2-Approximate Algorithm for Touring a Sequence of Pairwise Disjoint Simple Polygons

Shun LYU and Bo JIANG*

Dalian Maritime University, Linghai Road 1, Dalian, China

*Corresponding author

Keywords: Computational geometry, Touring polygons problem, Approximate algorithm, Simple polygon, Ratio.

Abstract. In this paper, a 2-approximate algorithm is described to answer the previously open problem “What is the complexity of the TPP for disjoint non-convex simple polygons” which is known to NP-hard. We provide a O(kn) approximate algorithm, where k is polygon counts, and n is the number of vertexes of the polygons, to efficiently find a path which is 2 times at most than the shortest path. To solve this problem, we transform all of simple polygons into corresponding convex polygons, then process the shortest path of convex polygons according to the parity of polygons sequence and finally obtain the approximate path of simple polygons.

Introduction

Touring Polygons Problem

A fundamental problem in computational geometry is to find the Euclidean shortest path from the start point s to target point t in the plane. In some applications, the path is often required to visit some constrained regions, such as zookeeper problem, safari problem, and watchman route problem [1]. Touring polygons problem (TPP)[2] is a general problem including some visiting properties mentioned above and it is described as follows:

Let $\pi$ be a plane, a sequence of pairwise disjoint simple polygons $P=(P_1, P_2, \ldots, P_k) \in \pi$. Two points $s, t \in \pi \setminus \bigcup_{i=1}^{k} P_i$. Simple polygon $P_i$ is a flat shape consisting of straight, non-intersecting line segments or ‘sides’ that are joined pairwise to form a closed path. Our goal is to compute the shortest path $L_{opt}=(s, p_1, p_2, \ldots, p_k, t)$ that starts at $s$, visiting each polygon $P_i (i=1, \ldots, k)$ in order, and ends at $t$. Note that the access point of each $p_i$ is the first point that the path intersects with the polygon $P_i$. See Figure 1.

Figure 1. An instance of TPP with the shortest path $L_{opt}=(s, p_1, p_2, p_3, p_4, p_5, t)$.

The cutting problem exists widely in clothing, machine manufacturing industry [3]. Suppose we have a sheet of some materials, such as glass or metal. There are a set of polygonal materials that must be cut out with a robot from a sheet. It is necessary to cut the edges of current polygon in a continuous cycle before the robot arrives to the next polygon. How to efficiently cut all of the materials can be transformed to obtain a minimum path that visits the border of all the polygons, since sum of the perimeter of each polygon is a constant. The cutting problem consequently is belonged in unconstrained TPP.
Current Work

TPP is a generalization of the traveling salesman problem (TSP) which is NP-hard if the order of the polygons is not given [2,4,5]. We focus on solving the problem where the polygons must be visited in order.

When all of the polygons are in order, convex, pairwise disjoint, and start point, target point are fixed, an $O(kn)$ algorithm to solve the problem is proposed in [6], where $n$ is the total number of vertexes of polygons. Once non-convex polygon is allowable, the touring problem will be NP-hard[4].

As a follow-up of this work, Pan X et al. provided a heuristic algorithm where the given convex polygons are not obligatory [7]. An $(1+\frac{(L_2-L_1)}{L})$-approximate solution is proposed in time $k(\varepsilon)O(n)$, where $n$ is the total number of vertexes of polygons, $k(\varepsilon)$ defines the numerical accuracy in dependency of a selected $\varepsilon > 0$. The approximate algorithm mainly adopts Rubber-band algorithm [8] (a algorithm based on iteration) and the theoretical upper time bound $k(\varepsilon)O(n)$ is replaced in practice by $O(n^2)$.

In this paper, we provide a algorithm named CHDP to achieve a 2-approximate solution and upper time bound $O(kn)$ in theory. We construct the convex hull of each simple polygon at first, and then we take the method of dividing the polygon in order and locating the edges in the reverse order to obtain the shortest path of convex hulls order, successively regulate odd polygons order and even polygons order in the end.

Our Work

In this paper, we propose an approximate algorithm to solve the unconstrained TPP, where simple input polygons have to satisfy the condition that their convex hulls are pairwise disjoint. The approximate path is less than 1.2 times of the shortest path after several practical experiments, and less than 2 times of the shortest path in theory if the distance between adjacent polygons is not short. That the distance is not short means polygon $P_{i,1}$ and polygon $P_{i+1}$ do not include in circle $O_i$ or intersect with circle $O_i$, where $O_i$ is denoted by segment $s_i$ of convex hull $CP_i$, $i=1, 2, 3, \ldots n$.

Construct Convex Hulls and Touring Convex Hulls Order

Touring disjoint polygons is NP-Hard [4], however, touring disjoint convex polygons can be solved in time $O(kn)[6]$. Therefore, we construct the convex hulls of each polygon and transfer simple polygons problem into convex polygons problem.

**Lemma 1.** The shortest path of touring simple polygons $P(P_1, P_2, \ldots, P_k)$ is not shorter than touring the convex hulls $C(P)$ ($CP_1$, $CP_2$, ..., $CP_k$) of corresponding polygons $P(P_1, P_2, \ldots, P_k)$, $|D_{cp-opt}| \leq |D_{p-opt}|$. $|D_{cp-opt}|$ is the shortest path of touring convex hulls $C(P)(CP_1, CP_2, \ldots, CP_k)$, and $|D_{p-opt}|$ is the shortest path of touring simple polygons $P(P_1, P_2, \ldots, P_k)$.

**Proof.** Assumed that there are a ordered set of polygons $P(P_1, P_2, \ldots, P_k)$, and corresponding convex hulls $C(P)(CP_1, CP_2, \ldots, CP_k)$. Define the shortest path for arbitrary polygons as $D_p(p_1, p_2, \ldots, p_k)$ and convex hulls as $D_{cp}(cp_1, cp_2, \ldots, cp_k)$. $p_i$, $cp_i$ is the initial point that the shortest path intersect with $P_i$, $C(P_i)$, $i=1, 2, \ldots, n$. Obviously, $p_i$ is on the frontier of $P_i$, and it could exist on the border of $C(P_i)$ or inside $C(P_i)$. If each point of $p_i$ is on the border of $C(P_i)$, then the distance $D_p(p_1, p_2, \ldots, p_k) = D_{cp}(cp_1, cp_2, \ldots, cp_k)$, $|D_{cp-opt}| = |D_{p-opt}|$. If there is a point $p_i$ inside $C(P_i)$, its incident line intersect $C(P)$ at $p_i^\prime$. Because of $D(p_i^\prime, p_i) + D(p_i, p_{i+1}) > D(p_i^\prime, p_{i+1})$ by triangle principle, we can get $D_{cp}(cp_1^\prime, cp_2^\prime, \ldots, cp_k^\prime) < D_p(p_1, p_2, \ldots, p_k)$, where $cp_1^\prime = p_1$ or $cp_i^\prime$= incident point $p_i^\prime$. $D_{cp}(cp_1^\prime, cp_2^\prime, \ldots, cp_k^\prime)$ is a path touring the convex hulls and consequently $D_{cp}(cp_1, cp_2, \ldots, cp_k) \leq D_{cp}(cp_1^\prime, cp_2^\prime, \ldots, cp_k^\prime) < D_p(p_1, p_2, \ldots, p_k)$, $|D_{cp-opt}| < |D_{p-opt}|$. See Figure 2.
Here, each convex hull is a convex polygon, therefore, we adopt the new algorithm on touring $C(P)(CP_1, CP_2, \ldots, CP_n)$.

Touring Pairwise Disjoint Simple Polygons

Obviously, the shortest path of touring convex hulls order is not the path of touring simple polygons. The key is updating the points which are not on the frontier of simple polygons. If we update each point one by one, each point $p_i'$ will update by an updated point $p_{i+1}$ and a non-update point $p_{i+1}'$ under the worst situation. The approximate path may be far from the shortest path of touring $C(P)$ under one time of iteration, because each update point $p_i$ is influenced by all previous update points. If we update the path by separately regulating odd polygons order and even polygons order, each point $p_i'$ will update by two non-update points $p_{i+1}'$, $p_{i+1}$ at first round, and update by two update points at second round. Each point is not influenced by other update points or is influenced once at most under the worst situation. This method can reduce the error effectively.

**Algorithm CHDP**

**Main procedure**

**Input:** a sequence of simple polygons $P(P_1, P_2, \ldots, P_n)$ in the plane $\pi$, two point $s$, $t$.

**Output:** the approximate path $L_a(s, p_1, p_2, \ldots, p_n, t)$ which starts at point $s=p_0=p_0'$, then visits the polygons $P(P_1, P_2, \ldots, P_n)$ at point $p_i$ in order, and ends at point $t=p_{n+1}=p_{n+1}$, $i=1, 2, \ldots, n$.

Step1. Apply Melkman algorithm[9] to compute $C(P)(CP_1, CP_2, \ldots, CP_n)$.

Step2. Compute the shortest path $L_a(s, p_1', p_2', \ldots, p_n', t)$ of touring $C(P)$ called touring convex hulls procedure[6,10].

Step3. For $i=1, 3, \ldots, n$, $i$ is odd

- If $p_i'$ exist on $P_i$ (also exist on $CP_i$), then $p_i'=p_i$;
- If $p_i'$ do not exist on $P_i$(exist on $CP_i$), find a point $q_i \in \partial P_i$, such that: $D(p_{i-1}', q_i)+D(q_i, p_{i+1}') = \min \{ D(p_{i-1}', p_i)+D(p_i, p_{i+1}), p_i \in \partial P \}$, update $p_i=q_i$.

For $i=2, 4, \ldots, n$, $i$ is even

- If $p_i'$ exist on $P_i$ (also exist on $CP_i$), then $p_i'=p_i$;
- If $p_i'$ do not exist on $P_i$(exist on $CP_i$), find a point $q_i \in \partial P_i$, such that: $D(p_{i-1}', q_i)+D(q_i, p_{i+1}) = \min \{ D(p_{i-1}, p_i)+D(p_i, p_{i+1}), p_i \in \partial P \}$, update $p_i=q_i$.

Step4. Output the approximate path $L_a(s, p_1, p_2, \ldots, p_n, t)$.

Step1, owing to all of certain polygons, Melkman algorithm compute $C(P)$ in linear time. Step2, the shortest path $L_a(s, p_1', p_2', \ldots, p_n', t)$ of touring $C(P)$ can be computed in $O(kn)$. Step3 and step4 output the approximate path in $O(n)$ time. Thus, CHDP algorithm runs in time $O(kn)$, where $k$ is the number of polygons and $n$ is the number of vertexes.
Approximate Path

We adopt ratio to describe the performance of approximate algorithm. In this paper, ratio is \( \frac{L_{opt}}{L_{approx}} \), where \( L_{opt} \) is the approximate path and \( L_{approx} \) is the shortest path. Ratio should be as low as possible.

Our algorithm must satisfy the condition that their convex hulls are pairwise disjointed; it means the distance between adjacent polygons should not be too short. See Figure 3a. In addition, there are still some situations that are thought impractical because of close distance between adjacent polygons. See Figure 3b.

![Diagram](image)

Figure 3. The distance of adjacent polygons can not be short.

Above all, we denote that circle \( O_{i} \) has diameter \( s_{i}, i=1, 2, ..., n \), polygon \( P_{i+1} \) and polygon \( P_{i+1} \) do not include in circle \( O_{i} \) or intersect with circle \( O_{i} \). There are three cases of update path contacted with \( CP_{i} \), which are as follows:

![Diagram](image)

Figure 4. Types of contact of update path with \( CP_{i} \).

**Case 1:**Passing through.
If local shortest path of touring convex hulls pass through \( CP_{i} \), the shortest path \( L_{o}(s, p_{1}{'} , p_{2}{'} , ..., p_{n}{'} , t) \) intersect \( P_{i} \) at point \( p_{i} \) inevitable. Update path is unchanged and \( p_{i} \neq p_{i} \). See Figure 5a.

**Case 2:** Vertex bending.
If \( p_{i}{'} \) is a bend point, it is on the frontier of \( P_{i} \). Thus \( p_{i} = p_{i}{'} \), update path is unchanged. See Figure 5b.

**Case 3:** Edge reflection.
If local shortest path reflect at point \( p_{i}{'} \), and \( p_{i}{'} \) is not on the frontier of \( P_{i} \). Update point \( p_{i} \) change, and local update path \( \langle p_{i-1}p_{i} \rangle + \langle p_{i}p_{i+1} \rangle \leq \sqrt{2}(\langle p_{i-1}p_{i} \rangle + \langle p_{i}p'_{i+1} \rangle) \). See Figure 5c.

**Lemma 2.** \( \langle p_{i-1}p_{i} \rangle + \langle p_{i}p_{i+1} \rangle \leq \sqrt{2}(\langle p_{i-1}p_{i} \rangle + \langle p_{i}p'_{i+1} \rangle) \).

**Proof.** \( |p_{i+1}p_{i}|-|p_{i}B|, |p_{i+1}p_{i}|-|p_{i}B|=|p_{i}B|, |p_{i+1}A|-|A B|, |p_{i+1}A|-|A B|=|A B|+|A p_{i-1} |\).

\( \angle p_{i+1}BC=a, \angle p_{i+1}AC=b, a \in (0, 90°), b \in (0, 90°) \).

\( \frac{|A B|+|A p_{i-1}|}{|p_{i+1}B|}=|p_{i+1}B| \times \cos \alpha + |A p_{i-1}| \times \cos \beta + |A p_{i+1}| \times \cos \beta + |p_{i}B| \times \sin \alpha \times \frac{\sin \alpha}{\sin \beta} \times |p_{i}B| \times \sin \beta \times \cos \gamma / |p_{i+1}B| \)

\( = \cos \alpha + \sin \alpha \times \frac{1-\cos \beta}{\sin \beta} = \cos \alpha + \sin \alpha \times \tan \frac{\beta}{2} \leq \cos \alpha + \sin \alpha = \sqrt{2} \sin 2\alpha \leq \sqrt{2} \)

**Theorem 1.** The approximate path do not exceed \( 2 \) times of the shortest path.
Proof. Assumed that \(|s, p_1'|=l_1, |p_1', p_2'|=l_2, \ldots, |p_{n'}, t|=l_{n+1}'. The shortest path \(L_c(s, p_1', p_2', \ldots, p_{n'}, t)\) of touring \(C(P)\), \(L_c=l_1+l_2+\ldots+l_{n+1}'. In addition, |s, p_1|=l_1, |p_1, p_2|=l_2, \ldots, |p_{n}, t|=l_{n+1}'. The approximate path \(L_a(s, p_1, p_2, \ldots, p_n, t)\), \(L_a=l_1'+l_2'+\ldots+l_{n+1}'. The shortest path of touring simple polygons \(P(P_1, P_2, \ldots, P_n)\) is called \(L_{opt}\), \(L_a \leq L_{opt}\). If \(p_i'\) do not exist on \(P_i\) (exist on \(CP_i\) and the edge \(s_i\)), \(r_i\) is the radius of circle \(O_i\), \(r_i \leq l_i, i=1, 2, \ldots, n\).

\[
\frac{L_a}{L_{opt}} \leq \frac{L_a}{L_c} = \frac{(l_1'+l_2'+\ldots+l_{n+1}')}{(l_1+l_2+\ldots+l_{n+1})} \\
= \frac{(l_1'+l_2'+\ldots+l_{n+1}')}{(l_1+l_2+\ldots+l_{n+1})} \\
\leq \sqrt{2} \left( l_1'+l_2'+\ldots+l_{n+1}' \right) / (l_1+l_2+\ldots+l_{n+1}) \\
\leq \sqrt{2} \left( l_1''+l_2''+\ldots+l_{n+1}'' \right) / (l_1+l_2+\ldots+l_{n+1}) \\
\leq \sqrt{2} \times \frac{L_{opt}}{L_c} \leq 2
\]

The algorithm provide a 2-approximate solution for unconstrained TPP in time \(O(nk)\), where \(k\) is the number of polygons and \(n\) is the number of vertexes.

The Implementation of Algorithm

The approximate algorithm has been implemented by program. We prove \(\frac{L_a}{L_{opt}} \leq 2\) through mathematical theory. On the one hand, majority of points \(p_i'\) exist on \(P_i\), \(i=1, 2, \ldots, n\). On the other hand, \(L_c\) is far less than \(L_{opt}\) in bad cases. \(\frac{L_a}{L_{opt}}\) is much less than 2 in practical result after several experiments with randomly polygons, see Table 1 and Figure 5.

<table>
<thead>
<tr>
<th>Polygon counts</th>
<th>(L_c)</th>
<th>(L_a)</th>
<th>(La/Lc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1603.36</td>
<td>1603.36</td>
<td>1.00</td>
</tr>
<tr>
<td>5</td>
<td>1346.93</td>
<td>1389.21</td>
<td>1.03</td>
</tr>
<tr>
<td>5</td>
<td>1693.86</td>
<td>1779.58</td>
<td>1.05</td>
</tr>
<tr>
<td>10</td>
<td>3676.51</td>
<td>3823.71</td>
<td>1.04</td>
</tr>
<tr>
<td>10</td>
<td>3635.48</td>
<td>4039.02</td>
<td>1.11</td>
</tr>
<tr>
<td>10</td>
<td>3536.89</td>
<td>3601.23</td>
<td>1.02</td>
</tr>
<tr>
<td>15</td>
<td>6998.16</td>
<td>7587.21</td>
<td>1.08</td>
</tr>
<tr>
<td>15</td>
<td>8278.29</td>
<td>9321.01</td>
<td>1.13</td>
</tr>
<tr>
<td>15</td>
<td>7690.23</td>
<td>7741.65</td>
<td>1.01</td>
</tr>
</tbody>
</table>

Figure 5. \(L_c=1094.46, L_a=1127.34\). \(L_c = 1.03\).
Conclusion

In this paper, we propose a 2-approximate solution for touring arbitrary simple polygons with the condition that the distance of pairwise polygons are not too short. Our algorithm gives an upper time bound $O(kn)$ in practice, where $k$ is the number of polygons and $n$ is the number of vertexes.

Touring simple polygons is an open problem. How to solve the problem that adjacent polygons are intersecting is an interesting research problem. How to precondition convex hulls if adjacent convex hulls are intersecting is also a research area and we are now working on it. In addition, there are some limits when we prove the ratio. How to reduce these limits and improve ratio are our future study as well.

Acknowledgements

This work is supported by Liaoning Province Natural Science Foundation (No.20170540147) and the General Project of Liaoning Province Science and Research (No. L2015105).

References


