PAPER
Exploring the Outer Boundary of a Simple Polygon

Qi WEI[†a)], *Member*, Xiaolin YAO[††], Luan LIU[†], *and* Yan ZHANG[†], *Nonmembers*

SUMMARY We investigate an online problem of a robot exploring the outer boundary of an unknown simple polygon P. The robot starts from a specified vertex s and walks an exploration tour outside P. It has to see all points of the polygon's outer boundary and to return to the start. We provide lower and upper bounds on the ratio of the distance traveled by the robot in comparison to the length of the shortest path. We consider P in two scenarios: convex polygon and concave polygon. For the first scenario, we prove a lower bound of 5 and propose a 23.78-competitive strategy. For the second scenario, we prove a lower bound of 5.03 and propose a 26.5-competitive strategy.
key words: computational geometry, online algorithm, path planning, exploration, competitive analysis

1. Introduction

A fundamental problem in computational geometry and robotics is that of exploring an unknown environment [1]–[3]. In many cases, the robot is equipped with a vision system that can continuously provides the visibility of its current position. The robot has to inspect an unknown environment based only on what it has seen so far and to return to the starting point. This problem has important applications [4]–[6], such as rescuing human beings in disaster area, exploring damaged nuclear base and so on.

Over the last two decades, many geometric and graph theoretic versions of this problem have been studied. A general model was introduced by Deng et al. [7]. The robot is placed in a polygon possibly with holes. It is equipped with a 360° vision system that can see infinite range as long as no obstacle blocks the view. It's task is to see all points in the polygon and to return to the start. Deng et al. [7] first showed that competitive strategy exists. For general polygon without holes, they proposed a 2016-competitive strategy inspired by a greedy offline approach. Later, Hoffmann et al. [8] improved the competitive ratio to 26.5. There is still a gap between the competitive ratio 26.5 and the lower bound 1.207 given by Deng et al. [7]. Deng et al. [7] also proposed a 2-competitive strategy for rectilinear polygon without holes. Later, Hammar et al. [9] improved the com-

petitive ratio to 3/2. They used L_1-metric to measure the length of the robot's path. Kleinberg [10] showed that the lower bound for this case is 5/4. Fekete et al. [11] studied the exploration of rectilinear polygon without holes under discrete visibility. They showed an $O(logA)$-competitive strategy, where A is the aspect ratio of the polygon. Premkumar et al. [12] used multiple robots to explore a rectilinear polygon without holes. They proposed a $\frac{2(\sqrt{2}p+\log p)}{1+\log p}$-competitive strategy, where p is the number of robots.

For the case that the polygon contains holes, Albers et al. [13] showed that the lower bound is $\Omega(h)$, where h is the number of the holes. Deng et al. [7] proposed an $O(h)$-competitive strategy for orthogonal polygons with h holes. Czyzowicz et al. [14] proposed an $O(P + D\sqrt{k})$-competitive strategy for exploring general polygon with holes, where P is the total perimeter of the region (including perimeters of holes), D is the diameter of the convex hull of the polygon, and k is the number of holes. Georges et al. [15] proposed an $(h+c_0)$!-competitive strategy for general polygon with holes under the assumption that each hole is marked with a special color, where c_0 is a constant, h is the number of holes. They also showed a lower bound 2.618 for this case. Disser et al. [16] considered the problem of mapping a polygon with holes using a compass.

Previous research mainly focuses on the exploration of the interior of the polygon. In this paper, we are interest in a new related model (i.e. the exploration of the outer boundary of the polygon). It can be motivated by the following application. Imagine that a robot wants to explore the shoreline of a lake or to know the shape of an obstacle during the exploration of an environment. In this model, the robot equipped with an unlimited vision system starts from a vertex of a simple polygon. The goal of it is to see all points of the polygon's outer boundary and to return to the start. The robot can not enter the polygon and has no previous knowledge of it. The major difference between previous work and ours is that the robot's path must be outside the polygon. We ask for the strategy that can compute a short exploration tour. The performance of the strategy is measured by the competitive ratio.

It is difficult to give such a competitive strategy. Because we should consider not only the correctness of the strategy but also the competitiveness of the strategy. The robot can not always afford to encircle the polygon [17]. If the polygon is a thin long one, the exploration tour is not related to its perimeter. This will lose the competitiveness.
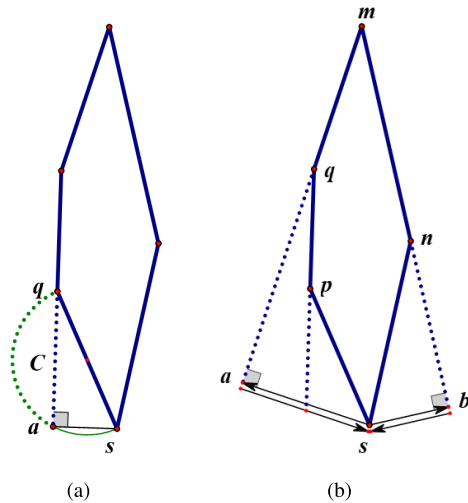
Fig. 1   (a) Visibility and the cut. (b) An optimal exploration tour.

See Fig. 1 (b) for an example. One might wonder if the problem can be solved by converting it into the problem of exploring a polygon with one hole by placing an outer boundary box. In fact we can not add this boundary box because the shape of the polygon is unknown in advance. Not to mention that there is no strategy with constant competitive ratio known for general polygons with holes.

This paper is organized as follows. In Sect. 2, we give the model of the outer boundary exploration problem and some preliminary results. In Sect. 3, we show lower bounds for both convex and concave polygons. In Sect. 4, we propose an exploration strategy for convex polygons together with the analysis. In Sect. 5, we propose and analyze a strategy for concave polygons. In Sect. 6, we make some concluding remarks.

2.   Preliminaries

This section formally defines the problem and gives some preliminary results.

2.1   Definitions

Let P be a simple polygon which has no holes. Let bd denote the outer boundary of P. The robot is modeled as a point. It is equipped with an unlimited 360° vision system. It can walk along a trajectory in arbitrary number of units. The unit is a constant distance specified by the robot's manufacturer. It's task is to walk an exploration tour T starting from s. T must be outside of P. Every point on bd must be visible from some point on T. We want T to be as short as possible. Let $SP(s, t)$ denote the shortest path outside P from s to a point t. If the robot walks $SP(s, q)$ to a vertex q of P, one of its adjacent polygon edges will not be visible until q is reached. The extension of this invisible edge is called a cut C of P with respect to s. See Fig. 1 (a) for an example.

When a vertex is visible to the robot, we say it is discovered. When the cut of a vertex is reached, we say it is

explored. Exploring bd of P is equivalent to exploring all vertices of P, i.e., to visiting all cuts of P with respect to s. Figure 1 (b) shows an optimal exploration tour with full knowledge of P. From this instance, we know that the exploration tour is not necessarily related to the perimeter.

In this paper, competitive ratio is used to measure the performance of the strategy,

$$r = sup_P \frac{|T(P)|}{|T_{opt}(P)|},$$

where $|T(P)|$ is the length of the robot's path, $|T_{opt}(P)|$ is the length of the shortest exploration path.

2.2   Preliminary Results

For exploring a vertex without loss of competitiveness, Hoffmann et al. [8] let the robot approach the vertex along a clockwise oriented circle $circ(s, t)$ spanned by s and t, where s is the local start and t is the target vertex. For analyzing the relation between the robot's path and the shortest path, they also define a geometric structure called angle hull. D is denoted by a polygon in the plane. The set of all points that can see two points of D at a right angle is called the angle hull of D, denoted by $AH(D)$. They showed that the perimeter of $AH(D)$ is at most $\pi/2$ times the perimeter of D if there is no further obstacles. Exploring a polygon P is equivalent to explore all of its reflect vertices. Hoffmann et al. [8] divides these vertices into two types: left reflect vertex and right reflect vertex. They show a strategy that can recursively explore groups of left and right reflect vertices and prove that the competitive ratio is 26.5.

For the problem of searching an unknown target on a line, Baezayates et al. [18] proposed a 9-competitive strategy called *doubling* and prove that it is optimal by showing a matching lower bound. The robot starts from the origin O and walks to the left and right along the line alternately. In the first step, the robot walks 2^0 units to the right. If the target is not reached, it walks back to the origin O. Then, it walks 2^1 units to the left. If the target is not reached, it walks back to the origin O. In the next steps, it walks twice the units of previous step (2^i units) to the right or left alternately, until the target is reached.

3.   Lower Bounds

In this section, we analyze the lower bound of the problem of exploring the outer boundary of a simple polygon. We consider the simple polygon in two scenarios: convex polygon and concave polygon.

Theorem 1. *The competitive ratio of exploring the outer boundary of a convex polygon is no less than 5.*

Proof. Since P may be a thin long one, the exploration tour is not necessarily related to the perimeter of P. See Fig. 1 (b) for an example. To analyze the lower bound, we construct a special instance shown in Fig. 2 (a) depending on the behavior of the strategy. We set that there is only one cut that

Fig. 2 (a) A lower bound instance for convex scenario. (b) A lower bound instance for concave scenario.

should be visited. Then, $T_{opt}(P)$ is only on the left side of the polygon. At the starting vertex s, the robot does not know which side the cut is located. Without loss of competitiveness, any deterministic competitive strategy will explore bd on both left and right sides. In this instance, when the cut $\overline{pa}$ is reached, bd is explored. We assume that the robot walks along $\overline{sa}$ to explore bd on the left side, and walks along $\overline{st}$ to explore bd on the other side, where $\overline{sa}$ is the shortest path from s to $\overline{pa}$. This can be seen as the problem of searching a point on a line, where s is the starting point and a is the target point. Baezayates et al. [18] showed that 9 is the lower bound on the competitive ratio for this problem. Thus, for any deterministic competitive strategy, we have

$$r = \frac{|T(P)|}{|T_{opt}(P)|}$$
$$\geq \frac{9|\overline{sa}| + |\overline{sa}|}{2\overline{sa}}$$
$$\geq 5,$$

the proof is complete. □

Theorem 2. *The competitive ratio of exploring the outer boundary of a concave polygon is at least 5.03.*

Proof. To prove the lower bound, we give a special instance shown in Fig. 2 (b). We show that any strategy will necessarily make a detour compared to the shortest path. In this instance, bd consists of two parts: bd^1, the boundary chain from vertex a to d which is in a rectangular triangle pocket; bd^2, the other parts of bd.

For the rectangular triangle pocket(bd^1), a is at the right angle, and there are two very small pockets at the other two corners. These two small pockets are formed such that the one at point c must be visited and the other one at point b is not. We call the line segment connecting vertex b and c *threshold*. When the *threshold* is reached, the robot can see which small pocket is still to visit. It is easy to see that the

lower bound for exploring bd^1 is $(1 + \sqrt{2})/2 \approx 1.2071$.

Next, we consider the exploration of bd^2. It is equivalent to explore the relative convex hull(RCH) of P. As discussed in the proof of Theorem 1, we set $T_{opt}(RCH(P))$ only on the left side of the polygon. When the robot reaches the cut $\overline{de}$, bd^2 is explored. Let $\overline{se}$ denote the shortest path from s to the cut $\overline{de}$. Since the exploration tour is not necessarily related to the perimeter of P, without loss of competitiveness, any deterministic competitive strategy will explore bd^2 on both left and right sides. This can be seen as the problem of searching a point on a line, where s is the starting point and e is the target point. Baezayates et al. [18] showed that 9 is the lower bound on the competitive ratio for this problem.

Let $T_{opt}(bd^1)$ and $T_{opt}(bd^2)$ denote the shortest path to explore bd^1 and bd^2 respectively. Thus, for any deterministic competitive strategy, we have

$$r = \frac{|T(P)|}{|T_{opt}(P)|}$$
$$\geq \frac{9|\overline{se}| + |\overline{ea}| + 1.2071|T_{opt}(bd^1)| + |\overline{sa}|}{|T_{opt}(P)|}$$
$$\geq \frac{9|\overline{se}| + |\overline{ea}| + 1.2071|T_{opt}(bd^1)| + |\overline{sa}|}{2|\overline{sa}| + |T_{opt}(bd^1)|}.$$

To achieve the worst case, assume that vertex a almost coincides with vertex d. In $\triangle sea$, let θ denote $\angle ase$ and $|\overline{sa}| = 1$, then $|\overline{se}| = \cos\theta$ and $|\overline{ea}| = \sin\theta$. The maximum value of $9|\overline{se}| + |\overline{ea}|$ is $9.06|\overline{sa}|$ by taking $\theta = 0.12$. In the view of the simple inequality $\frac{a+b}{c+d} \leq max(\frac{a}{c}, \frac{b}{d})$, we assume that $|T_{opt}(bd^1)|$ approaches zero in this instance. As above, we have

$$r \geq \frac{10.06|\overline{sa}| + 1.2071|T_{opt}(bd^1)|}{2|\overline{sa}| + |T_{opt}(bd^1)|}$$
$$\geq \frac{10.06|\overline{sa}|}{2|\overline{sa}|}$$
$$\geq 5.03,$$

thus, the proof is complete. □

4. Exploring the Outer Boundary of Convex Polygons

In this section, we first consider how to explore a single vertex on bd. Then, we consider the exploration of all vertices and give a 23.78-competitive strategy.

4.1 The Strategy

Exploring a single vertex on bd is a essential subtask of our strategy. It needs a little care. The robot can not walk straight to the vertex to reach the cut because the cut may passes by the robot's initial position very closely. This will lose competitiveness, see cut C in Fig. 1 (a) for an example. The optimal path is $\overline{sa}$. Instead, we use a clockwise half-circle to cope with this situation. Assume that the robot wants to explore the vertex q visible from the starting point

Algorithm 1 Explore the outer boundary of a convex polygon

```
1: procedure ExploreBDCX(P, s)
2:     maintain t_l and t_r;
3:     i ← 0;
4:     repeat
5:         j ← i%2;
6:         if j = 0 then
7:             ExploreVertex(P, s, t_l, cw, i);
8:         else
9:             ExploreVertex(P, s, t_r, ccw, i);
10:        end if
11:        i = i + 1;
12:    until t_l = t_r
13: end procedure
```

Algorithm 2 Explore a vertex on *bd*

```
1: procedure ExploreVertex(P, s, target, orientation, i)
2:     maintain the target;
3:     SP_1 ← 0;
4:     if i is not 0 or 1 then
5:         walk along the shortest path SP(s, return_last) from s to the
           return point of last step in this orientation.
6:         SP_1 ← SP(s, return_last);
7:     end if
8:     back ← the last vertex before target on SP(s, CP);
9:     walk 2^i − SP_1 units along circ(back, target);
10:    whenever back becomes invisible update back;
11:    return to s along the shortest path;
12: end procedure
```


Fig. 3 An instance of the strategy for convex polygons.

s, it walks a half-circle spanned by *q* and *s*, denoted by $circ(s, q)$. When the robot reaches the cut *C* of *q* at *a*, *q* is explored. With Thales' theorem, we know that the intersection point *a* is the point on *C* closest to *s*. The ratio between the length of the circular arc from *s* to *a* and $|sa|$ can be bounded by $\pi/2$.

Then we consider how to explore all vertices on *bd*. If *P* is a rectilinear polygon, the cut of each visible vertex is known. The robot can walk to the cut of next vertex in clockwise order. But for general polygons, this approach will lose competitiveness because *P* can be a thin long one, see Fig. 1 (b) for an example. The exploration path is not necessarily related to the perimeter of *P*. We use doubling strategy to cope with this situation. The robot walks along half-circle in cw/ccw orientation alternately, such that at each step *i*, it walks 2^i units in one orientation, comes back to the starting point, then walks 2^{i+1} units in the other orientation. It is ending when a vertex is discovered by the robot in both orientations.

Based on above analysis, we give the strategy for exploring the outer boundary of a convex polygon. We list the complete pseudo-code in the forms of Algorithm 1 and 2.

ExploreBDCX is the main program of our strategy. Its task is to direct the robot to explore the outer boundary of a given convex polygon. We use t_l and t_r to denote the robot's current target vertex on the left and right side respectively. During the exploration, the update of t_l and t_r will be maintained. When ExploreBDCX is called, t_l (t_r) is the first ver-

tex that is visible from *s* and has an invisible edge on the left(right) side. We call ExploreVertex to explore t_l and t_r in *cw* and *ccw* orientation alternately until t_l is equal to t_r, i.e. all vertices have been explored.

ExploreVertex is the sub program of our strategy. Its task is to direct the robot to explore a target vertex on *bd*. We use *CP* to denote the robot's current position. We use *back* to denote the last vertex before *CP* on $SP(s, CP)$. When ExploreVertex is called for the first time, the robot walks 2^0 units along $circ(back, t_l)$ in *cw* orientation. Then the robot returns to *s* along the shortest path. When ExploreVertex is called for the second time, the robot walks 2^1 units along $circ(back, t_r)$ in *ccw* orientation. Then the robot returns to *s* along the shortest path. When ExploreVertex is called for the third and later times, the robot first walks along the shortest path from *s* to the return point of last step in this orientation, and then walks the remaining units along $circ(back, target)$. On the exploration way, it may happen that the target(t_l or t_r) is explored and a new vertex after it is discovered. Then the target is updated to this new vertex. It may also happen that the robot loses sight of *back*, see point *d* in Fig. 3 for an example. In this case, we update *back*.

Figure 3 shows an instance of our strategy. Initially, t_l is *p* and t_r is *n*. The robot starts with a half-circle spanned by *s* and *p*. At point *a*, *p* is explored and t_l is updated to *q*. The robot goes on to explore *q*. Note that the half-circle spanned by *s* and *q* is passing through *a*. At point *b*, the robot has walked one unit in *cw* orientation. Then, it walks back to *s* along the shortest path. In the next step, the robot walks along a half-circle spanned by *s* and *n* in *ccw* orientation. At point *c*, *n* is explored and t_r is updated to *r*. The robot goes on to explore *r*. Note that the half-circle spanned by *s* and *r* is passing through *c*. At point *d*, *s* becomes invisible. *back* is updated to *n*. The robot walks along the half-circle spanned by *n* and *r*. At point *e*, *r* is explored and t_r is updated to *t*. At point *f*, the robot has walked two units in *ccw* orientation. Then, it walks back to *s* along the shortest path. In the third step, the robot walks along the shortest path from *s* to *b*.

Then, it walks along the half-circle spanned by s and q. At point g, q is explored and t_l is updated to t. At this time, t_l is equal to t_r, bd is explored. Then, the robot returns to s along the shortest path.

4.2 The Analysis

In this section, we analyze the performance of the strategy by competitive ratio.

Theorem 3. *The strategy for exploring the outer boundary of a convex polygon is 23.78-competitive.*

Proof. For convenience, we give an instance shown in Fig. 4. Without loss of generality, let e be the last explored vertex on bd. With our strategy, the robot discovers e at point e_r, and explores e at point e_l. We use SP_l (SP_r) to denote the shortest path from s to the extension of $\overline{ed}$ $(\overline{ef})$. Based on $T_{opt}(P)$, there are two cases to consider.

 Case 1. $T_{opt}(P)$ is less than the perimeter of P. We use SP_{lm} (SP_{rm}) to denote the left (right) part of $T_{opt}(P)$ and then $T_{opt}(P) = 2(SP_{lm} + SP_{rm})$. In the instance shown in Fig. 4, $SP_{lm}(SP_{rm})$ is the shortest path from s to the extension of $\overline{dc}(\overline{de})$, i.e., $\overline{sd_l}$ $(\overline{sg} + \overline{gd_r})$. Note that the last explored vertex of $T(P)$ may not be the same as that of $T_{opt}(P)$, see Fig. 4 for an example, the former is e and the latter is d. Then, $T_{opt}(P)$ may not be $2(SP_l + SP_r)$. Obviously, it can achieve a worse instance if $T_{opt}(P) < 2(SP_l + SP_r)$. In this case, either $|SP_{lm}| > |SP_l|$ or $|SP_{rm}| > |SP_r|$. Because exploring bd is equivalent to discovering a vertex on both left and right sides, and the length of the shortest path from s to discover a vertex is longer than that to discover its previous vertex on the same side, and the last explored vertex of $T_{opt}(P)$ is the descendant of the last explored vertex of $T(P)$ on either left or right side. See Fig. 4, the length of the shortest path to discover d is longer than that to discover e on the right side, i.e., $|SP_{rm}| > |SP_r|$. With the property of our strategy, we have $\frac{1}{2}|AH(SP_r)| < |AH(SP_l)| \le 2|AH(SP_r)|$, $|SP_l| < |AH(SP_l)|$, $|SP_r| < |AH(SP_r)|$, $|AH(SP_l)| \le \frac{\pi}{2} \times |SP_l|$ and $|AH(SP_r)| \le \frac{\pi}{2} \times |SP_r|$. Then, $|SP_l| < \pi \times |SP_r|$ and $|SP_r| < \pi \times |SP_l|$ both holds. If $|SP_{lm}| > |SP_l|$, $\frac{|SP_l|+|SP_r|}{|SP_{lm}|+|SP_{rm}|} <$

$\frac{(1+\pi)|SP_l|}{|SP_{lm}|} < 1 + \pi$. If $|SP_{rm}| > |SP_r|$, $\frac{|SP_l|+|SP_r|}{|SP_{lm}|+|SP_{rm}|} < \frac{(1+\pi)|SP_r|}{|SP_{rm}|} < 1 + \pi$. Thus, we have

$$\frac{|SP_l| + |SP_r|}{|SP_{lm}| + |SP_{rm}|} < 1 + \pi.$$

With the competitive ratio 9 of doubling strategy, the robot walks at most 9 times longer than the length of $AH(SP_l)$ when it reaches point e_l. Then, $|T(P)|$ is at most $9|AH(SP_l)| + |SP_l|$. We add the length of SP_l because the robot should return to the start point s. Also, the length of $AH(SP_l)$ is at most $\pi/2$ times longer than $|SP_l|$ and $|SP_l| < \pi \times |SP_r|$. Thus, we have

$$
\begin{aligned}
r &= \frac{|T(P)|}{|T_{opt}(P)|} \\
&\le \frac{9|AH(SP_l)| + |SP_l|}{2(|SP_{lm}| + |SP_{rm}|)} \\
&\le \frac{9|AH(SP_l)| + |SP_l|}{2(|SP_l| + |SP_r|)} \times (1 + \pi) \\
&\le \frac{9 \times \pi/2 \times |SP_l| + |SP_l|}{2(1 + \pi)|SP_l|} \times (1 + \pi) \times \pi \\
&\le 23.78.
\end{aligned}
$$

 Case 2. $T_{opt}(P)$ is the perimeter of P. We use R to denote the perimeter of P. As discussed in Case 1, we have

$$
\begin{aligned}
r &= \frac{|T(P)|}{|T_{opt}(P)|} \\
&\le \frac{9|AH(SP_l)| + |SP_l|}{|R|} \\
&\le \frac{9|AH(SP_l)| + |SP_l|}{|SP_l| + |SP_r|} \\
&\le \frac{9 \times \pi/2 \times |SP_l| + |SP_l|}{(1 + \pi)|SP_l|} \times \pi \\
&\le 11.48.
\end{aligned}
$$

Thus, the proof is complete. □

5. Exploring the Outer Boundary of Concave Polygons

In this section, we first propose a competitive strategy for concave polygons. And then analyze the performance of it.

5.1 The Strategy

We use two phases to explore bd of a concave polygon. **Phase 1.** Explore the boundary of the relative convex hull (RCH) of P which can be seen as the shape of a rubber band spanned around P, see Fig. 5 for an example. Since $RCH(P)$ is a convex polygon, we can use the strategy proposed in Sect. 4 to complete this task. **Phase 2.** Explore the pockets on bd. After phase 1, all pockets on bd have been found. We can compute the shortest path denoted by SP_c to connect each pocket. Since each pocket is a simple polygon, we use the 26.5-competitive strategy to explore each of them and use SP_c to connect them.

Fig. 4 An instance for the proof of theorem 3.

Algorithm 3 Explore the outer boundary of a concave polygon

1: **procedure** EXPLOREBDCV(P, s)
2:     ExploreBDCX($RCH(P)$, s) while maintaining the *PocketList*;
3:     compute SP_c;
4:     walk along SP_c to visit each pocket and use the
       26.5-competitive strategy to explore each pocket.
5:     return to s along SP_c;
6: **end procedure**
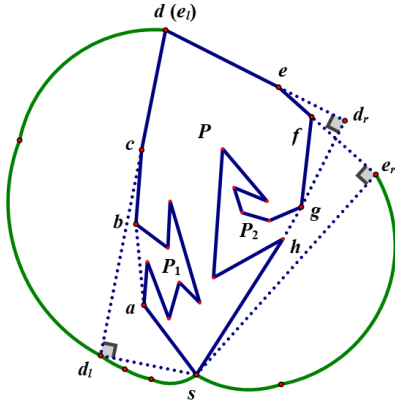
Fig. 5   An instance of the strategy for concave polygon.

Based on these two phases, we give the strategy for exploring the outer boundary of a concave polygon. The pseudo-code is listed in the form of Algorithm 3.

Figure 5 shows an instance of this strategy. Firstly, the robot explores the boundary of $RCH(P)$ with algorithm 1 and 2. e is the last explored vertex. The robot discovers e at point e_r and explores it at point e_l. During the exploration, pockets P_1 and P_2 found on bd are inserted into *PocketList*, as targets for next phase. After phase 1, the robot knows the shortest path SP_c. Then, the robot first walks along $\overline{sa}$ to vertex a and explores P_1 with the 26.5-competitive strategy. Then, it returns to s along $\overline{sa}$. Next, the robot walks along $\overline{sh}$ to vertex h and explores P_2 with the 26.5-competitive strategy. Then, it returns to s along $\overline{sh}$.

5.2   The Analysis

In this section, we analyze the performance of the strategy by competitive ratio.

Theorem 4. *The competitive ratio of the strategy for exploring the outer boundary of a concave polygon is 26.5.*

Proof. For convenience, we give an instance shown in Fig. 5. As above, we use two phases to explore P. Let $T^1(P)$ and $T_{opt}^1(P)$ denote the robot's path and shortest path in phase 1. Let $T^2(P)$ and $T_{opt}^2(P)$ denote the robot's path and shortest path in phase 2. Without loss of generality, let e be the last explored vertex on the boundary of $RCH(P)$ in phase 1. With our strategy, the robot discovers e at point e_r, and explores e at point e_l. We use SP_l (SP_r) to denote the shortest path from s to the extension of $\overline{ed}$ ($\overline{ef}$). Based on

$T_{opt}^1(P)$, there are two cases to consider.

Case 1. $T_{opt}^1(P)$ is less than the perimeter of $RCH(P)$. As discussed in theorem 3, we know that $T^1(P)$ is at most 23.78 times $T_{opt}^1(P)$. Let $T(P_i)$ and $T_{opt}(P_i)$ ($i = 1, 2, 3 \ldots n$) denote the robot's path and the shortest path to explore the pocket P_i on bd. Then, $|T_{opt}^2(P)| = |T_{opt}(P_1)| + |T_{opt}(P_2)| + \ldots + |T_{opt}(P_n)| + |SP_c|$. Since $T_{opt}(P)$ should explores both the boundary of $RCH(P)$ and the pockets on bd, we have $|T_{opt}(P)| \geq |T_{opt}^1(P)| + |T_{opt}(P_1)| + |T_{opt}(P_2)| + \ldots + |T_{opt}(P_n)|$ and $|T_{opt}(P)| \geq |SP_c| + |T_{opt}(P_1)| + |T_{opt}(P_2)| + \ldots + |T_{opt}(P_n)|$. Together with the 26.5-competitive strategy, we have

$$
\begin{aligned}
r &= \frac{|T(P)|}{|T_{opt}(P)|} \\
&= \frac{|T^1(P)| + |T^2(P)|}{|T_{opt}(P)|} \\
&\leq \frac{23.78|T_{opt}^1(P)| + 26.5(|T_{opt}(P_1)| + T_{opt}(P_2)) + |SP_c|}{|T_{opt}(P)|}
\end{aligned}
$$

Based on the relation between $|T_{opt}^1(P)|$ and $|SP_c|$, we consider two subcases.

Case 1.1. $|T_{opt}^1(P)| \leq |SP_c|$. With the simple inequality $\frac{a+b}{c+d} \leq max(\frac{a}{c}, \frac{b}{d})$, we have

$$
\begin{aligned}
r &\leq \frac{23.78|T_{opt}^1(P)| + 26.5(|T_{opt}(P_1)| + T_{opt}(P_2)) + |SP_c|}{|SP_c| + |T_{opt}(P_1)| + |T_{opt}(P_2)|} \\
&\leq \frac{24.78|SP_c| + 26.5(|T_{opt}(P_1)| + T_{opt}(P_2))}{|SP_c| + (|T_{opt}(P_1)| + |T_{opt}(P_2)|)} \\
&\leq 26.5.
\end{aligned}
$$

Case 1.2. $|T_{opt}^1(P)| > |SP_c|$. With the simple inequality $\frac{a+b}{c+d} \leq max(\frac{a}{c}, \frac{b}{d})$, we have

$$
\begin{aligned}
r &\leq \frac{23.78|T_{opt}^1(P)| + 26.5(|T_{opt}(P_1)| + T_{opt}(P_2)) + |SP_c|}{|T_{opt}^1(P)| + |T_{opt}(P_1)| + |T_{opt}(P_2)|} \\
&\leq \frac{24.78|T_{opt}^1(P)| + 26.5(|T_{opt}(P_1)| + T_{opt}(P_2))}{|T_{opt}^1(P)| + (|T_{opt}(P_1)| + |T_{opt}(P_2)|)} \\
&\leq 26.5.
\end{aligned}
$$

Case 2. $T_{opt}^1(P)$ is the perimeter of $RCH(P)$. As discussed in theorem 3, we know that $T^1(P)$ is at most 11.48 times $T_{opt}^1(P)$. Let R denote the perimeter of $RCH(P)$. Then, $|T_{opt}^1(P)| = |R|$, $|T_{opt}(P)| = |T_{opt}(P_1)| + |T_{opt}(P_2)| + \ldots + |T_{opt}(P_n)| + |R|$. Together with $|SP_c| \leq |R|$ and the 26.5-competitive strategy, we have

$$
\begin{aligned}
r &= \frac{|T(P)|}{|T_{opt}(P)|} \\
&= \frac{|T^1(P)| + |T^2(P)|}{|T_{opt}(P)|} \\
&\leq \frac{11.48|T_{opt}^1(P)| + 26.5(|T_{opt}(P_1)| + T_{opt}(P_2)) + |SP_c|}{|R| + (|T_{opt}(P_1)| + |T_{opt}(P_2)|)} \\
&\leq \frac{12.48|R| + 26.5(|T_{opt}(P_1)| + T_{opt}(P_2))|}{|R| + (|T_{opt}(P_1)| + |T_{opt}(P_2)|)}
\end{aligned}
$$

$$\leq 26.5$$

Thus, the proof is complete. $\qquad\square$

6. Conclusion

In this paper, we considered the outer boundary exploration of simple polygons. For convex polygons, we showed a lower bound of 5 and presented a 23.78-competitive strategy. For concave polygons, we proved a lower bound of 5.03 and proposed a 26.5-competitive strategy.

There is still a gap between the upper bound and the lower bound. It would be interesting to propose a new strategy to reduce the gap.

Acknowledgements

References

[1] S.K. Ghosh and R. Klein, "Online algorithms for searching and exploration in the plane," Computer Science Review, vol.4, no.4, pp.189–201, 2010.

[2] J. Chalopin, S. Das, Y. Disser, M. Mihalák, and P. Widmayer, "Mapping simple polygons: The power of telling convex from reflex," ACM Trans. Algorithms, vol.11, no.4, pp.1–16, 2015.

[3] K.-T. Foerster and R. Wattenhofer, "Lower and upper competitive bounds for online directed graph exploration," Theoretical Computer Science, vol.655, pp.15–29, 2016.

[4] J.P.L.S.D. Almeida, R.T. Nakashima, F. Neves-Jr, and L.V.R.D. Arruda, "Bio-inspired on-line path planner for cooperative exploration of unknown environment by a multi-robot system," Robotics and Autonomous Systems, vol.112, pp.32–48, 2019.

[5] D.P. Strom, I. Bogoslavskyi, and C. Stachniss, "Robust exploration and homing for autonomous robots," Robotics and Autonomous Systems, vol.90, pp.125–135, 2017.

[6] K. Doi, Y. Yamauchi, S. Kijima, and M. Yamashita, "Exploration of finite 2d square grid by a metamorphic robotic system," Proceedings of the International Symposium on Stabilization, Safety, and Security of Distributed Systems, vol.11201, pp.96–110, Springer International Publishing, Cham, 2018.

[7] X. Deng, T. Kameda, and C. Papadimitriou, "How to learn an unknown environment," Proceedings of the 32nd Annual Symposium of Foundations of Computer Science, pp.298–303, Springer Berlin Heidelberg, 1991.

[8] F. Hoffmann, C. Icking, R. Klein, and K. Kriegel, "The polygon exploration problem," SIAM J. Comput., vol.31, no.2, pp.577–600, 2002.

[9] M. Hammar, B.J. Nilsson, and M. Persson, "Competitive exploration of rectilinear polygons," Theoretical Computer Science, vol.354, no.3, pp.367–378, 2006.

[10] J.M. Kleinberg, "On-line search in a simple polygon," Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, pp.8–15, 1994.

[11] S.P. Fekete and C. Schmidt, "Polygon exploration with time-discrete vision," Computational Geometry, vol.43, no.2, pp.148–168, 2010.

[12] A.P. Premkumar, K. Yu, and P. Tokekar, "A geometric approach for multi-robot exploration in orthogonal polygons," Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics, vol.13, pp.896–911, Springer International Publishing, Cham, 2020.

[13] S.S.S. Albers and K. Kursawe, "Exploring unknown environments with obstacles," Algorithmica, vol.32, no.1, pp.123–143, 2002.

[14] J. Czyzowicz, D. Ilcinkas, A. Labourel, and A. Pelc, "Worst-case optimal exploration of terrains with obstacles," Information and Computation, vol.225, pp.16–28, 2013.

[15] R. Georges, F. Hoffmann, and K. Kriegel, "Online exploration of polygons with holes," Proceedings of the International Workshop on Approximation and Online Algorithms, vol.7846, pp.56–69, Springer Berlin Heidelberg, 2013.

[16] Y. Disser, S.K. Ghosh, M. Mihalák, and P. Widmayer, "Mapping a polygon with holes using a compass," Theoretical Computer Science, vol.553, pp.106–113, 2014.

[17] S.K. Ghosh, J.W. Burdick, A. Bhattacharya, and S. Sarkar, "Online algorithms with discrete visibility - exploring unknown polygonal environments," IEEE Robotics Automation Magazine, vol.15, no.2, pp.67–76, 2008.

[18] R.A. Baezayates, J.C. Culberson, and G.J.E. Rawlins, "Searching in the plane," Information and Computation, vol.106, no.2, pp.234–252, 1993.
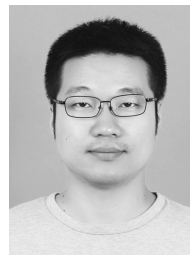
Qi Wei received his B.S. degree in mathematics from Liaoning University, China in 2005. He received his M.S. degree in software engineering from Dalian University of Technology, China in 2009. He received his Ph.D. degree in computer applied technology from Dalian Maritime University, China in 2016. He is currently working as an associate professor at Liaoning Normal University, China. His research interests include computational geometry and path planning of mobile robots.

Xiaolin Yao received her B.S. degree in mathematics from Liaoning University, China in 2005. She received her Ph.D. degree in information management from Dongbei University of Finance and Economics, China in 2016. She is currently working as an associate professor at Dalian Neusoft Institute of Information, China. Her research interests include Machine learning, design and analysis of algorithms.

Luan Liu received her B.S. degree in computer science and technology from Liaoning Normal University, China in 2020. She is currently an M.S. candidate at Liaoning Normal University, China. Her research interests include computational geometry and path planning of mobile robots.

Yan Zhang received her B.S. degree in computer science and technology from Liaoning Normal University, China in 2020. She is currently an M.S. candidate at Liaoning Normal University, China. Her research interests include computational geometry and path planning of mobile robots.