Simple Robot Free-Target Search in Rectilinear Streets

Mohammad Aletaha* Arash Vaezi* Mohammad Abouei Mehrizi† Mohammad Ghodsi‡

Abstract

We consider the problem of searching for an arbitrary (random) target in an unknown rectilinear-street using a simple robot. A simple polygon with respect to two given vertices u, v on the boundary is a street if the clockwise and counter-clockwise chains from u to v are weakly visible from each other. A simple robot, also called gap-detector, can only detect the discontinuities in the depth information (gaps) in a cyclical order. The goal is to design a strategy for a simple robot to find an arbitrary target t in a rectilinear-street, starting from u or v, with a minimum path length. We propose a strategy that guarantees a path which is at most $\sqrt{10}$ times longer than the shortest path.

1 Introduction

Path planning for robots in unknown environments is one of the fundamental problems in the fields of computational geometry, online algorithms, and robotics [4]. In path planning problems, a robot must find a target in a specified environment. Note that if we have the geometric map of the environment and the position of the target point, we can find the shortest-path easily. However, we consider the case that the robot does not have the information in advance. Such a robot should follow an online algorithm to find the target. A simple robot has access only to its local information about its surroundings. We denote the start and target point of the robot by s and t, respectively. The *competitive ratio* is the length of the path traveled by the robot from s to t, over the length of the shortest-path. A strategy is called *c-competitive* if its competitive ratio is at most c.

A *street* is a simple polygon with two distinct vertices u and v so that clockwise and counter-clockwise chains from u to v (L_{chain} and R_{chain} resp.) are mutually weakly visible [6]. In other words, every point on each chain is visible to at least one point on another one. A rectilinear-street is an orthogonal street (see Figure 1).

Using simple robots has quite a few advantages over 360° vision robots such as low cost, less sensitivity to failure, easy to replace and maintenance. Therefore, numerous types of simple robots have been characterized and deployed in path planning problems. [8, 12]

We use a simple point robot called *gap-detector*. To understand what a gap-detector robot can do we need to determine a sensor called a *gap-sensor*. A gap-sensor is a minimal sensing model that was introduced by Tovar et al. [12]. It can only detect the discontinuities in the depth information (gaps) in the robot's visibility region and reports them in a cyclical order. It can assign a label L (left-gap) or R (right-gap) to each gap depend on the portion of the environment hidden behind that gap (see Figure 1(a)). Tovar et al. proposed a data structure, called Gap Navigation Tree (GNT), to maintain and update the gaps that have been seen during the robot's movement. Also, the robot can detect the target whenever the target enters its visibility polygon. Later, this robot gets empowered by a 4-wind compass sensor [11]. This latter sensor can illustrate the four main directions (cardinal points: North, East, South, and West), and it can report which gap is between which two main directions or if it is collinear with a main direction.

A simple robot with those characteristics can only move toward the gaps, the compass directions, and the target (when it becomes visible). The robot can move around the polygon through an arbitrary number of steps. A step is a fixed distance specified by the robot's manufacturer. We assume the step is small enough compared to the scale of the given polygon. For the sake of simplicity, we assume that the given rectilinear polygon is based on a grid with unit distance d_g, and the robot's step is equal to d_g/k for any $k \in \mathbb{N}$.

As the robot moves, the combinatorial structure of its visibility region changes by the occurrence of four *critical events*. These critical events are: appearance, disappearance, split, and merge of gaps [12]. An appearance/disappearance event occur when the robot crosses an inflection-ray of a gap. Also, a split/merge event occur when the robot crosses the bitangent-complement of two polygon's reflex vertices (see Figure 2(a)). When a gap appears and the portion behind it was so far visible, we call it a *primitive-gap*. All other gaps are called *non-primitive-gaps*. The robot stores all of these information in GNT.

Previous Works. In 1992, Klein introduced street polygons [6]. He considered the problem of searching in a street, starting from u or v, for the other one. He

*Department of Computer Engineering, Sharif University of Technology, Tehran, Iran, `mohammadaletaha@ce.sharif.edu`, `avaezi@ce.sharif.edu`

†Gran Sasso Science Institute (GSSI), L'Aquila, Italy, `mohammad.aboueimehrizi@gssi.it`.

‡Sharif University of Technology and Institute for Research in Fundamental Sciences (IPM), Tehran, Iran, `ghodsi@sharif.edu`

presented a 5.73-competitive strategy and proved that the lower bound on the competitive ratio is $\sqrt{2}$. After several improvements, finally, Schuierer and Semrau [7] and Icking et al. [5] independently presented optimal strategies. The robot used in all previously mentioned works equipped with a 360° vision system. Such a robot can detect edges and vertices, measure the distances and angles, and move freely in any direction.

Bröcker and López-Ortiz considered two new types of search in streets called Position-Independent search [2]. In the first type, the robot starts from u (or v) and searches for an arbitrary target t on the boundary. In the second type, both of the start and target points are arbitrary points on the boundary. They presented 36.8 and 69.2-competitive strategies and proved the lower bounds of 9 and 11.78 for these two types, respectively. Bröcker and Schuierer showed that for the rectilinear-streets, one can achieve better competitive ratios [3]. In the first type, they presented an optimal 2.61-competitive strategy and proved a matching lower bound. In the second type, a 59.91-competitive strategy in L_1-metric is proposed.

For the first time, Tabatabaei and Ghodsi [10] considered the gap-detector robots for the Klein's problem [6]. They equipped the robot with a tool called pebble. A pebble is a detectable object that the robot can carry and put it everywhere in the polygon. They have presented an 11-competitive strategy using one pebble. They proved the competitive ratio can be improved to 9 using enough pebbles. Moreover, they showed considering rectilinear-streets, the gap-detector robot which is empowered by a compass achieves the optimal competitive ratio of $\sqrt{2}$. Wei et al. [13] and Tabatabaei et al. [9] independently presented 9-competitive strategies without using any pebbles. Furthermore, in [9] a 7-competitive randomized strategy has been proposed. Additionally, a lower bound of 9 (4.59) on the competitive ratio of all deterministic (randomized) strategies has been proved [9, 13]. Recently, Tabatabaei et al. showed that empowering the robot with a compass improves the competitive ratio to $3\sqrt{2}$ [11]. It also has been shown that if two simple robots cooperate with each other, the competitive ratio will decrease to 2 [1].

Our Contribution. We consider the first type of the Position-Independent search (mentioned above) introduced in [2] and call it *Free-Target search*. Inspiring from [3] in which the authors considered a 360° vision robot, we present a $\sqrt{10}$-competitive strategy using a gap-detector robot. Please note that despite 360° vision robots, gap-detector robot's vision and movement are strictly limited.

Problem Definition. Given a rectilinear-street polygon $\mathcal{P}$ (with respect to two vertices u, v), a simple robot $\mathcal{R}$ standing on a start vertex $s \in \{u, v\}$. Is there an online strategy with the minimum path for $\mathcal{R}$ to move from s and find a given target point t in $\mathcal{P}$? In the rest of the paper, we call it Free-Target Search (FTS) problem.

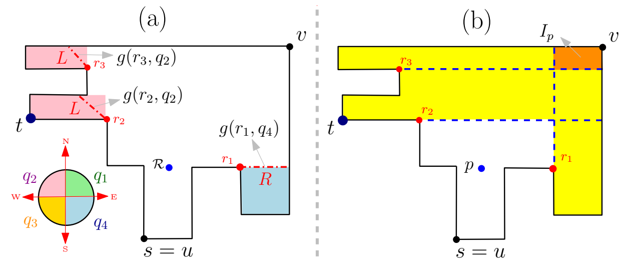

Figure 1: (a) A rectilinear-street, the robot $\mathcal{R}$, and the gaps (dashed lines). (b) Unexplored-regions of gaps at point p and their intersection area I_p.

2 Strategy

This section presents an online strategy for the FTS problem. Without loss of generality, assume that $s = u$. A robot $\mathcal{R}$ needs to explore $\mathcal{P}$ to find t. The target t may lie behind any gap. If the area behind a gap g is already visited by $\mathcal{R}$, the gap is primitive, the robot no longer needs to explore g. Hence, only non-primitive gaps are required to get visited. Hereinafter, when we use *gap*, we mean a non-primitive gap. As we mentioned earlier, the gap-detector robot has a compass sensor with four main directions. Each gap either falls between two main directions or is collinear with a main direction. Main directions partition the environment to four quadrants: NE, SE, SW, and NW. We assign each gap which falls between two directions to the related quadrant. Consider a gap g which is collinear with a main direction d, one side of d is visible to $\mathcal{R}$ and the other side is invisible. For a gap g which is collinear with d, we assign it to the quadrant contains the hidden part of g.

The strategy has three cases. Each case has an initiation-point and an end-point. A procedure called *case-analysis* determines which case the robot $\mathcal{R}$ should choose for its next step. From the beginning point (s), $\mathcal{R}$ should run case-analysis. At the end-point of each case, the robot $\mathcal{R}$ stops moving and runs case-analysis. Note that an end-point of a case is an initiation-point of the next case.

Whenever t is visible by $\mathcal{R}$, it stops and moves directly toward t regardless of its previous direction. Based on the position of $\mathcal{R}$ in $\mathcal{P}$, and the circumstances of the gaps around $\mathcal{R}$, there are three cases mentioned in the following. For a better presentation, based on the position of $\mathcal{R}$, each gap g is denoted by $g(r, q)$ which r is a reflex vertex that causes $\mathcal{R}$ not to see a part of $\mathcal{P}$, and q is a quadrant that g is assigned to it (see Figure 1(a)).

A case ends (or a new case initiates) when a gap ap-

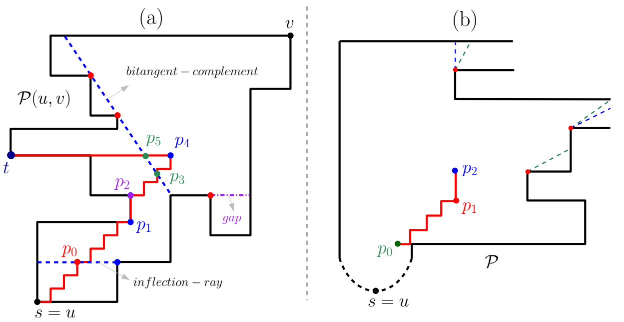

Figure 2: (a) Examples of critical events: at points p_0, p_2, p_3, p_5, the events disappearance, appearance, split, and merge occur, respectively. (b) Case 1.

pears or disappears. The robot $\mathcal{R}$ stops when a case is finished and run the case-analysis procedure. That is because the proposed strategy works based on the positions of gaps around $\mathcal{R}$. In fact, the number of quadrants the gaps are assigned to determines the strategy cases. So, an appearance or disappearance event might change the current case of $\mathcal{R}$.

Furthermore, note that the below-mentioned strategy guarantees that a merge or a split event can never change the current case of $\mathcal{R}$. That is because when either a merge or split event occurs for more than one gap, all such gaps must remain in the same quadrant. So, the current case will not change. Consequently, the robot considers those events only for updating GNT, but ignores them for case-analysis; in fact, only appearance and disappearance events will be considered for case-analysis.

Case 1: *If gap(s) is (are) assigned to one quadrant q (see Figure 2(b)).*
The robot moves alternatively toward two directions adjacent to q, i.e., one step toward a direction and one step toward another one. The robot $\mathcal{R}$ stops if a gap becomes collinear with any of its two main directions, say d, then $\mathcal{R}$ turns and moves directly toward d (see point p_1 in Figure 2(b)).

Case 2: *If gaps are assigned to two adjacent quadrants q and q' (see Figure 3(a)).*
The robot $\mathcal{R}$ moves along the main direction between q and q', e.g., if the quadrants be NE and NW, the robot moves toward N.

Case 3: *If gaps $g_i(r_i, q)$, $g_j(r_j, q')$, and $g_k(r_k, q'')$, $1 \leq i, k$ exist and at least some of them are located in two non-adjacent quadrants (q, q'') (see Figure 3(b)).*
In this case, the gaps are assigned to at most three quadrants. There are at least two opposite side quadrants q, q''. We call each gap which is assigned to q or q'' a *crucial-gap*.

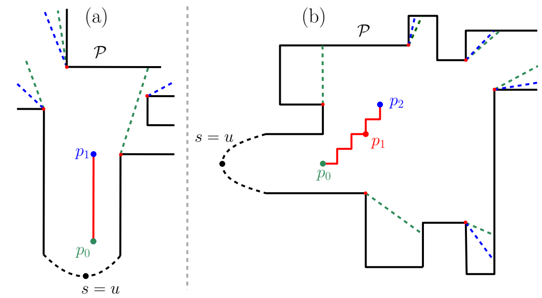

Figure 3: (a) Case 2 (b) Case 3.

If $j \geq 1$ (at least one gap is assigned to q'), then $\mathcal{R}$ moves alternatively between two main directions of q' (see Figure 3(b)).

Else there is no middle quadrant, and the opposite quadrants q and q'' must be either q_1, q_3 or q_2, q_4 (see Figure 1(a)). Using the compass, $\mathcal{R}$ can distinguish between these two. In such a case we need to find a quadrant to be in the placed of q'. This is a quadrant where it is not visited yet, and $\mathcal{R}$ should move towards q'. We set q' to be that quadrant, then $\mathcal{R}$ should move alternatively between two main directions of q'. In the following we will see how to find q':

If $q, q'' == q_1, q_3$, then choose an arbitrary $g_i(r_i, q)$ and see if g_i is a right-gap then q' equals to q_2. Otherwise, if g_i is a left-gap then q' equals to q_4.

Else if $q, q'' == q_2, q_4$, then choose an arbitrary $g_i(r_i, q)$ and see if g_i is a right-gap then q' equals to q_3. Otherwise, if g_i is a left-gap then q' equals to q_1.

3 Analysis

This section covers the proof of correctness and competitive ratio of FTS strategy mentioned in section 2. For every gap $g(r, q)$, r is a reflex vertex adjacent with two edges of $\mathcal{P}$, one of these two edges is visible to $\mathcal{R}$, and the other one is hidden. The extension of the hidden edge into the interior of $\mathcal{P}$ is called *inflection-ray* of r [10]. When $\mathcal{R}$ crosses the inflection-ray of r, $g(r, q)$ disappears accordingly. Every inflection-ray partitions $\mathcal{P}$ into two regions, one includes u and the other one contains v [3]. The latter one is called *unexplored-region* (see Figure 1(b)). Consider a point p as the current position of $\mathcal{R}$. We denote the intersection area of all unexplored-regions of the gaps at point p as I_p. Appendix A covers all the lemmas and observations that we used in the proof.

Theorem 1 *Using the strategy presented in Section 2, $\mathcal{R}$ always find t.*

Proof. The robot $\mathcal{R}$ always faces one of the three cases defined in Section 2 (based on Lemma 4 in Appendix A).

Consider the point t' from which $\mathcal{R}$ sees t. Thus it is sufficient to prove that $\mathcal{R}$ always directs to I_p for $p \in path(s,t')$, regarding those three cases. Let investigate each case as follows.

- Case 1: All of the gaps are located in a quadrant q, and $\mathcal{R}$ moves along an L_1-path through q. In fact, $\mathcal{R}$ must cross the inflection-ray of a gap. Since I_p is behind the unexplored-regions and $\mathcal{R}$ goes toward I_p, then $\mathcal{R}$ must cross an inflection-ray.

- Case 2: By the strategy, $\mathcal{R}$ moves toward a main direction d until an appearance/disappearance event happens. So similar to Case 1, $\mathcal{R}$ must cross an inflection-ray of a gap. In fact, at least one gap must get disappeared when $\mathcal{R}$ moves toward d and $\mathcal{R}$ reaches the inflection-ray of the disappeared gap. Since I_p (for any $p \in path(s,t')$) is behind all of the unexplored-regions, $\mathcal{R}$ must be directed to I_p. When $\mathcal{R}$ moves toward d, there must be a gap that get disappeared, otherwise $\mathcal{P}$ is not street. We claim that there are at least one gap whose inflection-ray is perpendicular to d. By contradiction, assume all inflection-rays of all gaps are collinear with d. Then, the intersection of the unexplored-regions of gaps must be null, and as a result $I_p = \emptyset$, which is not possible. So, there must be a gap that got disappeared while $\mathcal{R}$ moves toward d (see Figure 4(b)).

- Case 3: In this case, I_p (for any $p \in path(s,t')$) lies in the middle quadrant q'. The strategy leads $\mathcal{R}$ to move alternatively along two main directions adjacent to q' (the unexplored middle quadrant). So, $\mathcal{R}$ moves along an L_1-path toward I_p.

In all cases, we showed that $\mathcal{R}$ always pointed to I_p ($p \in path(s,t')$) and never got away from it. As a result, $\mathcal{R}$ must finally find t, and this concludes the proof. $\square$

The following theorem demonstrates that the competitive ratio of the strategy is $\sqrt{10}$. Moreover, Theorem 3 demonstrates that when $t = v$ the competitive ratio is $\sqrt{2}$. Again, Appendix B covers their proofs.

Theorem 2 *Given a rectilinear-street polygon $\mathcal{P}$ (u,v), and a simple robot $\mathcal{R}$, the strategy presented in Section 2 is $\sqrt{10}$-competitive.*

Theorem 3 *If $t = v$, the competitive ratio is $\sqrt{2}$ and it is optimal.*

4 Conclusion

We studied the problem of Free-Target Search in an unknown rectilinear-street for a simple robot. The robot starts from one of two distinguished points u or v and searches for an arbitrary target point t. We used a gap-detector robot that has a minimal sensing capability. Our strategy generates a path, starts at u (or v) to t, with a competitive ratio of $\sqrt{10}$. This paper opens several research lines. We plan to consider the problem when the robot starts from an arbitrary point on the boundary. Another research line is to study the problem in more general environments like streets and generalized-streets.

References

[1] M. Abouei Mehrizi, M. Ghodsi, and A. Tabatabaei. Robots' cooperation for finding a target in streets. In *International Conference on Topics in Theoretical Computer Science*, pages 30–43. Springer, 2015.

[2] C. A. Bröcker and A. López-Ortiz. Position-independent street searching. In *Workshop on Algorithms and Data Structures*, pages 241–252. Springer, 1999.

[3] C. A. Bröcker and S. Schuierer. Searching rectilinear streets completely. In *Workshop on Algorithms and Data Structures*, pages 98–109. Springer, 1999.

[4] S. K. Ghosh and R. Klein. Online algorithms for searching and exploration in the plane. *Computer Science Review*, 4(4):189–201, 2010.

[5] C. Icking, R. Klein, and E. Langetepe. An optimal competitive strategy for walking in streets. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 110–120. Springer, 1999.

[6] R. Klein. Walking an unknown street with bounded detour. *Computational Geometry*, 1(6):325–351, 1992.

[7] S. Schuierer and I. Semrau. An optimal strategy for searching in unknown streets. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 121–131. Springer, 1999.

[8] S. Suri, E. Vicari, and P. Widmayer. Simple robots with minimal sensing: From local visibility to global geometry. *The International Journal of Robotics Research*, 27(9):1055–1067, 2008.

[9] A. Tabatabaei, M. Aletaha, and M. Ghodsi. Competitive strategies for walking in streets for a simple robot using local information. In *The Third Iranian Conference on Computational Geometry*, pages 15–19, 2020.

[10] A. Tabatabaei and M. Ghodsi. Walking in streets with minimal sensing. *Journal of Combinatorial Optimization*, 30(2):387–401, 2015.

[11] A. Tabatabaei, M. Ghodsi, and F. Shapouri. Competitive strategy for walking in streets for an empowered simple robot. ICCG, 2019.

[12] B. Tovar, R. Murrieta-Cid, and S. M. LaValle. Distance-optimal navigation in an unknown environment without sensing distances. *IEEE Transactions on Robotics*, 23(3):506–518, 2007.

[13] Q. Wei, X. Tan, and Y. Ren. Walking an unknown street with limited sensing. *International Journal of Pattern Recognition and Artificial Intelligence*, 33(13):1959042, 2019.

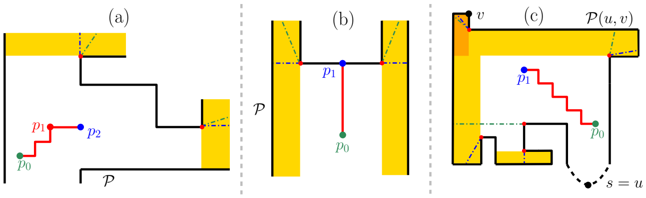

Figure 4: Examples of forbidden cases in which $\mathcal{R}$ sees some gaps at p but $I_p = \emptyset$. These cases does not occur through the robot's path in a street. If such a cases occur the polygon is not a street.

Appendix

A Correctness

This section covers the proof of correctness of the FTS strategy mentioned in section 2. For every gap $g(r, q)$, r is a reflex vertex adjacent with two edges of $\mathcal{P}$, one of these two edges is visible to $\mathcal{R}$, and the other one is hidden. The extension of the hidden edge into the interior of the polygon is called *inflection-ray* of r [10]. When $\mathcal{R}$ crosses the inflection-ray of r, $g(r, q)$ disappears accordingly. Every inflection-ray partitions the polygon into two regions, one includes u and the other one contains v [3].The latter one is called *unexplored-region* (see Figure 1(b)). Consider a point p as the current position of $\mathcal{R}$. We denote the intersection area of all unexplored-regions of the gaps at point p as I_p. In this section, we consider proving the following theorem.

Theorem 1: Using the strategy presented in section 2, $\mathcal{R}$ always find t.

Before proving the statement, let demonstrate some preliminaries.

Observation 1 *If $\mathcal{R}$ sees at least one gap at p, the point v must lie in I_p.*

Proof. By contradiction, let assume that v is in the unexplored-region of a gap $g(r_0, q)$ but not in I_p. As a result, the intersection of the boundary of $\mathcal{P}$ and I_p does not contain v. Also, we already know that I_p does not contain u. If a part of the boundary of $\mathcal{P}$ does not contains neither u nor v, then it must belongs to either one of L_{chain} or R_{chain} of $\mathcal{P}$. Consider a reflex vertex r_1 whose unexplored-region contains I_p but not v. As a result, the points on the hidden-edge of r_1 is not visible from the other chain, which contradicts the definition of a street polygon. □

Note that if there is at least one non-primitive gap, since I_p always contains at least v, $I_p \neq \emptyset$. So, the cases illustrated in Figure 4 can never occur.

Remark 2 *Since the polygon is rectilinear, for every gap $g(r, q)$, the unexplored-region of g can lie in q and at most one of its adjacent quadrants, but it never lies in the opposite (non-adjacent) quadrant of q.*

Observation 3 *Consider p as the current position of $\mathcal{R}$, if there exist gaps in two opposite quadrants (q, q''), I_p must be in the unexplored middle quadrant (q').*

Proof. Since $I_p \neq \emptyset$, the unexplored-regions of the gaps $g(r, q), g(r'', q'')$ must have an intersection region. The unexplored-regions of two opposite quadrants can only intersect in an unexplored quadrant named middle quadrant and denoted by q'. That is because their common region cannot be in q or q'' and must be somewhere in q' (see Figure 1(b)). □

In the next lemma, we show that regardless of the robot's position, the gaps can only lie in at most three quadrants.

Lemma 4 *At each point p of the robot's path, all the gaps must be located in at most three quadrants.*

Proof. Assume a situation where there are some gaps in three quadrants; w.l.o.g. assume that they are located in NE, NW, and SE. Hence, according to the Observation 3, I_p will be located in the middle quadrant, i.e., NW. By contradiction, assume there exists a gap $g(r, SW)$. As we saw in Remark 2, the unexplored-region of $g(r, SW)$ could be located in any of the quadrants except NW. Consequently, I_p is empty ($I_p = \emptyset$) and contradicts Observation 1 (see Figure 4(c)). □

Denote a path generated by the strategy from x to y by $path(x, y)$. We show that the strategy generates a path from $s = u$ to t ($path(s, t)$). Consider the point t' from which $\mathcal{R}$ sees t. It is sufficient to show that the strategy generates a path from s toward I_p for any $p \in path(s, t')$.

At each point $p \in path(s, t')$ if there exist at least one gap, we know v is in I_p. So, $\mathcal{R}$ travels a path from s toward v. Hence, either the whole polygon is cleared before reaching v, or v is reached by $\mathcal{R}$. At the latter case, $\mathcal{R}$ traveled a $path(s, v)$. Klien [6] proved that every path from $s = u$ to v explores the whole polygon. Therefore, at both cases, the whole polygon gets explored and $\mathcal{R}$ surely sees t.

Consider a point p_f on the boundary of last unexplored I_p $p \in path(s, t')$, when $\mathcal{R}$ reaches I_{p_f} all the gaps must disappear, and the whole polygon is clear. That is because v lies in I_{p_f}. So, the robot already passed over the path $path(s, p_f)$, and lies on p_f, and the only unexplored region of $\mathcal{P}$ (I_{p_f}) gets visible for $\mathcal{R}$. For every $p' \in path(p_f, t)$ $I_{p'} = \emptyset$. So, we only need to show that $\mathcal{R}$ always moves toward I_p for any $p \in path(s, t')$. We will show that considering the strategy's cases, the robot can eliminate each non-primitive-gap only once. Hence, the total path of $\mathcal{R}$ is finite, and the strategy will terminate.

Proof. [Proof of Theorem 1] Based on Lemma 4, $\mathcal{R}$ always faces one of the three cases defined in Section 2. Thus it is sufficient to prove that $\mathcal{R}$ always directs to I_p for any $p \in path(s, t')$, regarding all those three cases. We investigate each case separately, as follows.

- Case 1: All of the gaps are located in a quadrant q, and $\mathcal{R}$ moves along an L_1-path through q. In fact, $\mathcal{R}$ must cross the inflection-ray of a gap. Since I_p is behind the unexplored-regions and $\mathcal{R}$ goes toward I_p, then $\mathcal{R}$ must cross an inflection-ray.
- Case 2: By the strategy, $\mathcal{R}$ moves toward a main direction d until an appearance/disappearance event happens.

So similar to Case 1, $\mathcal{R}$ must cross an inflection-ray of a gap. In fact, at least one gap must get disappeared when $\mathcal{R}$ moves toward d and $\mathcal{R}$ reaches the inflection-ray of the disappeared gap. Since I_p (for any $p \in path(s,t')$) is behind all of the unexplored-regions, $\mathcal{R}$ must be directed to I_p. When $\mathcal{R}$ moves toward d, there must be a gap that get disappeared, otherwise $\mathcal{P}$ is not street. We claim that there are at least one gap whose inflection-ray is perpendicular to d. By contradiction, assume all inflection-rays of all gaps are collinear with d. Then, the intersection of the unexplored-regions of gaps must be null, and as a result $I_p = \emptyset$, which is not possible. So, there must be a gap that got disappeared while $\mathcal{R}$ moves toward d (see Figure 4(b)).

- Case 3: In this case, I_p (for any $p \in path(s,t')$) lies in the middle quadrant q'. The strategy leads $\mathcal{R}$ to move alternatively along two main directions adjacent to q' (the unexplored middle quadrant). So, $\mathcal{R}$ moves along an L_1-path toward I_p.

In all cases, we showed that $\mathcal{R}$ always pointed to I_p (p is a point in $path(s,t')$) and never got away from it. As a result, $\mathcal{R}$ must finally find t, and this concludes the proof. □

B Competitive Ratio

In this section, we intend to prove that the strategy stated in section 2 provides a competitive ratio of $\sqrt{10}$ for a simple robot $\mathcal{R}$.

During the robot's movement, whenever t becomes visible, $\mathcal{R}$ can detect it and move toward it. Let denote the last intersection of all unexplored-regions that $\mathcal{R}$ may see in its path ($path(s,t)$) by I_{p_f}, where p_f is a point on the boundary of last unexplored I_p (for any $p \in path(s,t')$).

Consider a situation where $\mathcal{R}$ lies on p_f. In such a case, t is located in I_{p_f} and $\mathcal{R}$ moves directly toward t. Note that if t is not in I_{p_f}, $\mathcal{R}$ must have seen it before reaching p_f.

In the following theorem, we show that all cases provide an L_1-shortest-path but Case 3. In fact, the competitive ratio for the first two cases is $\sqrt{2}$, while it is $\sqrt{10}$ for case 3.

Theorem 2: Given a rectilinear-street polygon $\mathcal{P}$ (u,v), and a simple robot $\mathcal{R}$, the strategy presented in Section 2 is $\sqrt{10}$-competitive.

Proof. When t is not visible to $\mathcal{R}$, it is hidden behind a gap $g(r,q)$. Consider a situation where t is located on the line containing an inflection-ray. Suppose $\mathcal{R}$ is not reached that inflection-ray yet, and the target point t is on the hidden edge $g(r,q)$. In this situation, $\mathcal{R}$ cannot see t until it reaches the inflection-ray of r. Regarding the positions of t and r, in other situations $\mathcal{R}$ might see t before it reaches the inflection-ray of r, and it will pass a shorter path to meet t (see Figure 5(a)). Hence, when t is on the hidden edge of a gap, $\mathcal{R}$ has to pass longer path concerning any other position of t.

For the first two cases, we show that when $\mathcal{R}$ reaches t, the robot's path length is at most equal to an L_1-shortest-path from s to t. This is because $\mathcal{R}$ approaches to quadrant(s) in which there exist gap(s). Note that R never gets away from gaps. We consider each case separately in the following.

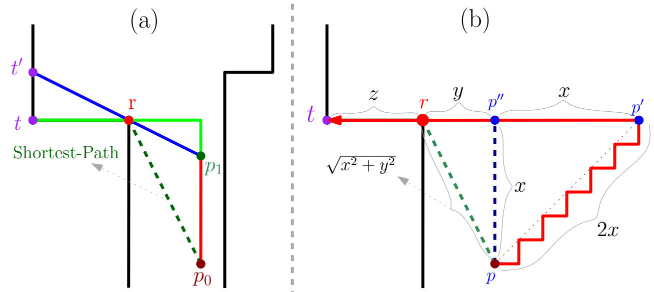

Figure 5: Worst case for finding t.

- Case 1: In this case, t is behind one of the gaps. Since all of the gaps are located in the same quadrant (q), and $\mathcal{R}$ moves toward q (one step in each direction alternatively), then its path length is equal to the L_1-shortest-path (see Figure 2(b)).
- Case 2: In this case, the gaps are in two adjacent quadrants q, q', and $\mathcal{R}$ moves toward the main direction between them. Since t is behind one of the gaps when R goes straight between q, q', it guarantees an L_1-path as $\mathcal{R}$ will never go away from t, and then its length is equal to the L_1-shortest-path (see Figure 3(a)).

In the above cases, $\mathcal{R}$ passes through an xy-monotone path, and the length of the path is equal to an L_1-shortest-path. If t is found during case 1 and 2, it can be showed the path of $\mathcal{R}$ can be extended by an L_1-shortest path to t. Hence, it guarantees the competitive ratio of $\sqrt{2}$ concerning the L_2-shortest-path.

Regarding case 3 (see Figure 3(b)), since I_p is in the middle quadrant q', $\mathcal{R}$ approaches to q'. If t is behind a gap in q', it will be similar to case 1, and the robot's path is an L_1-shortest-path. However, if t is behind one of the crucial gaps, $\mathcal{R}$ is getting away from t. So, the worst case is when the case 3 arises and t is behind a crucial gap; we denote the start point of such a case by p. After p, $\mathcal{R}$ may face many case 3 continuously because of some disappearance events. We consider all of them together and denote the endpoint of the last one by p'. We denote by p'' the point on the inflection ray of the reflex vertex r related to that crucial gap which is perpendicular to p. According to Figure 5(b), $\mathcal{R}$ travels $2x+x+y+z$ while the length of the shortest-path is $\sqrt{x^2+y^2}+z$. So, the competitive ratio is at most

$$\max\left\{\frac{2x+x+y+z}{\sqrt{x^2+y^2}+z}\right\}.$$

If we set $z=0$, by finding the function's extremum point, we get $\sqrt{10}$ as the maximum possible competitive ratio. Please note that if t is not visible at p', after p' $\mathcal{R}$ may face other cases until t is visible, but in those cases, the competitive ratio is less and greatest deviation occurs when t is found in a case 3. □

In the following theorem, we show that if the target point t is located in the same position of v, then the presented strategy guarantees an optimal competitive ratio.

Theorem 5 *If $t = v$, the competitive ratio is at most $\sqrt{2}$ and it is optimal.*

Proof. The proof is coming from Theorem 2 and Observation 1. In other words, in Observation 1 we showed that v always lies in I_p (for any $p \in path(s, t')$), and in Theorem 2 we showed that the strategy guarantees an L_1-shortest-path toward I_p. Since $t = v \in I_p$ Then it gives a competitive ratio of at most $\sqrt{2}$, which is optimal [6]. $\square$