Finding Approximately Convex Ropes in the Plane

Le Hong Trang\textsuperscript{1,4}, Nguyen Thi Le\textsuperscript{3}, and Phan Thanh An \textsuperscript{*2,4}

\textsuperscript{1}Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam
\textsuperscript{2}Institute of Mathematical and Computational Sciences (IMACS) and Faculty of Applied Science, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam
\textsuperscript{3}Institute of Mathematics, Vietnam Academy of Science and Technology (VAST), 18 Hoang Quoc Viet Road, Hanoi, Vietnam
\textsuperscript{4}Vietnam National University Ho Chi Minh City, Linh Trung Ward, Thu Duc City, Vietnam

Abstract

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 \(P\), where \(a\) is a vertex of the convex hull of \(P\) 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 \(P\). In this paper, the problem is reconstructed as the 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.

\textbf{Keywords:} Convex hull; approximate algorithm; convex rope; shortest path; non-convex optimization; geodesic convexity.

\textbf{MSC2010:} MSC 52A30; MSC 52B55; MSC 68Q25; 90C26; MSC 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

\textsuperscript{*}Corresponding author. E-mail: thanhan@hcmut.edu.vn
Figure 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 2: Contact points would not necessarily be in edges of the convex hull of the polygon.

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(i)). 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(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. 2). It turns out to consider that the endpoints of the convex rope can be visible from infinity (see in Sect. 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.

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, 5, 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 Sect. 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 sequence, there are some open questions related to the algorithm in [3] as follows

- Can use MMS 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 converges 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 (Sect. 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 1 and 2 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. Sect. 2 introduces the convex rope problem and three main factors of MMS. Sect. 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 is stated by results in Sect. 4. The algorithm is implemented in C++ and numerical results
are given and visualized in Sect. 5 to describe how our method works. Some advantages of MMS are established in Sect. 6. Geometrical properties and their proofs are arranged in Appendix.

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(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 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(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 Sect. 3.1.

We now recall some basic concepts and properties. For any points \( x, y \) in the plane, we denote \([x, y] := \{(1 - \lambda)x + \lambda y : 0 \leq \lambda \leq 1\}, [x, y] := \{x, y\} \setminus \{x\}, [x, y] := \{x, y\} \setminus \{y\}\). 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 \( \text{SP}(x, y) \), is the path of minimal length joining \( x \) and \( y \) in \( \mathcal{A} \). As shown in [7, 13], \( \text{SP}(x, y) \) 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 \( \pi \). In addition, these vertices are also reflex vertices of \( \text{SP}(x, y) \). Furthermore, let \( y' \in \mathcal{A} \), for any two line segments \( e \) of \( \text{SP}(x, y) \) and \( g \) of \( \text{SP}(x, y') \), 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] and Appendix.

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

Regarding MMS for finding the shortest path joining two points \( p \) and \( q \) in a domain \( \mathcal{D} \) [3, 5, 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;
Figure 3: The counterclockwise convex rope starting at $a$ and ending at $b$, the case (i): $a$ is a vertex of the convex hull of $P$ and the case (ii): $a$ is visible from infinity.

(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.

3.1 Constructing a simple polygon $D$

Take a rectangle $R$ that contains $P$ and has no edge touching $P$. 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 $R \setminus \text{int}P$ 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 $R \setminus \text{int}P$. Let $D$ 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 $b$; other, denoted by $B_2$, is the polyline starting at $b$ going along the boundary of $P$ via clockwise order and returning to $\tilde{b}$. Thus $D$ has the boundary $B_1 \cup B_2$ as shown in Fig. 3(ii).

The non-simple region $R \setminus \text{int}P$ is converted into the simple polygon $D$. By Remark 2 given in Appendix, the convex rope starting at $a$ and ending at $b$ is indeed the shortest path joining $a$ and $b$ in $D$. Therefore the convex rope problem is deduced to the shortest path problem in $D$. We can also model the convex rope problem as the shortest path problem in another polygon whose boundary includes the convex hull of $P$ instead of the rectangle $R$, see [8]. However computing the convex
Appendix, Collinear Condition (B) does not include the case 
and SP \[ \text{algorithm for finding the shortest paths in simple polygons. Throughout this paper, when we say } \]
and memory than that in an entire polygon. Here SP \( (\gamma \) a path \)
\( \gamma \) in next sections (3.3 and 3.4), we obtain an ordered set of points \( v \) set of shooting points \( a \). Such a point \( a \) is called a shooting point and \( \gamma \) 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 \( D_i \) and that in \( D \) 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 \( \text{SP}(a_i, a_{i+1}) \) could be computed by any known algorithm for finding the shortest paths in simple polygons. Throughout this paper, when we say \( \text{SP}(x, y) \) without further explanation, it means the shortest path joining \( x \) and \( y \) in the connected union of sub-polygons containing \( x \) and \( y \) of \( D \).

### 3.2 The factor (f1): partitioning the polygon \( D \)

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

\[
+ \xi_i = [u_i, v_i] \subset D \text{ such that } [u_i, v_i] \subset \text{int} D, \quad u_i \in B_1 \text{ and } v_i \in B_2, \text{ for } 1 \leq i \leq N; \\
+ \xi_i \text{ strictly separates } \tilde{b} \text{ and } b, \text{ for } 1 \leq i \leq N; \\
+ \xi_0 = \{\tilde{b}\}, \xi_{N+1} = \{b\}, [u_i, v_i] \cap [u_j, v_j] = \emptyset, \text{ for } i \neq j, 1 \leq i, j \leq N; \\
+ D_i \text{ is bounded by } B_1, B_2, \text{ cut segments } \xi_i \text{ and } \xi_{i+1}, \text{ for } 0 \leq i \leq N; \\
+ \text{int}D_i \cap \text{int}D_j = \emptyset, \text{ for } i \neq j, 0 \leq i, j \leq N \text{ and } D = \bigcup_{i=0}^{N} D_i. 
\]

A partition given by (1) is illustrated in Fig. 4(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 \( \xi_i \). For each iteration step, which is discussed carefully in next sections (3.3 and 3.4), we obtain an ordered set of points \( \{a_i \mid a_i \in \xi_i, i = 1, 2, \ldots, N\} \) and a path \( \gamma = \bigcup_{i=0}^{N} \text{SP}(a_i, a_{i+1}) \), where \( \text{SP}(a_i, a_{i+1}) \) is the shortest path joining \( a_i \) and \( a_{i+1} \) in \( D_i \) and \( a_0 = \tilde{b}, a_{N+1} = b \). Such a point \( a_i \) is called a shooting point and \( \gamma \) is called the path formed by the set of shooting points (see [2]).

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

\[(B) \text{ If } a_{i} \in [u_{i}, v_{i}], \text{ we say that } a_{i} \text{ satisfies the collinear condition when the upper angle created by } \text{SP}(a_{i-1}, a_{i}) \text{ and } \text{SP}(a_{i}, a_{i+1}) \text{ is equal to } \pi \text{ (i.e., } \angle (\text{SP}(a_{i-1}, a_{i}), \text{SP}(a_{i}, a_{i+1})) = \pi). \]

\[\text{If } a_{i} = v_{i}, \text{ we say that } a_{i} \text{ satisfies the collinear condition when the upper angle created by } \text{SP}(a_{i-1}, a_{i}) \text{ and } \text{SP}(a_{i}, a_{i+1}) \text{ is at least } \pi \text{ (i.e., } \angle (\text{SP}(a_{i-1}, a_{i}), \text{SP}(a_{i}, a_{i+1})) \geq \pi). \]

Let \( \gamma^* \) be the shortest path joining \( \tilde{b} \) and \( b \) in \( D \), i.e., \( \gamma^* = \text{SP}(\tilde{b}, b) \). According to Remark 2 in Appendix, Collinear Condition (B) does not include the case \( a_i = u_i \).
3.4 The factor (f3): updating shooting points

Suppose that in the $j$-iteration step, we have a set of shooting points $\{a_i\}_{i=1}^{N}$ and $\gamma = \bigcup_{i=0}^{N} \text{SP}(a_i, a_{i+1})$ 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 1 in Sect. 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 1 and 2. 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 $t_{i-1} \in \text{SP}(a_{i-1}, a_i), t_i \in \text{SP}(a_i, a_{i+1})$ such that $t_{i-1}, t_i$ are not identical to shooting points. We call such point $t_{i-1}$ (resp.) the temporary point with respect to $a_{i-1}$ ($a_i$, resp.). Let $a_{i}^{\text{next}} = \text{SP}(t_{i-1}, t_i) \cap [u_i, v_i]$.

The intersection point $a_{i}^{\text{next}}$ exists uniquely since $[u_i, v_i]$ divides $D$ into two parts, one that contains $t_{i-1}$ and one that contains $t_i$. We update shooting point $a_i$ to $a_{i}^{\text{next}}$ (see Fig. 4(iii)).

In particular, if $\text{SP}(a_i, a_{i+1})$ is a line segment, we choose $t_i$ as the midpoint of $\text{SP}(a_i, a_{i+1})$. Otherwise, $t_i$ is chosen as an endpoint of $\text{SP}(a_i, a_{i+1})$ that is not identical to $a_i$ and $a_{i+1}$. We can update

![Figure 4: (i): Illustration of partitioning $D$; (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 $a_{i}^{\text{next}} = \text{SP}(t_{i-1}, t_i) \cap \xi_i$; (iv): the update $\gamma^{\text{current}}$ (dashed polyline) to $\gamma^{\text{next}}$ (solid polyline).](image-url)
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 2 (in Sect. 4).

3.5 The Proposed Algorithm

\textbf{Input}: A simple polygon \( P, b \in P \) and its copy \( \tilde{b} \), an integer number \( N \geq 1 \).

\textbf{Output}: An approximate convex rope starting at \( \tilde{b} \) and ending at \( b \).

1: Construct a simple polygon \( D \) with the boundary \( B_1 \cup B_2 \) as shown in (A). \( \triangleright \) see Fig. 3(ii)

2: Divide \( D \) into \( D_i \) satisfying (1) by cutting segments \( [u_i, v_i] \), where \( u_i \in B_1, v_i \in B_2 \) (\( i = 1, \ldots, N \))

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 \( \{a_i\}_{i=0}^{N+1} \). \( \triangleright \) see Fig. 4(i)

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

5: Call Procedure \texttt{COLLINEAR\_UPDATE}(\gamma^{current}, \gamma^{next}, flag) \( \triangleright \) see Figs. 4(iii) and (iv)

6: If Collinear Condition(B) holds at all shooting points, i.e., \( \text{flag} = \text{true}, \) \textbf{stop} and \textbf{return} \( \gamma^{next} \).

\( \triangleright \) the shortest path is obtained by Proposition 1

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

8: \textbf{return} \( \gamma^{next} \).

For each iteration step, we need to check Collinear Condition (B) and update shooting points given by (C) to get a better path. Then Procedure \texttt{COLLINEAR\_UPDATE}(\gamma^{current}, \gamma^{next}, flag) of the proposed algorithm performs checking condition (B) and updating the shooting point \( a_i \) to \( a_i^{next} \).

\begin{verbatim}
1: procedure \texttt{COLLINEAR\_UPDATE}(\gamma^{current}, \gamma^{next}, flag)
2: \quad flag \leftarrow \text{true}
3: \quad \textbf{for} \ i = 0, 1, \ldots, N \ \textbf{do}
4: \quad \quad Take temporary points \( t_i \) on \( SP(a_i, a_{i+1}) \) such that
5: \quad \quad \quad if \( SP(a_i, a_{i+1}) \) is a line segment, \( t_i \) is the midpoint of \( SP(a_i, a_{i+1}) \),
6: \quad \quad \quad \quad for if not, \( t_i \) is an endpoint of \( SP(a_i, a_{i+1}) \) that is not identical to \( a_i \) and \( a_{i+1} \).
7: \quad \quad \quad if \( a_i \in]u_i, v_i[ \) then
8: \quad \quad \quad \quad \quad if \( \angle(SP(a_{i-1}, a_i), SP(a_i, a_{i+1})) = \pi \) then \( \triangleright \) Collinear Condition (B) is checked
9: \quad \quad \quad \quad \quad \quad \quad Set \( a_i^{next} \leftarrow a_i \)
10: \quad \quad \quad \quad \quad else \ \text{call} \ SP(t_{i-1}, t_i)
11: \quad \quad \quad \quad \quad \quad \quad Set \( a_i^{next} \) be the intersection point of \( SP(t_{i-1}, t_i) \) and \( [u_i, v_i] \) \( \triangleright \) due to (C)
12: \quad \quad \quad \quad \quad \quad \quad flag \leftarrow \text{false}
13: \quad \quad \quad else \( \triangleright \) \( a_i = v_i \)
14: \quad \quad \quad \quad if \( \angle(SP(a_{i-1}, a_i), SP(a_i, a_{i+1})) \geq \pi \) then \( \triangleright \) Collinear Condition (B) is checked
15: \quad \quad \quad \quad \quad \quad \quad Set \( a_i^{next} \leftarrow a_i \)
16: \quad \quad \quad \quad \quad else \ \text{call} \ SP(t_{i-1}, t_i)
17: \quad \quad \quad \quad \quad \quad \quad Set \( a_i^{next} \) the intersection point of \( SP(t_{i-1}, t_i) \) and \( [u_i, v_i] \) \( \triangleright \) due to (C)
18: \quad \quad \quad \quad \quad \quad \quad flag \leftarrow \text{false}
19: a_0^{next} = \tilde{b}, a_{N+1}^{next} = b
\end{verbatim}
4 The Correctness of the Proposed Algorithm

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

**Proposition 1.** Let $\gamma_{\text{current}}$ 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 $\gamma_{\text{current}} = \gamma^*$.

**Proposition 2.** Collinear Condition (B) holds at $a_i$ if and only if $\text{SP}(t_{i-1}, t_i) \cap [u_i, v_i] = a_i$, where $t_{i-1}$ and $t_i$ are temporary points w.r.t. $a_{i-1}$ and $a_i$.

**Theorem 1.** 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 $\gamma^*$ in Hausdorff distance (i.e., $d_H(\gamma^j, \gamma^*) \to 0$ as $j \to \infty$, where $\gamma^j$ be a path obtained by the algorithm in $j^{th}$-iteration step).

Since the function of the length of paths is lower semi-continuous but not continuous, the event $d_H(\gamma^j, \gamma^*) \to 0$ 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 below theorem.

**Theorem 2.** The sequence of lengths of paths obtained by the proposed algorithm has the following properties:

(a) The sequence is decreasing, i.e., Procedure \texttt{COLLINEAR\_UPDATE($\gamma_{\text{current}}, \gamma_{\text{next}}, \text{flag}$)} gives $l(\gamma_{\text{next}}) \leq l(\gamma_{\text{current}})$. If Collinear Condition (B) does not hold at some shooting point, then the inequality above is strict.

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

**Theorem 3.** 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 $b$.

The proofs of the above propositions and theorems are given in Appendix.

5 Numerical Experiments

In the implementation, assume that the boundary of $\mathcal{P}$ consists of two polylines with integer co-ordinates that are monotone w.r.t. $x$-coordinate axis, and the cutting segments $\xi_i$ 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 [4]. 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.

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 $a_i^{\text{next}}$ coincides with $a_i$ or not. Two ways are the same by Proposition 2. Here we use the latter way. We need a tolerance, say $\epsilon$, 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 in Appendix, the condition \( \| a_i^{next} - a_i \| < \epsilon \), \( 1 \leq i \leq N \) also applies to determine \( \{ \gamma^j \} \) converging to \( \gamma^* \). The actual runtime of our algorithm grows with the decreasing of \( \epsilon \). This effect is given in Table 1 and Fig 8, where \( \epsilon \) 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, 6, and 7 show the results when \( \epsilon = 10^0 \) and the number of vertices of \( P \) 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 \( P \).
Figure 5: Convex ropes of a polygon with 200 vertices: (i) the convex rope starting at a vertex of the convex hull of $P$ 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.

Figure 6: Convex ropes of a polygon with 400 vertices: (i) the convex rope starting at a vertex of the convex hull of $P$ 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.
Table 1: The actual runtime of the proposed algorithm and the length of corresponding convex ropes when $\epsilon$ changes, where the convex ropes of a polygon having 3000 vertices start and end at the same point that is visible from infinity.

| Tolerance $\epsilon$ | 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      |

Figure 7: Convex ropes of a polygon with 800 vertices: (i) the convex rope starting at a vertex of the convex hull of $P$ 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.

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 $D$ as in [3], we dealt with three open questions presented in Sect. 1. 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) [6], 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 a further research in the future.
Figure 8: The dependence of the runtime of the proposed algorithm and the length of corresponding convex ropes on $\epsilon$, where the convex ropes of a polygon having 3000 vertices starting and ending at the same point that is visible from infinity.

Acknowledgment

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.

References

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

[2] An, P. T. (2017). *Optimization Approaches for Computational Geometry*. Publishing House for Science and Technology, Vietnam Academy of Science and Technology, Hanoi, ISBN 978-604-913-573-6.

[3] An, P. T., Hai, N. N., Hoai, T. V. (2013). Direct multiple shooting method for solving approximate shortest path problems. *Journal of Computational and Applied Mathematics, 244*, 67–76.

[4] An, P. T., Hoai, T. V. (2012). Incremental convex hull as an orientation to solving the shortest path problem. *International Journal of Information and Electronics Engineering, 2*(5), 652–655.

[5] An, P. T., Trang., L. H. (2018). Multiple shooting approach for computing shortest descending paths on convex terrains. *Computational and Applied Mathematics, 37*(4), 1–31.

[6] An, P. T., Hai, N. N., Hoai, T. V., Trang., L. H. (2014). On the performance of triangulation-based multiple shooting method for 2D geometric shortest path problems. *Transactions on Large-Scale Data- and Knowledge-Centered Systems XVI, 45–56.*
[7] Chazelle, B. (1982). *A theorem on polygon cutting with applications*, Proc. 23rd IEEE Symposium on Foundations of Computer Science, Chicago, 339–349.

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

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

[10] Hai, N. N., An, P. T., & Huyen, P. T. T. (2019). *Shortest paths along a sequence of line segments in euclidean spaces*. Journal of Convex Analysis, 26(4), 1089-1112.

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

[12] Hoai, T. V., An, P. T., Hai, N. N. (2017). Multiple shooting approach for computing approximately shortest paths on convex polytopes. *Journal of Computational and Applied Mathematics*, 317, 235–246.

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

[14] Li, F, Klette, R. (2006). 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, Heidelberg, 280–291.

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

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

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

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

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

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

A Shortest paths and convex ropes

With the construction of $\mathcal{D}$ as shown by (A) in Sect. 3.1, vertices of $\mathcal{D}$ on $B_1$ except for $b$ and $\tilde{b}$ are convex (see Fig. 3(ii)). We have the following

**Remark 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, $\text{SP}(a, b)$ never touches $B_1$ of $\mathcal{D}$ except for $\tilde{b}$.

**Remark 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(ii)). Therefore from now on, we only focus on finding the shortest path joining $\tilde{b}$ and $b$ in $\mathcal{D}$.

B The proofs of propositions and theorems given in Sect. 4

Lemma 1 is presented without a proof (since the proof is similar to that of Proposition 2.19, [2]).

**Lemma 1.** Suppose that $\gamma$ is the shortest path joining two given points $p$ and $q$ in a polygon $A$ such that $\gamma$ 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 proof of Proposition 1:**

*Proof.* Let $\gamma_{\text{current}} = \bigcup_{i=0}^{N} \text{SP}(a_i, a_{i+1})$ 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 $\gamma_{\text{current}}$ and $\gamma^*$ are distinct. There are two distinct points $u, v$ which are common points to both $\gamma_{\text{current}}$ and $\gamma^*$ such that $\gamma_{\text{current}}(u, v) \cap \text{SP}(u, v) = \{u, v\}$, where $\gamma_{\text{current}}(u, v)$ is the sub-path of $\gamma_{\text{current}}$ from $u$ to $v$. Note that $\text{SP}(u, v)$ is also a sub-path of $\gamma^*$. Two points $u$ and $v$ always exist and may be identical to $\tilde{b}$ and $b$, respectively. Then $\gamma_{\text{current}}(u, v)$ and $\text{SP}(u, v)$ form a simple polygon with $m$ vertices $p_1 = u, p_2, \ldots, p_k = v, p_{k+1}, \ldots, p_m$ (see Fig. 9(i)). Let $\alpha_j$ be the internal angle at $p_j$, with $j = 1, 2, \ldots, m$. By the properties of shortest paths, if $p_j \in \text{SP}(u, v)$ then vertices of $\text{SP}(u, v)$ except for $u$ and $v$ are reflex vertices of $\text{SP}(u, v)$, i.e., $\alpha_j > \pi$. If $p_j \in \gamma_{\text{current}}(u, v)$, then $p_j$ is either a shooting point or a reflex vertex of $\text{SP}(a_i, a_{i+1})$ 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 $\gamma_{\text{current}}$, and thus $\alpha_j > \pi$. If $p_j$ is a reflex vertex of $\text{SP}(a_i, a_{i+1})$ 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 $(m - 2)\pi = \sum_{i=1}^{m} \alpha_i > \alpha_1 + (m - 2)\pi + \alpha_k > (m - 2)\pi$. The contradiction deduces $\gamma_{\text{current}} = \gamma^*$. \hfill $\square$

**The proof of Proposition 2:**
Proof. \( \Rightarrow \) Assuming that Collinear Condition (B) holds at \( a_i \), to show that \( \text{SP}(t_{i-1}, t_i) \cap [u_i, v_i] = a_i \), we just need to prove that \( \text{SP}(t_{i-1}, t_i) = \text{SP}(t_{i-1}, a_i) \cup \text{SP}(a_i, t_i) \) Assume the contrary, we use the same way of the proof of Proposition 1 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 \( \text{SP}(t_{i-1}, t_i) \cap [u_i, v_i] = a_i \). Therefore \( a_i \) belongs to \( \text{SP}(t_{i-1}, t_i) \). If \( a_i = v_i \), then \( \angle(\text{SP}(t_{i-1}, a_{i}^{\text{next}}), \text{SP}(a_{i}^{\text{next}}, t_i)) \geq \pi \) by the property of shortest paths. If \( a_i \in [u_i, v_i] \), then \( \angle(\text{SP}(t_{i-1}, a_{i}^{\text{next}}), \text{SP}(a_{i}^{\text{next}}, t_i)) = \pi \) due to Lemma 1. Thus Collinear Condition (B) holds at \( a_i \). \( \square \)

To prove Theorems 1, and 2 we need Proposition 3, Proposition 4 and Corollary 1. Next, assuming that a partition of \( D \) is given by (1), we have

**Proposition 3.** We have \( \gamma^* \subset \bigcup_{i=0}^{N} D_i \), where \( D_i \) is bounded by \( B_1, B_2 \) and the cut segments \( \xi_i \) and \( \xi_{i+1} \). Moreover there exists a set of points \( \{a_i^* \in \xi_i, i = 0, 1, \ldots, N + 1\} \), where \( a_0^* = \tilde{b}, a_{N+1}^* = b \), such that \( \gamma^* = \bigcup_{i=0}^{N} \text{SP}(a_i^*, a_{i+1}^*) \), in which \( \text{SP}(a_i^*, a_{i+1}^*) \) is the shortest path joining \( a_i^* \) and \( a_{i+1}^* \) in \( D_i \), for \( i = 0, 1, \ldots, N \).

**Proof.** Because of (1), \( \xi_i \) divides \( D \) into two parts, one contains \( \tilde{b} \) and the other contains \( b \). Hence the intersection of \( \gamma^* \) and \( \xi_i \) is not empty. As \( \gamma^* \) is the shortest path joining \( \tilde{b} \) and \( b \) in \( D \), 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 \( \gamma^* = \bigcup_{i=0}^{N} \text{SP}(a_i^*, a_{i+1}^*) \), where \( \text{SP}(a_i^*, a_{i+1}^*) \) is the shortest path joining \( a_i^* \) and \( a_{i+1}^* \) in \( D_i \), for \( i = 0, 1, \ldots, N \). Since the construction of \( D_i \), we deduce that \( \text{SP}(a_i^*, a_{i+1}^*) \subset D_i \).

The converse of Proposition 3 is not true for the general case. It means that if there is a set \( \{a_i \in \xi_i, i = 0, \ldots, N\}, \gamma^\text{current} = \bigcup_{i=0}^{N} \text{SP}(a_i, a_{i+1}) \), then it is not concluded that \( \gamma^\text{current} \) is \( \gamma^* \). However, if \( a_i \) satisfies Collinear Condition (B), for all \( i = 1, 2, \ldots, N \), we have \( \gamma^\text{current} = \gamma^* \) by Proposition 1.

**Remark 3.** Suppose that \( a_i \in \xi_i, a_{i+1} \in \xi_{i+1} \) are shooting points in some iteration step, and \( a_{i}^{\text{next}} \in \xi_{i}, a_{i+1}^{\text{next}} \in \xi_{i+1} \) are shooting points at the next iteration step of the proposed algorithm such that \( a_{i}^{\text{next}} \in [u_i, a_i] \) and \( a_{i+1}^{\text{next}} \in [a_{i+1}, u_{i+1}] \), see Figs. 4(iii) and (iv). Then we have

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

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

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

(a) \( \angle(\text{SP}(a_{i-1}, a_i), \text{SP}(a_i, a_{i+1})) = \pi \iff \hat{a}_i = a_i \);

(b) \( \angle(\text{SP}(a_{i-1}, a_i), \text{SP}(a_i, a_{i+1})) < \pi \iff \hat{a}_i \in [a_i, u_i[; \)

(c) \( \angle(\text{SP}(a_{i-1}, a_i), \text{SP}(a_i, a_{i+1})) > \pi \iff \hat{a}_i \in [v_i, a_i[. \)
Proof. When Collinear Condition (B) does not hold, the role of $\hat{a}_i$ and $a_i^{next}$ are the same. By Proposition 2, Collinear Condition (B) in Sect. 3.3, and the fact that $a_i \neq v_i$, we get (a).

b) ($\Rightarrow$) Suppose that $\angle (SP(a_{i-1}, a_i), SP(a_i, a_{i+1})) < \pi$ and $\hat{a}_i \notin [a_i, u_i]$. Because $SP(t_{i-1}, t_i)$ never touches $B_1$ of $D$, similarly to Remark 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 $a_i \neq \hat{a}_i$, $SP(t_{i-1}, t_i)$ and $SP(t_{i-1}, a_i) \cup SP(a_i, t_i)$ are thus distinct. Then there are two distinct points $u, v$ which are common points to both $SP(t_{i-1}, t_i)$ and $SP(t_{i-1}, a_i) \cup SP(a_i, t_i)$ such that $\sp u, v \cap (SP(u, a_i) \cup SP(a_i, v)) = \{u, v\}$. Note that $SP(u, v)$ is also a sub-path of $SP(t_{i-1}, t_i)$. Then $SP(u, v)$ and $SP(u, a_i) \cup SP(a_i, v)$ form a simple polygon in $D$ with $m$ vertices $p_1 = u, p_2, \ldots, p_t = a_i, p_{t+1}, \ldots, p_k = v, p_{k+1}, \ldots, p_m$. Let $\alpha_j$ be the internal angle at $p_j$, with $j = 1, 2, \ldots, m$, see Fig 9(i). 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 $\alpha_j = 2\pi - \angle (SP(a_{i-1}, a_i), SP(a_i, a_{i+1}))$, we have $\alpha_j > \pi$. Then $\alpha_j > \pi$ for $j \in \{1, 2, \ldots, m\} \setminus \{1, k, l\}$. It is similar to the proof of Proposition 1, a contradiction is obtained. Thus $\hat{a}_i \notin [a_i, u_i]$.

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

b) c) ($\Rightarrow$) For the sufficient condition of (b), ((c) can be considered similarly), assume that $\hat{a}_i \in [a_i, u_i]$. Then $\angle (SP(a_{i-1}, a_i), SP(a_i, a_{i+1})) < \pi$, for if not, $\angle (SP(a_{i-1}, a_i), SP(a_i, a_{i+1})) \geq \pi$ 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. \hfill $\square$

Proposition 4. For all $j$, we have $\gamma^{j+1}$ is above $\gamma^j$ when we view from $B_1$, i.e., $a_i^{prev} \in [a_i^{next}, u_i]$, for $i = 1, 2, \ldots, N$, where $\gamma^j$ 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 $\angle (SP(a_{i-1}, a_i), SP(a_i, a_{i+1})) \leq \pi$ due to Lemma 2(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 $\angle (SP(a_{i-1}, a_i), SP(a_i, a_{i+1})) > \pi$. 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, there is an index $i$ such that $\angle (\text{SP}(a_{i-1}, a_i), \text{SP}(a_i, a_{i+1})) > \pi$ and $a_i^{\text{next}} \notin [a_i, u_i]$, i.e.,

$$a_i^{\text{next}} \in [v_i, a_i] \quad (2)$$

Let $G_1$ be the closed region bounded by $[a_i^{\text{prev}}, a_i]$, $\text{SP}(a_i, t_{i-1})$ and $\text{SP}(t_{i-1}, a_i^{\text{prev}})$. Let $G_2$ be the closed region bounded by $[a_i^{\text{prev}}, a_i]$, $\text{SP}(a_i, t_i)$ and $\text{SP}(t_i, a_i^{\text{prev}})$, where $t_{i-1}$ and $t_i$ are temporary points w.r.t. $a_i^{\text{prev}}$ and $a_i^{\text{prev}}$ (see Fig. 9(ii)). Since $a_i \in [u_i, v_i]$, $\angle (\text{SP}(t_{i-1}, a_i), \text{SP}(a_i, t_i)) = \pi$ by Lemma 1. Next, let $[a_i, w_1]$ and $[a_i, w_2]$ be segments having the common endpoint $a_i$ of $\text{SP}(a_{i-1}, a_i)$ and $\text{SP}(a_i, a_{i+1})$, respectively. As $\angle (\text{SP}(t_{i-1}, a_i), \text{SP}(a_i, t_i)) = \pi$ and $\angle (\text{SP}(a_{i-1}, a_i), \text{SP}(a_i, a_{i+1})) > \pi$, there are at least one of two following cases that will happen:

$$[a_i, w_1] \text{ is below entirely } \text{SP}(t_{i-1}, a_i) \quad (3)$$

or $[a_i, w_2]$ is below entirely $\text{SP}(a_i, t_i)$ when we view from $u_i$.

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

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

**Lemma 3** (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 $d_H(\text{SP}(a, b), \text{SP}(c, d)) \leq \|a - d\| + \|b - c\| \leq 2 \max\{\|a - d\|, \|b - c\|\}$.

**Lemma 4** (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 $d_H(\text{SP}(a_n, b_n), \text{SP}(a, b)) \to 0$ as $n \to +\infty$.

**The proof of Theorem 1**:

**Proof**. Let $\gamma = \bigcup_{i=0}^{N} \text{SP}(a_i, a_{i+1})$, where $a_i \in [u_i, v_i]$ for $i = 1, 2, \ldots, N$ and $a_0 = b, a_N = b$. 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 1.

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

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 and the convergence of their subsequence. By the order of elements of $\{a_i^j\}_{j \in \mathbb{N}}$ as shown in Proposition 4, for all natural numbers large enough $j$, there exists $k \in \mathbb{N}$ such that $a_i^j \in [a_i^{k_j}, a_i^{(k+1)_j}]$. Furthermore, we also obtain $\{a_i^{k_j}\}_{k \in \mathbb{N}}$ converges on one side to $a_i^*$. For all $\delta > 0$, there exists $k_0 \in \mathbb{N}$ such that $\|a_i^{k_j} - a_i^*\| < \delta$, for $k \geq k_0$. As $a_i^j \in [a_i^{k_j}, a_i^{*}]$, for $j \geq j_k$, we get $\|a_i^j - a_i^*\| < \delta$, for $j \geq j_k$. This
clearly forces \( \{a^j_i\}_{j \in \mathbb{N}} \to a^*_i \) as \( j \to \infty \). This is the one-sided convergence, \( \gamma^j \) is thus below \( \gamma \) when we view from the boundary \( B_1 \).

ii. We next indicate that \( d_{BH}(\gamma^j, \gamma) \to 0 \) as \( j \to \infty \). According to item i, \( a^j_i \to a^*_i \) as \( j \to \infty \), for \( i = 1, 2, \ldots, N \). By Lemma 4, we get \( d_{BH} (SP(a^j_i, a^j_{i+1}) \cup SP(a^*_i, a^*_i)) \to 0 \) as \( j \to \infty \). Note that \( d_{BH}(A \cup B, C \cup D) \leq \max\{d_{BH}(A, C), d_{BH}(B, D)\} \), for all closed sets \( A, B, C \) and \( D \) in \( \mathbb{R}^2 \), we have \( d_{BH}(\gamma^j, \gamma) \leq \max_{0 \leq i \leq N}\{d_{BH} (SP(a^j_i, a^j_{i+1}) \cup SP(a^*_i, a^*_i))\} \). Therefore \( d_{BH}(\gamma^j, \gamma) \to 0 \) as \( j \to \infty \).

iii. Our next claim is that \( \hat{\gamma} \) is the shortest path joining \( \bar{b} \) and \( b \). According to Proposition 1, 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 \( a^j_i = v_i \) for all \( j \geq 0 \). Then Collinear Condition (B) holds at \( a^*_i \) for all \( j \geq 0 \), i.e., \( \angle (SP(a^j_{i-1}, a^j_i), SP(a^j_i, a^j_{i+1})) \geq \pi \), for all \( j \geq 0 \). Since \( \gamma^j \) is below \( \hat{\gamma} \) when we view from the boundary \( B_1 \) and \( \{a^j_{i-1}\}_{j \in \mathbb{N}} \to a^*_i \), \( \{a^j_{i+1}\}_{j \in \mathbb{N}} \to a^*_i \) as \( j \to \infty \), we get the upper angle of \( \hat{\gamma} \) at \( a^*_i \) is not less than \( \pi \), 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 \( \angle (SP(a^*_i, a^*_i), SP(a^*_i, a^*_i)) \) < \( \pi \) or \( \angle (SP(a^*_i, a^*_i), SP(a^*_i, a^*_i)) > \pi \).

Case 1: \( \angle (SP(a^*_i, a^*_i), SP(a^*_i, a^*_i)) < \pi \). To obtain a contradiction, we will show that there exists a natural number \( j_0 \) such that the updating \( a^j_i \) to \( a^{j+1}_i \) gives \( a^{j_0+1}_i \) \( \in (a^*_i, u_i) \). Let \( V \) be the set of all vertices of \( D \) which do not belong to \( \hat{\gamma} \). Set

\[
\mu = \min \{d(v, \hat{\gamma}) : v \in V \}.
\]

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. 10(i)). Through \( c_i \), we draw a polyline \( \mathcal{L} \) whose line segments are parallel to line segments of \( SP(a^*_i, a^*_i) \cup SP(a^*_i, a^*_i) \), respectively (see Fig. 10(ii)). 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 \( a^*_i \) 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 \( \triangle c_i dd' \) (see Fig. 10(i)). If not, we can take

\[
\epsilon_0 = \frac{\mu}{2K_0}, \quad \text{where} \quad K_0 \quad \text{is a large enough natural number.}
\]

For simplicity, assume that \( \triangle c_i dd' \) is an isosceles triangle, (see Fig. 10(i)).

Claim 1. Let \( \alpha^- = \angle (SP(a^*_i, a^*_i), [a^*_i, u_i]) \) and \( \alpha^+ = \angle ([a^*_i, u_i], SP(a^*_i, a^*_i)) \). In \( \triangle c_i dd' \), we obtain

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

The proof of Claim 1. Using the Law of Sines in a triangle.

Turning to the proof, for \( i = 1, 2, \ldots, N \), since \( \{a^j_i\}_{j \in \mathbb{N}} \to a^*_i \) as \( j \to \infty \), there exists \( j \in \mathbb{N} \) such that

\[
||a^j_i - a^*_i|| > \frac{\epsilon_0}{M}, \quad \text{for all} \quad j \geq j,
\]

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 \( \text{SP}(a_{i-1}^*, a_i^*) \) and \( \text{SP}(a_i^*, a_{i+1}^*) \) are contained in the region bounded by \( \mathcal{L}, [c_{i+1}, a_i^*], \text{SP}(a_{i-1}^*, a_i^*), \text{SP}(a_i^*, a_{i+1}^*), \) and \( [a_i^*, c_{i-1}] \), for all \( j \geq \overline{j} \), where \( \overline{j} \) is given by (7).

The 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. 10(ii)). One sees immediately that \( m_1 = 1 \) and \( \|c_{i+1} - a_{i+1}^*\| = \|c_i - a_i^*\| = \epsilon_0 \). By (7), we get \( \|a_i - a_i^*\| < \epsilon_0 \) and \( \|a_i - a_i^*\| < \epsilon_0 \). Therefore \( \|a_i^* - a_i^*\| < \|c_i - a_i^*\| / \|s_2 - a_i^*\| \) and \( \|a_i^* - a_i^*\| < \|c_i - a_i^*\| / \|s_2 - a_i^*\| \). Besides, \( \|a_i^* - a_i^*\| = \|s_2 - a_i^*\| / \|s_2 - a_i^*\| \cdot \|c_{i+1} - a_{i+1}^*\| = m_2 \cdot \|c_{i+1} - a_{i+1}^*\| \). Combining with (7) gives \( \|a_i^* - a_i^*\| < \frac{\epsilon_0}{M} = \frac{m_2}{M} \cdot \|c_{i+1} - a_{i+1}^*\| \leq \|c_{i+1} - a_{i+1}^*\| \). These results yield the claim.

**Figure 10: Illustration of the proof of Theorem 1**

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

Let two intersection points of \( \gamma_{j_0} \) and \( [d, d'] \) be \( t \) and \( t' \). By Lemma 3, \( d_K(\text{SP}(a_i^*, a_{i-1}^*), \text{SP}(a_i^*, a_{i+1}^*)) \leq 2 \max\{\|a_i^* - a_i^*\|, \|a_{i+1}^* - a_{i+1}^*\|\} \leq \mu / 2 \). Combining with (4), \( \text{SP}(a_i^*, a_{i+1}^*) \) does not contain vertices of \( V \) except the vertices belonging to \( \hat{\gamma} \). Therefore, all endpoints of \( \text{SP}(a_i^*, a_{i+1}^*) \) except \( a_i^* \) and \( a_{i+1}^* \) belong to \( \text{SP}(a_i^*, a_{i+1}^*) \). Note that the temporary point \( t_i \) of \( \gamma_{j_0} \) is taken as in step 4 of Procedure COLLINEAR UPDATE(\( \gamma_{current}, \gamma_{next}, flag \)). If \( \text{SP}(a_i^*, a_{i-1}^*) \) is a line segment, we have \( \|a_{i-1}^* - t_i\| \geq \frac{1}{2} d([u_i, v_i], [u_{i+1}, v_{i+1}]) \). Otherwise, \( t_i \) is one of the endpoints of \( \text{SP}(a_i^*, a_{i+1}^*) \) except \( a_i^* \) and \( a_{i+1}^* \), and thus \( t_i \in \hat{\gamma} \). Hence \( \|a_{i-1}^* - t_i\| \) is greater than a given constant. As \( \epsilon_0 \) is given by (5), we have \( t_i \notin [a_{i-1}^*, t] \). Similarly, we also get \( t_{i-1} \notin [a_{i+1}^*, t'] \). Because of the properties of shortest paths, \( \text{SP}(t_{i-1}, t_i) \) do not intersect with \([t, t'] \). In the sequence, \( \text{SP}(t_{i-1}, t_i) \) intersects \([u_i, v_i] \) in a point that is above \( a_i^* \) when we see from \( u_i \). This contradicts the result proved above in which \( \gamma_{j_0} \) is below \( \hat{\gamma} \).
**Case 2:** $\angle (\text{SP}(a_{i-1}^*, a_i^*), \text{SP}(a_i^*, a_{i+1}^*)) > \pi$. Since $a_i^* \neq v_i$, there exist two points $d \in \text{SP}(a_{i-1}^*, a_i^*)$, and $d' \in \text{SP}(a_i^*, a_{i+1}^*)$ 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 $\{a_i^*\}$ which is contained in $[e, a_i^*]$. Indeed, suppose that there exists $j_1 \in \mathbb{N}$ such that $a_{j_1}^* \in [e, a_i^*]$. Let two segments having the common endpoint $a_{j_1}^*$ of $\gamma^j$ be $[f, a_{j_1}^*]$ and $[a_{j_1}^*, f']$, where $f \in \text{SP}(a_{j-1}^*, a_{j_1}^*)$ and $f' \in \text{SP}(a_{j_1}^*, a_{j+1}^*)$. Then either $f = a_{j-1}^*$ 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 1, it follows that $\angle fa_{j_1}^*f' \leq \pi$. Because of $\angle ded' = \pi$, we obtain either $[a_i^*, f]$ crossing $[a_i^*, d]$ or $[a_i^*, f']$ crossing $[a_i^*, d']$ (see Fig. 10(iii)). This contradicts the proved result that $\gamma^j$ is below $\hat{\gamma}$ when we view from the boundary $B_1$.

Summarizing, combining 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 $\bar{b}$ and $b$. The proof is complete. □

**The proof of Theorem 2:**

**Proof.** (a) We have (see Figs. 4(iii) and (iv))

\[
I(\gamma_{\text{current}}) = \sum_{i=0}^{N} I(\text{SP}(a_i, a_{i+1}'))
\]

\[
= I(\text{SP}(a_0, t_0)) + \sum_{i=1}^{N} (I(\text{SP}(t_{i-1}, a_i)) + I(\text{SP}(a_i, t_i))) + I(\text{SP}(t_N, a_{N+1}))
\]

\[
\geq I(\text{SP}(a_0^{\text{next}}, t_0)) + \sum_{i=1}^{N} (I(\text{SP}(t_{i-1}, a_i^{\text{next}})) + I(\text{SP}(a_i^{\text{next}}, t_i))) + I(\text{SP}(t_N, a_{N+1}^{\text{next}}))
\]

\[
= \sum_{i=0}^{N} (I(\text{SP}(a_i^{\text{next}}, t_i)) + I(\text{SP}(t_i, a_{i+1}^{\text{next}})))
\]

\[
\geq \sum_{i=0}^{N} (I(\text{SP}(a_i^{\text{next}}, a_{i+1}^{\text{next}}))) = I\left(\bigcup_{i=0}^{N} \text{SP}(a_i^{\text{next}}, a_{i+1}^{\text{next}})\right) = I(\gamma^{\text{next}}).
\]

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

(b) Recall that $\gamma^* = \bigcup_{i=0}^{N} \text{SP}(a_i^*, a_{i+1}^*)$ is the shortest path joining $\bar{b}$ and $b$ in $\mathcal{D}$ and $\gamma^j = \bigcup_{i=0}^{N} \text{SP}(a_i^j, a_{i+1}^j)$ is the sequence of paths obtained by the algorithm, where $a_i^j$ ($a_i^j$, resp.) is the intersection point between $\gamma^*$ ($\gamma^j$, 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 1 (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,
\[ l(\text{SP}(a_j^i, a_{i+1}^i)) \to l(\text{SP}(a_i^*, a_{i+1}^*)), \text{ for } i = 0, 1, \ldots, N \text{ as } j \to \infty. \] It follows \( l(\gamma^j) \to l(\gamma^*) \) as \( j \to \infty \).

\textbf{The proof of Theorem 3:}

\textit{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 \( \gamma_{\text{current}} \) 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 \( \text{SP}(t_{i-1}, t_i) \) in sub-polygons \( \mathcal{D}_{i-1} \cup \mathcal{D}_i \) which call each vertex of \( \mathcal{D} \), time complexity of Procedure \text{COLLINEAR\_UPDATE}(\gamma_{\text{current}}, \gamma_{\text{next}}, \text{flag}) \) is linear. In summarizing, the algorithm runs in \( O(kn) \) time, where \( k \) is the number of iterations to get the required path. \( \square \)