This paper is presented at International Conference on Computer, Information Technology and Digital Media (CITaDim)At: Tehran, Iran .M. Eskandari, Y. Bahoo (2013) Blocking an Evader in a Polygon by a 2-modem Searcher.Zenodo.http://doi.org/10.5281/zenodo.4827646

DOI 10.5281/zenodo.4827646

Blocking an Evader in a Polygon by a 2-modem Searcher

M. Eskandari*, Y. Bahoo**
*Department of Mathematics, Alzahra University, eskandari@alzahra.ac.ir
**Department of Mathematics and Computer Science, Amirkabir University of Technology, bahoo@aut.ac.ir

Abstract: *In this paper, we introduce a version of Pursuit-Evasion problem in which the pursuer is a 2-modem which pursues an unpredictable evader in a polygonal environment. A 2-modem searcher is a wireless device whose radio signal can penetrate two walls. Furthermore, a query path inside the polygon is given so that the 2-modem moves on it. We want to find out that when the 2-modem moves on this path if evaders can move from an invisible region to another one without detecting by the pursuer or not. In the other word, we want to find out if the invisible regions merge together or not.*

Keywords: *Computational Geometry, Pursuit-Evasion, Visibility, 2-modem.*

1. Introduction

We introduce a new version of Pursuit-Evasion problem in which the pursuer is a 2-modem which pursues an unpredictable evader in a polygonal environment. Consider a simple polygon P is given and there are some evaders and a pursuer which moves continuously in it. The classical Pursuit-Evasion problem asks for planning the motion of the pursuer in a polygon to eventually see an evader. During the motion of the pursuer, some parts of polygon may be invisible for him; these invisible regions completely depend on the type of the pursuer and its position in P. Let p be an arbitrary point in P (as an initial position of the searcher). A maximal connected closed set of points inside P which are invisible for p is called a shadow of p. Actually the shadows of p are subpolygons of P. If the cardinality of the set of shadows of p (when the pursuer is located on p) is k, we denote these shadows by $S_i(p)$; $1 \leq i \leq k$. See Figure 1.

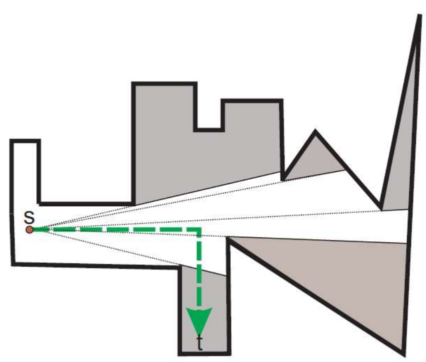

Fig. 1: The gray regions are the shadows of a given point s (an omnidirectional searcher -which can see all of around itself- is located on s).When the omnidirectional searcher moves continuously from s to t on a given path (which is presented by a dashed path), the shadows change.

As shown in [1], when the searcher moves continuously inside P, four geometric events may happen for its shadows: *merge*, *split*, *appear* and *disappear*. Moreover, when two disjoint shadows of p merge together and make one connected subpolygon, it is called the *merge* event. In contrast, when a shadow is divided into two components during the motion of the searcher, it is called the *split* event. Sometimes a shadow is destroyed when the searcher moves; this event is known as the *disappear* event and if a new shadow is created, we call it the *appear* event.

In [2], Guibas et al. studied the problem of maintaining the distribution of evaders that move out of view and inferring the location of these targets from combinatorial data extracted by searchers. In this paper, we consider a special type of searchers, 2-modems. As it is defined in [3], we call a wireless device whose radio signal can penetrate two walls, a 2-modem. A path inside P is given so that the 2-modem moves on it. In this paper, we want to find out if the *merge* event happens during the movement. The study of this problem is motivated by filtering perspective and robotics applications such as surveillance, as explained in [2]. At first according to the cell-decomposition which is introduced in [4], we introduce some candidate lines that crossing them may cause the *merge* event. In the next section, we will characterize exact lines which cause the *merge* event among candidate lines. Finally, we will propose an algorithm for reporting the *merge* event during the movement of 2-modem on a given path.

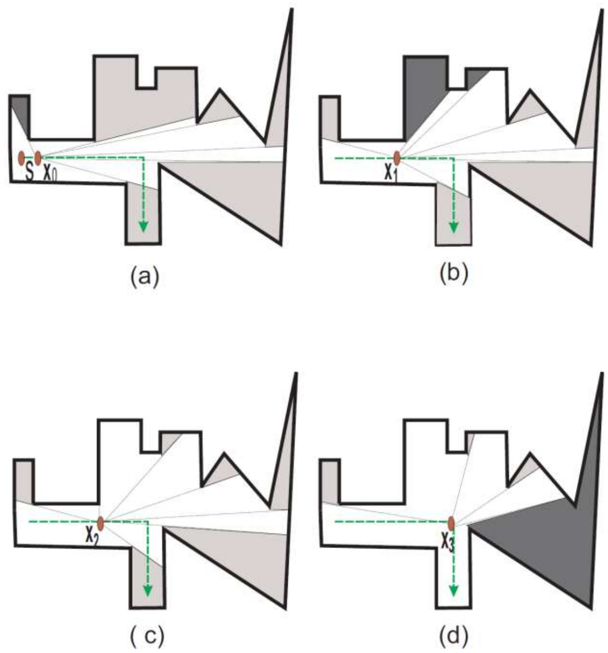

Fig. 2: The small disc represents the omnidirectional searcher (the searcher that can see all of around itself at the same time). (a) when a searcher moves from s to x_0, the black shadow will be appear (the *appear* event), (b) when it moves from x_0 to x_1, one shadow splits into two black shadows (*split* event), (c) when an omnidirectional searcher moves from x_1 to x_2, one shadow disappears (*disappear* event), (d) when it moves from x_2 to x_3, two different shadows merge together (*merge* event).

2. Preliminaries

In this section, we introduce some candidate lines that crossing them may cause the *merge* event. For this purpose, we will review the cell decomposition for 2-modem which is introduce in [4].

Definition: Let v and u be two vertices of P. The vertex u is a critical vertex for v if both of its edges are in the same half-plane bounded by the line uv.

This cell decomposition is constructed by three types of lines which are called *partition lines*:
i) For each reflex vertex of P, the extensions of its edges are drawn until they hit the boundary of P.
ii) For each pair of reflex vertices u and v which are critical for each other and $uv \in P$, we draw the line through uv until it hits ∂P and then omit uv.
iii) For each vertex v of P, we construct the 2-visibility polygon of v as defined in [3].

This cell decomposition guarantees when the 2-modem moves continuously inside a cell the combiniatorial representation of its shadows remains unchanged. In other word, while moving in a cell the *appear*, *disappear*, *merge* and *split* events may not occur. Consequently when the 2-modem transfers from one cell to another, the *appear*, *disappear*, *merge* and *split* events may happen. It means that crossing one of these partition lines may change the combinatorial representation of shadows. Also according to the Observation 1 in [4], crossing the third type of partition lines may cause the *merge* event. As we mentioned above, for constructing the third type of lines, for each vertex of the polygon P we construct the 2-visibility polygon. The 2-visibility polygon of a vertex $v \in P$ is a subpolygon of P which is visible by a 2-modem lied on v. For constructing this subpolygon, we have to consider a part of the line vu ($u \in P$) so that u is a critical vertex for the vertex v. For illustration of the lines of type 3 which are in the 2-visibility polygon of the vertices, we provide some examples. Let u be an arbitrary vertex of P and v be a critical vertex of u. In Figure 3, we illustrate the parts of the ray uv which is drawn in 2-cell decomposition by the bold pieces.

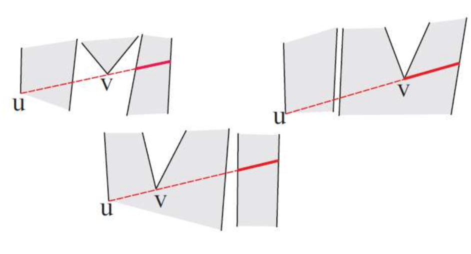

Fig. 3: The bold parts are drawn in 2-cell decomposition.

3. Characterizing the Partition Lines

Now we want to characterize the precise lines that crossing them results the *merge* event. As we mentioned in the previous section, the third type of lines may cause the *merge* event. Furthermore, these lines are constructed by drawing the 2-visibility polygon for each vertex v of P. Also a part of uv is considered as a type 3 partition line where u is one of the vertices of P which is critical for v. See Figure 3. For determining exact lines of type 3 which cause the *merge* event we reach the following observation;

Observation 1: A line of type 3 causes the *merge* event when two vertices (which construct this line) are critical for each other.

It means that if u and v are critical for each other and a part of uv is considered as a partition line of type 3, this partition line can cause the *merge* event. As a matter of fact, the *merge* event happens, because one of u and v may become invisible after crossing these lines. Suppose that u and v are two vertices of the polygon. u and v can be convex or reflex. Additionally u and v are called "same side" or "opposite side": If edges of the vertex u and v lie on the same half plane divided by the line uv, u and v are called in the same side. Otherwise, they are called the opposite side. See Figure 4. Consequently, the line that causes the merge event is one of these cases:
1) *RR: S*, two reflex vertices which are in the same side,
2) *RR: D*, two reflex vertices which are in the different side,
3) *RC: S*, a reflex and a convex vertex which are in the same side,
4) *RC: D*, a reflex and a convex vertex which are in the different side,
5) *CC: D*, two convex vertices which are in the same side.
6) *CC: S*, two convex vertices which are in the same side (this is impossible, because drawing a partition line of type three by u and v is impossible).

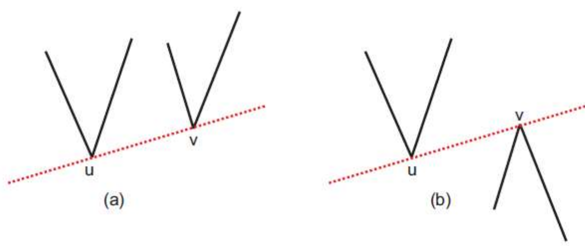

Fig. 4: (a) u and v are in the same side, (b) u and v are in the opposite sides.

A part of uv is considered as a partition line of type 3 if combinational representation of 2-visibile region changes after crossing that line. On the other hand, the 2-visibility changes when there are two other intersections between the line uv and P. We partition the line uv in three parts: the segment uv, a half-line emanates from u, a half-line emanates from v. These two intersections between uv and P can lie on different parts of the line uv. So these cases can happen:
i) Both intersections lie on the segment uv,
ii) One intersection lies on one of the half-lines and another one lie on the segment uv,
iii) Both intersections lie on one of the half-lines.

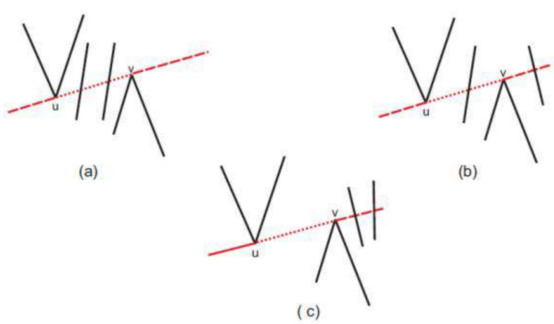

Fig. 5: (a), (b) and (c) illustrate the cases (i), (ii) and (iii) respectively.

Now according to above discussion we will study cases 1 to 6.

1) *RR: S*; two reflex vertices which are in the same side: in this case, we have to consider three different situations according to the positions of two intersections between uv and P. There are two half-planes bounded by uv. We consider the half-plan which both edges of u and v lie on it as a upper half-plan. Also another half-plan is considered as a lower half-plan.

1-i) When the intersections lie on the segment uv; in this case, there are two segments which are considered as a type three partition line. These partition lines lie on the line uv. See Figure 6. We suppose that the 2-modem crosses one of these lines toward the upper half-plane. Let m be an arbitrary point on the lower half-plan. We connect m to both v and u. We consider the segments emanate from u and v which lie on the line mu and mv respectively and hit the boundary of P. These two segments are called s and s'. It is tangible that the regions which are upper than s and s' are not visible for m. Now we move m toward the partition line as close as possible. So there are two parts inside P which are invisible for m. Since both u and v are visible for m, these two shadows are disjoint. But after crossing the partition line one of u and v becomes invisible for 2-modem. Consequently, two disjoint shadows will merge with each other. As a result, the *merge* event will happen.

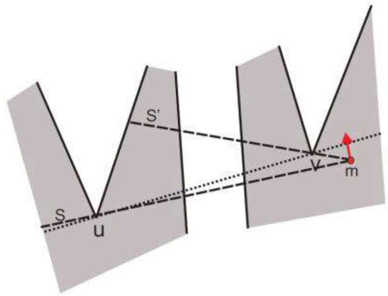

Fig. 6: The case (i) for RR: S (the grey regions refer to the onside of the polygon).

1-ii) The second case is when one intersection lies on the segment *uv* and another intersection occurs on one the half-line emanates from one of *u* or *v*. According to *u* and *v* which are reflex and the outer and the inner part of the polygon, this case cannot occur.

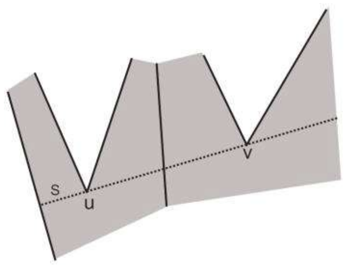

Fig. 7: The case (ii) for RR: S which is impossible.

1-iii) In this case, both intersections lie on one of the half-line emanating from one of *u* or *v* (e.g., *v*). The polygon is closed, so there is another intersection between the same half-line and *P*. We suppose a hypothetical walker who traverses the half line in the direction that go far from *v*. Then the region between the second and the third intersection is inside *P*. Also the segment between them which lies on the line *uv* is one of the third type of the partition line. We consider the 2-modem crosses this partition line toward the upper half-plan. Let *m* be an arbitrary point on the lower half-plan. We connect *m* to both *v* and *u*. We consider the segments emanate from *u* and *v* which lie on the line *mu* and *mv* respectively and hit the boundary of *P*. These two segments are called *s* and *s'*. It is tangible that the region upper than *s* and *s'* are invisible for *m*. Now we move *m* toward the partition line as close as possible. So there are two parts inside *P* which are invisible for *m*. Because both *u* and *v* are visible for *m*, these two shadows are disjoint. But after crossing the partition line one of *u* and *v* becomes invisible for 2-modem. Consequently two disjoint shadows will merge with each other. As a result the *merge* event will happen.

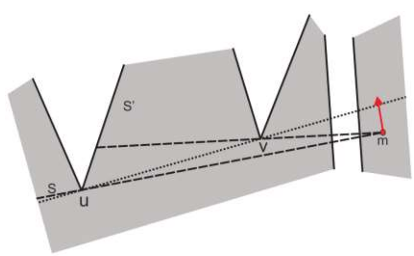

Fig. 8: The case (iii) for RR: S.

2) *RR: D*, two reflex vertices which are in the different side;

In this case, the merge event will happen in two cases which are shown in 9. In Figure 9 if the searcher crosses the partition line in direction of arrow, the merge event occurs. Each case can be concluded like above explanation for *RR:S*.

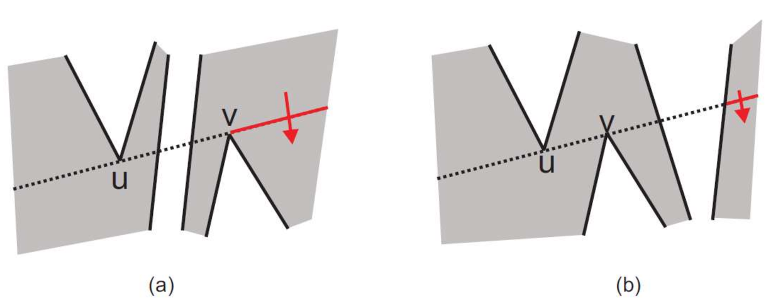

Fig. 9: Possible cases that the merge event occurs for RR: D.

3) *RC: S*, a reflex and a convex vertex which are in the same side;

3-i) When the intersections lie on the segment *uv*, according to the position of inside and outside of the polygon this case cannot happen.

3-ii) One intersection lies on one of the half-lines and another one lies on the segment *uv*, suppose that *u* is the convex vertex and *v* is the reflex vertex. Because *u* and *v* are not the same, the position of another intersection which lies on the half-line is important.

Because that the vertex v is a reflex vertex, one of these half-lines whose end point is v is one of the partition lines. Also because the polygon is closed, there is always one intersection on this half-line. Another half-line may intersect the polygon. If this half-line does not intersect P, crossing from other partition line does not cause the *merge* event and crossing can only cause the shadow becomes bigger or smaller than before. But if there is another intersection on this half-line, a part of it, is considered as a partition line so that crossing it in a specific direction causes the *merge* event. In other word, if before crossing that line both v and u were visible, after crossing that partition line v becomes invisible. As a result, two shadows that are disjoint by seeing v, after crossing the partition line, they merge together.

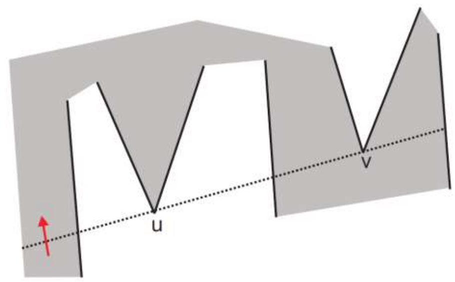

Fig. 10: The case (ii) for RC: S.

3-iii) Both intersections lie on one of the half-lines; according to the position of inside and outside of the polygon this case cannot happen.

4) *RC: D*, a reflex and a convex vertex which are in the different side;

4-i) When the intersections lie on the segment uv, this case cannot happen according to the inside and outside of the polygon.

4-ii) One intersection lies on one of the half-lines and another one lies on the segment uv, suppose that u is the convex vertex and v is the reflex vertex. In this case, only there is one case that if the 2-modem crosses the partition line in a specific direction (direction of arrow in Figure 11) the *merge* event will occur. It can be explained as the case 3-ii.

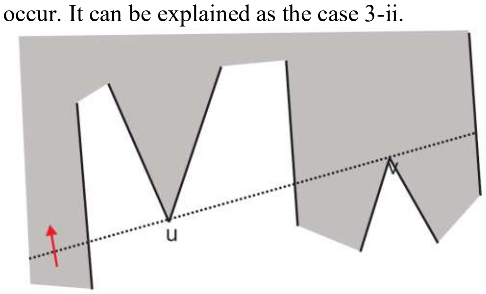

Fig. 11: The case (ii) for RC: D.

4-iii) Both intersections lie on one of the half-lines; according to the position of inside and outside of the polygon this case cannot happen.

5) *CC: D*, two convex vertices which are in the same side;

5-i) When the intersections lie on the segment uv, in this case, u and v are not 2-visible for each other. In other word, there is no partition line of type 3 which is constructed by these two vertices that crossing it causes the *merge* event.

5-ii) One intersection lies on one of the half-lines and another one lies on the segment uv, according to the position of inside and outside of P this case cannot happen.

5-iii) Both intersections lie on one of the half-lines; in this case, if the polygon only intersects one of the half-lines, a part of this half-line is considered as a partition line of type three, but crossing from this line only makes a shadow smaller or bigger. If both half-lines intersects P, a part of both of them is considered as a partition line. Moreover, if before crossing them both u and v are visible for the 2-modem, crossing this line causes that two shadows which are disjoint by seeing both vertices will merge, because after crossing this partition line one of vertices becomes invisible for the searcher. See Figure 12.

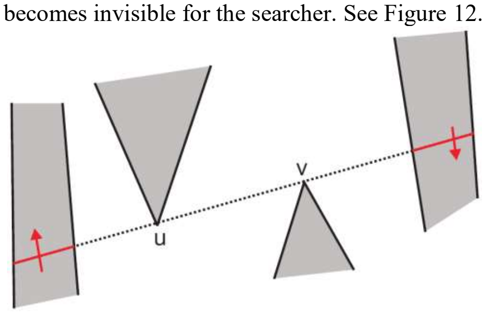

Fig. 12: The case (iii) for CC: D.

6) *CC: S*, two convex vertices which are in the same side;

6-i) When the intersections lie on the segment uv, in this case, u and v are not 2-visible for each other. In other word, there is no partition line of type 3 which is constructed by these two vertices that crossing it causes the *merge* event.

6-ii) One intersection lies on one of the half-lines and another one lies on the segment uv, according to the position of inside and outside of P this case cannot happen.

6-iii) Both intersections lie on one of the half-lines; in this case, crossing the partition line of type 3 may cause only that a shadow becomes smaller or bigger.

So in conclusion, there is no case of type 6 that cause the *merge* event.

At the end, all of lines which crossing them causes the *merge* event, are called *merge lines*.

4. Reporting the Merge Event on the Path

In this section we consider that a simple polygon P and a path inside it are given. This given path does not intersect ∂P. A 2-modem searcher moves on this path. We want to find out if during the 2-modem's movement, the merge event happens or not. Furthermore, if the merge event occurs, the point of the path which after crossing it two shadows will merge together has to be reported. For these purposes we use the characterization of the lines that according to the previous section crossing them in specific direction causes the merge event. It should be noted that for the merge event the searcher has to pass this line. Otherwise, only touching the line does not cause the merge event. We suppose that the query path is constructed by k connected segments. We consider each of these segments one by one respectively. S denotes one of these segments and S' is the next segment on the path. We find the intersections of S with the merge lines. The merge line which intersects S is called L. If this intersection happens in the middle of this S, it is obvious that for moving on this segment the 2-modem has to cross this line. If this intersection occurs at the end point of it, the problem is a little complicating. It is clear that this end point (we call it t) is either the starting of S on this path or the starting point of S' (unless this is the last segment). For the former case, if this end point and another end point of S lie on the same subpolygon which is divided by L, the merge event will not occurs. Otherwise, the merge event has to be reported for this point. In the latter case, the location of another endpoint of S' is important. We call it t'. Indeed, each merge line separates P into two subpolygons. If t and t' lie on the same subpolygon, the merge event will not occurs by moving on S', because the 2-modem only touches L. Thereafter we search the intersections of S' with the merge lines (except L if L intersects this line at the first end point of S').

5. Conclusion

In this study, we considered a new version of Pursuit-Evasion problem and characterized some lines which crossing them results that two invisible regions merge together. The number of these lines can be $O(n^2)$. Also if a query path which is constructed by k segments is given, in $O(kn^2)$ we can figure out if during the movement of the 2-modem on this path invisible regions will merge together or not.

References

[1] J. Yu and S. M. LaValle. Shadow information spaces: Combinatorial filters for tracking targets. *IEEE Transactions on Robotics*, 28(2):440-456, 2012.

[2] L. J. Guibas, J.-C. Latombe, S. M. LaValle, D. Lin, and R. Motwani. Visibility-based pursuit-evasion in a polygonal environment. *International Journal of Computational Geometry and Applications*, 9(5):471-494, 1999.

[3] O. Aichholzer, R. Fabila-Monroy, D. Flores-Pealoza, T. Hackl, C. Huemer, J. Urrutia, and B. Vogten-huber. Modem Illumination of Monotone Polygons. *Proc. 25th European Workshop on Computational Geometry EuroCG09,* pages 167-170, Brussels, Belgium, 2009.

[4] Y.Bahoo, A. Mohades, M. Eskandari, M. Sorouri. 2-modem Pursuit-Evasion Problem. Proc. *29th European Workshop on Computational Geometry Euro CG 2013*, pages 201-204, Braunschweig, Germany, 2013.