アルゴリズム 43−2
（1995. 1. 23）

New Algorithms for the Zoo-keeper Route Problem

Xuehou TAN
School of High-Technology for Human Welfare, Tokai University
317 Nishino, Numazu 410-03, Japan

Abstract

Let P be a simple polygon and let $\mathcal{P}$ be a set of convex polygons that lie in the interior of P and are attached to the boundary of P. The *zoo-keeper route problem* asks for a shortest route inside P that visits (but does not enter) each polygon in $\mathcal{P}$. Let n be the total number of vertices of polygon P and polygons in $\mathcal{P}$. We present two new algorithms for the zoo-keeper route problem; one is deterministic and runs in $O(kn)$ time where k is the maximum size of polygons in $\mathcal{P}'$, and the other gives an approximate solution that is guaranteed to be within $\pi/2$ times the optimal solution value and takes only linear time. Both of our results improve upon the previous results.

動物園経路問題に関する新しいアルゴリズム

譚　学厚

東海大学開発工学部

Pを単純多角形とし、P'をPの境界線に付けられる凸多角形の集合とする。動物園経路問題とは、P'の多角形をすべて訪れる最短経路を求めることである。nを多角形PとP'の各多角形の総辺数とする。本論文では動物園経路問題を解く二つの新しいアルゴリズムを与える。一つは決定的なアルゴリズムであり、O(kn)の時間を要する。ここでkはP'の多角形の最大サイズである。もう一つは線形時間で近似解法を与える。近似解は最適解のπ/2倍内であることが保証される。これらのアルゴリズムは以前の結果を改良する。

1 Introduction

Shortest paths are of fundamental importance in robotics and computational geometry. The *zoo-keeper route problem*, introduced by Chin and Ntafos [2], is defined as follows: Given a simple polygon P (the *zoo*) with a set $\mathcal{P}$ of disjoint convex polygons (the *cages*) inside it, each sharing one edge with the polygon P, find a shortest route inside P that starts from a given vertex s of P, visits (without entering) at least one point of each cage and finally returns back to s (Fig. 1). One may consider it as minimizing the route for a zoo-keeper to feed animals.

In some sense, the zoo-keeper route problem looks like the well known *Traveling Salesperson Problem* if we consider cage as cities. But, we actually know that the shortest zoo-keeper route has to visit cages in the order they appear in the boundary of P, since otherwise it would cross itself and could be shortened. On the other hand, the shortest zoo-keeper route is mainly determined by cages, rather than by zoo P. See also Fig. 1. (It is absolutely true when the given polygon P is convex.) If we consider cages as cities in the Traveling Salesperson Problem, then it is reasonable to assume that each cage has a small number of edges and the number of cages is very large. There is already such a research in which each cage is assumed to be a single edge [3].

Let n denote the total number of edges of polygon P and polygons in $\mathcal{P}$. With the unfolding and adjusting techniques, Chin and Ntafos gave an $O(n^2)$ time algorithm for the zoo-keeper route problem [2]. Recently, the result was improved to $O(n \log^2 n)$ by Hershberger and Snoeyink [9], using a complicated data structure for shortest-path queries in a simple polygon [8]. In this paper, we present two new algorithms for the zoo-keeper route problem. One is deterministic and runs in $O(kn)$ time, where k is the maximum size of polygons in $\mathcal{P}$, i.e., $k = max\{|P_i| \; |P_i \in \mathcal{P}\}$. As pointed out above, k is usually rather small, say, smaller than $O(\log n)$. The other gives an approximate solution, which is guaranteed to be within $\pi/2$ times the optimal solution value. The time complexity of the approximate algorithm is $O(n)$. Both of our results improve upon the previous known results.

2 Preliminaries

In order to solve the zoo-keeper route problem, Chin and Ntafos' algorithm [2] makes use of the unfolding method. It is an extension of the classical method of finding the shortest path between two points a and b, where the path must touch a given line L and both a and b lie in one side of L. To find such shortest path, we first reflect b across L to get the image b', then draw a straight line segment $\overline{ab'}$ from a to b', and finally fold back the portion of $\overline{ab'}$ lying in the other side of L to obtain the desired shortest path. If L is a line segment, the shortest path can also be found analogously.

The shortest zoo-keeper route touches at least one edge on each cage. Suppose we have a set of edges, one per cage. Then the corresponding optimum (not shortest) zoo-keeper route can be constructed by unfolding the interior (or triangulation) of $P - \mathcal{P}$ using each edge in the set as mirrors, then finding the shortest path between s and its image s' in the unfolded polygon and finally folding back the shortest path to get the zoo-keeper route. This unfolding process takes linear time [2]. To obtain the set of the edges with which the shortest zoo-keeper route makes contacts, Chin and Ntafos' algorithm first select an initial set of edges and then compute the initial zoo-keeper route using the unfolding method. Since the initial zoo-keeper route is not the shortest, it can be adjusted (or shortened) at some places. A zoo-keeper route is *clockwise (counter-clockwise) adjustable* at a vertex v_i of a cage P_x if and only if the incoming angle of the route with the supporting line of edge $\overline{v_i v_{i+1}}$ ($\overline{v_{i-1} v_i}$) is greater (smaller) than the outgoing angle. An adjustment involves a change in the edge set and calls the unfolding process once to compute the new shorter zoo-keeper route. In [2], the difference between the incoming and outgoing angles is defined as the *sliding tension* of the current route at cage P_x, which can be used as a measure of the adjustability of the route. Chin and Ntafos' algorithm always selects the adjustment with the maximum tension from possible candidates. In the adjusting process, the length of the current route decreases monotonically. When the current route can not be adjusted any more, it gives the shortest zoo-keeper route. Chin and Ntafos showed that the current route in their algorithm moves along the boundary of each cage in a single direction, i.e., the contact points of the current route with a cage P_i are well ordered on the boundary of P_i in the adjusting direction. This observation gives an $O(n^2)$ time algorithm, since it requires at most n adjustments before the shortest zoo-keeper route is found.

In the Section 3 of this paper, we make an important observation that all possible adjustments can be done once together, regardless of different sliding tensions. The current route in our algorithm moves along the boundary of each cage in a single direction, too. Thus, it gives us an $O(kn)$ time algorithm, where k is the maximum size of cages in $\mathcal{P}$. In Section 3, we further present a linear-time approximate algorithm, which is guaranteed to be within $\pi/2$ times the optimal solution value.

3 The deterministic algorithm

Our deterministic algorithm for computing shortest zoo-keeper routes is as follows:

Step 1. Index cages $P_1, \cdots, P_m$ along the boundary of zoo P in the clockwise order.

Step 2. Choose the leftmost edges of odd-numbered cages and the rightmost edges of even-numbered cages to form the initial set of edges.

Step 3. Compute the initial route R_0 for the edge set chosen in Step 2 and let $i = 0$.

Step 4. **While** the current route R_i is adjustable **do**

Perform all adjustments together, then compute the new zoo-keeper route R_{i+1} using the unfolding method and let $i \leftarrow i + 1$.

Step 5. Report the final (non-adjustable) route R_i as the shortest zoo-keeper route.

In the above algorithm, Steps 1 to 3 select a special set of edges, one per cage, to construct the initial zoo-keeper route. This set of edges is so selected that the initial route tends to be adjustable on each cage from one extreme (the leftmost or rightmost) vertex to the other.

The only difference between our and Chin-Ntafos' algorithms is Step 4. While Chin-Ntafos' algorithm performs adjustments one by one, our algorithm does all possible adjustments once together. The correctness of our algorithm is shown by the following lemma.

Definition 1 *The direction of an adjustment on a cage P_i is the shift direction of the contact points on the boundary of P_i.*

Lemma 1 *Let R_0 be the initial zoo-keeper route constructed in Steps 1 to 3. The adjustable direction on any cage can not be changed in Step 4.*

Proof. The initial route R_0 has the property that it makes contacts with each P_i at a point that is clockwise (i.e., towards higher indexed vertices) from the point of the contact of the shortest route if i is odd and counterclockwise (towards lower indexed vertices) from it if i is even. That is, if one unfolds both the initial route R_0 and the shortest route R, then the unfolded route R_0 lies in one side of the unfolded route R throughout its extent. Furthermore, since R_0 is locally optimum, the unfolded route R_0 forms a convex chain in the unfolded polygon. See Fig. 2.

Consider two adjacent adjustments. Observe that no matter which adjustment is performed first, the directions of both adjustments can not be changed. After one adjustment is done, the tension of the other adjustment is increased. Hence, such two adjustments can be performed together. Generally, all possible adjustments for R_0 can be performed together. Let R_1 be the new route after all adjustments for R_0 are done. Since we perform adjustments on each cage edge by edge, R_1 should retain as a convex chain in the unfolded polygon; otherwise the length of R_1 can not be shorter than that of R_0. Thus, R_1 still has the same property as R_0, that is, the unfolded version of R_1 is a convex chain that lies in one side of the unfolded route R. In the adjusting process, the current route R_i gradually moves close to the shortest route R, but can never go over R. Thus, all adjustment directions can not be changed in the adjusting process. It completes the proof. □

Theorem 1 *The time complexity of our deterministic algorithm is $O(kn)$, where k is the maximum size of all cages.*

Proof. Let k be the maximum size of all cages, i.e., $k = max\{|P_i| \; |P_i \in \mathcal{P}\}$. For the current route R_i, we perform all possible adjustments for R_i once together. Since no adjustment directions can be changed in the adjusting process, the unfolding method is applied at most k times. Hence, our algorithm runs in $O(kn)$ time. □

4 The approximate algorithm

The zoo-keeper route algorithms known so far (and our algorithm presented in Section 3) are all based on the unfolding method, which requires the multiplication of $O(n)$ transformation matrices. If the input coordinates have L bits of precision, then the output will have $O(nL)$ bits. Hence, these algorithms do not work well in most of practical examples.

In the following, we make two observations to give an approximate algorithm for the zoo-keeper route problem, which is guaranteed to be within $\pi/2$ times the optimal solution value. Our approximate algorithm does not depond on the unfolding method and takes only linear time. First, if we find out the shortest path from s to the boundary of a cage P_i, then the endpoint of the shortest path on P_i is very near to the point of P_i where the shortest zoo-keeper route visits (see also Fig. 1). In other words, if we connect these points by shortest paths, then we will get a zoo-keeper route that is close to the shortest one. Second, if any two adjacent segments of an unfolded zoo-keeper route are connected with an obtuse angle ($> 90^0$), then the route must be near to the shortest zoo-keeper route.

We now give an approximate algorithm based on the above observations. Let $s = s_0 = s_{m+1}$. We first find the shortest path from s to the boundary of P_1. Let s_1 be the other endpoint of this shortest path on P_1. Then we find the shortest path from s_1 to the boundary of P_2, and let s_2 be the other endpoint on P_2 and so on. In this way, we will get a sequence of shortest paths between points s_i and s_{i+1} ($0 \leq i \leq m$) and finally go back to s_{m+1} ($= s$). Putting all these shortest paths together gives a zoo-keeper route R'.

Consider the inner (or smaller) angle between two adjacent segments of the unfolded route R'. We now show that any inner angle is greater than 90^0 and smaller than 180^0. Since s_{i+1} is the other endpoint of the shortest path from s_i to the boundary of P_{i+1}, the shortest path usually has the right angle to the boundary of P_{i+1} at s_{i+1}. See Fig. 3. When we unfold route R' using the edges containing s_i as mirrors, any angle between two adjacent segments of the unfolded route R' must be greater than 90^0 and smaller than 180^0 (Fig. 3). Furthermore, since route R' has the right angle to the boundary of P_i at s_i, R' is adjustable on all cages P_i in the clockwise direction. Hence, the unfolded route R' is an obtuse zigzag path. See Fig. 4.

Let R be the shortest zoo-keeper route. In most cases, each segment of the unfolded route R' intersects with the unfolded route R. It may also be the case that some segments of the unfolded route R' do not intersect with the unfolded route R (since we can not determine whether s_i is left to the contact point of R with cage P_i or not). We call the unfolded route R', whose segments do not intersect with the unfolded route R at all, a *one-sided* path. To show that the length of R' is no longer than $\pi/2$ times the length of R, we first give the following two lemmas.

Lemma 2 *Let Z be an obtuse zigzag path with the source point s and the target point t on the x axis and let each segment of Z intersect with the x-axis. Then the length of Z is smaller than $\pi/2$ times the distance between s and t.*

Proof. The path Z and the x-axis together form many obtuse triangles with the longest edge on the x-axis. The lemma is simply proved by noticing that the length sum of two shorter edges of an obtuse triangle is smaller than $\pi/2$ times the length of the longest edge. □

Lemma 3 *[4] Let D_1, D_2, $\cdots$, D_k be circles all centered on the x-axis such that $D = \bigcup_{1 \leq i \leq k} D_i$ is connected. Then the boundary of D has length at most $\pi(x_r - x_l)$ where x_l and x_r are the least and greatest x-coordinates of D, respectively.*

Lemma 4 *The length of route R' is smaller than $\pi/2$ times the length of route R.*

Proof. If each segment of the unfolded route R' intersects with the unfolded route R, then we are done by applying Lemma 2. If some segments of the unfolded route R' do not intersect with the unfolded route R, then we can partition the unfolded route R' into pieces so that each piece is a one-sided path with respect to the unfolded route R. Since all inner angles between two segments of a one-sided path are greater than 90^0 and smaller than 180^0, we can find a circle for each segment so that the circle passes through two endpoints of the segment and the center lies on the unfolded route R. These circles are connected and the centers of them must lie in between two endpoints of the one-sided path. Lemma 3 then applies to the one-sided path because the half of the boundary of union of these circles that lies above (or below) the one-sided path has length at least as great as the path itself. It completes the proof. □

Theorem 2 *For any instance of the zoo-keeper route problem, the approximate algorithm gives a solution that is within $\pi/2$ times the optimal solution value. Furthermore, the approximate algorithm runs in $O(n)$ time.*

Proof. The first part of Theorem 2 is already proved in Lemma 4. Let us analyze the time complexity of our approximate algorithm. We first triangulate the interior of $P - \mathcal{P}$ using the linear time algorithm [1], and then repeatedly find the shortest path from s_{i-1} $(i \geq 1)$ to the boundary of P_i until $i = m + 1$. To efficiently find the shortest path from s_{i-1} to the boundary of P_i $(i \geq 1)$, we first figure out a new polygon P_i' that contains point s_{i-1} and polygon P_i. Let LS_i and RS_i denote the left *sleeve* from s_{i-1} to the head edge of P_i and the right *sleeve* from s_{i-1} to the tail edge of P_i, respectively. Sleeve LS_i (RS_i) is the union of the triangles, in which all shortest paths from s_{i-1} to any interior point of the head (tail) edge of P_i are contained. We define polygon P_i' as the union of LS_i and RS_i. (From the definition of LS_i and RS_i, $LS_i \bigcup RS_i$ is connected.) Clearly, LS_i, RS_i and P_i' can be found in the time linear to their sizes. Moreover, the sum of sizes of all polygons P_i' is still linear to n, since each triangle appears at most six times in these polygons P_i'. (Imagine that one walks in LS_i (or RS_i) from s_{i-1} to the head (or tail) edge of P_i. We call the first (last) edge of a triangle T encountered in the way the *entry* (*exit*) edge of T. A pair of entry and exit edges of T can appear at most twice in all polygons P_i'.) Hence, all of polygons P_i' can be obtained in $O(n)$ time.

Within polygon P_i', we can find the shortest path from s_{i-1} to the boundary of cage P_i in the time linear to the size of P_i'. Let us first introduce the concept of *shortest path maps.* In [7], the collection of all shortest paths from a source vertex to points inside P_i' is called the *shortest path map.* The shortest path map partitions the interior of P_i' into triangular regions, each with a distinguished vertex called the *apex.* The regions are so chosen that the shortest path from the source vertex to a point p passes through the apex of the region containing p. Not only the size of the shortest path map but also the time required to compute it is linear to the size of P_i' [7]. Based on the shortest path map of s_{i-1}, we can quickly know the region that contains the target point s_i (by comparing the lengths of the shortest paths from s_{i-1} to all vertices of the shortest path map that lie on the boundary of P_i) and then figure out the exact position of s_i on the boundary of P_i. After point s_i is fixed, the shortest path from s_{i-1} to s_i can simply be obtained. Thus, the time complexity of our approximate algorithm is $O(n)$. □

5 Conclusion

We have presented two new algorithms for the zoo-keeper route problem; one is deterministic and runs in $O(kn)$ time where k is the maximum size of cages in $\mathcal{P}'$, and the other finds an approximate solution that is guaranteed to be within $\pi/2$ times the optimal solution value and takes $O(n)$ time. Both of our results improve upon the previous results.

Acknowledgements

This work was partially supported by Grant in Aid for Scientific Research of the Ministry of Education, Science and Cultures of Japan.

References

[1] B. Chazelle, Triangulating a simple polygon in linear time, *Proceedings, 31th Annu. IEEE Symp. Found. of Comput. Sci.*, 220-229, 1990.

[2] W.P.Chin and S.Ntafos, The zookeeper route problem, *Inform. Sci.* **63**, 245-259, 1992.

[3] J. Czyzowicz *et al.*, The aquarium keeper's problem, presented at the Symposium on Data Structures and Algorithms, 1991.

[4] D.P.Dobkin, S.J.Friedman and K.J. Supowit, Delaunay graphs are almost good as complete graphs, *Discrete Comput. Geom.* **5**, pp. 399-407, 1990.

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

[6] L.Guibas and J. Hershberger, Optimal shortest path queries in a simple polygons, *Journal of Computer and System Sciences* **39**, 231-235, 1989.

[7] J.Hershberger, Finding the visibility graph of a simple polygon in time proportional to its size , *Proc. of the 3rd ACM Symp. on Comput. Geometry*, pp, 11-20, 1987.

[8] J.Hershberger, A new data structure for shortest path queries in a simple polygon, *Inform. Process. Lett.* **38**, 231-235, 1991.

[9] J.Hershberger and J.Snoeyink, An efficient solution to the zookeeper's problem, *Proc. of the 6th Canadian Conf. on Comput. Geom.*, 104-109, 1994.

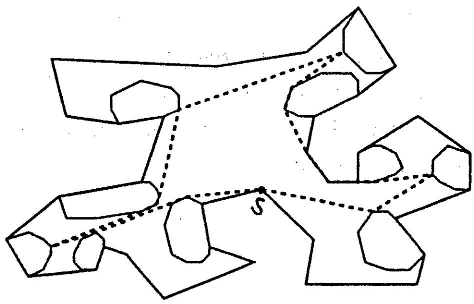

Fig. 1. A shortest zoo-keeper route.

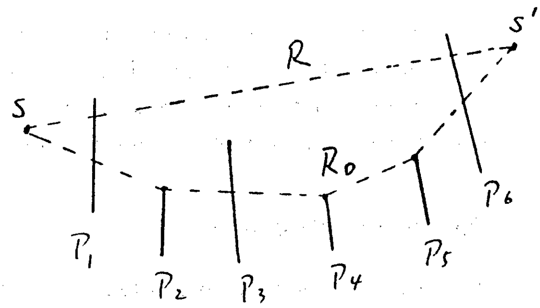

Fig. 2. Illustration for the proof of Lemma 1.

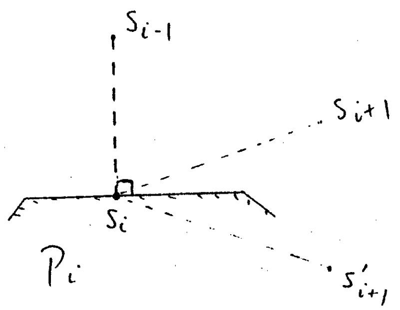

Fig. 3. Any angle between two adjacent segments of the unfolded route R' is obtuse.

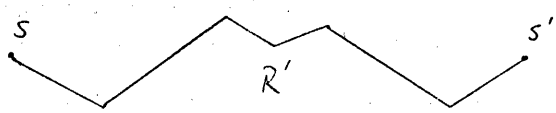

Fig. 4. The unfolded route R' is an zigzag path.