DIRECT MULTIPLE SHOOTING METHOD FOR FINDING APPROXIMATE SHORTEST PATHS IN POLYGONAL ENVIRONMENTS

Phan Thanh An
(joint work with Nguyen Ngoc Hai and Tran Van Hoai)

Institute of Mathematics
Vietnam Academy of Science and Technology
thanhan@math.ac.vn

ABSTRACT
We use the idea of the Direct Multiple Shooting Method (presented by H. G. Bock in Proceedings of the 9th IFAC world congress Budapest, Pergamon Press, 1984, for solving optimal control problems) to introduce an algorithm for solving some approximate shortest path problems in motion planning. The algorithm is based on a direct multiple shooting discretization that includes a collinear condition (a continuity condition type in the Direct Multiple Shooting Method), multiple shooting structure, and approximation conditions. In the case of monotone polygons, it is implemented by a C code and a numerical example shows that our algorithm significantly reduces running time and memory usage of the system.

Keywords: Approximation algorithm, direct multiple shooting method, geodesic convex set, motion planning, shortest path.

1. INTRODUCTION
The following problem is very classical in motion planning: find the Euclidean shortest path between two vertices in a polygonal environment. It has many applications (see [4] and [11]) and many analytical and geometrical properties in the case of simple polygons (see [8]). To date, solving shortest path problems inside simple polygons has usually relied on triangulation of the polygons and graph theory. The question: “Can one devise a simple $O(n)$ time algorithm for computing the shortest path between two points in a simple polygon (with n vertices), without resorting to a (complicated) linear-time triangulation algorithm?” raised by Mitchell [12], is still open. Also, approximation algorithms for solving shortest path problems rely on graph methods (see [1] and [7]). However, in [18], Sharir and Schorr introduced a method to solving shortest path problem by reducing to a collection of subproblems, each of which calls for the computation of the shortest path between a pair of points. But they have not specified how to combine such subproblems.

For solving optimal control problems, an indirect method, namely the Method of Orienting Curves was introduced by Phu in [13]-[14]. The idea of this method was used to introduce new algorithms for solving the convex hull problem in 2D ([3]-[4]) and in 3D ([2]), a version of shortest path problems. So far, no one uses direct methods (for optimal control problems) in solving shortest path problems. This is done and implemented in this paper.

2. DIRECT MULTIPLE SHOOTING METHOD FOR FINDING THE APPROXIMATE SHORTEST PATHS
A direct method, namely the all-at-once approach, or the Direct Multiple Shooting Method was presented by Bock in 1984 and developed recently by Diehl, Sager, Schlöder, etc (see [6], [9], [16], [17] and [19]). Continuity of the state trajectory between the multiple shooting intervals is required only at the solution of the problem.

In this paper, we use the idea of the Direct Multiple Shooting Method to introduce an algorithm for finding the approximate shortest path between two vertices inside a simple polygon without graph methods, but under the assumption [3]. It specifies how to combine subproblems to become the solution of shortest path problems. The continuity condition in the Direct Multiple Shooting Method becomes the collinear condition.
The approximation conditions of the solution originate from our Blaschke-type theorem [8]. Approximation of solutions is presented and approximate errors concentrate at shooting nodes only. This approach presents a step towards practical algorithms.

2.1. Some Geometrical Properties of Shortest Paths

A subset \( S \) of a simple polygon \( D \) is geodesic convex (with respect to \( D \)) if \( GP(x, y) \subset S \) whenever \( x, y \in S \). The geodesic convex hull of a subset \( S \) of a simple polygon \( D \), denoted \( \text{CH}_D(S) \), is the smallest geodesic convex set containing \( S \), i.e., the intersection of all geodesic convex sets (with respect to \( D \)) containing \( S \). Assume that \( \text{int}(\text{CH}_D(S)) \neq \emptyset \). For any points \( x \) and \( y \) in a simple polygon \( D \), there is a shortest path in \( D \) that connects \( x \) and \( y \). This path is unique and is a polyline (see [10] and [12]). We denote it by \( GP(x, y) \).

**Theorem 2.1.1.** Suppose that \( a \) and \( b \) are disjoint points of a simple polygon \( D \), \( \xi = [u, v] \) is a cut segment of \( D \) and the segment \([u, v]\) strictly separates \( a \) and \( b \). Let \( \alpha_u \) be the angle \( \angle GP(u, a), GP(u, b) \) which is an interior angle of the polygon \( \text{CH}_D(a, v, b, u) \) at the vertex \( u \) and \( \alpha_v \), the angle \( \angle GP(v, a), GP(v, b, u) \), which is an exterior angle of the polygon \( \text{CH}_D(a, v, b, u) \) at the vertex \( v \). Then,

i) \((\pi - \alpha_u)(\pi - \alpha_v) < 0 \) if \( GP(a, b) \) intersects \([u, v]\).

ii) If \( \alpha_u \geq \pi \) (\( \alpha_v \leq \pi \), respectively) then \( GP(a, b) \) goes through \( u \) (\( v \), respectively) and \( \alpha_u > \pi \) (\( \alpha_v < \pi \), respectively).

**Theorem 2.1.2.** (Monotonicity of angles on separate cut segments) Suppose that \( a \) and \( b \) are disjoint points of a simple polygon \( D \), \( \xi = [u, v] \) is a cut segment of \( D \) and the segment \([u, v]\) strictly separates \( a \) and \( b \), and \( x \in [u, v] \). Let \( \alpha_x \) be the angle \( \angle GP(x, a), GP(x, b) \), which is an interior angle of the polygon \( \text{CH}_D(a, v, b, x) \) at the vertex \( x \), \( GP(a, b) \) intersects \([u, v]\) at \( z \), and \( \alpha_u < \pi \). Then \( \alpha_u \) is monotone on \([u, z]\) and \([z, v]\) (i.e., \( \alpha_x_1 \leq \alpha_x_2 \leq \pi \) (\( \alpha_x_1 \geq \alpha_x_2 \geq \pi \), respectively) with \( x_1 \in [u, x_2] \subset [u, z] \) and \( \pi \leq \alpha_x_2 \leq \alpha_x_1 \) (\( \pi \geq \alpha_x_2 \geq \alpha_x_1 \), respectively) with \( x_2 \in [x_1, v] \subset [z, v] \).

We say a path \( X \) starting at the vertex \( a \) and ending at the vertex \( b \) of a polygon \( Y \) is clockwise (respectively) w.r.t. \( Y \) if the interior of this polygon lies to our left (right, respectively) as we step along \( X \), starting at \( a \).

2.2. Direct Multiple Shooting Discretization

In this section we describe the use of the idea of the direct multiple shooting method [9] for the discretization and parametrization of the shortest path problem. We consider the following shortest path problem:

Find the Euclidean shortest path \( Z \) between two vertices \( a \) and \( b \) in a simple polygon \( D = PQ \) in which \( \mathcal{P} \) (\( \mathcal{Q} \), respectively) is the polyline formed by vertices of the polygon from \( a \) to \( b \) (from \( b \) to \( a \), respectively) in counterclockwise order. We split the polygon \( \mathcal{P} \mathcal{Q} \) into subpolygons \( \mathcal{P}_i \mathcal{Q}_i \) by cut segments \( \xi_1, \ldots, \xi_k \) (i.e., a shooting grid) as follows:

\[
\xi_i = [u_i, v_i] \subset \mathcal{D} \text{ such that } \{u_i, v_i\} \subset \text{int} \mathcal{D} \quad \text{[u_i, v_i] strictly separates a and b}
\]

\( T_i \) is bounded by \( \mathcal{P}, \mathcal{Q} \) and

\[
\mathcal{P} \mathcal{Q} = \bigcup_{i=0}^{k} T_i \quad \text{int} T_i \cap \text{int} T_j = \emptyset \text{ for } i \neq j
\]

with \( \xi_0 := a, \xi_{k+1} := b \) (see Fig. [2]). From now we assume that \( \xi_1, \ldots, \xi_k \) satisfy the following "oracle":

\[
GP(u_i, u_{i+1}) \cap GP(v_i, v_{i+1}) \neq \emptyset \quad \text{for all } i = 1, \ldots, k - 1
\]

The basic concept of the direct multiple shooting method is to determine the shortest paths \( Z_i \), independently on each subpolygon \( \mathcal{P}_i \mathcal{Q}_i \). Suppose that \( GP(a, b) \) does not intersect \( \xi_j \) at \( u_i \) or \( v_i \). We choose shooting points \( a_j \) on the cut segments \( \xi_j \) as near as possible to \( GP(a, b) \); it would be ideal if the \( a_j \) actually lie on \( GP(a, b) \), i.e., the Collinear Condition

\[
\angle GP(a_i, a_{i+1}), GP(a_i, a_{i+1}) = \pi \text{ with } a_i \in \xi_i \cap \text{int} \mathcal{P} \mathcal{Q}, i = 1, \ldots, k (a_k = a \text{ and } a_{k+1} = b)
\]

holds true.

Note that the Collinear Condition is obtained from Theorem 2.1.2.

The Multiple Shooting Structure consists of the following conditions (with \( u \in \mathcal{P} \) and \( v \in \mathcal{Q} \)):

1. \( \xi = [u, v] \) is at non-corner case: \((\pi - \alpha_u)(\pi - \alpha_v) < 0 \). (Then, by Theorem 2.1.1 i), \( GP(a, b) \) intersects \([u, v]\),

2. \( \xi = [u, v] \) is at corner case: \( \alpha_u \geq \pi \) (\( \alpha_v \leq \pi \), respectively). (Then, by Theorem 2.1.1 ii), \( GP(a, b) \) goes through \( u \) (\( v \), respectively)).

It is an open problem whether faster procedures than the straightforward one just sketched exist for solving the combinatorial shortest path problem (see [18]). It is noteworthy that this combinatorial shortest path problem is at least solvable in finite time (Prop. 3.4 [15]).
Figure 1: \( \text{GP}(x, y, \xi, \mu) \) finds the shortest path \( \mathcal{Z} \) between points \( x \in \xi \) and \( y \in \mu \) in the simple polygon bounded by polylines \( P, Q, \) and cut segments \( \xi \) and \( \mu \).

2. New Algorithm

Procedure \( \text{GP}(x, y, \xi, \mu) \) (\( \text{GP} \) stands for Geodesic Path)

Given a simple polygon bounded by \( P, Q \) and boundary edges \( \xi, \mu \), and points \( x \in \xi \) and \( y \in \mu \).

\( \text{GP}(x, y, \xi, \mu) \) finds the shortest path \( \mathcal{Z} \) between points \( x \) and \( y \) in the polygon.

\( \text{GP}(x, y, \xi, \mu) \) can be determined without triangulation of the simple polygon \( P Q \) by [5] (see Fig. 1). The polygon \( P Q \) is divided into subpolygons \( T_i \) by cut segments \( \xi_1, \ldots, \xi_k \) \((k \geq 1)\) satisfying [1]-[4]. Fix \( k \geq 1 \).

Algorithm 2.3

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

Find: the shortest path \( \mathcal{Z} \) between \( a \) and \( b \) inside \( P Q \).

1. Divide the polygon \( P Q \) into suitable subpolygons \( T_i \) by cut segments \( \xi_1, \ldots, \xi_k \) satisfying [1]-[3].

   Set \( j = 0 \). Initial shooting points \( a_i^j \in \xi_i, \) \( i = 1, \ldots, k \).

2. Call \( \text{GP}(a_i^j, a_{i+1}^j, \xi_i, \xi_{i+1}) \) to get the shortest path \( \mathcal{Z}_i^j \) connecting \( a_i^j \) and \( a_{i+1}^j \) in \( T_i \). Check if all \( \mathcal{Z}_i^j \) \((i = 1, \ldots, k)\) satisfy simultaneously the Multiple Shooting Structure:

   If either \([a_i^j, a_{i+1}^j]\) is at non-corner case (i.e., Multiple Shooting Structure 1.) and the Collinear Condition holds true \((\angle \text{GP}(a_i^j, a_{i-1}^j), \text{GP}(a_i^j, a_{i+1}^j) = \pi)\), or \([a_i^j, a_{i+1}^j]\)

   is at corner case (i.e., Multiple Shooting Structure 2.)

   then \( \mathcal{Z} = \bigcup_{i=1}^k \mathcal{Z}_i^j \). STOP. Else, refine shootings \( a_i^j \in \xi_i \) to ensure that Collinear Condition holds true. Set \( j = j + 1 \), goto 2.

Algorithm 2.3 is illustrated by Fig. 2. The number \( k \) of cut segments \( \xi_i, \) \( i = 1, \ldots, k \), is specified by the user.

3. NUMERICAL RESULTS

The algorithm in Sect. 2.3 is implemented in C code. The algorithm for finding shortest paths \( \mathcal{Z}_i^j \) in subpolygons \( \text{GP}(a_i^j, a_{i+1}^j, \xi_i, \xi_{i+1}) \) and its implementation are due to [5]. \( P \) and \( Q \) are monotone with respect to \( x \)-coordinate axis and no 3 vertices of \( P Q \) are collinear, the cut segments \( \xi_1, \ldots, \xi_k \) are parallel to \( y \)-coordinate axis, and each cut segment starts from a vertex of \( P \)

Thus [1] is satisfied. As we see from Fig. 3 (where \(|P| = 100000, |Q| = 80000 \) and \( \epsilon = 10^{-3} \)), our algorithm significantly reduces the running time as we increase the number \( k \) of cut segments.

If the algorithm for finding shortest paths \( \mathcal{Z}_i^j \) in subpolygons \( \text{GP}(a_i^j, a_{i+1}^j, \xi_i, \xi_{i+1}) \) and for finding the
Figure 3: The running time of our algorithm decreases significantly from 399.125 seconds to 1.94997 seconds as the number $k$ of cut segments increases from 0 to 2500. Here, $k = 0$ means the shortest path between $a$ and $b$ inside the polygon is determined by [5] without the Direct Multiple Shooting Method.

shortest path $GP(a,b)$, is determined by [10] in which, the triangulation of the corresponding polygon is from [15]. Algorithm [2,3] also significantly reduces the peak memory usage of the system as we increase the number $k$ of cut segments (see Table 1). Here, they are implemented on Ubuntu Linux operating system with platform Pentium(R) 4, core 2, 3GHz, 1GB Memory and 3GB Swap space.

4. REFERENCES


Table 1: Both the running time of our algorithm and the peak memory usage of the system significantly reduce as cut segments are used.

<table>
<thead>
<tr>
<th>Number of cut segments</th>
<th>Time (in sec.)</th>
<th>Peak Memory (in MB)</th>
<th>Peak Memory (in MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(freed on nodes)</td>
<td>(freed on tree)</td>
</tr>
<tr>
<td>0</td>
<td>20858.53</td>
<td>0.80</td>
<td>4839.91</td>
</tr>
<tr>
<td>5</td>
<td>3641.00</td>
<td>0.13</td>
<td>2040.96</td>
</tr>
<tr>
<td>10</td>
<td>1988.84</td>
<td>0.07</td>
<td>610.29</td>
</tr>
<tr>
<td>100</td>
<td>298.82</td>
<td>0.00</td>
<td>7.85</td>
</tr>
<tr>
<td>1000</td>
<td>283.01</td>
<td>0.00</td>
<td>0.09</td>
</tr>
</tbody>
</table>


