A Complexity Result for the Pursuit-Evasion Game of Maintaining Visibility of a Moving Evader

Rafael Murrieta-Cid[†], Raul Monroy[‡], Seth Hutchinson[*] and Jean-Paul Laumond[**]

†Centro de Investigación	‡Tecnológico de Monterrey	*Beckman Institute	**LAAS-CNRS
en Matemáticas (CIMAT), México	Campus Edo. de México	University of Illinois	Toulouse France
murrieta@cimat.mx	raulm@itesm.mx	seth@uiuc.edu	jpl@laas.fr

Abstract— In this paper we consider the problem of maintaining visibility of a moving *evader* by a mobile robot, the *pursuer*, in an environment with obstacles. We simultaneously consider bounded speed for both players and a variable distance separating them. Unlike our previous efforts [11], we give special attention to the combinatorial problem that arises when searching for a solution through visiting several locations. We approach evader tracking by decomposing the environment into convex regions. We define two graphs: one is called the mutual visibility graph (MVG) and the other the accessibility graph (AG). The MVG provides a sufficient condition to maintain visibility of the evader while the AG defines possible regions to which either the pursuer or the evader may go. The problem is framed as a non cooperative game. We establish the existence of a solution, based on a $k - Min$ approach, for the following givens: the environment, the initial state of the evader and the pursuer, including their maximal speeds. We show that the problem of finding a solution to this game is NP-complete.

I. INTRODUCTION

In this paper, we consider the problem of maintaining visibility of a moving *evader* by a mobile robot, called the *pursuer*. As usual, the environment is filled with obstacles and the speed of the participants is bounded; however, the participants may be separated by an arbitrary distance. We approach evader tracking by decomposing the environment into convex regions.

Pursuit-evasion can be defined in several ways. One formulation requires *finding* the evader with one or more mobile pursuers that sweep the environment so that the evader does not eventually sneak into an area that has already been explored. Deterministic [14], [18], [4], [17] and probabilistic [21], [5], [8] algorithms have been developed in this vein. An alternate formulation requires the pursuers achieve the goal of *catching* the evader; that is, moving to a contact configuration or closer than a given distance [7]. These problems are related to ours but are not the same. In our setting, the evader is initially positioned within the pursuer's field of view and the goal is to make the pursuer keep visibility of the moving evader.

We have already developed motion strategies for evader tracking [11], analyzing two main scenarios: one where the distance between the pursuer and the evader is variable but the speed of both player is unbounded; and other where the speed of both the evader and the pursuer is bounded but the distance between the pursuer and the evader is constant. In this paper, we present a more general formulation, in which,

we simultaneously consider bounded speed for both players and a variable distance separating them.

In [11], the case of bounded speed for both the pursuer and the evader is only analyzed for a constant distance between the pursuer and the evader. Therefore, under that assumption of constant distance, a slower pursuer will always be defeated by the evader, even in an environment without obstacles. In this paper we analyze the formulation of variable distance between the evader and pursuer considering that both players move with bounded speed. Now a slower pursuer may be able to maintain visibility of the evader even in an polygonal environment with obstacles, which is simply connected. To find a solution a careful inspection on the map and the initial position of both participants is required to determine the existence of a solution. In this paper we provide such analysis.

In [11], we were able to establish sufficient conditions for escape by the evader (note that if the evader can escape an infinitely fast pursuer, then it will naturally escape a pursuer with finite speed), but we were unable to determine sufficient conditions under which the pursuer could maintain visibility of the evader. In the present paper, we provide sufficient conditions for surveillance by exploiting the concept of strong mutual visibility between regions in the convex decomposition of the environment.

Furthermore, in [11], we did not consider the combinatorial problem inherent to any strategy that considers visiting several locations, for the case of bounded pursuer and evader speeds. In this paper, we provide such analysis and we also provide complexity results to the problem.

Thus, in summary, the research reported in this paper differs from our previous efforts in the following main contributions:

1) We solve evader tracking for the case where the speed of each participant is bounded and where they may separate one another a variable distance.
2) We address the combinatorial problem inherent to any strategy that considers visiting several locations, for the case of bounded pursuer and evader speeds.
3) We consider the case of a pursuer slower than the evader; and
4) We provide complexity results to the problem.

II. Previous Work

Others have also been interested in keeping track of a moving evader. In [10] game theory is applied to formulate the problem and an algorithm that operates by maximizing the probability of future visibility of the evader is presented.

More recently, an extended version of the problem, where multiple evaders and pursuers are involved, has attracted increasing attention. In [13] a method that accomplishes this task but restricted to uncluttered environments is proposed. The method works by minimizing the total time in which targets escape surveillance from a robot team member. In [9] an approach that maintains visibility of several evaders using mobile and static sensors is proposed. It applies a metric for measuring the degree of occlusion, based on the average mean free path of a random line segment.

More related to ours is the work of [1], which shows how to efficiently (low-polynomial) compute an optimal reply path for the pursuer that counteracts *a given evader movement*. This work does not deal with the problem of deciding whether or not there is an evader path that escapes surveillance, not even for the special case where the evader follows a fixed policy.

It follows that while the problem of keeping track of a moving evader has been largely investigated, the decision problem—answering whether or not the evader may escape—has not been addressed, at least for the case where the speed of each participant is bounded and the surveillance distance varies. Answering this question is one of the goals of this paper.

III. Problem Definition

The evader and the pursuer are represented as points. Obstacles are modeled as polygons and the environment is known a priori. Each participant accurately knows his position at all times, is equipped with an omni-directional sensor and is limited to move at bounded speed. We assume an antagonistic evader who moves continuously but not unpredictably for he follows a fixed policy: travel the shortest path to escape pursuer surveillance. Other than this, no kinematic nor dynamic constraints are imposed on the pursuer or the evader motions.

Notice that it would be misleading to conclude that by making the evader stick to an escaping policy our problem is no longer a game. To begin with, policies are popular in games. For instance, in tic-tac-toe an unbeatable strategy starts by systematically choosing the center bean in the grid. What makes our problem a non cooperative game is that the evader and pursuer have antagonistic goals [19]: The evader aims to maximize gain by seeking for a time to escape, t_e strictly smaller than the time to prevent escaping, t_{pe}; while the pursuer aims to minimize loss by keeping $t_{pe} \leq t_e$.

A. Strong Mutual Visibility

Let the environment be divided into convex regions. We define the visibility of a participant in terms of the visibility of the region where he is. Two regions are *strongly mutually visible* if every point belonging to any of the two regions is able to see all the points of the other region. The pursuer maintains strong mutual visibility of the evader, if it is within the same region where the evader is or if they both are in regions that are strongly mutually visible. Thus, maintaining strong mutual visibility of the evader amounts to maintaining visibility of the entire region where it is.

Strong mutual visibility is stronger than classical visibility, where the participants see one another if the line segment between them does not cross an obstacle at any point other than the endpoints [16]. So if on a given scenario there is a solution to evader tracking under strong mutual visibility then there also is a solution under classical one. This implication does not reverse in general. Indeed, upon classical visibility, it is not clear what the pursuer should do when the line of sight between him and the evader is in contact with an obstacle. Fig. 1 shows an example of this situation. Note that there is a conflict on what the pursuer should strive towards: either minimizing the shadow region so as to prevent escaping or minimizing the distance so as to prevent a further, second occlusion.
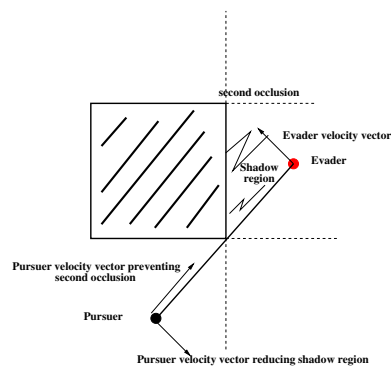

Fig. 1. Classical visibility

This issue has been already noticed in [6], who proposed a heuristic that causes the pursuer to move in the direction of the summation of the vector that locally minimizes the shadow region and the vector that minimizes the distance to the vertex giving rise to the shadow. Improving upon this result, [2] presented a local minimum risk function, called the vantage time, used to drive a greedy motion planning strategy. Neither [6] nor [2] established whether or not their motion strategy guarantees that the pursuer is always able to prevent the evader from escaping.

Under our definition of strong mutual visibility the conflict in deciding if the pursuer should concentrate its effort to minimize the shadow region or to minimize the distance so as to prevent a further second occlusion does not exist.

In this paper, we will introduce an approach that determines whether or not it is possible for a pursuer to maintain strong mutual visibility of a moving evader, addressing the question: can the evader escape? Further, whenever a solution exists, we will find a motion plan for the pursuer that guarantees surveillance of the moving evader under the given assumptions.

IV. Environment partition

We divide the environment into convex regions. Convexity ensures that a robot with omnidirectional sensing is able to see all the points within the region of residence. Our convex partition is similar to the region decomposition produced by the lines of the aspect graph [15] in 2D using perspective projection. Lines forming one such a graph are of two types: inflection rays and bi-tangent rays. An inflection ray emerges from a reflex vertex (a corner of an internal angle greater than π) and terminates when it reaches a polygonal barrier of the environment. A bi-tangent ray connects two visible reflex vertices. The bi-tangent ray is extended outward from the pair of bi-tangent points (the reflex vertices) and it terminates also when it reaches an obstacle.

A property of an aspect graph is that any time one line is crossed a new segment of the environment appears (from the position where the aspect graph line was crossed). Note that the aspect graph says nothing about the visible area inside the polygon. In order to consider visibility in the interior of the polygon we use the strong mutual visibility notion defined in section III-A.

Our convex partition of the environment is an aspect graph partitioning plus an additional feature, namely: we connect every pair of bi-tangent vertices. Thus, in our partition bi-tangent rays are extended both outwards and inwards from a pair of bi-tangent points. Consequently, the polygonal environment is partitioned into more smaller convex regions than the standard aspect graph, see Fig. 2.
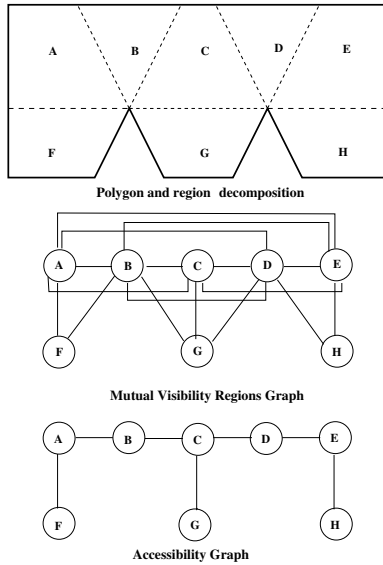

Polygon and region decomposition

Mutual Visibility Regions Graph

Accessibility Graph

Fig. 2.   Environment partition and resulting graphs

A. Two graphs modeling the environment

Our partition of the environment yields two graphs. The first one we call the *mutual visibility graph* (MVG) and the second one the *accessibility graph* (AG). Nodes in these graphs are regions. In an MVG, two regions are connected if they are strongly mutually visible. The MVG enables each participant to know which regions are potential candidates to attempt to escape. These regions we call *escapable regions*. Symmetrically regions where the pursuer should move to prevent the evader from escaping we call *prevention-from-escape*. In an AG, two regions are connected if they share a region boundary bigger than one single point.

An MVG therefore provides a sufficient condition to maintain visibility of the evader while an AG defines the possible transitions that both pursuer and evader can carry out between regions. Note that what counts as an escapable (respectively prevention-from-escape) region depends on the current regions where both the evader and the pursuer are.

B. Finding strongly mutually visible regions

To define whether or not two regions are strongly mutually visible we use a convex hull computation [12]. Formally we say that regions A and B are strongly mutually visible if and only if $\text{int}[\text{convex-hull}(A \cup B)] \subset W$, where W is the polygon representing the workspace. In Fig. 2, regions A and E are strongly mutually visible; by contrast, regions C and H are not.

V. Paths to move between regions

Our pursuit-evasion problem can be abstracted to a graph: whether or not it is solvable amounts to whether or not the graph enjoys some properties. However, the problem still has a geometric aspect, namely: *finding paths to move across regions. This problem corresponds to assigning the appropriate weights to the graph edges*. To escape, the evader should move to an *escapable region*. His task is to find a path to one such a region and travel it in a time strictly smaller than the time it would take the pursuer to prevent the escape. This path will end in a point on the boundary of an escapable region. Often this point corresponds to a reflex vertex.

A. Bounded speed

Thus, the solution to our game depends on two times: the time taken for the evader to reach an escapable region, t_e, and the time taken for the pursuer to reach a prevention-from-escape region, t_{pe}. For the pursuer to keep the evader from escaping, the constraint $t_e \geq t_{pe}$ must be satisfied at all times. Considering that both pursuer and evader travel a given path at saturated speed, this constraint can be defined in terms of distances and relative velocities. Thus, it is possible to define the problem as a constraint over the distances to be traveled, namely:

$$d_e \geq d_{pe} \frac{V_e^{\max}}{V_p^{\max}} \tag{1}$$

where d_e is the distance to escape (the distance between the evader and the boundary of an escapable region), d_{pe} is the distance to keep the evader from escaping (the distance between the pursuer and the boundary of a prevention-from-escape region), and $V_e^{\max}$ and $V_p^{\max}$ are respectively the maximal speed of the evader and the pursuer. Clearly, if the pursuer is faster than the evader, then it will have a winning strategy in a larger number of environments.

This formulation *holds for polygons with or without holes*. However, in polygons with holes a faster evader can always escape pursuer surveillance following a simple strategy: turn around the nearest hole. Conversely, a faster pursuer, without surveillance distance constraint, may apply another simple strategy: catch the evader (moving to a configuration in contact with it) and then stick to it.

However, in polygons without holes, it is possible for a slower pursuer to keep visibility of a faster evader. For instance, for an environment containing one single corner, if the pursuer is at the corner, then it needs not move at all to avoid the evader from escaping. Even for more complex polygons, provided they have no holes, a slower pursuer may always maintain visibility of a faster evader. So a careful inspection on the map and the initial position of both participant is required to determine the existence of a solution. Our approach is also able to find a winning pursuer motion strategy if a solution exists.

B. Evader and pursuer paths: Local optimality vs. global validity

By definition, the visibility constraint is initially satisfied. Therefore, an action must be taken by the evader to break it. This yields a sequence of motion decision elements, each of which is a pair representing the action of the evader and the corresponding response of the pursuer.

At any step, starting from their current location, the evader moves aiming at reaching the boundary of an escapable region, and, accordingly, the pursuer reacts moving towards the boundary of a prevention-from-escape region. The graphs AG and MVG are used to determine every *feasible motion sequence* yielding a *feasible ordering* to visit regions. That is, the sequences of escapable regions and prevention-from-escape regions (using the MVG) while moving the players through adjacent regions (using the AG). To decide whether or not the evader can escape at a given sequence step i, we find the shortest evader path to all escapable regions. If for all these paths there is a pursuer path satisfying (1) then the evader cannot escape at step i.

Note that assuming that the evader is moving along the shortest path to escape is a sufficient condition to determine the existence of a solution. The evader may consider other, larger alternative paths at hand to break pursuer visibility. However, if the pursuer is able to maintain visibility, even though the evader is traveling along the shortest path, then he will always win at this step (see Proposition 6.1).

In a path plan covering more than one single step, it is crucial to ensure that the final condition at step i meets the initial condition of step $i + 1$. Hence a long term plan must satisfy constraint (1) for all the feasible permutations of the elements of the sequence (sequence of adjacent regions). In fact (1) can be decomposed into several parts and, thus, we have a set of constraints that have to be satisfied for all i:

$$d_e^{(i)} \geq d_{pe}^{(i)} \frac{V_e^t}{V_p^t}, \ \forall t \tag{2}$$

C. Example Scenario Contrasting Local and Global Paths

It is important to underline that a response of the pursuer that is *locally optimal* for step i may break the visibility along the whole sequence; that is, the pursuer will lose. In order to better clarify our statements we use the example scenario depicted in Fig. 3. There, thinner dashed lines indicate the environment partition, black solid bold lines the pursuer paths, bold (red) dashed lines the evader paths, and light (cyan) solid lines indicate invalid pursuer paths. Partial paths and regions are labeled. Path weights, corresponding in this case to path lengths, are also indicated. For this example, the pursuer is supposed to be twice as fast as the evader.
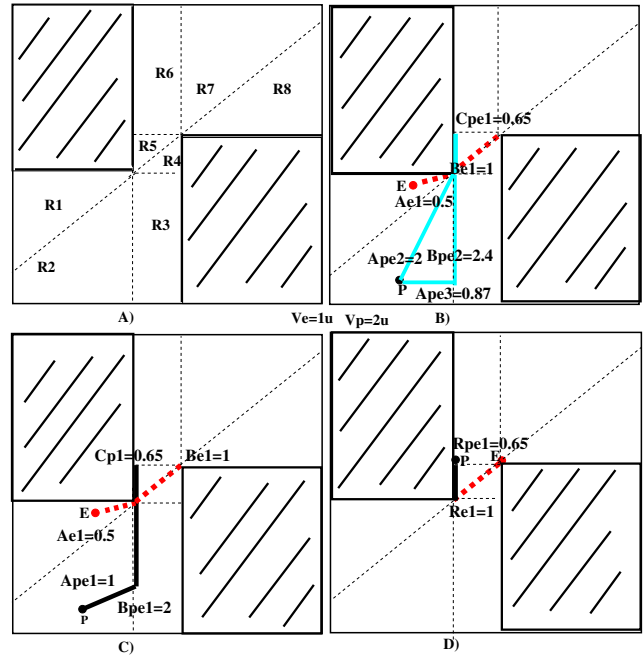

Fig. 3.   Local Optimality vs. Global Validity: an Example Scenario

Fig. 3 A) shows the partition of the environment, its regions and their labels. Fig. 3 B) shows the initial position of the evader, in $R1$, and the pursuer, in $R2$. Given that these regions are strongly mutually visible, the initial visibility condition is satisfied. Fig. 3 B) shows also the pursuer's two locally optimal paths. If the pursuer follows either of them it will lose the game, as explained by the following reasoning. First, notice that $R4$ and $R5$ are the closest escapable regions that the evader can reach. So $Ae1$ is the shortest path the evader could take to reach the boundaries of these two regions. The pursuer locally optimal response is to move along a straight line perpendicular to the boundary of region $R3$ (inflection ray of the aspect graph), path $Ape3$. However, if the pursuer commits to such a path then he will lose in the second step. This is because the evader can then move along the visibility graph (the shortest path) connecting regions $R5$ and $R8$ traveling path $Be1$ and, in that case, the pursuer will *not* be able to reach region $R6$ in time traveling path $Bpe2$. This is in turn because the total cost of the pursuer path is $\frac{(Ape3=0.87)+(Bpe2=2.4)}{2} = \frac{3.27}{2} = 1.64$, while that of the

evader path is only $(Ae1 = 0.5) + (Be1 = 1) = 1.5$.

In an alternative case, the pursuer may identify that the shortest path at step $i = 2$ corresponds to the concatenation of paths $Ape2$ and $Cpe1$, with a total cost of $\frac{Ape2(=2)+Cpe1(=0.65)}{2} = 1.32$. The cost of the evader path, also at step $i = 2$, is greater, 1.5. But these pair of paths will still make the pursuer lose, only that at step $i = 1$. To see this notice that the cost of the pursuer path at step $i = 1$ is $Ape2 = \frac{2}{2} = 1$ while that of the evader is $Ae1 = 0.5$. Hence, following either of these two motion paths the pursuer fails.

However, there exists a winning path for the pursuer, shown in Fig. 3 C). This wining path satisfies the constraints for the two steps of the sequence. It is composed of paths $Ape1(= \frac{1}{2})$ and $Bpe1(= \frac{2}{2})$. The costs of each individual path components respectively are $Ae1 = 0.5$ and $Be1 = 1$. So the pursuer should move along this path, represented by the black bold solid line in Fig. 3 C).

Assuming both the environment depicted in Figure 3 and that the players have chosen their best tries, they end up at the configuration depicted in Figure 3 D). Note that the evader tries to go back to $R1$, traveling along the visibility graph. The game at this point reaches a *steady state*. It does not matter what the evader does, it is sufficient for the pursuer to move over path $Rpe1$.

Note that the evader may not move at saturated speed. Then, if the pursuer moves at saturated speed, condition (2) may not hold anymore and thus the evader could simply move back to an occlusion and escape. To get around this situation, whenever the evader does not move at saturated speed, the pursuer must move at a corresponding speed so that the speed ratio constraint is satisfied. This is achievable since both players instantaneously know the full state of one another.

VI. A $K - Min$ APPROACH TO ESTABLISH THE EXISTENCE OF A SOLUTION

We now aim to determine if there is a winning motion strategy for the pursuer. We start proving a simple result:

Proposition 6.1: Consider that the evader trajectory is known in advance. If the evader moves traveling the shortest path to an escapable region and if the pursuer can keep strong mutual visibility of the evader at all times, then there will always be a pursuer solution for these and any other evader paths, including those that are not the shortest ones.

Proof: Let us assume that the evader is not traveling the shortest paths between the escapable regions then the time to escape can only increase. If the pursuer is able to maintain strong mutual visibility of the evader that follows the path associated to the minimum time to escape then it can maintain strong mutual visibility of the evader that travels any other path having associated a larger time to escape. The result follows. ∎

Let us introduce two concepts. An *escape point* is a point on the boundary of an escapable region. A *prevention-from-escape point* is a point over the *valid* set of pursuer's locations on the boundary of prevention-from-escape regions.

That is, points that can be reached satisfying constraint $t_{pe}^{(i)} \leq t_e^{(i)}$.

For any instance of the problem, there is a set of escape points that the evader should reach in order to escape from the pursuer and a collection of sets of prevention-from-escape points, one for each prevention-from-escape region. These sets are defined for every feasible sequence of evader escapable regions. If a solution pursuer path exists, there will be a set of possible valid pursuer positions over the target region boundary. The crux behind determining a global path consists of concatenating partial local paths connecting these sets.

Algorithm 1 determines the sets of valid pursuer positions. It works by iteratively computing the distance between frontiers of prevention-from-escape regions. It returns a pursuer solution path, if any.

Algorithm 1 Calculate a valid pursuer path

Input: Environment partition, MVG, AG, pursuer's initial position, shortest evader sub paths.

Output: Pursuer Path (list of segments).

repeat

    1. Determine the closest point from the pursuer position (element $i - 1$) to the closest point over the border of the prevention-from-escape region i, associated with the escapable region where the evader is aiming. Define the pursuer subpath as the straight line connecting the pursuer position $i - 1$ and the closest point over the prevention-from-escape region i.

    if region i is not the last element of the sequence **then**

        2. Find the two end-points over the border of the *prevention-from-escape region* i delimiting the valid set that the pursuer can reach at least at the time that the evader reaches its closest escape point over the i-th *escapable region* (note that the two end-points define a segment).

        3. Let $|d|$ be the closest distance between segment i (defining the valid set) and the closest border of the next *prevention-from-escape region* $i + 1$ of the sequence. Call the beginning point, over the valid set i at $|d|$ distance from the *prevention-from-escape region* $i + 1$ the $local - end$.

        4. Modify the path concatenating elements $i - 1$ and i by making the final local path point equal to $local - end$.

    end if

    5. $i \leftarrow i + 1$

until last element of the sequence is reached.

If the pursuer speed is not big enough the resulting valid set will be empty. In other words, the pursuer will not reach the region boundary in time to prevent the evader from escaping.
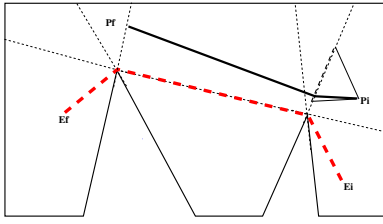
Fig. 4. The pursuer paths

Our algorithm to compute pursuer paths is similar to the dynamic programming principle of minimal cumulative cost:

$$L^*(\pi_k) = \min_{u_k,...,u_K}\left\{\sum_{i=k}^{K} l(x_i,u_i) + l_f(x_f)\right\} \quad (3)$$

where $L^*(\pi_k)$ is the minimal cumulative cost for policy π; u_i the pursuer control at step i; x_i the state of the i-th element of the sequence; $l(x_i,u_i)$ the transition cost from i to $i+1$; K is the length of the sequence and where $l_f(x_f)$ is the cost of the final state, which is taken to be ∞ if the evader escapes and 0 otherwise. However, our approach differs in that we consider *only* valid pursuer paths; that is, paths for which $t_{pe}^{(i)} \leq t_e^{(i)}$ holds. Our algorithm embodies a look-ahead strategy, which aims to minimize the surveillance cost in the next step, $i+1$, at the cost of finding the valid extrema paths satisfying the constraint in the current state i. Note that x_i (the state of the i-th element of the sequence) encodes the constraint $t_{pe}^{(i)} \leq t_e^{(i)}$.

We illustrate the workings of our algorithm using the scenario depicted in Fig. 4. There, we use dotted lines to delimit environment partitions;[1] P_i and P_f (respectively E_i and E_f) to denote the initial and final position of the pursuer (respectively the evader); and (red) dashed lines to denote the shortest paths that the evader may choose to escape. Our algorithm first defines the set of positions (over region boundaries) that the pursuer can reach at least at the same time that the evader reaches an escapable region (cf. step 2); we call these positions *valid*. In Fig. 4, the set of valid positions appears as a dashed bold line over the inflection ray that stems out of the reflex vertex that is next to the pursuer initial position.

Then, the algorithm computes the closest distance between the segment encoding valid positions over the border of the prevention-from-escape region i and the border of the next prevention-from-escape region $i+1$ (cf. step 3). By forward propagation, the end point of a given element of the sequence, i, becomes the starting point of the next element of the sequence, $i+1$, such that the path in the next element $i+1$ is of minimal length—ie. time—(cf. step 4).

In Fig. 4, we use black (thinner) lines to denote the valid extrema paths and bold, solid black lines to denote the paths that are selected. Notice that we use region boundaries as targets to be reached by either participant. Our algorithm explores all feasible motion (sub)paths. Since, for a finite

[1]Note that we are not dividing the environment into a grid.

environment, the number of feasible subpaths is finite then the algorithm shall terminate.

Our approach is of type k-min for two reasons. First (min), the paths that the evader travels are of minimal time. Second (k), the paths that the pursuer should travel to find a solution must have a cost time such that $t_{pe}^{(i)} - t_e^{(i)} \leq k$, for $k = 0$.
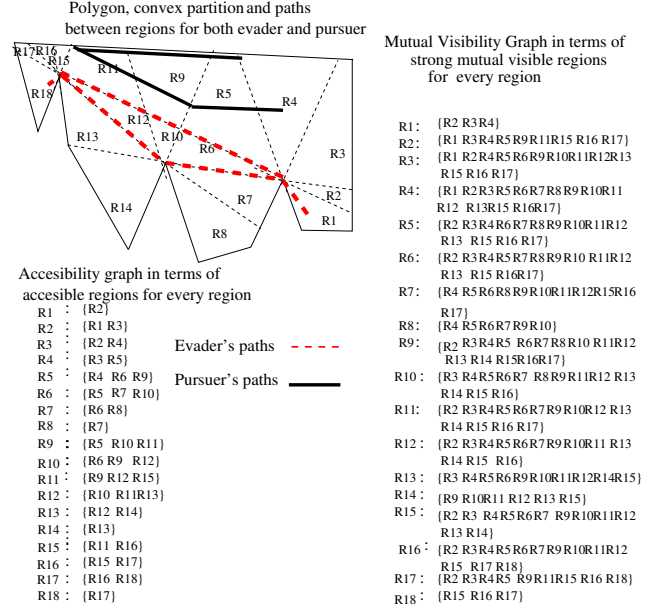
VII. COMPLEXITY RESULTS

Fig. 5. The Geometry: The map, the partition, the paths

Whether or not the pursuer or evader has a winning strategy amounts to checking the cost over the edges connecting strongly mutually visible regions. The cost associated with every edge is given simply by $t_{pe}^{(i)} - t_e^{(i)}$. Thus, to decide which player wins we need to check every feasible ordering: if they all have a non-positive cost then the pursuer wins, otherwise it looses.

The problem of deciding which player wins is NP-complete. The *bottleneck traveling salesman* problem, known to be NP-complete [3], can be reduced to the problem of establishing a feasibility cost over the edges connecting the strongly mutually visible regions. This issue will be illustrated using the example scenario depicted in Fig. 5.

Fig. 5 depicts an environment with 18 regions, each of which is delimited with dotted lines. The pursuer and the evader are respectively assumed to initially be in regions R4 and R1. Fig. 5 displays both the accessibility graph and the strong mutual visibility graph. Nodes in these graphs are regions, labeled R1,..., R18. For each node, we write $R_i : \mathcal{R}\,(1 \leq i \leq 18)$ to indicate that whenever $R_j \in \mathcal{R}$ there exists an edge connecting R_i with R_j. Fig. 5 portrays the paths that each participant considers in an attempt to beat its opponent. The paths that are explored by the evader (respectively the pursuer) are denoted with dashed (solid) lines.

First pair of paths
Evader path 1
$R1 \overset{ec1}{\to} R2 \overset{0}{\to} R3 \overset{0}{\to} R4 \overset{0}{\to} R5 \overset{0}{\to} R6 \overset{0}{\to} R7 \overset{0}{\to} R6 \overset{ec2}{\to}$ $R10 \overset{0}{\to} R12 \overset{0}{\to} R13 \overset{0}{\to} \mathbf{R14} \overset{0}{\to} R13 \overset{0}{\to} R12 \overset{ec3}{\to} R11 \overset{}{\to}$ $R15 \overset{0}{\to} R16 \overset{0}{\to} R17 \overset{0}{\to} \mathbf{R18}$
Pursuer path 1
$R4 \overset{pc1}{\to} R5 \overset{pc2}{\to} \mathbf{R9} \overset{pc3}{\to} R11 \overset{pc4}{\to} \mathbf{R15}$
Simplification of pair of paths 1
Evader path 1
$t_e^1 = ec1 + ec2, t_e^2 = ec3$ $R1 \overset{t_e^1}{\to} \mathbf{R14} \overset{t_e^2}{\to} \mathbf{R18}$
Pursuer path 1
$t_{pe}^1 = pc1 + pc2, t_{pe}^2 = pc3 + pc4$ $R4 \overset{t_{pe}^1}{\to} \mathbf{R9} \overset{t_{pe}^2}{\to} \mathbf{R15}$
Second pair of paths
Evader path 2
$R18 \overset{0}{\to} R17 \overset{0}{\to} R16 \overset{0}{\to} R15 \overset{0}{\to} R11 \overset{ec4}{\to} R9 \overset{ec5}{\to} R5 \overset{ec6}{\to} R4 \overset{0}{\to}$ $R3 \overset{0}{\to} R2 \overset{0}{\to} \mathbf{R1}$
Pursuer path 2
$R15 \overset{0}{\to} R11 \overset{pc5}{\to} R9 \overset{pc6}{\to} R5 \overset{pc7}{\to} \mathbf{R4}$
Simplification of pair of paths 2
Evader path 2
$t_e^3 = ec4 + ec5 + ec6$ $R18 \overset{t_e^3}{\to} \mathbf{R1}$
Pursuer path 2
$t_{pe}^3 = pc5 + pc6 + pc7$ $R15 \overset{t_{pe}^3}{\to} \mathbf{R4}$

TABLE I

PATHS COST AND THEIR SIMPLIFICATION

Table I describes the paths, together with their associated costs, that are considered by each participant. The adjacency between region is obtained from the accessibility graph (AG). The escapable regions and their associated prevention-from-escape region are obtained from the mutual visibility graph (MVG). The paths cost for the pursuer are computed using algorithm 1. The evader paths are of minimal length and hence of minimal cost to move between regions.

The paths are paired, for the path selected by the pursuer comes as a response to the moves taken by the evader. For each (sub)path, we write $R_i \overset{c}{\to} R_j$ to denote a transition from R_i to R_j at cost c. Note that in some transitions $c = 0$. This is because, when robots move around a corner (reflex vertices), the distance to (and hence the cost of moving across) the regions that share a point with that corner is zero. This is in turn due to that our robots are points; naturally, if this were not the case, it is possible to consider a cost different to zero (related for instance to robot size). In table I, escapable regions and their associated prevention-from-escape regions appear typed in boldface. Such regions are key. The pursuer must reach a prevention-from-escape region at least at the same time that the evader reaches the intended escapable region.

Paths can be simplified by collapsing transitions of cost zero and by adding the cost of traveling between adjacent regions, which are not escapable or prevention-from-escape. This naturally yields paths that are more informative in that the new transitions take respectively the participants to a region that is escapable or prevention-from-escape.

Simplified paths are then used to build a decision diagram (cf. Fig 6), with which we can determine what participant has a winning strategy. The evader has a winning strategy if there exists a path in the decision diagram for which the constraint $t_{pe}^i - t_e^i \leq 0$ is broken. Conversely, the pursuer has a winning strategy if every path in the decision diagram preserves the constraint. Notice that the paths cost must be checked to reach a decision. This is the rationale behind the proof of the following proposition.

Proposition 7.1: The problem of deciding whether or not the pursuer is able to maintain strong mutual visibility of the evader is NP-complete.

Proof: In order to make a formal reduction, we use the concept of strong mutual visibility among regions. So, we consider a set of locations, each location corresponds to the border of a region in our environment partition (which can be reached faster than any point inside the region), visiting each of which comes with a cost, given by $t_{pe}^i - t_e^i$, for every feasible element i of the sequence (sequence of adjacent regions). We assume, without loss of generality, that both players move at saturated speed.

The reduction consists of defining the cost between the locations as the edge weights of the bottleneck traveling salesman problem. Answering whether there is a tour of locations whose largest edge cost is $\leq k$ will solve both problems: the bottleneck traveling salesman problem and deciding which player will win. A pursuer solution corresponds to $k = 0$. A polynomial time algorithm capable of solving these instances of our decision problem would also solve all instances of the bottleneck traveling salesman in polynomial time. Therefore, our decision problem is NP-complete. ∎

Note that this complexity result applies to both players: the evader ought to work as hard to defeat the pursuer. To get around the need of a computationally unfeasible algorithm, the players may use an on-line strategy. An on-line strategy computes a motion plan for the next h future stages, and replans in the next iteration for the following h future stages. Typically, h is a very small number. To determine if a given time horizon will produce a winner strategy (in the case that one exists), it is necessary to know the time horizon of the other player and its computational power.

VIII. DISCUSSION AND CONCLUSION

In this paper, we have addressed the problem of maintaining visibility of a moving evader by a pursuer, in an environment with obstacles, where the speed of each participant is bounded and there is not an *a priori* limit as to the distance that may separate them. This problem is related to the problem of *finding* an evader with one or more mobile pursuers [7], [8], [20], but is *not* the same. In this latter problem, the evader is not initially visible to the pursuer (while in ours it is); once the evader is found keeping visibility is not attempted (while in ours it is.)

In this paper we have introduced two relations defined over regions: strong mutual visibility and accessibility. Thanks to these relations, we can model the problem of maintaining visibility of a moving evader by means of a pair of graphs.
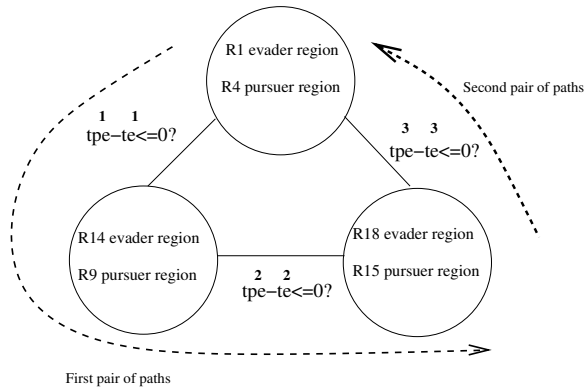
Fig. 6. The final graph with geometric equivalence (one instance of our problem) corresponding to the Bottleneck Traveling Salesman problem

Thus, finding a solution to the pursuit-evasion problem amounts to checking the paths cost for the evader to reach escapable regions and for the pursuer to reach prevention-from-escape regions.

We have established the existence of a solution, given the initial state of the evader and the pursuer, their maximal speeds, the distance separating them and the environment. We have defined the set of pursuer locations over the region boundaries that satisfy strong mutual visibility. We have proposed a $k - Min$ approach to establish the existence of a solution. We have also addressed the combinatorial problem inherent to any strategy that considers visiting several locations.

Finally, we have shown that the problem of deciding whether or not the pursuer is able to maintain strong mutual visibility of the evader is NP-complete. It would be misleading to conclude that [1]'s results and ours contradict one another. In [1] the authors want to find the pursuer path associated to *one given evader path*. That work did not deal with the problem of deciding whether or not some (at least one) of all possible evader paths will yield an escaping path, not even for the case where the evader follows a fixed policy.

Strong mutual visibility is only a sufficient condition to maintain classical visibility. However, we believe that our analysis gives insight to the problem of maintaining classical visibility between two robots. Our problem is a simplified instance of the general problem of maintaining classical visibility. Since a global solution to this simplified instance is already NP complete, the general problem should be at least of that complexity class.

In our current formulation, uncertainty is not considered at all. Ongoing research is concerned with determining whether or not pursuit-evasion under non deterministic uncertainty is decidable. In particular, we target the case where the pursuer does not know the global evader paths; it only knows the instantaneous evader state —position and velocity.

REFERENCES

[1] A. Efrat, H. H. Gonzalez-Baños, S. G. Kobourov, and L. Palaniappan. Optimal Motion Strategies to Track and Capture a Predictable Target. *Proc. IEEE Int. Conf. on Robotics and Automation*, Taipei, Taiwan, pp. 411-423, 2003.
[2] T. Bandyopadhyay, Y. Li, M.H. Ang and D. Hsu, A Greedy Strategy for Tracking a Locally Predictable Target among Obstacles, *In Proc IEEE Int. Conf. on Robotics and Automation*, 2006.
[3] Garey, M.R. and D.S. Johnson, *Computers and Intractability,* W. H. Freeman and Company, 1979.
[4] L. Guibas, J.-C. Latombe,S.M. LaValle, D. Lin, R. Motwani, Visibility-based pursuit-evasion in a polygonal environment, *In Proc 5th Workshop on Algorithms and Data Structures*, 1997.
[5] J. Hespanha, M. Prandini, and S. Sastry, Probabilistic Pursuit-Evasion Games: A one-step Nash approach, *in proc Conference on Decision and Control*, 2000.
[6] H.H. González, C.-Y. Lee and J.-C. Latombe, Real-Time Combinatorial Tracking of a Evader Moving Unpredictably Among Obstacles, *In Proc IEEE Int. Conf. on Robotics and Automation*, 2002.
[7] V. Isler, S. Kannan, and S. Khanna, Locating and Capturing an Evader in a Polygonal Environment. *In Sixth International Workshop on the Algorithmic Foundations of Robotics*, 2004.
[8] V. Isler, S. Kannan, and S. Khanna, Randomized Pursuit-Evasion in a Polygonal Environment. *IEEE Transactions on Robotics*, 5(21):864–875, 2005.
[9] B. Jung and G. Sukhatme. Tracking targets using multiple robots: the effect of environment occlusion. In *Journal Autonomous Robots*, vol. 12 pp. 191-205, 2002.
[10] S.M. LaValle, H.H. González-Baños, C. Becker and J.-C. Latombe, Motion Strategies for Maintaining Visibility of a Moving Target. *In Proc IEEE Int. Conf. on Robotics and Automation*, 1997.
[11] R. Murrieta-Cid, T. Muppirala, A. Sarmiento, S. Bhattacharya and S. Hutchinson. Surveillance Strategies for a Pursuer with Finite Sensor Range. *International Journal of Robotics Research*, Vol 26, No 3 pp. 233-253, March 2007.
[12] J. O'Rourke, *Computational Geometry In C*. Cambridge University Press, 2000.
[13] L. Parker. Algorithms for Multi-Robot Observation of Multiple Targets. In *Journal Autonomous Robots*, vol. 12 pp. 231-255, 2002.
[14] T.D. Parsons Pursuit-evasion in a graph. *In Y. Alani and D.R. Lick, editors, Theory and Application of Graphs*, pages 426-441, Springer-Verlag, Berlin, 1976.
[15] S. Petitjean, D. Kriegman and J. Ponce, Computing exact aspect graphs of curved objects: algebraic surfaces. *Int Journal on Computer Vision, 9:231-255*, Dec 1992.
[16] T. C. Shermer, "Recent results in art galleries," *Proc. of the IEEE*, vol. 80, no. 9, pp. 1384–1399, Sep 1992.
[17] S. Shas, S. Rajko and S.M. LaValle, Visibility-based pursuit-evasion in an unknown planar environment *Int. Journal on Robotics Research, 23(1):3-26*, January 2004.
[18] I. Suzuki and M. Yamashita Searching for a mobile intruder in a polygonal region. *SIAM J. Comput*, 21(5):863-888, October 1992.
[19] A. Rappoport. *Two-Person Game Theory*. Dover, 1966.
[20] B Tovar and S. M. LaValle Visibility-based Pursuit-Evasion with Bounded Speed, *Proc. Workshop on the Algorithmic Foundations of Robotics (WAFR)*,2006.
[21] R. Vidal, O. Shakernia, H.Jin, D. Hyunchul and S. Sastry, Probabilistic Pursuit-Evasion Games: Theory, Implementation, and Experimental Evaluation, *IEEE Trans. Robotics and Automation*, 18(5):662-669, October, 2002.