Metadata of the chapter that will be visualized in SpringerLink

Book Title	Combinatorial Optimization and Applications	
Series Title		
Chapter Title	Optimal Strategy for Walking in Streets with Minimum Number of Turns for a Simple Robot	
Copyright Year	2014	
Copyright HolderName	Springer International Publishing Switzerland	
Corresponding Author	Family Name	**Tabatabaei**
	Particle	
	Given Name	**Azadeh**
	Prefix	
	Suffix	
	Division	Department of Computer Engineering
	Organization	Sharif University of Technology
	Address	Tehran, Iran
	Email	atabatabaei@ce.sharif.edu
Author	Family Name	**Ghodsi**
	Particle	
	Given Name	**Mohammad**
	Prefix	
	Suffix	
	Division	School of Computer Science, Institute for Research in Fundamental Sciences (IPM)
	Organization	Sharif University of Technology
	Address	Tehran, Iran
	Email	ghodsi@sharif.edu

Abstract	We consider the problem of walking a simple robot in an unknown street. The robot that cannot infer any geometric properties of the street traverses the environment to reach a target t, starting from a point s. The robot has a minimal sensing capability that can only report the discontinuities in the depth information (gaps), and location of the target point once it enters in its visibility region. Also, the robot can only move towards the gaps while moving along straight lines is cheap, but rotation is expensive for the robot. We maintain the location of some gaps in a tree data structure of constant size. The tree is dynamically updated during the movement. Using the data structure, we present an *online* strategy that generates a search path for the robot with optimal number of turns.

Keywords (separated by '-')	Computational geometry - Minimum link path - Simple robot - Street polygon - Unknown environment

Author Proof

Optimal Strategy for Walking in Streets with Minimum Number of Turns for a Simple Robot

Azadeh Tabatabaei[1(✉)] and Mohammad Ghodsi[2]

[1] Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
atabatabaei@ce.sharif.edu

[2] School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Sharif University of Technology, Tehran, Iran
ghodsi@sharif.edu

Abstract. We consider the problem of walking a simple robot in an unknown street. The robot that cannot infer any geometric properties of the street traverses the environment to reach a target t, starting from a point s. The robot has a minimal sensing capability that can only report the discontinuities in the depth information (gaps), and location of the target point once it enters in its visibility region. Also, the robot can only move towards the gaps while moving along straight lines is cheap, but rotation is expensive for the robot. We maintain the location of some gaps in a tree data structure of constant size. The tree is dynamically updated during the movement. Using the data structure, we present an *online* strategy that generates a search path for the robot with optimal number of turns.

Keywords: Computational geometry · Minimum link path · Simple robot · Street polygon · Unknown environment

1 Introduction

Due to many real life applications, path planning in unknown environments is considered as a fundamental problem in robotics, computational geometry and online algorithms [2,12,17]. A robot based on the information gathered from its tactile sensors moves in the environment until it achieves its target. Neither the geometric map of the environment nor the location of the target point are known to the robot. The volume of information provided to the robot depends on the strength of its sensors. Employing a simple robot with a simple sensing model has many advantages such as: low cost of hardware, being applicable to many situations, and being robust against sensing uncertainty and noise [3–5,20].

Here a simple robot with an abstract sensor that can only detect the order of discontinuities in the depth information (or gaps) in its visibility region is

© Springer International Publishing Switzerland 2014
Z. Zhang et al. (Eds.): COCOA 2014, LNCS 8881, pp. 1–12, 2014.
DOI: 10.1007/978-3-319-12691-3_9

Author Proof

considered. Each discontinuity represents a portion of the environment that is not visible to the robot (Fig. 1). A label of L or R is assigned to each gap g depending on which side of the gap the hidden region is. Also, the robot recognizes a target point t as it enters in its omnidirectional and unbounded field of view. Moving along straight lines is cheap, but rotation is expensive for the robot. The robot using the information gathered through the sensor starts navigating a street environment from a start point s to reach a target t. A street is a simple polygon P with two vertices s and t such that the counter-clockwise polygonal chain R_{chain} from s to t and the clockwise one L_{chain} from s to t are mutually weakly visible. In other words each point on the left chain is visible from at least one point on the right chain and vice versa [8], (Fig. 1.a). Note that minimizing the number of turns is an essential criterion in path planning for such robot. This problem is also known as the shortest path problem in the link metric, in some literatures [10,13].

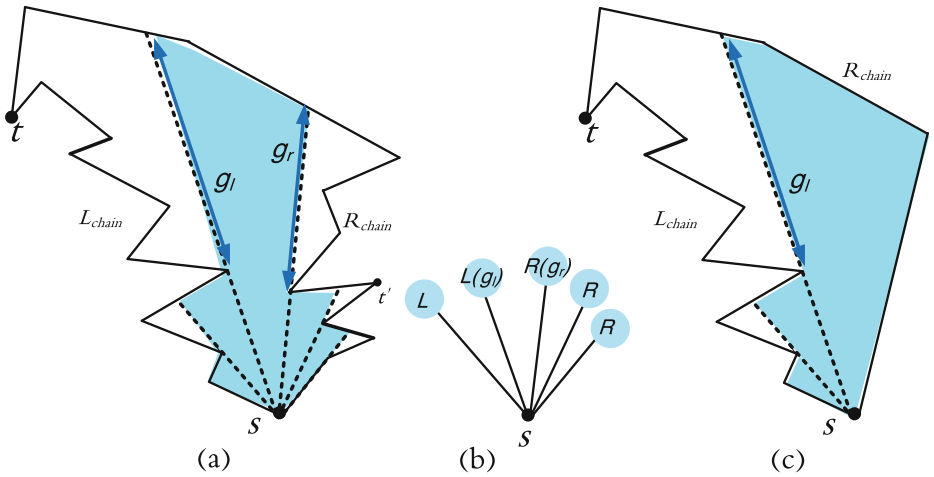

Fig. 1. (a) A street in which L_{chain} is the left chain and R_{chain} is the right chain. The colored region is the visibility polygon of the point robot at the start point s. (b) The position of discontinuities in the depth information detected by the sensor at the start point in (a). (c) A street that has only left gaps at the start point.

The simple robot system applied in this paper was first presented by Lavalle *et al.* [19]. They proposed Gap Navigation Tree (GNT) as a mean to maintain location of the gaps sensed by the robot for navigating the unknown scene. The topological changes of the robot's visibility region, along a path, modify the position of the gaps. Once the GNT is completed, it encodes the shortest path from the current position of the robot to any place in a simply connected environment. Also, it is proved in [18] that, using this data structure, locally optimal searching can be achieved. An online algorithm for the well-known visibility problem pursuit-evasion in an unknown simply connected environment is

Author Proof

proposed by Guilamo *et al.* [4,14] using GNT. An optimal search algorithm is offered for a disc robot, using GNT, in a simply connected environment [11]. A competitive strategy is presented for walking in streets for a point robot that is equipped with the gap sensor in [16] such that the generated path is at most 11 times longer the shortest path.

Other minimal sensing models have been introduced by other researchers. Suri *et al.* [15] offered a robot system in which the robot can only sense the combinatorial (non-metric) properties of its surroundings. The sensor detects the vertices of the polygon in its visibility region, and can distinguish if there is an edge between consecutive vertices of the region. Their robot is equipped with a compass in [1]. Katsev *et al.* [7] introduced a simple robot that performs wall-following motions and can traverse the interior of the scene only by following a direction that is parallel to an edge of the environment. Despite of the minimal capabilities, all of them shown that their robot can provide many geometric reasoning and executes many non-trivial tasks such as counting vertices, solving pursuit-evasion problems and mapping a polygon.

The problem of walking in unknown streets with minimum number of turns was first studied in [9]. They presented an optimal online algorithm for a robot with an on-board vision system based on the map that was provided by the robot of its visibility region. In other words their robot has access to the map of its visibility region in the street. In this research, our robot is a simple robot that cannot realize any geometric properties of the surroundings such as coordinates, angles or distances in contrast with the robot system applied in [9] for this problem. Our simple robot traverses interior of the street detecting the positions of the gaps while it can only move towards them. Even with such a confined sensing model, we show that the robot can traverse an optimal path as well as the robot employed in [9]. We present a tree data structure of constant size that maintains the essential data for leading the robot. The tree dynamically changes during the movement.

2 The Sensing Model and Motion Primitive

A point robot starts exploring an un known street until the target t is achieved, starting from s. The robot is equipped with a sensor that detects each discontinuities in depth information that referred as gaps. The sensor reports a cyclically ordered location of the gaps in its visibility region, (Fig. 1). Also, the robot allocates a label of L or R (left or right) to each gaps. Each label shows direction of the hidden region behind a gap relative to the robot's heading [18]. The robot can only track the gaps and records their topological changes. These changes are: appearance, disappearance, merging, and splitting of gaps. The appearance and disappearance events happen when the robot crosses the inflection rays (Fig. 2). Each appearance event generates a gap that corresponds to a portion of the environment that was so far visible, and now is invisible. A gap that is generated by an appearance event, during the movement, is called a primitive gap and the other gaps are non-primitive gaps. The merge and split events occur when the robot crosses the bitangent complements (Fig. 2).

Author Proof

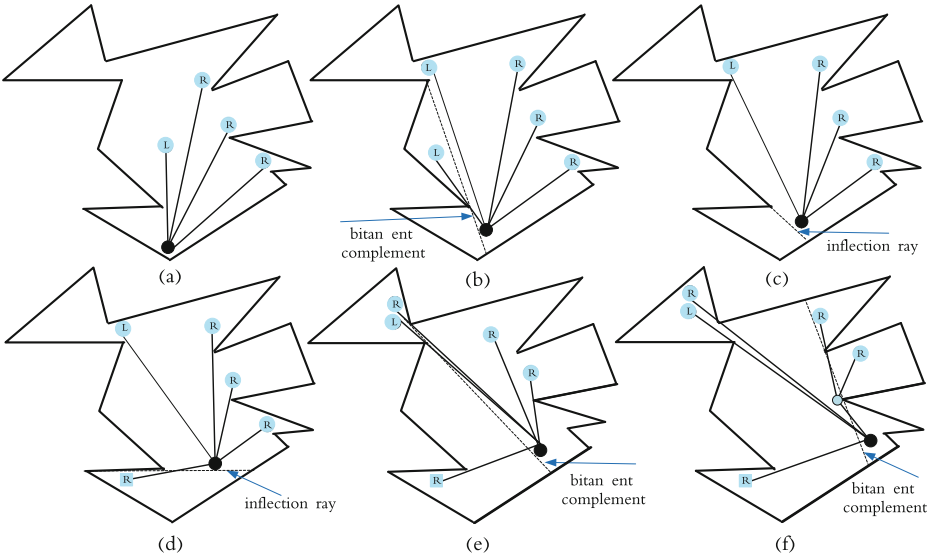

Fig. 2. Illustration of the dynamically changes of the gaps as the robot moves towards a gap. The dark circle denotes the location of the robot, and squares and other circles display primitive and non-primitive gaps respectively. (a) Existing gaps at the beginning. (b) A split event. (c) A disappearance event. (d) An appearance event. (e) Another split event. (f) A merge event.

In order to cover entire region, the robot follows the non-primitive gaps; the region hidden behind the primitive gaps has already been covered. The robot moves along a straight line towards non-primitive gaps, and may rotate as a critical event occurs. Also, the robot makes a turn when a wall of the environment is hit. As the target enters in the robot's visibility region, the robot orients its heading with the target, and walks towards it. Note that moving along straight lines is cheap, but rotation is expensive. At the point in which there is no non-primitive gap the entire environment has been observed by the robot.

3 Algorithm

Here we explain our strategy for leading the robot in a street to reach the target t, starting from s, such that the number of turns in the robot's search path is as small as possible. At each point of the search path, the sensor detects the target t or reports a set of gaps with the label of L or R (l-gap and r-gap for abbreviation).

Definition 1. *[16] In the set of non-primitive l-gaps, the gap which is in the right side of the others is called most advanced left gap and is denoted by g_l. Analogously, in the set of non-primitive r-gaps, the gap which is in left side of the others is called the most advanced right gap and is denoted by g_r (Fig. 1.b).*

Author Proof

We use the following property of streets in searching the target t.

Lemma 1. *[8,16] On any point of the robot search path, the target t is hidden behind one of the most advanced gaps unless t is visible to robot.*

Proof. Let the target be hidden behind another gap, for example t' in Fig. 1.a is behind an r-gap. Then, the points that are immediately behind g_r are not visible from any point on the R_{chain} that connects s to t'. This is a contradiction with the definition of street.

From Lemma 1, the target is constantly behind the most advanced gaps. So, at the start point s, there is at least one of g_l and g_r unless t is visible from s. If only one of them exists, the robot moves towards the gap in order to cover the region that is behind it (Fig. 1.c). The case in which both of advanced gaps exist is called a funnel case [6,16]. When a funnel situation arises at the start point, the robot moves towards one of the most advanced gaps, for example g_l, to cover the region hidden behind it (Fig. 1.a). Observe that moving along an advanced gap in a funnel situation also decreases the region that is behind the other advanced gap. As the robot traverses interior of the street, g_l and g_r dynamically change. During the traversing, we hold the essential data for memorizing the location of two gaps in a tree. The tree has only two branches that address paths towards g_l and g_r. At the beginning, the start point is root of the tree and g_l and g_r are its children that are circularly ordered around the root. Always the robot's location is root of the tree. The critical events (appearance, disappearance, merge, and split) that change the structure of the robot's visibility region, dynamically change branches of the tree, g_l and g_r, (Fig. 3). The tree data structure is updated as follows:

1. When the robot crosses a bitangent complement of g_r/g_l and another r-gap/l-gap, then g_r/g_l splits and will be replaced by the r-gap/l-gap, (point 1 in Fig. 3).
2. When the robot crosses a bitangent complement of $g_r/\ g_l$ and an l-gap/r-gap, then g_r/g_l splits into two gaps. g_l/g_r will be replaced by the l-gap/r-gap, (point 2 in Fig. 3 and point 2 in Fig. 4.a).
3. Each appearance event generates a gap that hides a portion of the street that already was visible. Such gap is a primitive gap. So, the data structure does not update by arising an appearance event, (point 3 in Fig. 3). A primitive gap changes the data structure when a most advanced gap merges with the gap; the most advanced gap will be a child of the primitive gap in the data structure, (point 4 in Fig. 3).
4. When the robot crosses over an inflection ray, each of g_l or g_r which is adjacent to the ray, disappears and is eliminated, (point 5 in Fig. 3 and point 1 in Fig. 4.a).

The dynamically changes of the data structure, as the robot traverses the street, are illustrated in Fig. 3.

Our main idea for reducing the number of turns (links) of the search path is maximum use of a selected direction. In other words the robot continues moving

Author Proof

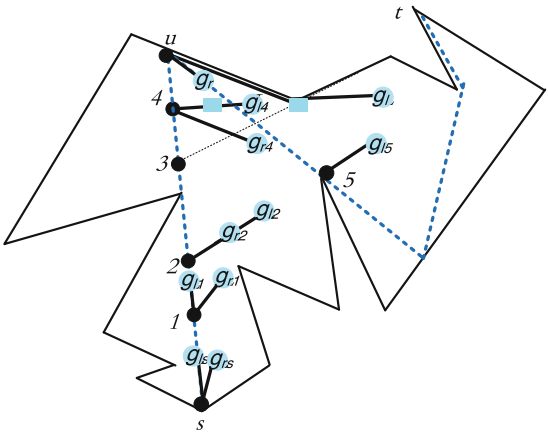

Fig. 3. g_{rs} and g_{ls} are the most advanced gaps at the start point s. g_{ri} and g_{li} are the most advanced gaps at point i. The dotted path that connects s to t is the robot search path. Both of the two most advanced gaps are active at the start point. At point 1, event (1) arises that updates g_r. At point 2, event (2) arises that updates g_l and sets it as an inactive gap. At point 4, event (3) arises. At point 5, the right most advanced gap disappears.

along a selected direction as long as at least one of the hidden region behind the two gaps decreases. In our strategy, each decision for turning is only based on the critical events (appearance, disappearance, merge, and split) which change the robot's visibility region while in the presented strategy in [9] the robot based on the available map of its visibility region selects a direction to move, and makes its decisions for turning. We refer to a most advanced gap as an *active* gap if further movement along the selected direction allows the robot to see more of the hidden region behind the gap.

At the start point, if the funnel situation arises, the robot moves towards g_l to cover the hidden region behind it. By this movement the hidden region behind g_r also decreases. So, by the definition, g_l and g_r are active (Fig. 3). If the other situation arises, in which there exists only on advanced gap, the robot moves towards it to cover the region behind this gap. So, the gap is also an active gap, by the definition. During the movement a gap may switches from being active to being inactive and vice versa. Some of the above events which update g_l and g_r may exchange an advanced gap from being active to being inactive and vice versa. In the following, the events that exchange a gap from being active to being inactive and vice versa is explained in both conditions, funnel condition and the other condition.

– In a funnel condition, the active most advanced gaps may switch from being active to being inactive as soon as the condition ends. It ends if one of these events occurs: (i) The robot enters a point in which the two most advanced gaps are collinear, (point 1 in Fig. 4.b). Further movement along the current

Author Proof

direction constructs a new funnel case. One of the most advanced gaps of this funnel is a most advanced gap of the previous funnel, so this gap remains active. The other most advanced gap of new funnel is revealed via splitting the gap of the previous funnel. Further movement along the current direction does not allow the robot to see more of the hidden region behind the gap. So, the gap is set as inactive gap, (point * and gap * in Fig. 4.b). (ii) The robot enters a point in which one of the two active gaps disappears. Note that at this point, only one of the most advanced gaps exists and it is active, (point 1 in Fig. 4.a).

– When there exist only one of the most advanced gaps, for example only g_l exists, a funnel situation starts via splitting g_l into g_l and an r-gap. The most advanced gap of this funnel are g_l and the r-gap. So the r-gap is current g_r. Further movement along the current direction does not allow the robot to see more of the hidden region behind g_r. So, this gap is set as inactive gap. g_l remains active, (point 2 in Fig. 4.a). The situation in which only g_r exists is symmetric, and an inactive g_l may be generated analogously.

The robot continues to walk along the selected direction and makes a turn as often as each of the following conditions occurs.

1. The robot hits a point u on a wall such that it cannot proceed further. Recall that the counter-clockwise polygonal chain from s to u or the clockwise one from s to u or both are weakly visible from each simple path that connects s to u [9]. So, by arising a disappearance event or a split event, the most advanced gap that lies on the chain has become inactive before reaching the hit point u. At the hit point, the robot moves towards the gap that has not become inactive. Also at the turn point, the existing most advanced gaps will be set as active gaps again, (point u in Fig. 3).
2. The robot achieves a point in which none of the most advanced gaps are active. Further movement along the current direction does not allow the robot to see more hidden region behind the gaps. One of the situations below has arisen.
   – The existing active most advanced gap merges with an inactive gap (point 3 in Fig. 4.a and point 2 in Fig. 4.b). At this point, the robot turns towards the merged gap and the gap will be set as an active gap.
   – The existing active most advanced gap disappears, (point 2 in Fig. 5). At this point, the robot turns towards the existing advanced gap and the gap will be set as an active gap.

4 Analysis of the Algorithm

In this section, we enumerate the number of the links of the generated path by our strategy for the robot to reach the target t starting from s. Also we prove a competitive ratio for our strategy. Assume that $SP(s,t)$ is the Euclidean shortest path from s to t. An edge u_iu_j of $SP(s,t)$ is called an eave if u_{i-1} and u_{j+1} lie on the different sides of the line that connects u_i to u_j [10] (Fig. 5). First we consider a simple case in which the shortest path between s and t has no eave; $SP(s,t)$ has only right turns or only left turns.

Author Proof

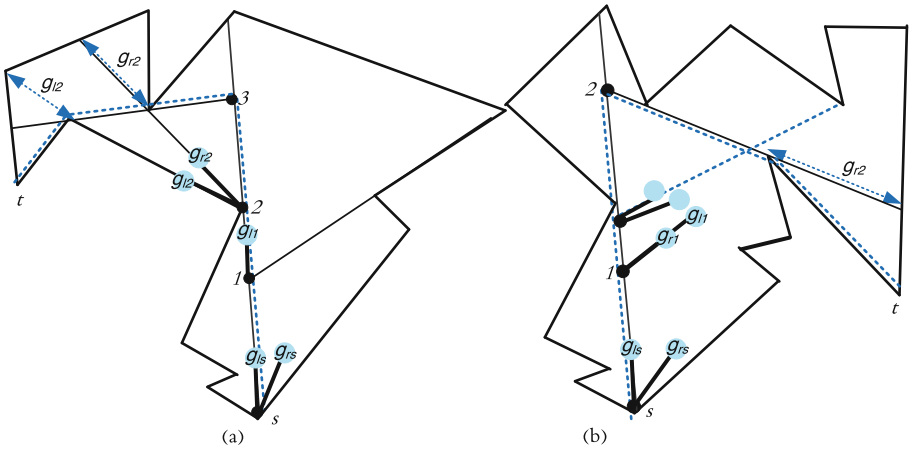

Fig. 4. The dotted path is the robot search path from s to t. (a) A disappearance event occurs at point 1. At point 2, event (1) arises that updates g_l, also an event (2) arises that updates g_r and sets g_r as an inactive gap. At point 3, the active gap g_l merges with the inactive gap g_r. (b) At point 2 the active gap g_r merges with a non-primitive gap.

Lemma 2. *Consider the case that $SP(s,t)$ has only left turns and t is not in the robot's visibility region at the start point s. Our strategy generates a path for the robot to reach the target t with an optimal number of links.*

Proof. Since t is not visible, the robot moves towards g_l. If the target is visible from a point along this direction, by the strategy, the robot does not turn before reaching the point. So, the robot turns only at the point and moves towards t. In the case when t is not visible from any point along the direction, the robot can make a turn when either condition (1) or condition (2) for turning occurs, (Fig. 6.a). Here the key observation is that at each of the turn points, $z_1, z_2, ..., z_n$, both of the left tangent and the right tangent to $SP(s,t)$ lie inside the street (Fig. 6.a). Any minimum link path between s and t must intersect the left tangents to $SP(s,t)$ from $z_1, z_2, ..., z_n$ [10]. Whereas no link can intersect more than one of the left tangents, the path is a minimum link path.

Lemma 3. *If $SP(s,t)$ has only right turns and t is not in the robot's visibility region at the start point s, then our strategy generates a path for the robot to reach the target t that has at most one link more than the optimal path.*

Proof. At the start point s there is two situations: In the first situation, some l-gap exist at the start point. So, the robot moves towards g_l. The case that t is visible from a point on this direction is similar to the corresponding case in the proof of Lemma 2. So, the robot moves from the point towards t. When t is not visible from any point along the direction, g_l becomes inactive at some point x. So either case (1) or case (2) for turning arises, the robot makes a right turn

Author Proof

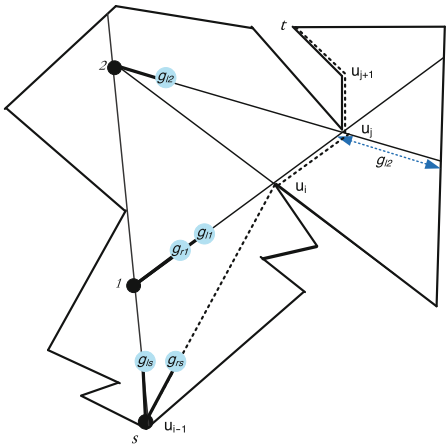

Fig. 5. At the start point both of g_l and g_r are active. At point 1, g_l becomes inactive. At point 2, g_r becomes inactive, as well as g_l. So, the robot makes a turn at this point. The dotted path is the shortest path from s to t and u_iu_j is an eave.

towards the current g_r. Similarly at next turn points the robot makes a right turn. A similar argument to the proof of Lemma 2 shows that the generated path from the first turn point to t has minimum number of links. Then the robot achieves t with at most one additional turn than the optimal (Figs. 4.b and 6.b). In the second situation, no l-gap exists at the start point. The robot moves towards g_r. This condition is symmetric to the situation of Lemma 2 and the number of the robot's turns to achieve the goal is optimal. Hence the claim is proved at the both first and second situations.

Now we consider the general case in which $SP(s,t)$ has both left and right turns. Following property of a street is used as a guideline for obtaining the competitive ratio of our algorithm.

Lemma 4. *[10] There exists a minimum link path that contains all eaves of* $SP(s,t)$.

In the general case, there are some eaves in the path. Let u_iu_j be the first eave of the path. The shortest path from s to u_i has only right turns or only left turns. If we extend the eave to the street boundary of both side, by our strategy, the robot achieves a point x_i on the first extension with at most one link more than the optimal path, see Fig. 7.

Assume that the $SP(s,t)$ makes left turns from s to u_i and makes a right turn at u_j. The robot, after passing through the point x_i, turns left as soon as each of the conditions (1) or (2) for turning arises and crosses the eave. by our strategy, g_l becomes inactive before the robot achieves the next turn point. So, the robot turns towards the current g_r and crosses the other extension of the eave, x_j. Thus the robot traverses from first extension of an eave to the other

Author Proof

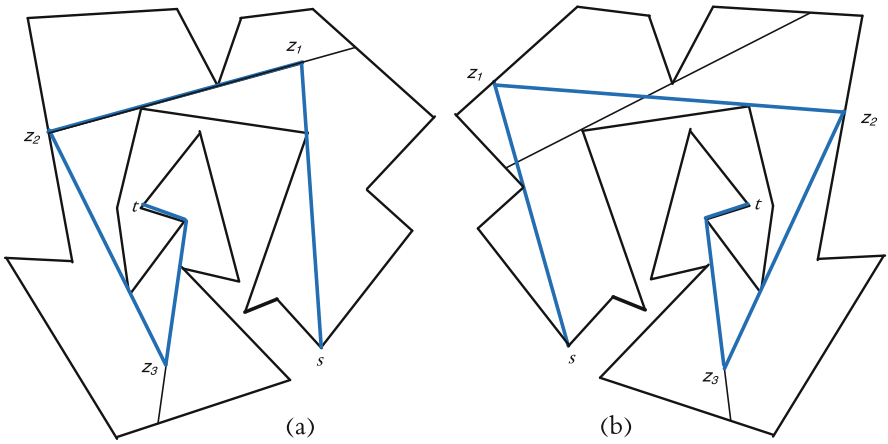

Fig. 6. The bold path is the robot search path. The left and right tangent from each turn point z_i lies inside the street. (a) Shortest path from s to t has only left turns. (a) Shortest path from s to t has only right turns.

extension of the eave with at most two links. Above discussion and Lemma 4 prove the main result of this paper.

Theorem 1. *The robot achieves the target t, starting from s, with at most $m + 1 + e$ links where m is the link distance between s and t and e is the number of eaves in $SP(s,t)$. Also, the generated path by our strategy is optimal.*

Proof. From Lemma 3, the robot may make an additional turn if $SP(s,t)$ turns right at the start point to reach a point on the extension of the first eave. Above discussion shows that the robot makes an additional turn for traversing from first extension to the other extension for every eaves. Since shortest path between two consecutive eave has only right/left turns, the arguments used in the proof of Lemma 2 proves that robot traverses the distance with optimal number of links. So, the robot achieves the target using at most $m + 1 + e$ links. The number of links is equal to the number of the links of the optimal path proposed in [9] for a robot with on-board vision system. Hence this result is optimal for our robot with the minimal sensing capability.

Theorem 2. *Our online strategy terminates while a search path from s to t with the competitive ratio of $2 - 1/m$ is generated for the simple robot, using a constant size memory space.*

Proof. In order to find the target, the robot moves towards the most advanced gaps. So all things that we maintain, during the traversing, are the location of the two gaps in the tree data structure of constant size. Although new most advanced gaps may reveal via splitting during the movement, the number of such events are finite; each corresponds to a crossing over a bitangent complement. The robot is one step closer to reach the target as soon as a most advanced gap

Author Proof

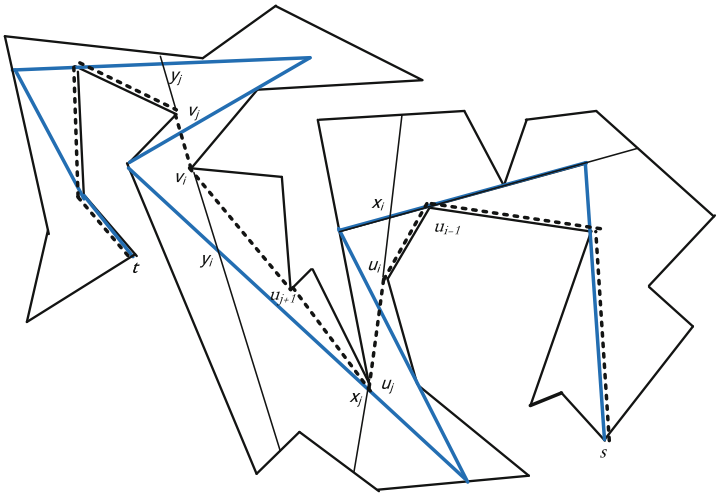

Fig. 7. The general case in which shortest path from s to t has both left and right turns. Bold path is the robot search path.

disappears. So, the strategy should terminate while the target is achieved. From Lemma 4, $e \leq m-2$. Then, the number of links of the path is at most $2m-1$, from Theorem 1. Hence the competitive ratio is $2-1/m$.

5 Conclusions

In this paper, we considered the problem of walking in streets for a point robot that has a minimal sensing capability. The robot can only detect the gaps and the target in the street. Also, moving along straight lines is cheap, but rotation is expensive for the robot. We proposed an online search strategy that generates a search path for such robot, using a tree data structure of constant size. The robot, starting from s, traverses the street to reach the target t with at most $2m-1$ links, where m is the link distance between s and t. In other words our strategy has optimal competitive ratio of $2-1/m$. Proposing a competitive search strategy for the simple robot in more general classes of polygons is an attractive open problem.

References

1. Disser, Y., Ghosh, S.K., Mihalk, M., Widmayer, P.: Mapping a polygon with holes using a compass. Theor. Comput. Sci. (in press, corrected proof) (Available online 18 December 2013)
2. Fekete, S.P., Mitchell, J.S.B., Schmidt, C.: Minimum covering with travel cost. J. Comb. Optim. **24**, 32–51 (2003)

Author Proof

3. Ghannadpour, S.F., Noori, S., Tavakkoli-Moghaddam, R.: A multi-objective vehicle routing and scheduling problem with uncertainty in customers request and priority. J. Comb. Optim. **28**, 414–446 (2012)
4. Guilamo, L., Tovar, B., LaValle, S.M.: Pursuit-evasion in an unknown environment using gap navigation trees. In: Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robot's and Systems, (IROS 2004), vol. 4, pp. 3456–3462. IEEE, September 2004
5. Hammar, M., Nilsson, B.J., Persson, M.: Competitive exploration of rectilinear polygons. Theor. Comput. Sci. **354**(3), 367–378 (2006)
6. Icking, C., Klein, R., Langetepe, E.: An optimal competitive strategy for walking in streets. In: Meinel, C., Tison, S. (eds.) STACS 1999. LNCS, vol. 1563, pp. 110–120. Springer, Heidelberg (1999)
7. Katsev, M., et al.: Mapping and pursuit-evasion strategies for a simple wall-following robot. IEEE Trans. Robot. **27**(1), 113–128 (2011)
8. Klein, R.: Walking an unknown street with bounded detour. Comput. Geom. **1**(6), 325–351 (1992)
9. Kumar Ghosh, S., Saluja, S.: Optimal on-line algorithms for walking with minimum number of turns in unknown streets. Comput. Geom. **8**(5), 241–266 (1997)
10. Kumar Ghosh, S.: Computing the visibility polygon from a convex set and related problems. J. Algorithms **12**(1), 75–95 (1991)
11. Lopez-Padilla, R., Murrieta-Cid, R., LaValle, S.M.: Optimal gap navigation for a disc robot. In: Frazzoli, E., Lozano-Perez, T., Roy, N., Rus, D. (eds.) Algorithmic Foundations of Robotics X. STAR, vol. 86, pp. 123–138. Springer, Heidelberg (2013)
12. Lpez-Ortiz, A., Schuierer, S.: On-line parallel heuristics, processor scheduling and robot searching under the competitive framework. Theor. Comput. Sci. **310**(1), 527–537 (2004)
13. Mitchell, J.S., Rote, G., Woeginger, G.: Minimum-link paths among obstacles in the plane. Algorithmica **8**(1–6), 431–459 (1998)
14. Sachs, S., LaValle, S.M., Rajko, S.: Visibility-based pursuit-evasion in an unknown planar environment. Int. J. Robot. Res. **23**(1), 3–26 (2004)
15. Suri, S., Vicari, E., Widmayer, P.: Simple robots with minimal sensing: from local visibility to global geometry. Int. J. Robot. Res. **27**(9), 1055–1067 (2008)
16. Tabatabaei, A., Ghodsi, M.: Walking in streets with minimal sensing. In: Widmayer, P., Xu, Y., Zhu, B. (eds.) COCOA 2013. LNCS, vol. 8287, pp. 361–372. Springer, Heidelberg (2013)
17. Tan, X., Bo, J.: Minimization of the maximum distance between the two guards patrolling a polygonal region. Theor. Comput. Sci. **532**, 73–79 (2014)
18. Tovar, B., Murrieta-Cid, R., LaValle, S.M.: Distance-optimal navigation in an unknown environment without sensing distances. IEEE Trans. Robot. **23**(3), 506–518 (2007)
19. Tovar, B., LaValle, S.M., Murrieta, R.: Optimal navigation and object finding without geometric maps or localization. In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'03, vol. 1, pp. 464–470. IEEE, September 2003
20. Xu, Y., et al.: The canadian traveller problem and its competitive analysis. J. Comb. Optim. **18**(2), 195–205 (2009)

MARKED PROOF

Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

Instruction to printer	*Textual mark*	*Marginal mark*
Leave unchanged	··· under matter to remain	
Insert in text the matter indicated in the margin		New matter followed by or
Delete	through single character, rule or underline or through all characters to be deleted	or
Substitute character or substitute part of one or more word(s)	through letter or through characters	new character or new characters
Change to italics	under matter to be changed	
Change to capitals	under matter to be changed	
Change to small capitals	under matter to be changed	
Change to bold type	under matter to be changed	
Change to bold italic	under matter to be changed	
Change to lower case	Encircle matter to be changed	
Change italic to upright type	(As above)	
Change bold to non-bold type	(As above)	
Insert 'superior' character	through character or where required	or under character e.g. or
Insert 'inferior' character	(As above)	over character e.g.
Insert full stop	(As above)	
Insert comma	(As above)	
Insert single quotation marks	(As above)	or and/or or
Insert double quotation marks	(As above)	or and/or or
Insert hyphen	(As above)	
Start new paragraph		
No new paragraph		
Transpose		
Close up	linking characters	
Insert or substitute space between characters or words	through character or where required	
Reduce space between characters or words	between characters or words affected	