HAL open science

Robots' Cooperation for Finding a Target in Streets

Mohammad Abouei Mehrizi, Mohammad Ghodsi, Azadeh Tabatabaei

To cite this version:

Mohammad Abouei Mehrizi, Mohammad Ghodsi, Azadeh Tabatabaei. Robots' Cooperation for Finding a Target in Streets. 1st International Conference on Theoretical Computer Science (TTCS), Aug 2015, Tehran, Iran. pp.30-43, 10.1007/978-3-319-28678-5_3 . hal-01446262

HAL Id: hal-01446262

https://inria.hal.science/hal-01446262v1

Submitted on 25 Jan 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Robots' Cooperation for Finding a Target in Streets

Mohammad Abouei Mehrizi, Mohammad Ghodsi, and Azadeh Tabatabaei

[1] Department of Computer Engineering, Sharif University of Technology
abouei@ce.sharif.edu
[2] Sharif University of Technology and School of Computer Science Institute for Research in Fundamental Sciences (IPM) ghodsi@sharif.edu
[3] Department of Computer Engineering, Sharif University of Technology
atabatabaei@ce.sharif.edu

Abstract. We study the problem of finding a target t from a start point s in street environments with the cooperation of two robots which have a minimal sensing capability; that is, robots do not know any information about the workspace including information on distances, edges, coordinates, angles etc. They just can detect the discontinuities in the visibility region of their location. The robots can detect target point t as soon as it enters their visibility region and have communication peripherals to send messages to each other. Our aim is to minimize the length of the path passed by the robots. We propose an online algorithm for robots such that they move in the workspace and find the target. This algorithm generates a search path from a start point s to a target point t such that the distance traveled by the robots is at most 2 times longer than the shortest path. Also, we prove that this ratio is tight.

Keywords: Computational Geometry, Visibility, Motion Planning, Minimal Sensing, Multi Robot's Cooperation.

1 Introduction

The problem of finding a target in an environment is a fundamental problem in computational geometry and robotics [5,7,10]. This problem is known as exploration which appears in many applications where the environment is unknown and no geometric map of the scene is available. In this problem, the robot's sensors are the only input device to gather information. Depending on the ability of the sensors, information will be exact. Simple sensor robots have many benefits such as low cost, less sensitivity to failure, stable against noise etc [3].

In this research robots have a limited ability in sensing the environment. They have an abstract sensor to detect discontinuities of the work space (called gaps) in their visibility region. Discontinuity means a portion of the environment that is not visible to the robot (Figure 1). The robot can assigning a label L or R to every gap g. When robot is scanning the environment counter clockwise, if a gap is sensed such that discontinuity is near to far then gap is a right gap

(labeled by R), otherwise it is a left gap (labeled by L). In Figure 1 (a) A and B are right gaps while C and D are left gaps. Also, the robots are capable of recognizing a target point t when it is in the robots' visibility region.

Robots do not need any data structure to store the topological map of the environment and just store some fixed data for navigation. They have to move towards the gaps to cover the hidden region behind each gap. Note that the robots do not have to measure any angles or distances to the walls of the scene or infer their position but have an infinite and omnidirectional field of view. Robots' cooperation will be done by sending messages to each other. Cooperation is necessary for some situation. For example, when one robot find out its path is along the shortest path (from s to t), it send a message and help the other robot to find the correct path. Messages are some kind of data known for both robots. In this paper, the workspace is assumed to be a restricted simple polygon called a street.

Definition 1. *A simple polygon P with two vertices s and t is called a street if the counter-clockwise polygonal chain R_{chain} from s to t and the clockwise chain L_{chain} from s to t are mutually weakly visible [6].*

Definition 1 means each point on the left chain L_{chain} must be visible from at least one point on the right chain R_{chain} and vice versa (Figure 1). A street polygon is also known as L-R visible polygon [2].

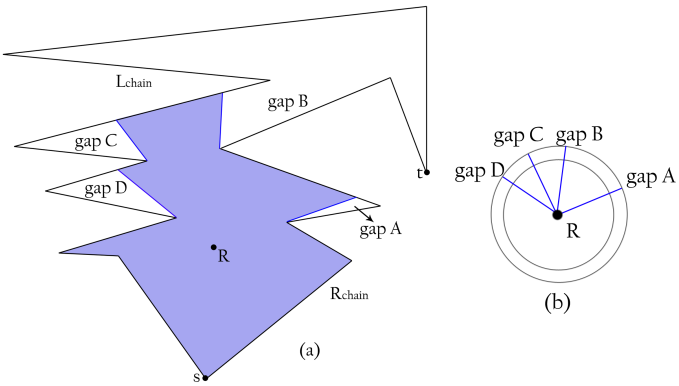

Fig. 1. (a) A street in which L_{chain} is the left chain and R_{chain} is the right chain. The colored region is the visibility polygon of the point robot R in the street. (b) The position of discontinuities in the depth information detected by the sensor. Note that robot cannot detect the angular information of gaps, it just can detect the position of gap and move toward them.

This paper will present an algorithm such that two point robots that are equipped with a gap sensor and communication device start navigating from s

to reach target t. The robots have no geometric map of the scene and have to make decisions to achieve the target only based on the information gathered through the sensor and received messages.

This paper is organized as follows. Section 2 will review related works, Section 3 will present structures and the model used; Section 4 will present preliminaries. Section 5 will present the algorithm, and Section 6 will present the conclusions.

2 Related Works

Because of application and usefulness of path finding in online problems, it has recently received much attention from researchers. Klein proposed the first competitive online strategy for searching a target point in a street [6]; The robot employed in [6] is equipped with a 360 degree vision system. Also, it can measure each angle or distance to the walls of the street. As the robot moves, a partial map is constructed from what has been seen so far. Klein proved an upper bound of 5.72 for the competitive ratio (the ratio of the length of the traversed path to the shortest path from s to t) of this problem. Also, it was proved later that there is no strategy with the competitive ratio less than $\sqrt{2}$ for this problem.

A strategy similar to Klein's with the competitive ratio of $\pi + 1$ has been introduced in [8,9] which is robust under small navigation errors. Other researchers have presented several algorithms with the competitive ratios between $\sqrt{2}$ and the upper bound of 5.72 [7,9]. Icking et al. presented an optimal strategy with the competitive ratio of $\sqrt{2}$ [5].

The limited sensing model that we use in this paper was first introduced by Lavalle et al. [13]. Gap Navigation Tree (GNT) has been proposed to maintain and update the gaps seen along the navigating path. This tree is built by detecting the discontinuities in the depth information and updated by the topological changes of the information. The topological changes are appearances, disappearances, merges, and splits of gaps.

Another minimal sensing model was introduced by Suri and Vicari [11] for a simple robot. They assume that the robot can only sense the combinatorial (non-metric) properties of their surroundings. The sensor can detect vertices of the polygon in its visibility region, and can report if there is a polygon edge between consecutive vertices. The information is maintained in two combinatorial vectors, called the combinatorial visibility vector (cvv) and the point identification vector (piv). Despite minimal capability, they show that the robot can obtain enormous geometric reasoning and can accomplish many non-trivial tasks.

Tabatabaei and Ghodsi [23] presented an online algorithm to find the target point t from the start point$s$ with one minimal sensing robot in the street environment. Beacuse their model is minimal sensing their problem is more difficult than Kleins problem. Their strategy is doubling to find the target such that the robot will move 11 times as long as the shortest path with just one pebble and if the robot has access to many pebbles, this ratio reduces to 9. In the doubling strategy, the robot moves back and forth on the line, such that at each stage i, it walks 2^i steps in one direction, comes back to the origin, walks 2^{i+1} steps

in the opposite direction until the target is reached. Also pebble is some thing portable and distinguishable by the robot. Showing that this ratio is optimal in this model, they presented an optimal strategy for walking in streets with minimum number of turns [14].

Path finding for multiple robots where paths are collision-free is still an interesting and difficult problem which has been studied for many years [19,20,21,22]. Jingjin Yu and Steven M. LaValle reported a reduction from a special case of multi agent path planning to network flow [18]. They introduced a multiflow based ILP algorithm for planning optimal, collision-free paths for multiple robots on graphs, introducing complete ILP algorithms to solve time optimal and distance optimal multi-robot path planning problems [16]. They also proved that computing a minimum total arrival time, a minimum makespan, or a minimum total distance solution is NP-complete for some kinds of cooperation multi-robot path planning [17].

Luna and Bekris proposed an algorithm that guarantees completeness for a general class of problems without any assumptions about the graph's topology. Their algorithm required two primitive operations: push where agents move towards their goals and do this to arrive a point where no progress can be made. swap allows two agents to swap their positions. Their approach can address any solvable instance in a graph of size n where there are at most $n-2$ agents [20].

Mikael Hammar et al. [15] have presented constant competitive strategies to explore a rectilinear simple polygon in the L_1 metric with one or more robots. They also proved there is no deterministic strategy for path exploration of a rectilinear polygon with one robot that has competitive ratio $2-\epsilon$ for any $\epsilon > 0$. They also showed there are no deterministic strategies for exploring a rectilinear polygon with two or three robots having smaller competitive ratio than $\frac{2}{3}$ but their results are not tight.

In this paper we propose an online search strategy for two point robots equipped with the gap sensor and a communication peripheral like Bluetooth, Wi-Fi with wireless antenna, etc with respect to expansion of environment. The aim of the algorithm is to minimize the minimum path length of the two robots from the start point s to the target point t in a street workspace. We will show the minimum search path which is generated by the strategy is at most 2 times as long as the shortest path that is tight for this situation. To our knowledge, this is the first result providing some competitive ratio for two robots walking in streets in a minimal sensing model.

3 The sensing model and communications

3.1 Gap sensor

Gap sensor used in this paper is a visual sensing model. At any position q of the environment, a cyclically ordered location of the depth discontinuities in the visibility region of the point is what the robot's sensor detects, as shown in (Figure 1). When the robot reports the discontinuities counterclockwise from a

visibility region, it assigns a left label to a transition from far to near and assigns a right label to a transition from near to far [12] (Figure 2). The robot can only walk towards the gaps, this restriction cannot get us into trouble and despite this constraint we will present a 2-approximation algorithm.

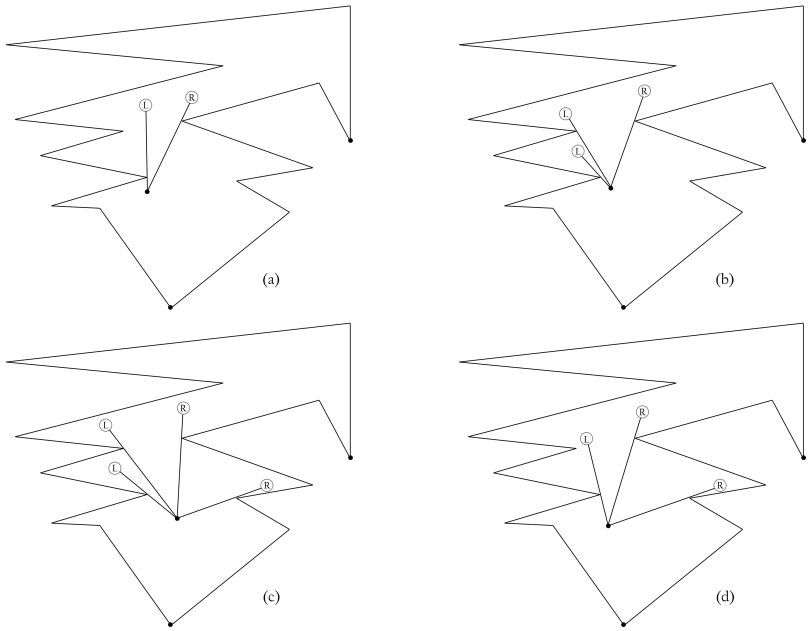

Fig. 2. The dark circle denotes the location of the robot. (a) Existing gaps at the beginning. (b) A split event. (c) An appearance event. (d) A merge event.

GNT data structure has been introduced as a means for the robot system to navigate in an unknown scene [12]. This structure is bassed on four critical events. In the strategy proposed here, the robots do not need to store GNT; they just pay attention to the events which will change the number of gaps. Each robot just stores two special gaps (later will discuss about these gaps) and sets its orientation to one of them and moves toward it until a critical event occurs.

3.2 Communications

In the algorithm (presented in Section 5) we will use two robots to find the target. To achieve a good ratio, robots are equipped with communication peripherals for sending data together. The type of communication device depends on whether the expansion of workspace can be detected. For example, if robots will not have more than a 100 meter distance, we can use Bluetooth as the communication

device. If more than 100 meters is needed, we can use Wi-Fi devices and wireless network antenna prepared in the workspace.

In this research we will not explain communication methods and their potential difficulties, like message failure. We assume that robots can have communication together and send messages to each other as well as possible.

3.3 Motion Primitive

Robots in their movement can sense gaps and their labels (right or left) in counter-clockwise order. Gaps' labels are not unique and just show that the gap is a left gap (L) or a right one (R) (Figure 2). These gaps may change during the robots' movement. Robots can set their direction to a gap and move toward it. During walking they cannot detect geometric properties of the environment. They can only walk to gaps until a critical event occurs or the gap disappears; then they will make a decision and find a direction for movement and walk around the workspace. Also, robots can detect and move toward target t as soon as it enters the visibility region.

4 Preliminaries

In this paper we use the notation used in [14]; some of the definitions and lemmas that will be used in our strategy are mentioned here.

At each point p of the robots' search paths, their sensor either sees the target or achieves a set of gaps with the label of L or R (*l-gap* and *r-gap* for abbreviation). If the target is visible, the robot moves towards the target to reach it. When the robots report the position of the gaps, they should move towards the gaps to find the target.

Definition 2. *In the set of l-gaps, the gap that is on the right side of the others is called most-advanced left gap and is denoted by g_l. Analogously, in the set of r-gaps, the gap that is on the left side of the others is called most-advanced right gap and is denoted by g_r [14].*

Note: Accourding to the Definition 2, we know in visibility region of robots, all of the right gaps are near together and then all of the left gaps are near to each other (counter clockwise). Definition 2 means when robot is scanning its visibility region counter clockwise, the last right gap is called most-advanced right gap and the first left gap is called most-advanced left gap. In any street polygon (circular, general etc.) this is correct.

Lemma 1. *At any point of the robot's search paths, if the target is not visible, then it is behind one of the most-advanced gaps [14].*

As the robots move in the environment, g_l and g_r may dynamically change. The critical events in which the structure of the robot's visibility region changes can also change g_l and g_r. In the next section, we show how the critical events

change the left most advanced gaps such that a sequence of the left most advanced gaps, $[g_{l1}, g_{l2}, ..., g_{lm}]$, appears in the robots' visibility region, while exploring the street. Similarly, a sequence of the right most advanced gaps, $[g_{r1}, g_{r2}, ..., g_{rn}]$, may occur.

At each point, if there is exactly one of the two gaps (g_l or g_r), then the goal is hidden behind that gap. Thus, there is no ambiguity and the robots move toward the gap. If both g_r and g_l exist, then the target is hidden behind one of these gaps. This case is called a *funnel*. As soon as a robot enters a point in which both g_r and g_l exist, a funnel situation starts. This case continues until one of g_l or g_r disappears or until they become collinear. When a robot enters a point in which there is a funnel situation, it will wait for the other robot. In this situation the only non-trivial case in this navigation occurs. Before describing our strategy, we state some features of a street and the gaps that are applied in the algorithm.

When the robot enters a funnel situation, there are two convex chains in front of it: the left convex chain that lies on the left chain (L_{chain}) of the street, and the right convex chain that lies on the right chain (R_{chain}) of the street. The two chains have the following main property.

Lemma 2. *When a funnel situation starts, the shortest path from s to t lies completely on the left convex chain, or on the right convex chain of the funnel [14].*

Theorem 1. *For any vertex $v_j \in L_{chain}(or, v_j \in R_{chain})$, the shortest path from s to v_j makes a left turn (respectively, a right turn) at every vertex of L_{chain} (respectively, R_{chain}) in the path [4].*

Lemma 3. *Each of the two convex chains, in a funnel situation, contains a point at which the funnel situation ends or a new funnel situation starts [14].*

Note: Lemmas 1, 2 and 3 are about the features of environment (street polygon) and are not related to the robots[4].

5 Algorithm

In this section we explain our movement strategy for the robots to move in the environment (street polygon) from s to t. We use R_1 for the first robot and R_2 for the second one. Note that both robots start at point s, and if R_1(respectively, R_2) achieves t, it will send a message (*target-founded*) to the R_2(respectively, R_1); when R_2(respectively,R_1) receives the message, it stops searching. This moment represents the end of movement for both robots.

Obviously, this strategy minimizes the robots' search paths. That is, our aim is:

$$Minimize : Min\{path_1, path_2\}$$

[4] Capabilities or number of robots does not have any relation to these lemmas.

Such that $path_1$ is the length of the path traversed by R_1 and $path_2$ is the length of the path traversed by R_2 from s to t.

As mentioned, since our sensing model is minimal, then robots decide to move in the environment based on the information gathered by their sensor. The intuition behind the algorithm is when robots are in the funnel situation, then according to Lemma 1, when the target is visible from neither R_1 nor R_2, then the target is behind one of the most-advanced gaps. Because robots do not know which gap is correct, then R_1 will move to g_l and R_2 will move to g_r; since we refer to R_1 as R_{left} and R_2 as R_{right}. If there is just one of g_l or g_r, for each robot, it will get moving toward that gap.

In the set of *l-gaps*, the gap which is on the right side of the others for R_{left} is called *the most left robot advanced left gap* and is denoted by g_{ll}; In the set of *r-gaps*, the gap which is on the left side of the others for R_{left} is called *the most left robot advanced right gap* and is denoted by g_{lr}. Analogously, In the set of *l-gaps*, the gap which is on the right side of the others for R_{right} is called *the most right robot advanced left gap* and is denoted by g_{rl}; in the set of *r-gaps*, the gap which is on the left side of the others for R_{right} is called *the most right robot advanced right gap* and is denoted by g_{rr} (Figure 3).

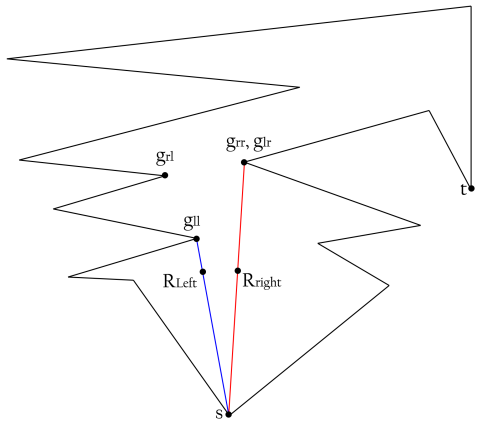

Fig. 3. The red line is the direction of R_{right} and blue line is the direction of R_{left} from the start point s. R_{right} moves to g_{rr} and R_{left} moves to g_{ll}. In the visibility region of R_{right}, gaps are different from those in the visibility region of R_{left} .

We know in each funnel situation either R_{left} or R_{right} passes the shortest path. If there is just one advanced gap (g_l or g_r), the target is behind it, and then both robots move toward it. If a robot finds out that its path is along the shortest path, it will send a message to the other robot and the other will correct its direction.

5.1 Critical Events

In any funnel situation, both robots start from one point and search the workspace. During the movement, some critical events may occur for each robot. Critical events usually occur when the visibility region of a robot changes such that it makes a change to the direction of robots. These events may happen for either R_{left} or R_{right} and change their direction. Events that may occur while the robots are moving and the corresponding operations are mentioned bellow.

i When the R_{right} crosses a bitangent complement of g_{rl} and an *r-gap*, then g_{rr} will be replaced by the *r-gap* (Figure 4.a).
ii When the R_{left} crosses a bitangent complement of g_{lr} and an *l-gap*, then g_{ll} will be replaced by the *l-gap* (Figure 4.b).
iii When the R_{right} crosses a bitangent complement of g_{rl} and another *l-gap*, then g_{rl} splits and will be replaced by the *l-gap* (Figure 4.c).
iv When the R_{left} crosses a bitangent complement of g_{lr} and another *r-gap*, then g_{lr} splits and will be replaced by the *r-gap* (Figure 4.d).
v When the R_{right} crosses a bitangent of g_{rr} and another *r-gap*, at the point in which g_{rr} disappears, g_{rr} will be replaced by the *r-gap* (disappearance and split events occur simultaneously). In this situation, if there are more than one gap similar to the *r-gap*, g_{rr} will be replaced by the one which is on the left side of the others (Figure 4.e).
vi When the R_{left} crosses a bitangent of g_{ll} and another *l-gap*, at the point in which g_{ll} disappears, g_{ll} will be replaced by the *l-gap* (disappearance and split events occur simultaneously). In this situation, if there are more than one gap similar to the *l-gap*, g_{ll} will be replaced by the one which is on the right side of the others (Figure 4.f).
vii When the R_{left}/R_{right} crosses over an inflection ray, each of g_{ll}/g_{rl} or g_{rl}/g_{rr} which is adjacent to the ray, disappears and is eliminated from the detected gaps automatically.

If R_{right} goes to g_{rr} and event i occurs there will be two different situations (let $r_{right} \in$ *r-gap* be the right gap that event i occured for it):

- $\| R_{right} - g_{rl} \| < \| R_{right} - r_{right} \|$. This situation is shown in Figure 4 (a), after replacing g_{rr}, R_{right} will change its direction to g_{rl} (g_{rl} and g_{rr} are collinear) and keep going until it touches it. When R_{right} touches g_{rl}, there may occur some critical events that update g_{rr} or/and g_{rl}. Then R_{right} will continue its routing by going toward g_{rr}.
- $\| R_{right} - g_{rl} \| > \| R_{right} - r_{right} \|$. In this situation R_{right} is shown in Figure 5 after replacing the g_{rr}, like before, R_{right} will change its direction to g_{rl} (g_{rl} and g_{rr} are collinear) and keep going until it touches it. But when R_{right} goes to g_{rl} there will occur two different events.
  - g_{rr} disappears and there is no split event (no new *r-gap* (g_{rr}) will be created): in this situation R_{right} moves toward to g_{rl}) until an split event occurs and g_{rr} is found (Figure 5 (a)).

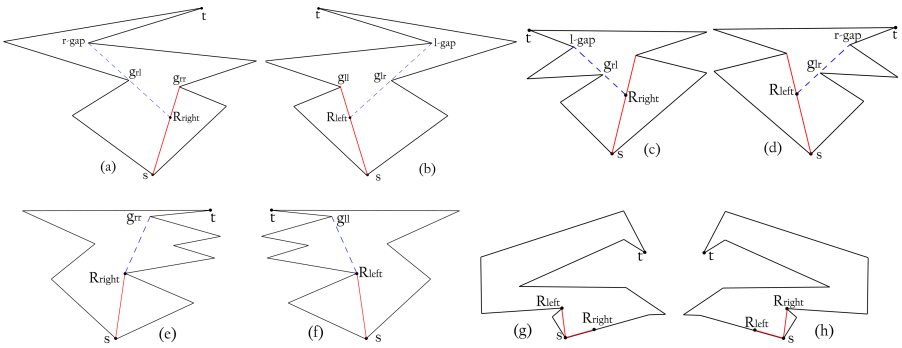

Fig. 4. Critical events will occur during the robots movement. (a) R_{right} finds a new *r-gap* in its visibility region which is on left side of g_{rr}. (b) R_{left} finds a new *l-gap* in its visibility region which is on right side of g_{ll}. (c) Split event occurs for *l-gaps* of R_{right}. (d) Split event occurs for *r-gaps* of R_{left}. (e) Split event occurs for *r-gaps* of R_{right}. (f) Split event occurs for *l-gaps* of R_{left}. (g) Critical message event occurs for R_{right}; when R_{left} arrives at point g_{ll}, g_{ll} disappears and a new right gap will appear. It finds out that its path is along the shortest path and sends *bad-direction* message to R_{right}, and R_{right} corrects its direction. (h) Like (g) critical message event occurs for R_{left}.

- When the robot arrives at g_{rr}, event v happens and g_{rr} will update. In this situation R_{right} will move toward g_{rl} until touches it. Then R_{right} will keep going to the g_{rr} (Figure 5 (b)).
  Note: In Figure 5 (b) just two gaps are visible at start point s, because robots are capable of detecting the discontinuities of environment and other reflex vertices do not create any gap, then robots cannot distinguish them. When R_{right} is going to g_{rr} no critical event occur (appearance a new gap is not a critical event) until R_{right} crosses bitangent complement of g_{rl} and an *r-gap* (event i), then the above situation will be done.

When R_{left} goes to g_{ll} and event ii occurs, R_{left} will act like R_{right} as mentioned above.

Note: When R_{right} crosses a bitangent complement of g_{rl} and an *r-gap*, it knows that it has to change its direction to g_{rl} and it does not need to measure any distance; It just change its direction to g_{rl} and keep going, during movement some critical events will happen and R_{right} can update its important gaps (g_{rl}, g_{rr}).

5.2 Message Events

These events occur when a robot receives a message from the other, then it will manipulate it and do corresponding operations. These message events are mention here.

1. When the R_{left} crosses a bitangent of g_{ll} and an *r-gap*, at the point in which g_{ll} disappears and the *r-gap* appears due to split event (disappearance and

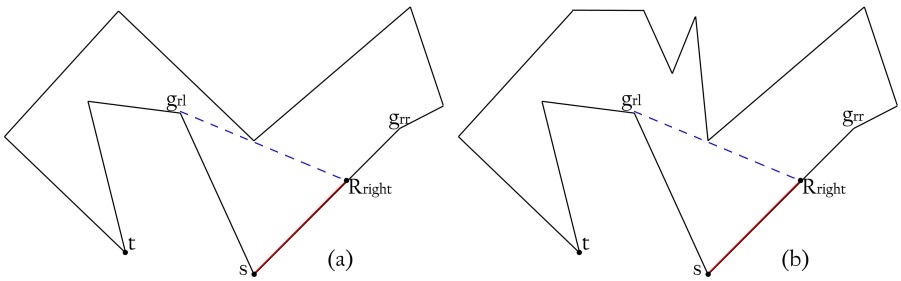

Fig. 5. R_{right} crosses the bitangent of g_{rl} and an *r-gap* such that $\| R_{right}$ - *r-gap*$\| < \| R_{right} - g_{rl} \|$. (a) When R_{right} goes to the g_{rl} there will not be any right gap. (b) When R_{right} goes to g_{rl} there will be right gap and g_{rr} is found.

split events occur simultaneously), R_{left} sends a bad-direction message to R_{right} , then R_{right} will change its direction to g_{rl} (Figure 4.g).

2. When the R_{right} crosses a bitangent of g_{rr} and an l-gap, at the point in which g_{rr} disappears and the *l-gap* appears, R_{right} sends a *bad-direction* message to R_{left} , then R_{left} will change its direction to g_{lr} (Figure 4.h).

Both robots will keep going until the current funnel situation ends and arrives at a point which is the start of next funnel situation. Then the earlier robot will send a *funnel-situation* message to another one and waits there until the other robot arrives. When a robot receives a *funnel-situation* message it will keep going until it arrives at the new funnel situation, then it sends a *start-funnel* to the other robot and both do this algorithm for new funnels until finding the target t.

From the start point of any funnel situation, R_{left} just saves g_{ll}, g_{lr} and moves toward g_{ll}; and R_{right} just saves g_{rr},g_{rl} and moves toward g_{rr}. Note that in the start point s, the gaps coincide for both robots, but when they move in the environment, their gaps may be different from each other (Figure 3). R_{left} will keep going to g_{ll} and R_{right} will keep going to g_{rr} all of the time, as often as each of the following events occurs (robots will ignore the other events):

- When one of the critical events that are mentioned in 5.1 occurs.
- When one of the message events that are mentioned in 5.2 occurs.

Note: Some other events may occur during robots' movement, but they are not important for us. It means, robots just pay attention to mentioned events in Section 5.1 and 5.2, robots will ignore the other events.

5.3 Analysis

Lemma 4. *In the strategy mentioned in Section 5, robot R_{right}/R_{left} will change its direction only when the robot crosses bitangent of g_{rl}/g_{lr} and an r-gap/l-gap, or receives a message event.*

Proof. Directly concludes from the critical and message events presented in Section 5.1 and 5.2.

Lemma 5. *Assume both robots are in point s and are in the funnel situation, R_{left} moves toward g_{ll} and R_{right} moves toward g_{rr}, if R_{left}/R_{right} passes along the shortest path to arrive at g_{ll}/g_{rr}, then R_{right}/R_{left} will pass a detour with the length of at most $3 \parallel s - g_{ll}/g_{rr} \parallel$ to arrive at g_{ll}/g_{rr} .*

Proof. According to Lemma 2 either R_{left} or R_{right} passes along the shortest path. Without loss of generality, assume R_{left} passes along the shortest path and R_{right} goes in a bad direction. Let $a = \parallel s - g_{ll} \parallel$, we know R_{left} after traversing a distance of length a, understands that it has passed the shortest path and sends *bad-direction* message to R_{right}, R_{right} passes the distance a (like R_{left}), in the worst case R_{right} has to come back to the point s and then moves to g_{ll}. Then it will pass at most $3 \parallel s - g_{ll} \parallel$. This is true when R_{right} passes the shortest path and R_{left} goes in a bad direction.

Both robots will pass a path with at most 4 times as long as the shortest path because in any funnel situation one robot is along the shortest path and the other will pass a detour at most 3 times as long as the shortest path (according to Lemma 5).

Theorem 2. *$Min\{path_1, path_2\} \leq 2path_{optimal}$, such that R_{left} passes along $path_1$, R_{right} passes along $path_2$ and $path_{optimal}$ is shortest path from s to t.*

Proof. let α be the length of the shortest path from s to t. We know in any funnel situation one robot is along the shortest path. To maximize the length of $Min\{path_1, path_2\}$ we should take $\mid path_1 \mid = \mid path_2 \mid$ and R_{left} passes at most $\frac{\alpha}{2}$ along the shortest path and detours $\frac{3\alpha}{2}$. Then R_{left} passes at most $\frac{\alpha}{2} + \frac{3\alpha}{2} = 2\alpha$. This is the case for R_{right} too; in Figure 6, a tight example for $2 - \epsilon (\epsilon > 0)$ ratio is shown.

6 Conclusion

In this research, we proposed an online algorithm for two simple robots, such that they find the target t from the start point s. Minimal sensing means that robots do not have any sense from the environment and can only detect the discontinuities (known as gap), and move to their directions. Also, the robots can detect the target t, as soon as it enters their visibility region. We prepared an algorithm to find target such that there exists one robot whose path from s to t is at most 2 times as long as the shortest path; this ratio is tight. Introducing more general classes of polygons which admit competitive searching with minimal sensing multi robots is an interesting open problem.

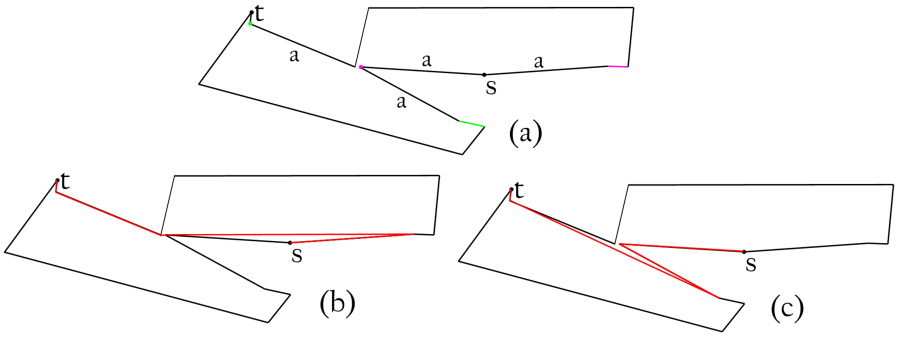

Fig. 6. (a) The workspace is shown, the pink segment on the R_{chain} is visible just from the pink point on the L_{chain} and the green segment on the L_{chain} is visible just from the green point on the R_{chain}. The shortest path from s to t has a length of $2a + \epsilon_0$. (b) The red line shows the path of R_{right} according to the presented algorithm. As we can see, R_{right} will pass a path with almost $4a + \epsilon_0 - \epsilon_1$ length. (c) The red line shows R_{left} path according to the presented algorithm. As we can see, R_{left} will pass a path with almost $4a + \epsilon_0 - \epsilon_2$ length too. Then the algorithm is tight for ratio $2 - \epsilon(\epsilon > 0)$.

References

1. Baezayates, R. A., Culberson, J. C., Rawlins, G. J.: Searching in the plane. Information and Computation, 106(2), 234-252 (1993).
2. Das, G., Heffernan, P. J., Narasimhan, G.: LR-visibility in polygons. Computational Geometry, 7(1), 37-57 (1997).
3. Gfeller, B., Mihalak, M., Suri, S., Vicari, E., Widmayer, P.: Counting targets with mobile sensors in an unknown environment. In: Algorithmic Aspects of Wireless Sensor Networks (pp. 32-45). Springer Berlin Heidelberg (2008).
4. Ghosh, S. K.: Visibility algorithms in the plane. Cambridge University Press (2007).
5. Icking, C., Klein, R., Langetepe, E.: An optimal competitive strategy for walking in streets. In: STACS 99 (pp. 110-120). Springer Berlin Heidelberg (1999, January).
6. Klein, R.: Walking an unknown street with bounded detour. In: Foundations of Computer Science, 1991. Proceedings., 32nd Annual Symposium on (pp. 304-313) (1991, October).
7. Kleinbergt, J. M.: On-line search in a simple polygon. In: Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms (No. 70, p. 8). SIAM (1994).
8. Lopez-Ortiz, A.: On-line target searching in bounded and unbounded domains. University of Waterloo (1996).
9. Lopez-Ortiz, A., Schuierer, S.: Simple, efficient and robust strategies to traverse streets. In Proc. 7th Canad. Conf. on Computational Geometry (1995).
10. Mitchell, J. S.: Geometric shortest paths and network optimization. Handbook of computational geometry, 334, 633-702 (2000).
11. Suri, S., Vicari, E., Widmayer, P.: Simple robots with minimal sensing: From local visibility to global geometry. In: The International Journal of Robotics Research, 27(9), 1055-1067, (2008).
12. Tovar, B., Murrieta-Cid, R., LaValle, S. M.: Distance-optimal navigation in an unknown environment without sensing distances. Robotics, IEEE Transactions on, 23(3), 506-518, (2007).

13. Tovar, B., LaValle, S. M., Murrieta, R.: Optimal navigation and object finding without geometric maps or localization. In: Robotics and Automation, 2003. Proceedings. ICRA'03. In:IEEE International Conference on (Vol. 1, pp. 464-470), (2003, September).
14. Tabatabaei, A., Ghodsi, M.: Optimal Strategy for Walking in Streets with Minimum Number of Turns for a Simple Robot. In: Combinatorial Optimization and Applications (pp. 101-112). Springer International Publishing, (2014).
15. Hammar, M., Nilsson, B. J., Persson, M.: Competitive exploration of rectilinear polygons. In: Fundamentals of Computation Theory (pp. 234-245). Springer Berlin Heidelberg, (2003, January).
16. Yu, J., LaValle, S. M.: Planning optimal paths for multiple robots on graphs. In: Robotics and Automation (ICRA), 2013 IEEE International Conference on (pp. 3612-3617), (2013, May).
17. Yu, J., LaValle, S. M.: Structure and Intractability of Optimal Multi-Robot Path Planning on Graphs. In: AAAI, (2013, June).
18. Yu, J., LaValle, S. M.: Multi-agent path planning and network flow. In: Algorithmic Foundations of Robotics X (pp. 157-173). Springer Berlin Heidelberg, (2013).
19. Erdmann, M., Lozano-Perez, T.: On multiple moving objects. Algorithmica, 2(1-4), 477-521, (1987).
20. Luna, R., Bekris, K. E.: Push and swap: Fast cooperative path-finding with completeness guarantees. In: IJCAI (pp. 294-300), (2011, July).
21. Ryan, M. R. K.: Exploiting subgraph structure in multi-robot path planning. Journal of Artificial Intelligence Research, 497-542, (2008).
22. Silver, D.: Cooperative Pathfinding. In AIIDE (pp. 117-122), (2005, June).
23. Tabatabaei, A., Ghodsi, M.: Walking in Streets with Minimal Sensing. Journal of Combinatorial Optimization `http://link.springer.com/article/10.1007/s10878-014-9791-4`, (2014)