polygon such that the convex rope problem is referred to as the problem of finding the shortest path joining two points in the polygon. In the second phase, three factors (f1)–(f3) will be applied for the shortest path problem.

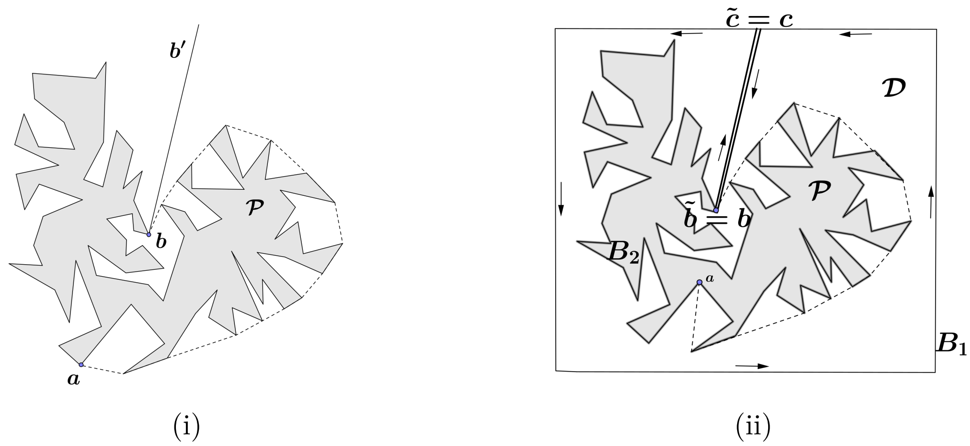


Figure 3.1: The counterclockwise convex rope starting at a and ending at b, the case (i): a is a vertex of the convex hull of $\mathcal{P}$ and the case (ii): a is visible from infinity.

3.1. Constructing a simple polygon $\mathcal{D}$

Take a rectangle R that contains $\mathcal{P}$ and has no edge touching $\mathcal{P}$. Since b is visible from infinity, the ray bb' always intersects with R at only one point, say c. The line segment $[b, c]$ partitions the non-simple region $R \setminus \text{int}\mathcal{P}$ into a simple region, similar to that of [19] as follows.

(A) We insert $[b, c]$ and its copy $[\tilde{b}, \tilde{c}]$ into the descriptions of the boundary of the region $R \setminus \text{int}\mathcal{P}$. Let $\mathcal{D}$ be a new simple polygon that is determined by a closed polyline including two parts: one part, denoted by B_1, is a polyline starting at $\tilde{b}$, crossing $\tilde{c}$, going along the boundary of R with the direction of arrows to come c and going to b; the other part, denoted by B_2, is the polyline starting at b going along the boundary of $\mathcal{P}$ via clockwise order and returning to $\tilde{b}$. Thus $\mathcal{D}$ has the boundary $B_1 \cup B_2$ as shown in Fig. 3.1(ii).

The non-simple region $R \setminus \text{int}\mathcal{P}$ is converted into the simple polygon $\mathcal{D}$. By Remark A.2 given in the Appendix, the convex rope starting at a and ending at b

is indeed the shortest path joining a and b in $\mathcal{D}$. Therefore the convex rope problem is deduced to the shortest path problem in $\mathcal{D}$. We can also model the convex rope problem as the shortest path problem in another polygon whose boundary includes the convex hull of $\mathcal{P}$ instead of the rectangle $\mathcal{R}$, see [8]. However computing the convex hull of $\mathcal{P}$ needs linear time whereas the construction of rectangle $\mathcal{R}$ takes constant time when input points are within a given range. According to Remark A.2 in the Appendix, from now on we focus on finding the shortest path joining $\tilde{b}$ and b in $\mathcal{D}$.

3.2. The factor (f1): partitioning the polygon $\mathcal{D}$

We split the polygon $\mathcal{D}$ into sub-polygons $\mathcal{D}_i$ by N cutting segments $\xi_1, \cdots, \xi_N$ ($N \geq 1$) as follows

$$\begin{aligned}
&+\ \xi_i = [u_i, v_i] \subset \mathcal{D} \text{ such that }]u_i, v_i[\subset \operatorname{int}\mathcal{D},\ u_i \in B_1 \text{ and } v_i \in B_2,\ \text{ for } 1 \leq i \leq N;\\
&+\ \xi_i \text{ strictly separates } \tilde{b} \text{ and } b,\ \text{for } 1 \leq i \leq N;\\
&+\ \xi_0 = \{\tilde{b}\}, \xi_{N+1} = \{b\}, [u_i, v_i] \cap [u_j, v_j] = \emptyset,\ \text{for } i \neq j,\ 1 \leq i, j \leq N;\\
&+\ \mathcal{D}_i \text{ is bounded by } B_1, B_2,\ \text{cut segments } \xi_i \text{ and } \xi_{i+1},\ \text{for } 0 \leq i \leq N;\\
&+\ \operatorname{int}\mathcal{D}_i \cap \operatorname{int}\mathcal{D}_j = \emptyset,\ \text{for } i \neq j,\ 0 \leq i, j \leq N \text{ and } \mathcal{D} = \textstyle\bigcup_{i=0}^{N} \mathcal{D}_i.
\end{aligned} \tag{1}$$

A partition given by (1) is illustrated in Fig. 3.2(i). In the first step, we take a point in each cutting segment. Two consecutive initial points are connected by the shortest path in the corresponding sub-polygon. The initial path is received by combining these shortest paths. For convenience, these initial points are chosen at endpoints v_i of ξ_i. For each iteration step, which is discussed carefully in next sections (3.3 and 3.4), we obtain an ordered set of points $\{a_i | \, a_i \in \xi_i, i = 1, 2, \ldots, N\}$ and a path $\gamma = \cup_{i=0}^{N} \mathrm{SP}(a_i, a_{i+1})$, where $\mathrm{SP}(a_i, a_{i+1})$ is the shortest path joining a_i and a_{i+1} in $\mathcal{D}_i$ and $a_0 = \tilde{b}, a_{N+1} = b$. Such a point a_i is called a *shooting point* and γ is called *the path formed by the set of shooting points* (see [2]).

It is clear that the shortest path joining a_i, a_{i+1} in $\mathcal{D}_i$ and that in $\mathcal{D}$ are identical. However, in the implementation, finding the shortest paths in these sub-polygons makes more efficient use of time and memory than that in an entire polygon. Here $\mathrm{SP}(a_i, a_{i+1})$ could be computed by any known algorithm for finding the shortest paths in simple polygons. Throughout this paper, when we say $\mathrm{SP}(x, y)$ without further explanation, it means the shortest path joining x and y in the connected union of sub-polygons containing x and y of $\mathcal{D}$.

3.3. The factor (f2): establishing and checking the collinear condition

Firstly, we present a so-called *collinear condition* inspired by Corollary 4.3 in [10]. Recall that $a_i \in [u_i, v_i]$, for all $i = 1, 2, \ldots, N$, where $a_0 = \tilde{b}, a_{N+1} = b$.

The *upper angle* created by $\mathrm{SP}(a_{i-1}, a_i)$ and $\mathrm{SP}(a_i, a_{i+1})$ with respect to $[u_i, v_i]$, denoted by $\angle(\mathrm{SP}(a_{i-1}, a_i), \mathrm{SP}(a_i, a_{i+1}))$, is defined to be the angle of the polyline $\mathrm{SP}(a_{i-1}, a_i) \cup \mathrm{SP}(a_i, a_{i+1})$ at a_i containing the ray $a_i u_i$. For each $i = 1, 2, \ldots, N$, the collinear condition states that:

(B) If $a_i \in]u_i, v_i[$, we say that a_i satisfies the collinear condition if the upper angle created by $\mathrm{SP}(a_{i-1}, a_i)$ and $\mathrm{SP}(a_i, a_{i+1})$ is equal to π, which can equivalently be written as $\angle(\mathrm{SP}(a_{i-1}, a_i), \mathrm{SP}(a_i, a_{i+1})) = \pi$.

If $a_i = v_i$, we say that a_i satisfies the collinear condition if the upper angle created by $\mathrm{SP}(a_{i-1}, a_i)$ and $\mathrm{SP}(a_i, a_{i+1})$ is at least π, or, equivalently, $\angle(\mathrm{SP}(a_{i-1}, a_i), \mathrm{SP}(a_i, a_{i+1})) \geq \pi$.

Let γ^* be the shortest path joining $\tilde{b}$ and b in $\mathcal{D}$, i.e., $\gamma^* = \mathrm{SP}(\tilde{b}, b)$. According to Remark A.2 in the Appendix, the Collinear Condition (B) does not include the case $a_i = u_i$.

3.4. The factor (f3): updating shooting points

Suppose that in j^{th}-iteration step, we have a set of shooting points $\{a_i\}_{i=1}^N$ and $\gamma = \cup_{i=0}^N \mathrm{SP}(a_i, a_{i+1})$ is the path formed by the set of shooting points, where $a_0 = \tilde{b}, a_{N+1} = b$. If Collinear Condition (B) holds at all shooting points, by Proposition 4.2 in Section 4, the path obtained by the algorithm is the shortest, and our algorithm stops. Otherwise, we update shooting points to get a path formed by the set of new shooting points in which the sequence of paths obtained converges to the shortest path, according to Theorems 4.11 and 4.12. Assuming that Collinear Condition (B) does not hold at all shooting points, we update shooting points as follows

(C) For $i = 1, 2 \ldots, N$, fix $t_{i-1} \in \mathrm{SP}(a_{i-1}, a_i), t_i \in \mathrm{SP}(a_i, a_{i+1})$ such that t_{i-1}, t_i are not identical to shooting points. We call such point t_{i-1} (t_i, resp.) the *temporary point* with respect to a_{i-1} (a_i, resp.). Let $a_i^{next} = \mathrm{SP}(t_{i-1}, t_i) \cap [u_i, v_i]$.
The intersection point a_i^{next} exists uniquely since $[u_i, v_i]$ divides $\mathcal{D}$ into two parts, one that contains t_{i-1} and one that contains t_i. We update shooting point a_i to a_i^{next} (see Fig. 3.2(iii)).

In particular, if $\mathrm{SP}(a_i, a_{i+1})$ is a line segment, we choose t_i as the midpoint of $\mathrm{SP}(a_i, a_{i+1})$. Otherwise, t_i is chosen as an endpoint of $\mathrm{SP}(a_i, a_{i+1})$ that is not identical to a_i and a_{i+1}. We can update shooting points in which the collinear condition does not hold and keep the remaining shooting points. But the proposed algorithm will update all shooting points. Because if a_i satisfies Collinear Condition (B), then the update gives $a_i^{next} = a_i$ due to Proposition 4.3 (in Section 4).

3.5. The proposed algorithm

Require: A simple polygon $\mathcal{P}$, $b \in \mathcal{P}$ and its copy $\tilde{b}$, an integer number $N \geq 1$.

Ensure: An approximate convex rope starting at $\tilde{b}$ and ending at b.

1: Construct a simple polygon $\mathcal{D}$ with the boundary $B_1 \cup B_2$ as shown in (A), ▷ see Fig. 3.1(ii).

2: Divide $\mathcal{D}$ into $\mathcal{D}_i$ satisfying (1) by cutting segments $[u_i, v_i]$, where $u_i \in B_1$, $v_i \in B_2$ $(i = 1, \ldots, N)$, ▷ see Fig. 3.2(i).

3: Take an ordered set $\{a_i\}_{i=0}^{N+1}$ initial shooting points consisting of $\tilde{b}$, v_i, $(i = 1, 2, \ldots, N)$ and b. Let $\gamma^{current}$ be the path formed by $\{a_i\}_{i=0}^{N+1}$, ▷ see Fig. 3.2(ii).

4: $flag \leftarrow true$, $a_i^{next} \leftarrow a_i$ and γ^{next} is the path formed by $\{a_i^{next}\}_{i=0}^{N+1}$.

5: Call Procedure COLLINEAR_UPDATE$(\gamma^{current}, \gamma^{next}, flag)$, ▷ see Figs. 3.2(iii) and (iv).

6: If Collinear Condition(B) holds at all shooting points, i.e., $flag = true$, **stop** and **return** γ^{next}, ▷ the shortest path is obtained by Proposition 4.2

7: Otherwise, i.e, $flag = false$, then $\gamma^{current} \leftarrow \gamma^{next}$ and **go to** step 5.

8: **return** γ^{next}.

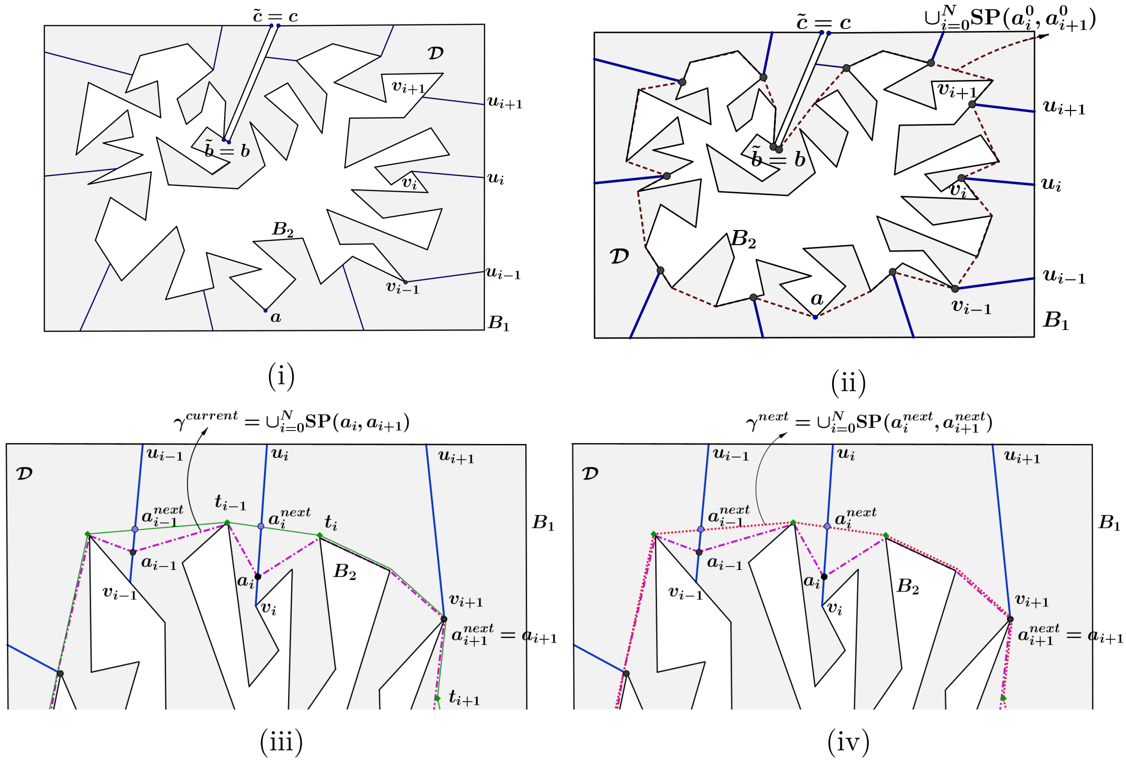


Figure 3.2: (i): Illustration of partitioning $\mathcal{D}$; (ii): the set of initial shooting points consisting of $\tilde{b}$, $v_i, (i = 1, 2, \ldots, N)$ and b (black big dots) and the path formed by the set (dashed polyline); (iii): illustration of taking temporary points and finding $a_i^{next} = \mathrm{SP}(t_{i-1}, t_i) \cap \xi_i$; (iv): the update $\gamma^{current}$ (dashed polyline) to γ^{next} (solid polyline).

For each iteration step, we need to check Collinear Condition (B) and update shooting points given by (C) to get a better path.

Then the Procedure COLLINEAR_UPDATE($\gamma^{current}, \gamma^{next}, flag$) of the proposed algorithm performs checking condition (B) and updating the shooting point a_i to a_i^{next}.

4. The correctness of the proposed algorithm

Lemma 4.1 is presented without a proof (since the proof is similar to that of Proposition 2.19, [2]).

Lemma 4.1. *Suppose that γ is the shortest path joining two given points p and q in a polygon $\mathcal{A}$ such that γ intersects a cutting segment $[u, v]$ at a point, say a. If $a \in]u, v[$, then $\angle(\gamma(p, a), \gamma(a, q)) = \pi$.*

The following proposition shows that the path satisfying Collinear Condition (B) is the shortest path γ^* joining $\tilde{b}$ and b.

Proposition 4.2. *Let $\gamma^{current}$ be a path formed by a set of shooting points, which joins $\tilde{b}$ and b in $\mathcal{D}$. If Collinear Condition* (B) *holds at all shooting points, then $\gamma^{current} = \gamma^*$.*

procedure COLLINEAR_UPDATE($\gamma^{current}, \gamma^{next}, flag$)
  $flag \leftarrow true$
  for $i = 0, 1, \ldots, N$ **do**
    Take temporary points t_i on $\mathrm{SP}(a_i, a_{i+1})$ such that if $\mathrm{SP}(a_i, a_{i+1})$ is a line segment, t_i is the midpoint of $\mathrm{SP}(a_i, a_{i+1})$, for if not, t_i is an endpoint of $\mathrm{SP}(a_i, a_{i+1})$ that is not identical to a_i and a_{i+1}.
    if $a_i \in]u_i, v_i[$ **then**
      if $\angle(\mathrm{SP}(a_{i-1}, a_i), \mathrm{SP}(a_i, a_{i+1})) = \pi$ **then** ▷ Collinear Condition (B) is checked
        Set $a_i^{next} \leftarrow a_i$
      else call $\mathrm{SP}(t_{i-1}, t_i)$
        Set a_i^{next} be the intersection point of $\mathrm{SP}(t_{i-1}, t_i)$ and $[u_i, v_i]$ ▷ due to (C)
        $flag \leftarrow false$
    else ▷ $a_i = v_i$
      if $\angle(\mathrm{SP}(a_{i-1}, a_i), \mathrm{SP}(a_i, a_{i+1})) \geq \pi$ **then** ▷ Collinear Condition (B) is checked
        Set $a_i^{next} \leftarrow a_i$
      else call $\mathrm{SP}(t_{i-1}, t_i)$
        Set a_i^{next} the intersection point of $\mathrm{SP}(t_{i-1}, t_i)$ and $[u_i, v_i]$ ▷ due to (C)
        $flag \leftarrow false$
  $a_0^{next} = \tilde{b},\ a_{N+1}^{next} = b$

Proof. Let $\gamma^{current} = \cup_{i=0}^{N} \mathrm{SP}(a_i, a_{i+1})$ where $\{a_i\}_{i=1}^{N}$ is a set of shooting points and $a_0 = \tilde{b}, a_{N+1} = b$. Assume the contrary that $\gamma^{current}$ and γ^* are distinct. There are two distinct points u, v which are common points to both $\gamma^{current}$ and γ^* such that $\gamma^{current}(u, v) \cap \mathrm{SP}(u, v) = \{u, v\}$, where $\gamma^{current}(u, v)$ is the sub-path of $\gamma^{current}$ from u to v. Note that $\mathrm{SP}(u, v)$ is also a sub-path of γ^*. Two points u and v always exist and may be identical to $\tilde{b}$ and b, respectively. Then $\gamma^{current}(u, v)$ and $\mathrm{SP}(u, v)$ form a simple polygon with m vertices $p_1 = u, p_2, \ldots, p_k = v, p_{k+1} \ldots, p_m$ (see Fig. 4.1).

Let α_j be the internal angle at p_j, with $j = 1, 2, \ldots, m$. By the properties of shortest paths, if $p_j \in \mathrm{SP}(u, v)$ then vertices of $\mathrm{SP}(u, v)$ except for u and v are reflex vertices of $\mathrm{SP}(u, v)$, i.e., $\alpha_j > \pi$. If $p_j \in \gamma^{current}(u, v)$, then p_j is either a shooting point or a reflex vertex of $\mathrm{SP}(a_i, a_{i+1})$ with some index $i \in \{1, 2, \ldots, N\}$. If p_j is a shooting point, then $\alpha_j \geq \pi$ since Collinear Condition (B) holds at all shooting points. The equality does not occur, because p_j is a vertex of $\gamma^{current}$, and thus $\alpha_j > \pi$.

If p_j is a reflex vertex of $\mathrm{SP}(a_i, a_{i+1})$ with some index $i \in \{1, 2, \ldots, N\}$, we also get $\alpha_j > \pi$. Hence $\alpha_j > \pi$ for $j \in \{1, 2, \ldots, m\} \setminus \{1, k\}$. Since the sum of the measures of the internal angles of the simple polygon with m vertices is $(m-2)\pi$, we have $(m-2)\pi = \sum_{j=1}^{m} \alpha_i > \alpha_1 + (m-2)\pi + \alpha_k > (m-2)\pi$. The contradiction deduces $\gamma^{current} = \gamma^*$. □

Proposition 4.3. *Collinear Condition* (B) *holds at* a_i *if and only if we have* $\mathrm{SP}(t_{i-1}, t_i) \cap [u_i, v_i] = a_i$, *where* t_{i-1} *and* t_i *are temporary points w.r.t.* a_{i-1} *and* a_i.

Proof. ($\Rightarrow$) Assuming that Collinear Condition (B) holds at a_i, to show that $\mathrm{SP}(t_{i-1}, t_i) \cap [u_i, v_i] = a_i$, we just need to prove that $\mathrm{SP}(t_{i-1}, t_i) = \mathrm{SP}(t_{i-1}, a_i) \cup \mathrm{SP}(a_i, t_i)$. Assume the contrary, we use the same way of the proof of Proposition 4.2 and the property that the sum of the internal angles in a simple polygon with m vertices is $(m-2)\pi$ to obtain a contradiction.

($\Leftarrow$) Assume that $\mathrm{SP}(t_{i-1}, t_i) \cap [u_i, v_i] = a_i$. Therefore a_i belongs to $\mathrm{SP}(t_{i-1}, t_i)$. If $a_i = v_i$, then $\angle(\mathrm{SP}(t_{i-1}, a_i^{next}), \mathrm{SP}(a_i^{next}, t_i)) \geq \pi$ by the property of shortest paths. If $a_i \in]u_i, v_i[$, then $\angle(\mathrm{SP}(t_{i-1}, a_i^{next}), \mathrm{SP}(a_i^{next}, t_i)) = \pi$ due to Lemma 4.1. Thus Collinear Condition (B) holds at a_i. □

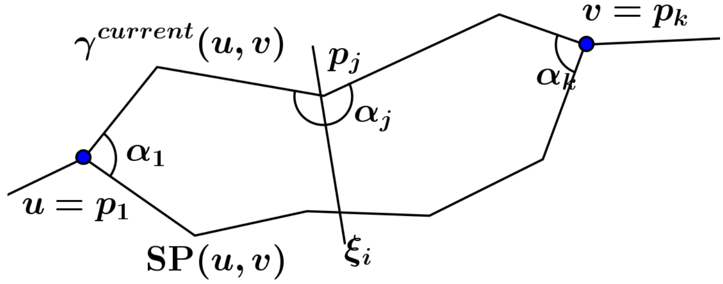


Figure 4.1: Illustration of the proof of Proposition 4.2.

To state and prove Theorems 4.11 and 4.12 we need Proposition 4.4, Proposition 4.7 and Corollary 4.8. Next, assuming that a partition of $\mathcal{D}$ is given by (1), we have

Proposition 4.4. *We have $\gamma^* \subset \cup_{i=0}^N \mathcal{D}_i$, where $\mathcal{D}_i$ is bounded by B_1, B_2 and the cut segments ξ_i and ξ_{i+1}. Moreover there exists a set $\{a_i^* \in \xi_i,\ i = 0, 1, \ldots, N+1\}$ of points, where $a_0^* = \tilde{b}$, $a_{N+1}^* = b$, such that $\gamma^* = \cup_{i=0}^N \mathrm{SP}(a_i^*, a_{i+1}^*)$, in which $\mathrm{SP}(a_i^*, a_{i+1}^*)$ is the shortest path joining a_i^* and a_{i+1}^* in $\mathcal{D}_i$, for $i = 0, 1, \ldots, N (N \geq 1)$.*

Proof. Because of (1), ξ_i divides $\mathcal{D}$ into two parts, one contains $\tilde{b}$ and the other contains b. Hence the intersection of γ^* and ξ_i is not empty. As γ^* is the shortest path joining $\tilde{b}$ and b in $\mathcal{D}$, the intersection is only one point. Let $a_i^* = \gamma^* \cap \xi_i$, for $i = 1, 2, \ldots, N$, where $a_0^* = \tilde{b}, a_{N+1}^* = b$. Then $\gamma^* = \cup_{i=0}^N \mathrm{SP}(a_i^*, a_{i+1}^*)$, where $\mathrm{SP}(a_i^*, a_{i+1}^*)$ is the shortest path joining a_i^* and a_{i+1}^* in $\mathcal{D}$, for $i = 0, 1, \ldots, N$. Since the construction of $\mathcal{D}_i$, we deduce that $\mathrm{SP}(a_i^*, a_{i+1}^*) \subset \mathcal{D}_i$. □

The converse of Proposition 4.4 is not true for the general case. It means that if there is a set $\{a_i \in \xi_i\}_{i=0}^N$, $\gamma^{current} = \cup_{i=0}^N \mathrm{SP}(a_i, a_{i+1})$, then e cannot conclude that $\gamma^{current}$ is γ^*. However, if a_i satisfies Collinear Condition (B), for all $i = 1, 2, \ldots, N$, we have $\gamma^{current} = \gamma^*$ by Proposition 4.2.

Remark 4.5. Suppose that $a_i \in \xi_i$, $a_{i+1} \in \xi_{i+1}$ are shooting points in some iteration step, and $a_i^{next} \in \xi_i, a_{i+1}^{next} \in \xi_{i+1}$ are shooting points at the next iteration step of the proposed algorithm such that $a_i^{next} \in [a_i, u_i]$ and $a_{i+1}^{next} \in [a_{i+1}, u_{i+1}]$, see Figs. 3.2(iii) and (iv). Then we have

(a) $\mathrm{SP}(a_i, a_{i+1})$ and $\mathrm{SP}(a_i^{next}, a_{i+1}^{next})$ are convex with convexity facing towards B_1.

(b) The segment $\mathrm{SP}(a_i^{next}, a_{i+1}^{next})$ is entirely contained in the polygon bounded by $[u_i, a_i]$, $\mathrm{SP}(a_i, a_{i+1})$, $[a_{i+1}, u_{i+1}]$, and B_1. Thus we say that $\mathrm{SP}(a_i^{next}, a_{i+1}^{next})$ is above $\mathrm{SP}(a_i, a_{i+1})$ when we view from B_1.

Lemma 4.6. *Recall that the upper angle created by* $\mathrm{SP}(a_{i-1}, a_i)$ *and* $\mathrm{SP}(a_i, a_{i+1})$ *w.r.t.* $[u_i, v_i]$ *is denoted by* $\angle(\mathrm{SP}(a_{i-1}, a_i), \mathrm{SP}(a_i, a_{i+1}))$*. For* $i = 1, 2, \ldots, N$*, we suppose that* $a_i \in]u_i, v_i[$*. Let* $\hat{a}_i = SP(t_{i-1}, t_i) \cap [u_i, v_i]$*, where* t_{i-1} *and* t_i *are temporary points w.r.t.* a_{i-1} *and* a_i*. Then we have*

(a) $\angle(\mathrm{SP}(a_{i-1}, a_i), \mathrm{SP}(a_i, a_{i+1})) = \pi \Leftrightarrow \hat{a}_i = a_i$*;*

(b) $\angle(\mathrm{SP}(a_{i-1}, a_i), \mathrm{SP}(a_i, a_{i+1})) < \pi \Leftrightarrow \hat{a}_i \in]a_i, u_i[$*;*

(c) $\angle(\mathrm{SP}(a_{i-1}, a_i), \mathrm{SP}(a_i, a_{i+1})) > \pi \Leftrightarrow \hat{a}_i \in [v_i, a_i[$*.*

Proof. When Collinear Condition (B) does not hold, the role of $\hat{a}_i$ and a_i^{next} are the same. By Proposition 4.3, Collinear Condition (B) in Section 3.3, and the fact that $a_i \neq v_i$, we get (a).

(b) ($\Rightarrow$) Suppose that $\angle(\mathrm{SP}(a_{i-1}, a_i), \mathrm{SP}(a_i, a_{i+1})) < \pi$ and $\hat{a}_i \notin]a_i, u_i]$. Because $\mathrm{SP}(t_{i-1}, t_i)$ never touches B_1 of $\mathcal{D}$, similarly to Remark A.1, then $\hat{a}_i \neq u_i$. Therefore $\hat{a}_i \in [v_i, a_i]$. Clearly, the collinear condition does not hold at a_i. Therefore $a_i \neq \hat{a}_i$, $\mathrm{SP}(t_{i-1}, t_i)$ and $\mathrm{SP}(t_{i-1}, a_i) \cup \mathrm{SP}(a_i, t_i)$ are thus distinct. Then there are two distinct points u, v which are common points to both $\mathrm{SP}(t_{i-1}, t_i)$ and $\mathrm{SP}(t_{i-1}, a_i) \cup \mathrm{SP}(a_i, t_i)$ such that $\mathrm{SP}(u, v) \cap (\mathrm{SP}(u, a_i) \cup \mathrm{SP}(a_i, v)) = \{u, v\}$. Note that $\mathrm{SP}(u, v)$ is also a sub-path of $\mathrm{SP}(t_{i-1}, t_i)$. Then $\mathrm{SP}(u, v)$ and $\mathrm{SP}(u, a_i) \cup \mathrm{SP}(a_i, v)$ form a simple polygon in $\mathcal{D}$ with m vertices $p_1 = u, p_2, \ldots, p_l = a_i, p_{l+1} \ldots, p_k = v, p_{k+1}, \ldots, p_m$. Let α_j be the internal angle at p_j, with $j = 1, 2, \ldots, m$, see Fig 4.1. According to the property of shortest paths, $\alpha_j > \pi$ for $j \in \{1, 2, \ldots, m\} \setminus \{1, k, l\}$. If $j = l$, due to $\alpha_j = 2\pi - \angle(\mathrm{SP}(a_{i-1}, a_i), \mathrm{SP}(a_i, a_{i+1}))$, we have $\alpha_j > \pi$. Then $\alpha_j > \pi$ for $j \in \{1, 2, \ldots, m\} \setminus \{1, k\}$. It is similar to the proof of Proposition 4.2, a contradiction is obtained. Thus $\hat{a}_i \in]a_i, u_i[$.

(c) ($\Rightarrow$) Suppose that $\angle(\mathrm{SP}(a_{i-1}, a_i), \mathrm{SP}(a_i, a_{i+1})) > \pi$ and $\hat{a}_i \notin]v_i, a_i[$. A contradiction can be deduced in much the same way as the above proof. Therefore $\hat{a}_i \in]v_i, a_i[$.

(b)(c) ($\Leftarrow$) For the sufficient condition of (b), ((c) can be considered similarly), assume that $\hat{a}_i \in]a_i, u_i[$. Then we have $\angle(\mathrm{SP}(a_{i-1}, a_i), \mathrm{SP}(a_i, a_{i+1})) < \pi$, because otherwise, $\angle(\mathrm{SP}(a_{i-1}, a_i), \mathrm{SP}(a_i, a_{i+1})) \geq \pi$ deduces that $\hat{a}_i \in [v_i, a_i]$ by the necessary conditions of (a) and (c) that are proved above, which is impossible. Hence the proof is complete. □

Proposition 4.7. *For all* j*, we have* γ^{j+1} *is above* γ^j *when we view from* B_1*, i.e.,* $a_i^{j+1} \in [a_i^j, u_i[$*, for* $i = 1, 2, \ldots, N$*, where* γ^j *is the path obtained in* j^{th}*-iteration step of the algorithm.*

Proof. We give a proof by induction on j. The statement holds for $j = 1$ due to taking initial shooting points $a_i^0 = v_i$, for $i = 1, 2, \ldots, N$. Let $\{a_i^{prev}\}_{i=1}^N, \{a_i\}_{i=1}^N$ and $\{a_i^{next}\}_{i=1}^N$ be the sets of shooting points corresponding to $k - 1^{th}$, k^{th} and $k + 1^{th}$-iteration steps of the algorithm, respectively ($k \geq 1$). Assuming that, for $i = 1, 2, \ldots, N$, $a_i \in [a_i^{prev}, u_i[$, we next prove that $a_i^{next} \in [a_i, u_i[$.

If we have $\angle(\mathrm{SP}(a_{i-1}, a_i), \mathrm{SP}(a_i, a_{i+1})) \leq \pi$, due to Lemma 4.6 (a) and (b), we obtain $a_i^{next} \in [a_i, u_i[$. Thus we just need to prove that $a_i^{next} \in [a_i, u_i[$ in the case $\angle(\mathrm{SP}(a_{i-1}, a_i), \mathrm{SP}(a_i, a_{i+1})) > \pi$. If $a_i = v_i$, then Collinear Condition (B) holds at a_i then $a_i^{next} = a_i \in [a_i, u_i[$.

If $a_i \neq v_i$, suppose contrary to our claim, that there is an index i such that we have $\angle(\mathrm{SP}(a_{i-1}, a_i), \mathrm{SP}(a_i, a_{i+1})) > \pi$ and $a_i^{next} \notin [a_i, u_i[$, i.e.,

$$a_i^{next} \in [v_i, a_i[\tag{2}$$

Let G_1 be the closed region bounded by $[a_i^{prev}, a_i]$, $\mathrm{SP}(a_i, t_{i-1})$ and $\mathrm{SP}(t_{i-1}, a_i^{prev})$. Let G_2 be the closed region bounded by $[a_i^{prev}, a_i]$, $\mathrm{SP}(a_i, t_i)$ and $\mathrm{SP}(t_i, a_i^{prev})$, where t_{i-1} and t_i are temporary points w.r.t. a_{i-1}^{prev} and a_i^{prev} (see Fig. 4.2). Since $a_i \in]u_i, v_i[$, $\angle(\mathrm{SP}(t_{i-1}, a_i), \mathrm{SP}(a_i, t_i)) = \pi$ by Lemma 4.1.

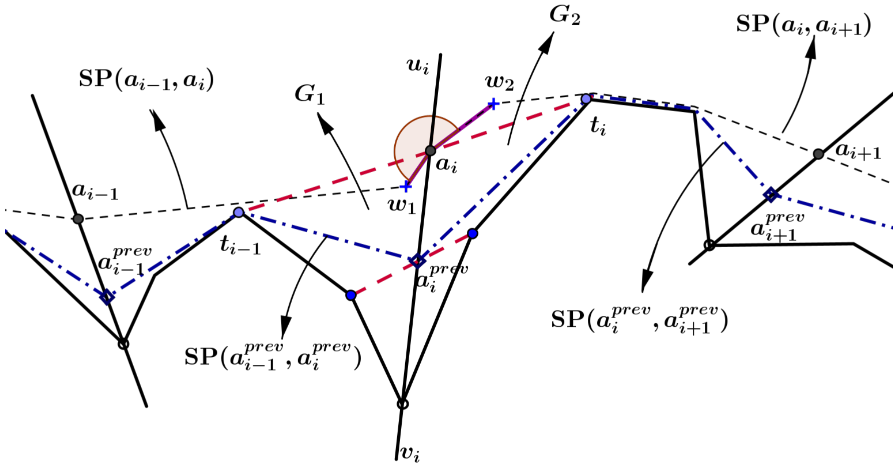


Figure 4.2: Illustration of the proof of Proposition 4.7 in the case $a_i^{prev} \neq a_i$

Next, let $[a_i, w_1]$ and $[a_i, w_2]$ be segments having the common endpoint a_i of the segments $\mathrm{SP}(a_{i-1}, a_i)$ and $\mathrm{SP}(a_i, a_{i+1})$, respectively. As $\angle(\mathrm{SP}(t_{i-1}, a_i), \mathrm{SP}(a_i, t_i)) = \pi$ and $\angle(\mathrm{SP}(a_{i-1}, a_i), \mathrm{SP}(a_i, a_{i+1})) > \pi$, there are at least one of two following cases that will happen:

$$]a_i, w_1] \text{ is below entirely } \mathrm{SP}(t_{i-1}, a_i) \tag{3}$$
$$\text{or }]a_i, w_2] \text{ is below entirely } \mathrm{SP}(a_i, t_i) \text{ when we view from } u_i.$$

Then $\mathrm{SP}(a_i, t_i)$, $\mathrm{SP}(a_i, a_{i+1})$ and $\mathrm{SP}(a_i^{prev}, t_i)$ do not cross each other, due to the properties of shortest paths. According to Remark 4.5(a), for any temporary point $t_i \in \mathrm{SP}(a_i^{prev}, a_{i+1}^{prev})$, there is a polygon bounded by $[a_i^{prev}, a_i]$, $[a_{i+1}, a_{i+1}^{prev}]$, $\mathrm{SP}(a_i, a_{i+1})$, and $\mathrm{SP}(a_i^{prev}, a_{i+1}^{prev})$ which contains entirely $\mathrm{SP}(a_i, t_i)$. Thus $\mathrm{SP}(a_i, a_{i+1})$ does not intersect with the interior of G_2. Similarly, $\mathrm{SP}(a_{i-1}, a_i)$ does not also intersect with the interior of G_1. These things contradict (3), which completes the proof. □

Corollary 4.8. *Let $\gamma^{current}$ be the path obtained in some iteration step of the algorithm, where $\gamma^{current} = \cup_{i=0}^{N} \mathrm{SP}(a_i, a_{i+1})$.*
If $a_i \in]v_i, u_i[$, then we have $\angle(\mathrm{SP}(a_{i-1}, a_i), \mathrm{SP}(a_i, a_{i+1})) \leq \pi$, for $i = 1, 2, \ldots, N$.

Lemma 4.9. [Proposition 5.1, [3]] *If a, b, c, and d are points in a simple polygon $\mathcal{P}$ such that $[a, d], [b, c] \subset \mathcal{P}$, then*

$$d_{\mathcal{H}}(\mathrm{SP}(a, b), \mathrm{SP}(c, d)) \leq \|a - d\| + \|b - c\| \leq 2\max\{\|a - d\|, \|b - c\|\}.$$

Lemma 4.10. [Lemma 3.1, [9]] *If $\{a_n\}$ and $\{b_n\}$ are sequences of points in a simple polygon, $a_n \to a$ and $b_n \to b$, then $d_{\mathcal{H}}(\mathrm{SP}(a_n, b_n), \mathrm{SP}(a, b)) \to 0$ as $n \to +\infty$.*

Theorem 4.11. *The algorithm is convergent. It means that if the algorithm stops after a finite number of iteration steps (i.e., Collinear Condition (B) holds at all shooting points), we obtain the shortest path, otherwise, the sequence of paths* $\{\gamma^j\}$ *converges to* γ^* *in Hausdorff distance (i..e.,* $d_{\mathcal{H}}(\gamma^j, \gamma^*) \to 0$ *as* $j \to \infty$*, where* γ^j *is a path obtained by the algorithm in the* j^{th}*-iteration step).*

Proof. Let $\gamma^j = \cup_{i=0}^{N} \mathrm{SP}(a_i^j, a_{i+1}^j)$, where $a_i^j \in [u_i, v_i]$ for $i = 1, 2, \ldots, N$ and $a_0^j = \tilde{b}$, $a_N^j = b$. If the algorithm stops after a finite number of iterations, the path obtained is the shortest. We thus just need to consider the case that $\{\gamma^j\}_{j\in\mathbb{N}}$ is infinite, according to Proposition 4.2.

For each $i = 1, 2, \ldots, N$, since $[u_i, v_i]$ is compact and $\{a_i^j\}_{j\in\mathbb{N}} \subset [u_i, v_i]$ there exists a sub sequence $\{a_i^{j_k}\}_{k\in\mathbb{N}} \subset \{a_i^j\}_{j\in\mathbb{N}}$ such that $\{a_i^{j_k}\}_{k\in\mathbb{N}}$ converges to a point, say a_i^*, in $\mathbb{R}^2$ with $\|.\|$ as $k \to \infty$. As $[u_i, v_i]$ is closed, $\{a_i^{j_k}\}_{k\in\mathbb{N}} \subset [u_i, v_i]$, and $\{a_i^{j_k}\}_{k\in\mathbb{N}} \to a_i^*$, we get $a_i^* \in [u_i, v_i]$. Write $a_0^* = \tilde{b}$ and $a_{N+1}^* = b$. Set $\hat{\gamma} = \cup_{i=0}^{N} \mathrm{SP}(a_i^*, a_{i+1}^*)$.

i. We begin by proving the convergence of whole sequences,i.e., $\{a_i^j\}_{j\in\mathbb{N}} \to a_i^*$ as $j \to \infty$, for $i = 1, 2, \ldots, N$, based on the order of elements of $\{a_i^j\}_{j\in\mathbb{N}}$ shown in Proposition 4.7 and the convergence of their subsequence. By the order of elements of $\{a_i^j\}_{j\in\mathbb{N}}$ as shown in Proposition 4.7, for all natural numbers large enough j, there exits $k \in \mathbb{N}$ such that $a_i^j \in [a_i^{j_k}, a_i^{j_{k+1}}]$. Furthermore, we also obtain $\{a_i^{j_k}\}_{k\in\mathbb{N}}$ converges on one side to a_i^*. For all $\delta > 0$, there exists $k_0 \in \mathbb{N}$ such that $\|a_i^{j_k} - a_i^*\| < \delta$, for $k \geq k_0$. As $a_i^j \in [a_i^{j_{k_0}}, a_i^*]$, for $j \geq j_{k_0}$, we get $\|a_i^j - a_i^*\| < \delta$, for $j \geq j_{k_0}$. This clearly forces $\{a_i^j\}_{j\in\mathbb{N}} \to a_i^*$ as $j \to \infty$. This is the one-sided convergence, γ^j is thus below $\hat{\gamma}$ when we view from the boundary B_1.

ii. We next indicate that $d_{\mathcal{H}}(\gamma^j, \hat{\gamma}) \to 0$ as $j \to \infty$. According to item **i**, $a_i^j \to a_i^*$ as $j \to \infty$, for $i = 1, 2, ..., N$. By Lemma 4.10, $d_{\mathcal{H}}\left(\mathrm{SP}(a_i^j, a_{i+1}^j), \mathrm{SP}(a_i^*, a_{i+1}^*)\right) \to 0$ as $j \to \infty$. Note that $d_{\mathcal{H}}(A \cup B, C \cup D) \leq \max\{d_{\mathcal{H}}(A, C), d_{\mathcal{H}}(B, D)\}$, for all closed sets A, B, C and D in $\mathbb{R}^2$, we have

$$d_{\mathcal{H}}(\gamma^j, \hat{\gamma}) \leq \max_{0\leq i\leq N}\{d_{\mathcal{H}}\left(\mathrm{SP}(a_i^j, a_{i+1}^j), \mathrm{SP}(a_i^*, a_{i+1}^*)\right)\}.$$

Therefore $d_{\mathcal{H}}(\gamma^j, \hat{\gamma}) \to 0$ as $j \to \infty$.

iii. Our next claim is that $\hat{\gamma}$ is the shortest path joining $\tilde{b}$ and b. According to Proposition 4.2, we need to prove that Collinear Condition (B) holds at all a_i^*, for $i \in \{1, 2, \ldots, N\}$. Conversely, suppose that there exists an index $i \in \{1, 2, \ldots, N\}$ such that Collinear Condition (B) does not hold at a_i^*. If $a_i^* = v_i$ then $a_i^j = v_i$ for all $j \geq 0$. Then Collinear Condition (B) holds at a_i^j for all $j \geq 0$, i.e., $\angle\left(\mathrm{SP}(a_{i-1}^j, a_i^j), \mathrm{SP}(a_i^j, a_{i+1}^j)\right) \geq \pi$, for all $j \geq 0$. Since γ^j is below $\hat{\gamma}$ when we view from the boundary B_1 and $\{a_{i-1}^j\}_{j\in\mathbb{N}} \to a_{i-1}^*$, $\{a_{i+1}^j\}_{j\in\mathbb{N}} \to a_{i+1}^*$ as $j \to \infty$, we get the upper angle of $\hat{\gamma}$ at a_i^* is not less than π, which contradicts to Collinear Condition (B) not holding at a_i^*. Thus $a_i^* \neq v_i$. As Collinear Condition (B) does not hold at a_i^*, we conclude $\angle\left(\mathrm{SP}(a_{i-1}^*, a_i^*), \mathrm{SP}(a_i^*, a_{i+1}^*)\right) < \pi$ or $\angle\left(\mathrm{SP}(a_{i-1}^*, a_i^*), \mathrm{SP}(a_i^*, a_{i+1}^*)\right) > \pi$.

Case 1: $\angle\left(\mathrm{SP}(a_{i-1}^*, a_i^*), \mathrm{SP}(a_i^*, a_{i+1}^*)\right) < \pi$. To obtain a contradiction, we will show that there exists a natural number j_0 such that the updating $a_i^{j_0}$ to $a_i^{j_0+1}$ gives $a_i^{j_0+1} \in (a_i^*, u_i]$.

Let V be the set of all vertices of $\mathcal{D}$ which do not belong to $\hat{\gamma}$. Set

$$\mu = \min\{d(v,\hat{\gamma}) : v \in V\}. \tag{4}$$

Because V is finite, $d(v,\hat{\gamma}) > 0$, for $v \in V$, we get $\mu > 0$. Let $\epsilon_0 = \mu/4$. Since $a_i^* \neq v_i$, there is a point in $[v_i, a_i^*]$, say c_i, such that $\|c_i - a_i^*\| = \epsilon_0$ (see Fig. 4.3). Through c_i, we draw a polyline $\mathcal{L}$ whose line segments are parallel to line segments of $\mathrm{SP}(a_{i-1}^*, a_i^*) \cup \mathrm{SP}(a_i^*, a_{i+1}^*)$, respectively (see Fig. 4.4(i)).

Let us denote by c_{i-1} (c_{i+1}, resp.) the point of intersection of $\mathcal{L}$ and the line going through $[u_{i-1}, v_{i-1}]$ ($[u_{i+1}, v_{i+1}]$, resp.). Since a_i^* is very closed to c_i, w.l.o.g we can suppose that there is a line going through a_i^* and creating with $\mathcal{L}$ an triangle, say $\triangle c_i d d'$ (see Fig. 4.3). If not, we can take

$$\epsilon_0 = \frac{\mu}{2^{K_0}}, \text{ where } K_0 \text{ is a large enough natural number.} \tag{5}$$

For simplicity, assume that $\triangle c_i d d'$ is an isosceles triangle, (see Fig. 4.3).

Claim 1. *Let* $\alpha^- = \angle\left(SP(a_{i-1}^*, a_i^*), [a_i^*, u_i]\right)$ *and* $\alpha^+ = \angle\left([a_i^*, u_i], SP(a_i^*, a_{i+1}^*)\right)$. *In* $\triangle c_i d d'$, *we obtain*

$$\|c_i - d\| = \epsilon_0 . \frac{\sin\left(\frac{1}{2}(\pi - \alpha^- - \alpha^+)\right)}{\sin\left(\frac{1}{2}(\pi - \alpha^- + \alpha^+)\right)}. \tag{6}$$

Proof of Claim 1. Use the Law of Sines in a triangle. □

We return to the proof. Since $\{a_i^j\}_{j\in\mathbb{N}} \to a_i^*$ as $j \to \infty$ for $i = 1, 2, \ldots, N$, there exists $\overline{j} \in \mathbb{N}$ such that

$$\|a_i^j - a_i^*\| < \frac{\epsilon_0}{M}, \text{ for all } j \geq \overline{j}, \tag{7}$$

where M is a given real number which is chosen as in Claim 2 below and note that M only depends on the segments $[u_{i-1}, v_{i-1}], [u_i, v_i]$, and $[u_{i+1}, v_{i+1}]$.

Claim 2. *If* $[u_{i-1}, v_{i-1}]$ *is parallel to* $[u_i, v_i]$, *then we set* $m_1 = 1$. *Otherwise let the point of intersection of two lines going through* $[u_{i-1}, v_{i-1}]$ *and* $[u_i, v_i]$ *be* s_1 *and we set* $m_1 = \|s_1 - a_i^*\|/\|s_1 - a_{i-1}^*\|$. *Similarly, if* $[u_{i+1}, v_{i+1}]$ *is parallel to* $[u_i, v_i]$, *then we set* $m_2 = 1$. *Otherwise let the point of intersection of two lines going through* $[u_{i+1}, v_{i+1}]$ *and* $[u_i, v_i]$ *be* s_2 *and we set* $m_2 = \|s_2 - a_i^*\|/\|s_2 - a_{i+1}^*\|$. *Take* $M = \max\{1, m_1, m_2\}$. *Then* $\mathrm{SP}(a_{i-1}^j, a_i^j)$ *and* $\mathrm{SP}(a_i^j, a_{i+1}^j)$ *are contained in the region bounded by* $\mathcal{L}, [c_{i+1}, a_{i+1}^*], \mathrm{SP}(a_{i-1}^*, a_i^*) \cup \mathrm{SP}(a_i^*, a_{i+1}^*)$, *and* $[a_{i-1}^*, c_{i-1}]$, *for all* $j \geq \overline{j}$, *where* $\overline{j}$ *is given by* (7).

Proof of Claim 2. For simplicity, we only prove the case that $[u_{i-1}, v_{i-1}]$ is parallel to $[u_i, v_i]$ and two lines going through $[u_{i+1}, v_{i+1}]$ and $[u_i, v_i]$ intersect at s_2 (see Fig. 4.4(i)). One sees immediately that $m_1 = 1$ and $\|c_{i-1} - a_{i-1}^*\| = \|c_i - a_i^*\| = \epsilon_0$.

By (7), we get $\|a_i^j - a_i^*\| < \epsilon_0$ and $\|a_{i-1}^* - a_{i-1}^j\| < \epsilon_0$. Therefore

$$\|a_i^j - a_i^*\| < \|c_i - a_i^*\| \text{ and } \|a_{i-1}^* - a_{i-1}^j\| < \|c_{i-1} - a_{i-1}^*\|.$$

Besides, $\epsilon_0 = \|c_i - a_i^*\| = \dfrac{\|s_2 - a_i^*\|}{\|s_2 - a_{i+1}^*\|}.\|c_{i+1} - a_{i+1}^*\| = m_2.\|c_{i+1} - a_{i+1}^*\|$.

Combined with (7) this gives us

$$\|a^j_{i+1} - a^*_{i+1}\| < \frac{\epsilon_0}{M} = \frac{m_2}{M}.\|c_{i+1} - a^*_{i+1}\| \leq \|c_{i+1} - a^*_{i+1}\|.$$

These results yield the claim. □

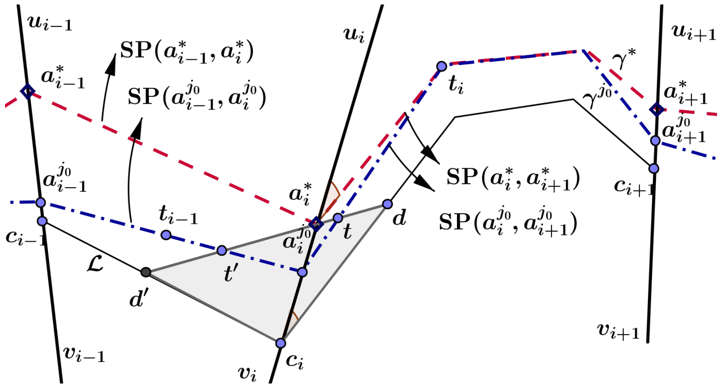
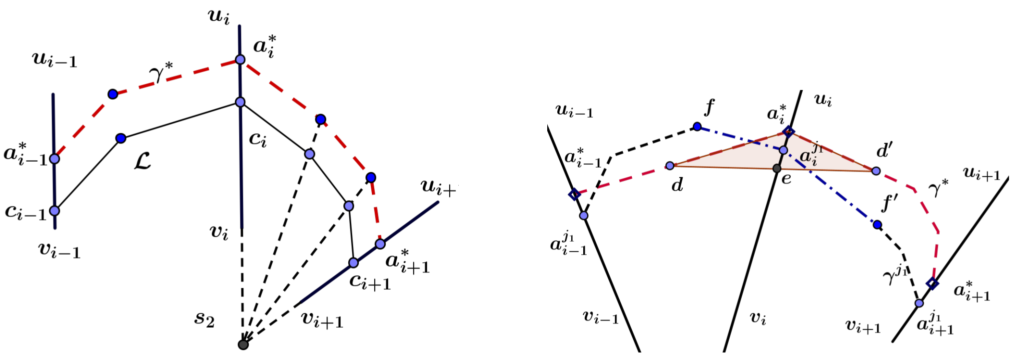


Figure 4.3: Illustration of the proof of Theorem 4.11.



(i) (ii)

Figure 4.4: Illustration of the proof of Theorem 4.11

We now turn to the proof of the theorem. We have $\angle\left(\mathrm{SP}(a^*_{i-1}, a^*_i), \mathrm{SP}(a^*_i, a^*_{i+1})\right) < \pi$ and the fact that $\{a^j_{i-1}\}$, $\{a^j_i\}$, and $\{a^j_{i+1}\}$ converge on one side to a^*_{i-1}, a^*_i and a^*_{i+1} as $j \to \infty$, respectively. There exists a natural number $j_0 \geq \overline{j}$ such that $\angle\left(\mathrm{SP}(a^{j_0}_{i-1}, a^{j_0}_i), \mathrm{SP}(a^{j_0}_i, a^{j_0}_{i+1})\right) < \pi$. Then the algorithm updates $a^{j_0}_i$ to $a^{j_0+1}_i$. *We are now in a position to show* $a^{j_0+1}_i \in (a^*_i, u_i]$.

Let two intersection points of γ^{j_0} and $[d, d']$ be denoted by t and t'. By Lemma 4.9, $d_{\mathcal{H}}\left(\mathrm{SP}(a^{j_0}_i, a^{j_0}_{i+1}), \mathrm{SP}(a^*_i, a^*_{i+1})\right) \leq 2\max\{\|a^{j_0}_i - a^*_i\|, \|a^{j_0}_{i+1} - a^*_{i+1}\|\} \leq \mu/2$. Combined with (4), $\mathrm{SP}(a^{j_0}_i, a^{j_0}_{i+1})$ does not contain vertices of V except the vertices belonging to $\hat{\gamma}$. Therefore, all endpoints of $\mathrm{SP}(a^{j_0}_i, a^{j_0}_{i+1})$ except for $a^{j_0}_i$ and $a^{j_0}_{i+1}$ belong to $\mathrm{SP}(a^*_i, a^*_{i+1})$. Note that the temporary point t_i of γ^{j_0} is taken as in step 4 of Procedure COLLINEAR_UPDATE($\gamma^{current}, \gamma^{next}, flag$). If $\mathrm{SP}(a^{j_0}_i, a^{j_0}_{i+1})$ is a line segment, we have $\|a^{j_0}_i - t_i\| \geq \frac{1}{2}d([u_i, v_i], [u_{i+1}, v_{i+1}])$. Otherwise, t_i is one of the endpoints of $\mathrm{SP}(a^{j_0}_i, a^{j_0}_{i+1})$ except for $a^{j_0}_i$ and $a^{j_0}_{i+1}$, and thus $t_i \in \hat{\gamma}$. Hence $\|a^{j_0}_i - t_i\|$ is greater than a given constant. As ϵ_0 is given by (5), we have $t_i \notin [a^{j_0}_i, t]$. Similarly, we also get $t_{i-1} \notin [a_i^{j_0}, t']$. Because of the properties of shortest paths, $\mathrm{SP}(t_{i-1}, t_i)$ do not intersect with $[t, t']$. In consequence, $\mathrm{SP}(t_{i-1}, t_i)$ intersects $[u_i, v_i]$ in a point that is above a_i^* when we view from u_i. This contradicts the result proved above in which γ^{j_0+1} is below $\hat{\gamma}$ when we view from B_1.

Case 2: $\angle\left(\mathrm{SP}(a_{i-1}^*, a_i^*), \mathrm{SP}(a_i^*, a_{i+1}^*)\right) > \pi$. Since $a_i^* \neq v_i$, there exist two points $d \in \mathrm{SP}(a_{i-1}^*, a_i^*)$, and $d' \in \mathrm{SP}(a_i^*, a_{i+1}^*)$ such that three points d, d', and a_i^* form a triangle that completely belongs to $\mathcal{D}$ but does not contain any vertex of $\mathcal{D}$, $[u_{i-1}, v_{i-1}] \cap \triangle dd'a_i^* = \emptyset$, and $[u_{i+1}, v_{i+1}] \cap \triangle dd'a_i^* = \emptyset$. Let $e = [d, d'] \cap [u_i, v_i]$.

We claim that there is no element of $\{a_i^j\}$ which is contained in $[e, a_i^*]$. Indeed, suppose that there exists $j_1 \in \mathbb{N}$ such that $a_i^{j_1} \in [e, a_i^*]$. Let two segments having the common endpoint $a_i^{j_1}$ of γ^{j_1} be $[f, a_i^{j_1}]$ and $[a_i^{j_1}, f']$, where $f \in \mathrm{SP}(a_{i-1}^{j_1}, a_i^{j_1})$ and $f' \in \mathrm{SP}(a_i^{j_1}, a_{i+1}^{j_1})$. Then either $f = a_{i-1}^{j_1}$ or f is a vertex of $\mathcal{D}$, and hence $f \notin \triangle dd'a_i^*$. Similarly, we get $f' \notin \triangle dd'a_i^*$. According to Corollary 4.8, it follows that $\angle f a_i^{j_1} f' \leq \pi$. Because of $\angle ded' = \pi$, we obtain either $[a_i^j, f]$ crossing $[a_i^*, d]$ or $[a_i^j, f']$ crossing $[a_i^*, d']$ (see Fig. 4.4(ii)). This contradicts the proved result that γ^{j_1} is below $\hat{\gamma}$ when we view from the boundary B_1.

Summarizing, combining the two cases gives that Collinear Condition (B) holds at a_i^*, for $i = 1, 2, \ldots, N$. Hence $\hat{\gamma}$ is the shortest path joining $\tilde{b}$ and b. The proof is complete. □

Since the function of the length of paths is lower semi-continuous but not continuous, the event $d_{\mathcal{H}}(\gamma^j, \gamma^*) \to 0$ as $j \to \infty$ does not ensure the convergence of the sequence of lengths of these paths. However, this holds for the sequence of paths obtained by the proposed algorithm. This is shown in the theorem below.

Theorem 4.12. *The sequence of lengths of paths obtained by the proposed algorithm has the following properties:*

(a) *The sequence is decreasing, i.e., the procedure*

$$\textsc{Collinear\_Update}(\gamma^{current}, \gamma^{next}, flag)$$

yields $l(\gamma^{next}) \leq l(\gamma^{current})$. If Collinear Condition (B) *does not hold at some shooting point, then the inequality above is strict.*

(b) *The sequence converges to $l(\gamma^*)$.*

Proof. (a) We have (see Figs. 3.2(iii) and (iv))

$$
\begin{aligned}
l(\gamma^{current}) &= \sum_{i=0}^{N} l(\mathrm{SP}(a_i, a_{i+1})) \\
&= l(\mathrm{SP}(a_0, t_0)) + \sum_{i=1}^{N} \left(l(\mathrm{SP}(t_{i-1}, a_i)) + l(\mathrm{SP}(a_i, t_i))\right) + l(\mathrm{SP}(t_N, a_{N+1})) \\
&\geq l(\mathrm{SP}(a_0^{next}, t_0)) + \sum_{i=1}^{N} l\left(\mathrm{SP}(t_{i-1}, t_i)\right) + l(\mathrm{SP}(t_N, a_{N+1}^{next})) \\
&= l(\mathrm{SP}(a_0^{next}, t_0)) + \sum_{i=1}^{N} \left(l(\mathrm{SP}(t_{i-1}, a_i^{next})) + l(\mathrm{SP}(a_i^{next}, t_i))\right) + l(\mathrm{SP}(t_N, a_{N+1}^{next}))
\end{aligned}
$$

$$= \sum_{i=0}^{N} \left(l(\mathrm{SP}(a_i^{next}, t_i)) + l(\mathrm{SP}(t_i, a_{i+1}^{next})) \right)$$
$$\geq \sum_{i=0}^{N} \left(l(\mathrm{SP}(a_i^{next}, a_{i+1}^{next})) \right) = l\left(\cup_{i=0}^{N} \mathrm{SP}(a_i^{next}, a_{i+1}^{next}) \right) = l(\gamma^{next}).$$

If Collinear Condition (B) does not hold at some shooting point a_i, then first inequality is strict. Thus $l(\gamma^{current}) > l(\gamma^{next})$.

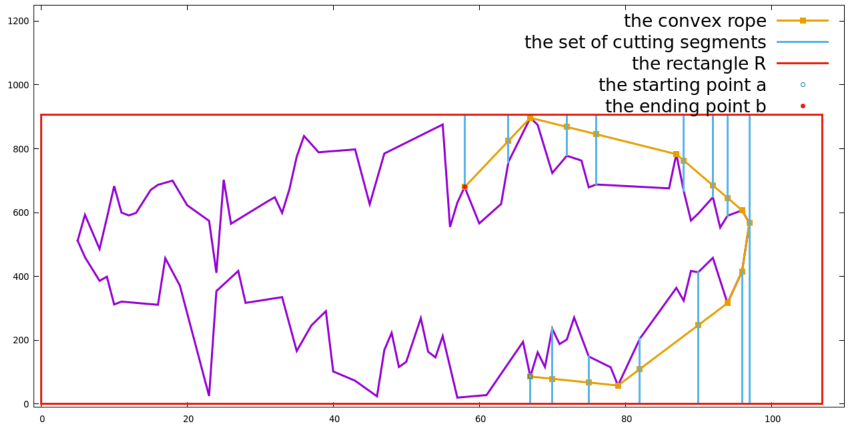
(b) Recall that $\gamma^* = \cup_{i=0}^{N} \mathrm{SP}(a_i^*, a_{i+1}^*)$ is the shortest path joining $\tilde{b}$ and b in $\mathcal{D}$ and $\gamma^j = \cup_{i=0}^{N} \mathrm{SP}(a_i^j, a_{i+1}^j)$ is the sequence of paths obtained by the algorithm, where a_i^* (a_i^j, resp.) is the intersection point between γ^* (γ^j, resp.) and the corresponding cutting edge. If the algorithm stops after a finite number of iterations, the proof is trivial. Otherwise, repeat the proof of Theorem 4.11 (item **i**) we get $a_i^j \to a_i^*$ as $j \to \infty$, for $i = 1, \ldots, N$, where $a_0^j = a_0^*$ and $a_{N+1}^j = a_{N+1}^*$. According to [[9], p. 542], the geodesic distance between two points x and y in a simple polygon, which is measured by the length of the shortest path joining two points x and y in the simple polygon, is a metric on the simple polygon and it is continuous as a function of both x and y. That is, $l(\mathrm{SP}(a_i^j, a_{i+1}^j)) \to l(\mathrm{SP}(a_i^*, a_{i+1}^*))$, for $i = 0, 1, \ldots, N$ as $j \to \infty$. It follows $l(\gamma^j) \to l(\gamma^*)$ as $j \to \infty$. □

Theorem 4.13. *Given a fixed number N of cutting segments, the proposed algorithm runs in $O(kn)$ time, where n is the number of vertices of $\mathcal{P}$, k is the number of iterations to get the required path joining $\tilde{b}$ and b.*

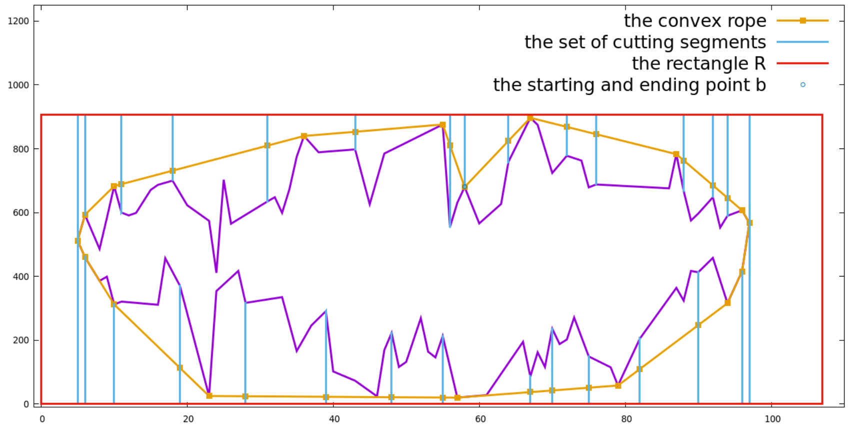
Proof. Steps 1 and 2 of the algorithm need $O(1)$ and $O(n)$ time, respectively. In step 3, there are at most two procedures of finding shortest paths in sub-polygons which call each vertex of $\mathcal{D}$. Since the problem of finding the shortest path joining two points in a simple polygon can be solved in linear time, computing the initial path $\gamma^{current}$ formed by the set of initial shooting points takes $O(n)$ time. Steps 6 and 7 need $O(1)$ time. We need to show that step 5 needs $O(n)$ time. For each iteration step, there are N cutting segments, where N is a given constant number, then the computing sequentially the angles at shooting points is done in constant time. In step 8 and 14 of for-loop in the procedure, since there are at most two procedures for finding the shortest path $\mathrm{SP}(t_{i-1}, t_i)$ in sub-polygons $\mathcal{D}_{i-1} \cup \mathcal{D}_i$ which call each vertex of $\mathcal{D}$, time complexity of Procedure COLLINEAR_UPDATE($\gamma^{current}, \gamma^{next}, flag$) is linear. In summarizing, the algorithm runs in $O(kn)$ time, where k is the number of iterations to get the required path. □

5. Numerical experiments

In the implementation, assume that the boundary of $\mathcal{P}$ consists of two polylines with integer coordinates that are monotone w.r.t. x-coordinate axis, and the cutting segments ξ_i are parallel to y-coordinate axis. The monotone condition appears because we use a procedure for finding the shortest path joining two consecutive shooting points inside sub-polygons $\mathcal{D}_i$ as shown in [5]. The algorithm is implemented in C++ programming language. The codes are compiled and executed by GNU Compiler Collection under platform Ubuntu Linux 18.04, Processor 2.50GHz Core i5.



(i)



(ii)

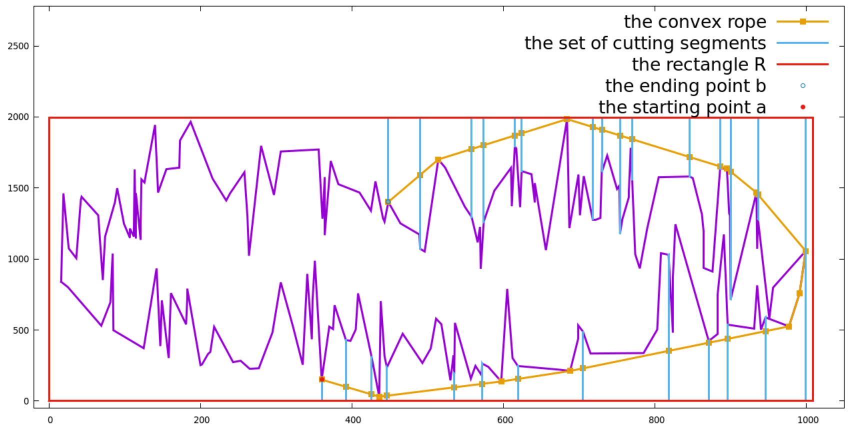
Figure 5.1: Convex ropes of a polygon with 200 vertices: (i) the convex rope starting at a vertex of the convex hull of $\mathcal{P}$ and ending at a point that is visible from infinity; (ii) the convex rope starting and ending at the same point that is visible from infinity.

To check whether Collinear Condition (B) holds or not at a shooting point a_i, we calculate the upper angle at a_i or determine if a_i^{next} coincides with a_i or not. Two ways are the same by Proposition 4.3. Here we use the latter way. We need a tolerance, say ϵ, to check for the coincidence of points when implementing. Collinear Condition (B) holds at a_i if $\|a_i^{next} - a_i\| < \epsilon$, for all $1 \leq i \leq N$. Moreover, by Lemma 4.10, the condition $\|a_i^{next} - a_i\| < \epsilon$, $1 \leq i \leq N$ also applies to determine $\{\gamma^j\}$ converging to γ^*. The actual runtime of our algorithm grows with the decreasing of ϵ. This effect is given in Table 5.4 and Fig 5.5, where ϵ changes from 10^0 to 10^{-9}, and the problem is to find an approximate convex rope starting at the copy $\tilde{b}$ of a fixed point b and ends at b of the polygon having 3000 vertices and 200 cutting segments. Figs. 5.1, 5.2, and 5.3 show the results when $\epsilon = 10^6$ and the number of vertices of $\mathcal{P}$ is 100, 400, and 800, respectively. Herein, the convex ropes start at a and end at b, where a is a visible infinity point that plays the same role as b ($a = \tilde{b}$) or a vertex of the convex hull of $\mathcal{P}$.

6. Conclusion

We presented the use of MMS with three factors (f1)–(f3) for approximately solving the convex rope problem.

Although the "oracle" condition does not hold for polygon $\mathcal{D}$ as in [3], we dealt with three open questions presented in Section 1.



(i)

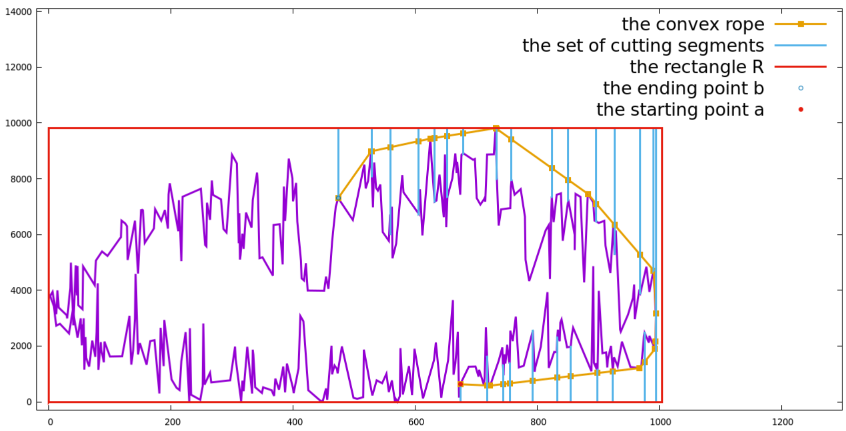
(ii)

Figure 5.2: Convex ropes of a polygon with 400 vertices: (i) the convex rope starting at a vertex of the convex hull of $\mathcal{P}$ and ending at a point that is visible from infinity; (ii) the convex rope starting and ending at the same point that is visible from infinity.

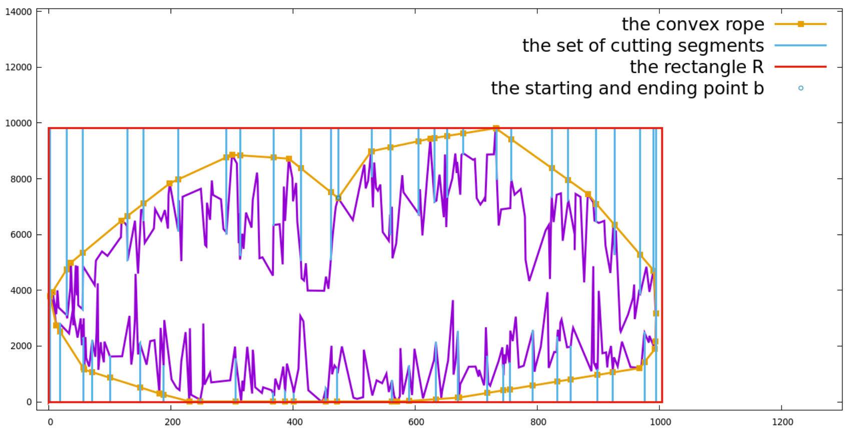
We proved that the shortest path is well determined if the collinear condition holds at all shooting points. Otherwise, the sequence of paths obtained from new shooting points by the update of the method converges in Hausdorff distance to the shortest path. The proposed algorithm was implemented in C++ and some numerical examples were given to show that the method is suitable for the problem. On the other side, as an advantage of MMS by the factor (f1) [4], instead of solving the problem for the whole formed polygon, we deal with a set of smaller partitioned subpolygons. Consequently, the memory consuming of the algorithm should be low. This is useful when considering to deploy the method in robotic applications with limited computing resources. This will be the topic of further research in the future.

Acknowledgments. This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number T2022-20-01. The first and third authors acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study. The second author was funded by Vingroup Joint

Stock Company and supported by the Master/ PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), Vingroup Big Data Institute (VinBigdata), code VINIF.2021.TS.068.



(i)



(ii)

Figure 5.3: Convex ropes of a polygon with 800 vertices: (i) the convex rope starting at a vertex of the convex hull of $\mathcal{P}$ and ending at a point that is visible from infinity; (ii) the convex rope starting and ending at the same point that is visible from infinity.

A. Appendix

With the construction of $\mathcal{D}$ as shown by (A) in Section 3.1, vertices of $\mathcal{D}$ on B_1 except for b and $\tilde{b}$ are convex (see Fig. 3.1(ii)). We have the following

Remark A.1. The shortest path joining a and b in $\mathcal{D}$ does not depend on the rectangle R, i.e, we can replace R with any rectangle that contains $\mathcal{P}$ and has no edge touching $\mathcal{P}$ to obtain another polygon $\mathcal{D}'$ in which shortest paths joining a and b in $\mathcal{D}$ and $\mathcal{D}'$ are the same. Furthermore, $\mathrm{SP}(a, b)$ never touches B_1 of $\mathcal{D}$ except for b.

Remark A.2. The convex rope starting at a and ending at b of $\mathcal{P}$ can be obtained by solving directly the problem of finding the shortest path joining a and b inside $\mathcal{D}$. If a is a vertex of the convex hull of $\mathcal{P}$, then the shortest path contains a and never touches the boundary B_1 of $\mathcal{D}$ except for $\tilde{b}$ and b. More generally, if a is visible from infinity, then $\tilde{b}$ plays the same role as a (see Fig. 3.1(ii)). Therefore from now on, we only focus on finding the shortest path joining $\tilde{b}$ and b in $\mathcal{D}$.

Tolerance ϵ	The runtime (secs)	Length of paths
10^{0}	14.26	432322.554
10^{-1}	44.33	432312.019
10^{-2}	106.96	432311.361
10^{-3}	257.45	432311.340
10^{-4}	398.17	432311.339
10^{-5}	553.05	432311.339
10^{-6}	693.02	432311.339
10^{-7}	881.29	432311.339
10^{-8}	1053.78	432311.339
10^{-9}	1217.04	432311.339

Table 5.4: The actual runtime of the proposed algorithm and the length of corresponding convex ropes when ϵ changes, where the convex ropes of a polygon having 3000 vertices start and end at the same point that is visible from infinity.

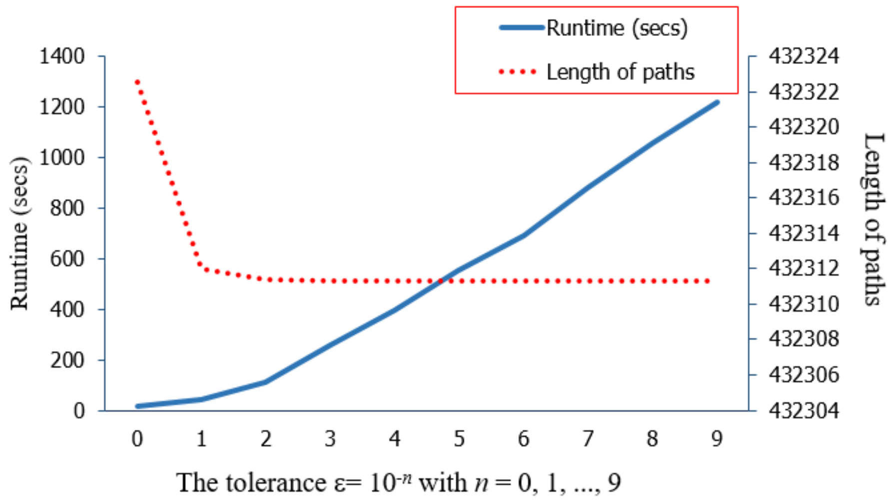


Figure 5.5: The dependence of the runtime of the proposed algorithm and the length of corresponding convex ropes on ϵ, where the convex ropes of a polygon having 3000 vertices starting and ending at the same point that is visible from infinity.

References

[1] P. T. An: *Reachable gasps on a polygon of a robot arm: finding the convex rope without triangulation*, J. Robotics Automation 25/4 (2010) 304–310.

[2] P. T. An: *Optimization Approaches for Computational Geometry*, Publishing House for Science and Technology, Vietnam Academy of Science and Technology, Hanoi (2017).

[3] P. T. An, N. N. Hai, T. V. Hoai: *Direct multiple shooting method for solving approximate shortest path problems*, J. Comp. Appl. Math. 244 (2013) 67–76.

[4] P. T. An, N. N. Hai, T. V. Hoai, L. H. Trang: *On the performance of triangulation-based multiple shooting method for 2D geometric shortest path problems*, Trans. Large-Scale Data Knowledge Centered Syst. 16 (2014) 45–56.

[5] P. T. An, T. V. Hoai: *Incremental convex hull as an orientation to solving the shortest path problem*, Int. J. Inf. Elect. Eng. 2/5 (2012) 652–655.

[6] P. T. An, L. H. Trang: *Multiple shooting approach for computing shortest descending paths on convex terrains*, Comp. Appl. Math. 37/4 (2018) 1–31.

[7] B. Chazelle: *A theorem on polygon cutting with applications*, in: Proc. 23rd IEEE Symposium on Foundations of Computer Science, Chicago (1982) 339–349.

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

[9] N. N. Hai, P. T. An: *Blaschke-type theorem and separation of disjoint closed geodesic convex sets*, J. Optimization Theory Appl. 151/3 (2011) 541–551.

[10] N. N. Hai, P. T. An, P. T. T. Huyen: *Shortest paths along a sequence of line segments in Euclidean spaces*, J. Convex Analysis 26/4 (2019) 1089–1112.

[11] P. J. Heffernan, J. S. B. Mitchell: *Structured visibility profiles with applications to problems in simple polygons*, in: Proc. 6th Annual ACM Symposium on Computational Geometry (1990) 53–62.

[12] T. V. Hoai, P. T. An, N. N. Hai: *Multiple shooting approach for computing approximately shortest paths on convex polytopes*, J. Comp. Appl. Math. 317 (2017) 235–246.

[13] D. T. Lee, F. P. Preparata: *Euclidean shortest paths in the presence of rectilinear barriers*, Networks 14/3 (1984) 393–410.

[14] F. Li, R. Klette: *Finding the shortest path between two points in a simple polygon by applying a rubberband algorithm*, in: Pacific-Rim Symposium on Image and Video Technology, Springer, Berlin (2006) 280–291.

[15] T. Lozano-Perez: *Automatic planning of manipulator transfer movements*, IEEE Trans. Systems Man Cybernetics SMC-11/10 (1981) 681–698.

[16] A. A. Melkman: *On-line construction of the convex hull of a simple polyline*, Inf. Processing Letters 25 (1987) 11–12.

[17] M. A. Peshkin, A. C. Sanderson: *Reachable grasps on a polygon: the convex rope algorithm*, IEEE J. Robotics Automation 2/1 (1986) 5–58.

[18] P. Song, Z. Fu, L. Liu: *Grasp planning via hand-object geometric fitting*, The Visual Computer 34/2 (2018) 257–270.

[19] G. T. Toussaint: *Computing geodesic properties inside a simple polygon*, Revue d'Intelligence Artificielle 3/2 (1989) 9–42.

[20] Z. Xue, J. M. Zoellner, R. Dillmann: *Automatic optimal grasp planning based on found contact points*, in: IEEE International Conference on Advanced Intelligent Mechatronics (2008) 1053–1058.