Evasive Navigation of an Autonomous Mobile Robot in Hostile Unknown Environments

Y. Veksler and E. D. Rimon*

Technion, Israel Institute of Technology,
*rimon@technion.ac.il

Abstract. This paper considers on-line navigation of an autonomous mobile robot in an unknown hostile environment. The hostility of the environment is modeled as the requirement that the robot avoids traveling through open areas where it can be easily noticed. To move evasively through the unknown environment, the robot is allowed to move along obstacle boundary segments and locally shortest lines connecting obstacles in exposed areas of the environment. An *evasive motion graph* consisting of these path segments is used to compute the optimal off-line navigation path. Based on the optimal off-line path, an evasive on-line navigation algorithm called E-Nav is described. The E-Nav algorithm constructs a *partial evasive motion graph* in the currently known environment measured by the robot during navigation to the target. The competitive performance of the E-Nav algorithm is shown to be *log-quadratic* in the optimal off-line evasive path length. The E-Nav algorithm is demonstrated with a step-by-step execution example.

1 Introduction

Mobile robots are expanding into increasingly more demanding and diverse applications. Many of these applications require the robot to traverse unknown environments populated by obstacles whose layout must be measured by the robot during task execution. Navigation under limited knowledge of the environment appears in ground and aerial package delivery in office spaces, industrial facilities, hospitals and large urban areas. Goal seeking algorithms also appear in security applications such as patrolling installations [1, 5] and special operations [15].

Early papers on sensor-based mobile robot navigation considered navigation with position and tactile sensors [14, 20]. Subsequent works considered navigation using vision and laser scanners [10, 16]. With the exception of Blum [3, 4], the performance of these algorithms was typically analyzed in terms of geometric parameters of the environment. Lacking competitive bounds, these algorithms can be fooled to generate lengthy paths. This caveat led to CBUG [6, 7] that transformed the mentioned algorithms into competitive navigation algorithms. Using expanding search ellipses, the modified algorithms are *quadratic* in the optimal off-line path length, which is also the universal lower bound on this problem.

All of these works do not put any restrictions on the mobile robot during navigation. In the *evasive navigation problem,* a mobile robot must reach a target along a path that is hidden from observers whose position is unknown to the robot. Park [17] considers stealth navigation to a target that invades a fully known environment. The target is also the observer, and the robot must reach the target along a path that avoids detection by the invader. Ravela [18] considers stealth navigation over unknown 3-D terrain with known observer locations at any given time. By treating the observers' visible regions as hostile areas, an evasive path is computed and updated based on measurements of the environment and observers. Lavalle [13] and Isler [8, 9] consider *intruder detection* by mobile robots, while Bandyopadhyay [2] considers *stealth tracking.* These algorithms can perhaps be adapted to the intruder's perspective, thus solving dynamic versions of the stealth navigation problem, which is addressed by Stiffler [21].

This paper considers the *on-line evasive navigation problem,* where the mobile robot observes and accumulates information on the environment during navigation to the target. The algorithm described in this paper, E-Nav, navigates a mobile robot through a hostile unknown environment. The robot has no prior information about the location of potential threats in the environment. The algorithm constructs an *evasive motion graph* based on the currently known environment. By traveling exclusively along this graph, the robot avoids areas in the environment that result in prolonged exposure, and thus exhibits evasive behavior during navigation to the target. The E-Nav algorithm is shown to have a *log-quadratic* upper bound in terms of the optimal evasive off-line path length. This competitive upper bound is close to the universal lower bound over all on-line navigation algorithms that use the same sensors, which is shown in Ref. [22] to be *quadratic* in terms of the optimal evasive off-line path length.

The paper is structured as follows. Section 2 describes the evasive navigation problem under local exposure constraints. Section 3 describes a scheme for computing the optimal off-line navigation path under local exposure constraints. The E-Nav algorithm is described in Section 4, followed by a step-by-step execution example. Section 5 analyzes the competitive performance of the E-Nav algorithm. The conclusion summarizes additional results as well as future research.

2 The On-Line Evasive Navigation Problem

The *on-line evasive navigation problem* is defined as follows. A mobile robot modeled as a disc of size D is located at a start point S. The robot must reach a specified target point T in a planar environment populated by stationary obstacles whose layout is unknown to the robot. The start and target are located at concealed positions adjacent to obstacle boundaries. The robot is assumed to have an ideal position sensor (particle filters can help achieve this assumption), and an omnidirectional scanner that detects obstacles visible from the robot's current position (Figure 1). The robot stores a map of the observed environment in the form of a polygonal approximation of the obstacles (Figure 1).

The *hostility* of the environment is modeled as the requirement that the robot avoids traveling through open areas where it can be easily detected by potential threats whose position is unknown to the robot. Thus, when the robot moves in open areas between obstacles it is considered to be *exposed.* An intuitive way to quantify the risk of detection at a given robot position is based on the minimal distance between the robot and surrounding obstacles. As the minimum distance increases, the robot becomes more exposed and can be seen from many directions

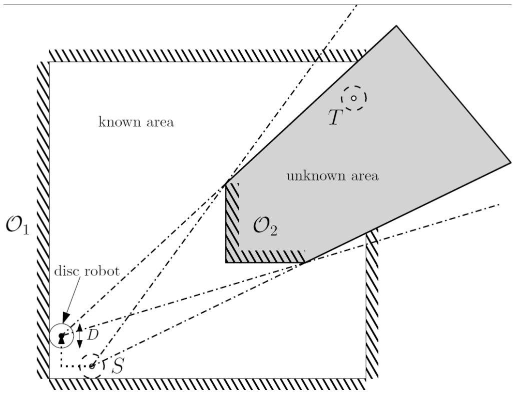


Fig. 1: An omnidirectional scanner mounted at the robot center measures distance to the obstacles and surrounding walls during navigation.

at once. Moreover, it will take the robot longer time to take cover once detected. As the robot approaches an obstacle boundary, it becomes less exposed, until it touches the boundary of an obstacle and is then considered to be *concealed*.

To avoid exposure to hostile agents while navigating in an unknown environment, the robot strives to limit its exposure time and tries to remain concealed as much as possible. An intuitive measure for the probability of detection is the total time the robot is exposed throughout the navigation task. The *total exposure time,* t_{exp}^{tot}, is defined as the total time the robot is *not* in contact with an obstacle (and therefore exposed) along the robot path from S to T. For simplicity the robot is assumed to move with uniform speed, and therefore the shortest path is also the fastest. When the robot moves with uniform speed, t_{exp}^{tot} is proportional to the total length of the *exposed path segments* along the robot path from S to T. Under the total exposure measure, the robot strives to minimize the *total length* of the exposed path segments along its path from S to T.

The total exposure measure often prefers paths that pass through large exposed areas between neighboring obstacles, rather than paths with a large number of short exposed transitions between obstacles (see [22] for an example). This can result in lengthy movements through individual open areas, which is undesirable in hostile environments when the location of threats is unknown to the robot. The *local exposure measure* restricts the local maneuvers of the robot in open areas as follows. Let α_{AB} be an exposed path segment between concealed positions A to B. The *local exposure time* is the time it takes the robot to move from A to B along the path α_{AB}. When the robot moves with uniform speed, the local exposure time is proportional to the length of α_{AB}. By limiting the local exposure time and assuming uniform speed, the *local exposure path length* is also limited. The latter is defined as follows.

Local exposure length: *Let α_{AB} be an exposed path segment between two concealed positions of the robot. The* **local exposure length** *is the length $l(\alpha_{AB})$.*

This paper studies evasive navigation under local exposure constraint. The robot is given a user specified maximally allowed inter-obstacle exposure, L_{max}. Evasive navigation under the *local exposure constraint* is modeled as the requirement

$$l(\alpha_{AB}) \leq L_{max}$$

over all exposed path segments during navigation from S to T. The robot must reach the target along local exposure path segments that satisfy $l\left(\alpha_{AB}\right) \leq L_{max}$, or report that such a path is not feasible.

3 Optimal Evasive Off-Line Navigation

This section describes a computation scheme that strives to minimize the robot's total path length from S to T while satisfying the local exposure constraint, L_{max}. The scheme is based on an *evasive motion graph* for the robot which requires the following definitions.

The *c-space* of a disc robot consists of the (x, y) coordinates of its center. The *c-obstacle* induced by a stationary obstacle $\mathcal{O}_i$, $\mathcal{CO}_i$, is the set of all configurations at which the robot penetrates the obstacle. The boundary of $\mathcal{CO}_i$ represents configurations at which the robot touches the obstacle. When two or more obstacles are close to each other, their c-obstacles may overlap, creating a single *traceable obstacle*. Thus, an environment containing obstacles $\mathcal{O}_1, \ldots, \mathcal{O}_k$ is partitioned into subsets of traceable obstacles $Q_1, \ldots, Q_m \quad (m \leq k)$, such that the obstacles of each Q_j induce overlapping c-obstacles. For a disc robot of size D, obstacles located less than D apart would belong to the same traceable obstacle. The boundary of a traceable obstacle represents configurations at which the robot is in contact with one or more physical obstacles that define the traceable obstacle. This boundary forms the following *contact preserving curve.*

Contact preserving curve: *Let a disc robot circumnavigate a traceable obstacle Q_j while maintaining contact with its boundary. The contiguous curve traced by the robot's center throughout this motion forms a* **c-space contact preserving curve** *of Q_j, denoted C_j.*

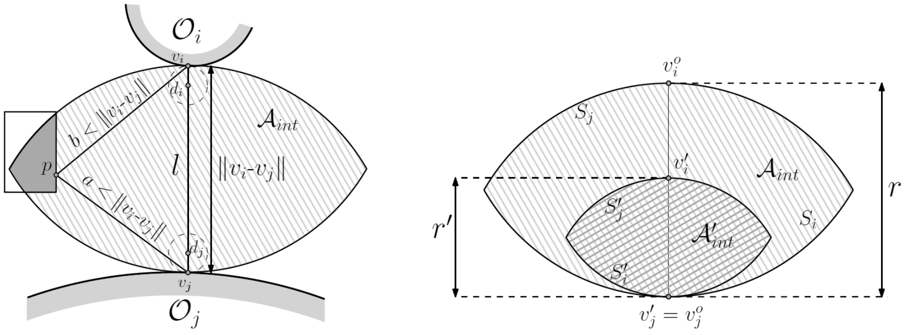
In order for the robot to be minimally exposed while moving between obstacles, the evasive off-line path will consist of obstacle-hugging segments and minimum-length transition lines between obstacles. The transition lines are next defined.

Potential transition line: *Let C_i and C_j be two c-space contact preserving curves. A* **potential transition line**, *l_{tr}, is a locally shortest line segment connecting two c-obstacle boundary points d_i and d_j on C_i and C_j.*

A potential transition line has minimum length relative to local variations of its endpoints along C_i and C_j. It is perpendicular to the contact preserving curves at its endpoints. The set of potential transition lines in the environment is next filtered so that there will be no obstacle points inside the *interference area* of the remaining transition lines in this set. The *interference area* is defined as follows.

Interference area: *Let l_{tr} be a potential transition line with endpoints d_i and d_j. Let v_i and v_j be the physical obstacle contact points at d_i and d_j. The* **interference area**, *$\boldsymbol{\mathcal{A}_{int}}$, is the intersection region of two discs centered at $\boldsymbol{v_i}$ and $\boldsymbol{v_j}$ of radius $\|\boldsymbol{v_i} - \boldsymbol{v_j}\|$, which are tangent to the obstacles at $\boldsymbol{v_i}$ and $\boldsymbol{v_j}$* (Figure 2(a)).

Note that if the disc robot happens to touch another obstacle point, this point can be ignored. At any point within the interference area, $p \in \mathcal{A}_{int}$, the distance from p to the obstacle points v_i and v_j is *shorter* than $\|v_i - v_j\|$ (Figure 2(a)). The interference area induced by any other two obstacle points located on the



(a) (b)

Fig. 2: (a) At an obstacle point p inside $\mathcal{A}_{int}$, the lines connecting p to v_i and v_j are shorter than $\|v_i - v_j\|$. (b) The relation $\mathcal{A}'_{int} \subseteq \mathcal{A}_{int}$ stated in Lemma 3.1.

v_i–v_j segment is entirely contained in $\mathcal{A}_{int}$ induced by v_i and v_j, as stated in the following lemma.

Lemma 3.1. *Let $\mathcal{A}_{int}$ be the interference area induced by two obstacle points v_i and v_j. Let v_i' and v_j' be two obstacle points located inside the line segment v_i-v_j, such that v_i' is closer to v_i and v_j' is closer to v_j. Let $\mathcal{A}'_{int}$ be the interference area induced by v_i' and v_j'. Then $\mathcal{A}'_{int} \subseteq \mathcal{A}_{int}$* (Figure 2(b)).

The proof can be visualized in Figure 2(b) and is therefore omitted. Any potential transition line whose interference area contains other obstacle points is *redundant,* since shorter exposure transition lines connect these points. Hence, all redundant transition lines will be deleted. Hereby, all non-redundant transition lines will be simply called *transition lines* and denoted $l_{tr} : d_i \rightarrow d_j$, where d_i and d_j are the endpoints of the transition line l_{tr}. The filtering of unnecessary long transition lines ensures that the remaining transition lines are disjoint as stated in the following lemma.

Lemma 3.2. *Let $l_{tr}^{(1)} : d_1 \rightarrow d_2$ and $l_{tr}^{(2)} : d_3 \rightarrow d_4$ be two distinct transition lines. Let v_1, v_2, v_3, v_4 be the obstacles contact points when the robot is at d_1, d_2, d_3, d_4. Then the line segment v_1-v_2 does not intersect the line segment v_3-v_4 and therefore $l_{tr}^{(1)}$ and $l_{tr}^{(2)}$ are disjoint.*

Proof Sketch: Assume that $l_{tr}^{(1)}$ and $l_{tr}^{(2)}$ *do* intersect. It can be shown that if $v_1, v_2 \notin \mathcal{A}_{int}^{34}$ ($l_{tr}^{(2)}$ is non-redundant), then the distance from either v_3 or v_4 to both points v_1 and v_2 is shorter than $\|v_1\text{-}v_2\|$. Hence, either $v_3 \in \mathcal{A}_{int}^{12}$ or $v_4 \in \mathcal{A}_{int}^{12}$, implying that the transition line $l_{tr}^{(1)}$ is redundant. Since all transition lines are non-redundant, $l_{tr}^{(1)}$ and $l_{tr}^{(2)}$ must be disjoint. □

The *evasive motion graph* is defined as follows.

Evasive motion graph: *Let a known planar environment contain traceable obstacles $Q_1, \ldots, Q_m$. The* **evasive motion graph, EMG,** *is a graph whose edges consist of all contact preserving curves $C_1, \ldots, C_m$ and all transition lines. The nodes of the graph are all intersection points between $C_1, \ldots, C_m$ and the transition lines, as well as S and T* (Figure 3).

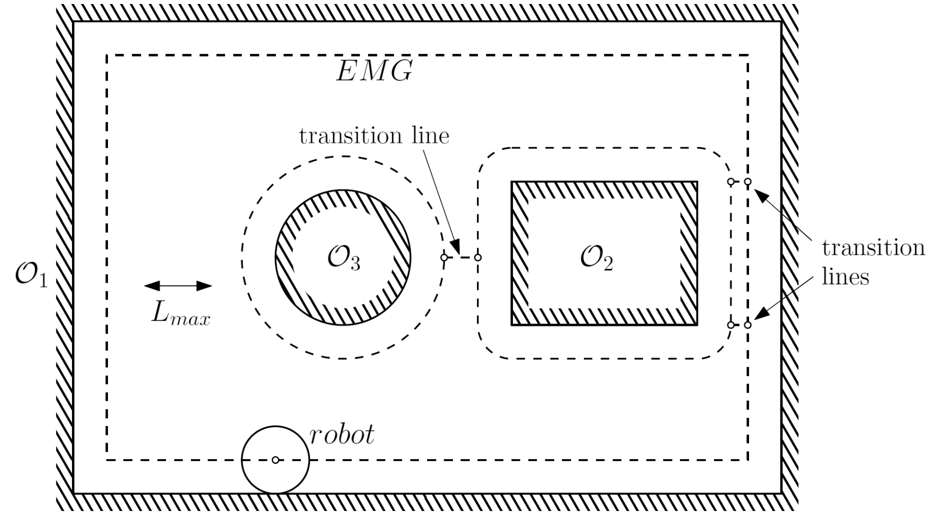


Fig. 3: The EMG graph for an environment populated by known obstacles. Only transition lines of length less than L_{max} are shown.

As illustrated in Figure 3, the EMG forms a planar graph based on the following argument. A transition line l_{tr} does not intersect any contact preserving curve C_j, as this would imply that the transition line passes through a traceable obstacle Q_j, which by definition means that l_{tr} is redundant. Since no two transition lines intersect according to Lemma 3.2, the EMG forms a planar graph.

To quantify the size of the EMG, consider a polygonal environment with n vertices and n edges. The environment contains $O(n)$ contact preserving curves. The number of potential transition lines is $O(n)$, since each transition line crosses the *Voronoi diagram* of the environment at a local minimum of the distance to the obstacles along the respective Voronoi arc. It can be verified that each arc of the Voronoi diagram contains at most one such local minimum point. Since there are $O(n)$ Voronoi arcs [11], the EMG contains $O(n)$ nodes and $O(n)$ edges.

The computation of the optimal off-line path requires construction of the EMG for the given environment. This step requires computation of the c-obstacles associated with the disc robot. One next computes all potential transition lines between neighboring c-obstacles. This computation can be done in $O\left(n \log n\right)$ steps by building and searching the Voronoi diagram of the environment [11]. The redundancy check for all potential transition lines can be done in $O(n^2)$ steps. All non-redundant transition lines are added to the EMG, which is used to compute the following path under the local exposure constraint.

Optimal off-line path: *The optimal local exposure off-line path, l^*_{local}, is defined as the shortest path among all paths α from S to T along the EMG for which no transition line l_{tr} is longer than L_{max}:*

$$l^*_{local} = \min\left\{l\left(\alpha\right) \,:\, \alpha \in EMG, \quad l(l_{tr}) \leq L_{max}\right\} \tag{1}$$

where $l(l_{tr})$ is the length of l_{tr}, $l(\alpha)$ is the total length of the path α from S to T.

The optimal off-line path, l^*_{local}, is the shortest path from S to T in the EMG that satisfies the local exposure constraint, L_{max}, between neighboring obstacles in the environment. To compute this path, all transition lines l_{tr} with length

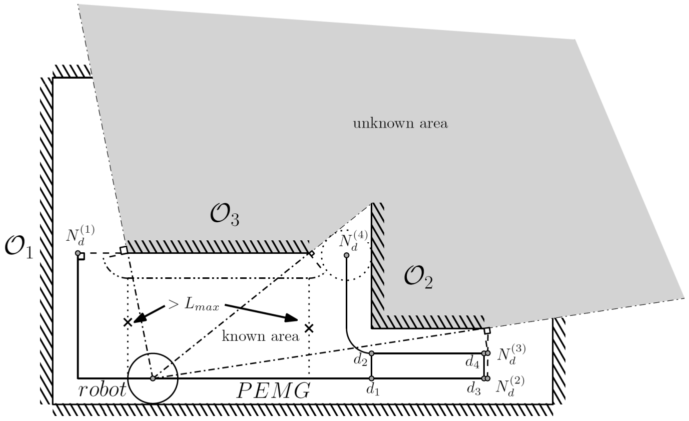


Fig. 4: Initial PEMG of an unknown environment before the robot starts moving.

greater than L_{max} are purged from the EMG. This stage takes $O(n)$ steps. The shortest path from S to T can now be found in the EMG using standard graph search algorithms such as Dijkstra. The purged EMG contains $O(n)$ nodes and $O(n)$ edges, hence l^*_{local} can be computed in $O(n \log n)$ steps.

4 The On-line Evasive Navigation Algorithm

The on-line algorithm described in this section is based on the off-line scheme. The section starts with an overview of the on-line algorithm, followed by a description of the *partial evasive motion graph,* then pseudo-code and a detailed execution example of the on-line algorithm.

Overview of the algorithm: The E-Nav algorithm works as follows. 1) The robot searches for an evasive path to the target in a series of expanding ellipses having focal points at S and T. 2) The robot constructs a *partial evasive motion graph,* PEMG, based on the currently known environment. 3) The robot deletes from the PEMG redundant transition lines and adds temporary transition lines that maintain connectivity of the PEMG in the current search ellipse. 4) The robot moves to the nearest unexplored endpoint of the current PEMG within the current search ellipse. 5) If none exists, the search ellipse is enlarged until it contains an unexplored endpoint of the current PEMG. The robot is assumed to scan the environment during movement between the nodes of PEMG. These steps repeat until either the target T becomes a node of the current PEMG (in which case the robot moves to T along the PEMG), or until the target is found to be unreachable and the robot returns to the start point S.

The initial ellipse area, $\mathcal{A}_0$, is chosen as the *larger* of two ellipses: 1) The smallest ellipse with focal points S and T that contains a rectangle with side lengths $\frac{\sqrt{3}}{2}D \times \frac{3}{2}D$. 2) The smallest ellipse with focal points S and T that contains the area swept by the robot while moving along a straight line from S to T. The ellipse determined by condition (1) is important for competitiveness (Section 5). If the robot cannot continue exploring the environment within the current search ellipse, an enlarged search ellipse with focal points S and T is formed by *doubling* its area $\mathcal{A}_i = 2\mathcal{A}_{i-1} = 2^i \mathcal{A}_0$.

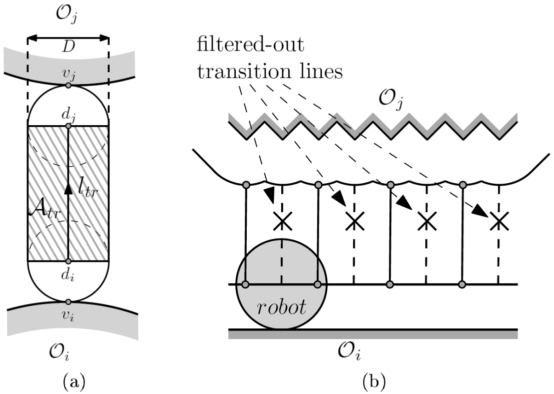


Fig. 5: (a) Transition area $\mathcal{A}_{tr}$ of a transition line l_{tr}. (b) New transition lines are filtered out when they intersect the area $\mathcal{A}_{tr}$ of existing transition lines.

Construction of the PEMG: During execution of the on-line algorithm, the robot constructs a *partial evasive motion graph,* PEMG, based on the currently known environment (Figure 4). The PEMG is constructed based on *conclusive knowledge* of the environment. In particular, no new nodes of the PEMG will be formed inside edges of the current PEMG as a result of subsequent measurements of the environment. The PEMG is constructed as follows.

1. Compute all segments of the known contact preserving curves $C_1, \ldots, C_m$ of the obstacles in the partial map of the environment.
2. Delete segments of the contact preserving curves whose minimal distance to *unexplored regions* is smaller than L_{max} (Figure 4). The cut-off points of the contact preserving curves form *decision nodes* of the PEMG.
3. Compute all potential transition lines between the contact preserving curves. Delete transition lines whose length exceeds L_{max}.
4. Filter out redundant transition lines. If there is not enough information to decide whether a transition line is non-redundant, its endpoint is added as a decision node to the current PEMG.
5. Filter out new transition lines that intersect the transition area of existing transition lines (see text and Figure 5(b)).
6. Add *temporary transition lines* to maintain connectivity of the PEMG within the current search ellipse (see text and Figure 6).

The *current PEMG* is defined as the connected subgraph computed in steps 1–6 that includes the current robot position. As long as the target has not become a node of the current PEMG, the robot moves to the nearest unvisited decision node, denoted $N_d^{(i)}$, of the current PEMG. As the robot proceeds with the evasive navigation task, decision nodes can disappear or change location when additional information about the environment becomes available. In step 5, the removal of unnecessary long transition lines is based on the following *transition area.*

Transition Area: *Let l_{tr} be a transition line with endpoints v_i and v_j. The* **transition area** *of this line, $\boldsymbol{\mathcal{A}_{tr}}$, is the area swept by the robot while traversing $\boldsymbol{l_{tr}}$.*

Any new transition line that intersects the transition area of existing transition lines is filtered out in step 5 (Figure 5(b)). This filtering is crucial for establishing the competitiveness of the algorithm in Section 5. The current PEMG forms a connected graph that contains the current robot position. However, the portion of the PEMG within the current search ellipse may *not* maintain connectivity after step 5. Hence, *temporary transition lines* are added in step 6. These temporary transition lines are built close to the boundary of the current search ellipse as illustrated in Figure 6. Their length is locally minimal among all feasible transition lines in their vicinity. It can be shown that all temporary transition lines of length smaller than L_{max} suffice to maintain connectivity of the PEMG in the search ellipse [22]. Temporary transition lines are either changed or deleted when the search ellipse is enlarged in subsequent steps of the on-line algorithm.

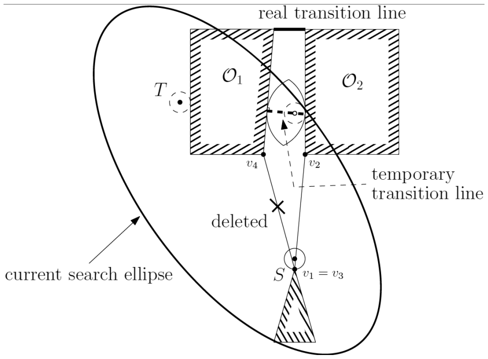


Fig. 6: Temporary transition lines ensure connectivity of the PEMG inside the current search ellipse. These lines are eventually deleted from the PEMG.

Algorithm 1 E-Nav: On-Line Evasive Navigation Algorithm

```
Sensors: Position and omnidirectional obstacle detection sensors.
Input: Target point $T$, maximum local exposure length $L_{max}$.
Initialization: Set $i = 1$. Set initial search ellipse with focal points $S$ and $T$.
Repeat
  1. Set $j = 1$. Set $V_1$ as the robot's current decision node.
  2. Delete all temporary transition lines from the current PEMG.
  3. Repeat
     (a) Construct the traceable curves $C_1, \ldots, C_m$ for the currently known map.
     (b) If $T$ is inside an entire c-obstacle in the known map: return to $S$ and STOP
         (target is unreachable).
     (c) Update the PEMG based on the current map of the environment.
     (d) If $T \in PEMG$ and there is a path from $V_j$ to $T$ inside the current search ellipse
         of area $\mathcal{A}_i$: Move to $T$ and STOP (target is found).
     (e) If there are no unvisited decision nodes in the current PEMG: return to $S$ and
         STOP (target is unreachable).
     (f) If all unvisited decision nodes are inaccessible within the current search ellipse:
         exit the current loop and go to step 4.
     (g) Move to the decision node closest to the current robot position in the PEMG.
         Update the map of the environment as the robot moves to new decision node.
     (h) Set $V_{j+1}$ as the decision node to which the robot moved. Set $j = j + 1$.
  4. Double the search ellipse area: $\mathcal{A}_{i+1} = 2\mathcal{A}_i$. Set $i = i+1$ (end of outer repeat loop)
```

The E-Nav algorithm is summarized as Algorithm 1. A step-by-step execution example of the algorithm is next described.

Execution example: Fig. 7(a) shows the robot at the start point S located on the obstacle $\mathcal{O}_1$. The target point T is located on the obstacle $\mathcal{O}_5$. At this initial stage, the robot is *not* allowed to make direct transition from $\mathcal{O}_1$ to $\mathcal{O}_5$, as this transition is longer than L_{max}. There is only one decision node inside the initial search ellipse, $N_d^{(1)}$, and the robot moves to $N_d^{(1)}$ as seen in Fig. 7(b). The robot next scans the environment and constructs a transition line between $\mathcal{O}_1$ and $\mathcal{O}_3$. The interference area of this transition line, $\mathcal{A}_{int}$, overlaps the obstacle $\mathcal{O}_2$ and is therefore filtered out (Fig. 7(b)). The robot now moves along the shorter transition line no. 1 to decision node $N_d^{(2)}$, located on $\mathcal{O}_2$ (Fig. 7(b)). The next stage is depicted in Fig. 7(c). The robot *cannot* continue to any new decision node while remaining inside the current search ellipse. The search ellipse is therefore enlarged. The larger ellipse contains two new decision nodes, $N_d^{(3)}$ and $N_d^{(4)}$ (Fig. 7(c)). Additionally, the decision node $N_d^{(5)}$ on $\mathcal{O}_4$ is now reachable. The robot moves to the closest decision node, $N_d^{(3)}$, located on $\mathcal{O}_3$, while passing through transition line no. 2 (Fig. 7(c)).

The next stage is depicted in Fig. 7(d). The robot discovers three new decision nodes: $N_d^{(6)}$ on the current obstacle $\mathcal{O}_3$ and $N_d^{(7)}$ on $\mathcal{O}_4$. The closest decision node on the current PEMG is $N_d^{(7)}$. The path to this node passes through transition line no. 3 (Fig. 7(d)-(e)). At $N_d^{(7)}$ the robot constructs two new transition lines (Fig. 7(e)). One of these transition lines connects $\mathcal{O}_3$ with $\mathcal{O}_5$. This transition line lies *outside* the current search ellipse. Hence, a temporary transition line no. 5 is built as the shortest line connecting $\mathcal{O}_3$ with $\mathcal{O}_5$ inside the current search ellipse (Fig. 7(e)). The other new transition line connects $\mathcal{O}_4$ with $\mathcal{O}_5$. This new transition line intersects the transition area, $\mathcal{A}_{tr}$, of the existing transition line no. 4 (Fig. 7(e)), and is therefore filtered out.

To reach the closest unvisited decision node in the current PEMG, $N_d^{(8)}$ located on $\mathcal{O}_5$, the robot moves along transition line no. 4 back to $\mathcal{O}_3$. The robot then moves along the temporary transition line no. 5 to $\mathcal{O}_5$ (Fig. 7(f)). When the robot reaches the new decision node, $N_d^{(8)}$, it finds that no new decision nodes lie in the current search ellipse. The search ellipse is therefore enlarged once more (Fig. 7(f)). Consequently, the temporary transition line no. 5 is deleted and replaced by the true transition line no. 5 (Fig. 7(f)). The last step is depicted in Fig. 7(g). The robot circumnavigates the obstacle $\mathcal{O}_5$ to the decision node $N_d^{(9)}$, where it detects the target T. The target becomes a node of the current PEMG, and the robot moves along the shortest PEMG path to T (Fig. 7(g)). As seen in Fig. 7(h), the evasive navigation path taken by the robot from S to T has been completed with very little exposure of the robot to the open environment. The optimal off-line path is also shown in Fig. 7(h).

5 Competitive Upper Bound on the E-Nav Algorithm

The E-Nav algorithm is next analyzed using the notion of *competitiveness.* It compares the solution of an on-line algorithm, $l(alg)$, to the optimal off-line solution, l_{opt}, over all instances of the problem. It usually takes the form $l(alg) \leq c_1 l_{opt} + c_0$. Here we need the following generalized notion of competitiveness.

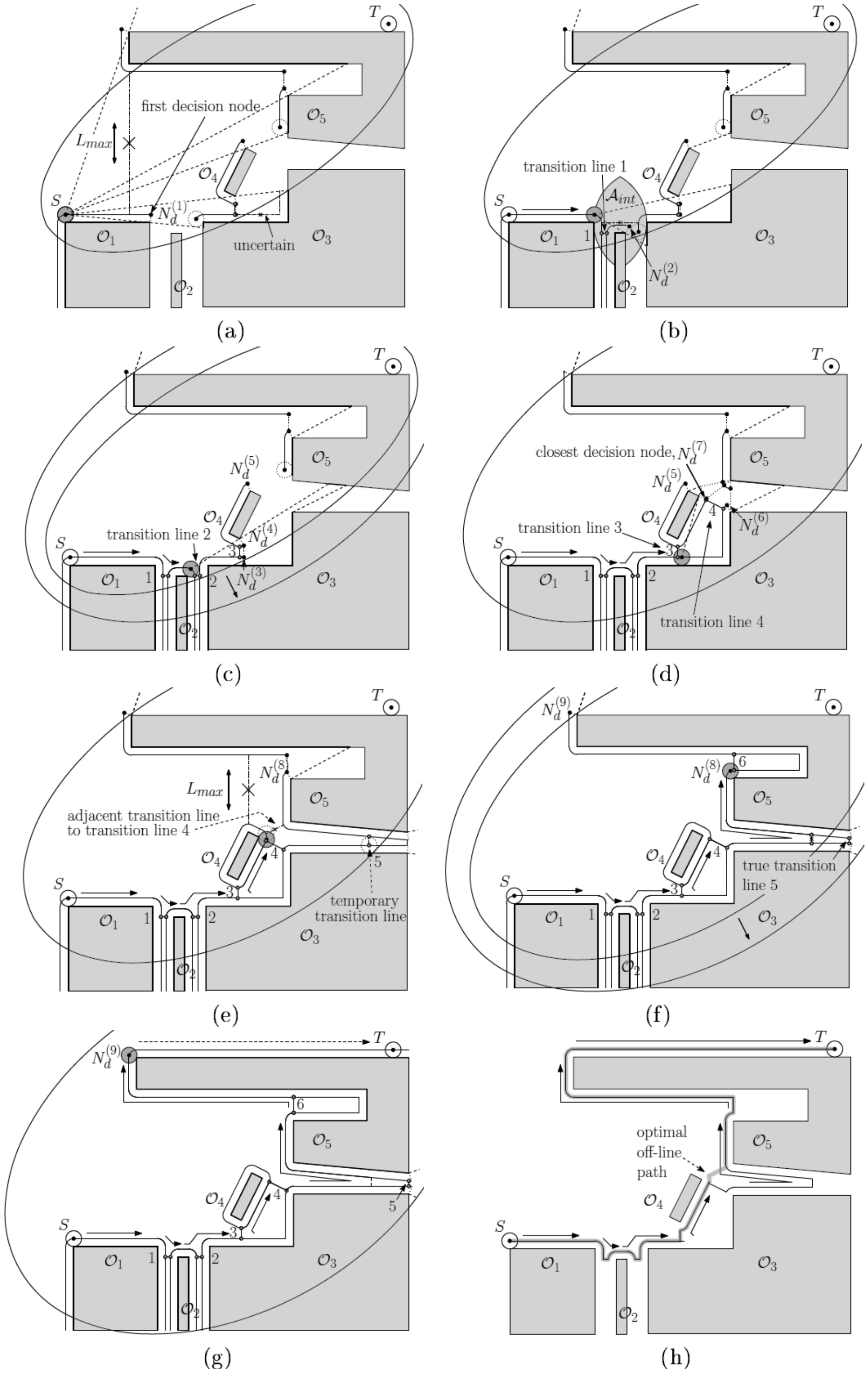


Fig. 7: Execution example of the E-Nav algorithm (see text). The search ellipse area is only approximately doubled in each step to better illustrate the algorithm.

Generalized competitiveness [6]: *Let $\mathcal{T}$ be an on-line mobile robot task. An algorithm for $\mathcal{T}$ is $f(l_{opt})$-competitive when its path length is bounded by a scalable function $f(l_{opt})$ over all instances of $\mathcal{T}$. In particular, $l(alg) \le c_2 l_{opt}^2 + c_1 l_{opt} + c_0$ is* **quadratic competitiveness,** *where c_0, c_1, c_2 are positive constants that depend only on apriori information such as the robot size D.*

Scalability means that the inequality $l(alg) \le f(l_{opt})$ is invariant under change of length units. The competitiveness of E-Nav will be analyzed in two steps. First we obtain an upper bound on the E-Nav algorithm path length within the i^{th} search ellipse, denoted l_i(E-Nav). As shown below, this upper bound is

$$l_i(\text{E-Nav}) \le (1 + \log_2 N_i) \cdot Weight_i(\text{PEMG}) \tag{2}$$

where N_i is the number of *decision nodes* the robot visited in the i^{th} search ellipse, and $Weight_i$(PEMG) is the sum of all edge lengths of the PEMG in the i^{th} search ellipse. Summing the bound of Eq. (2) over all search ellipses gives the *log-quadratic* competitive bound on the E-Nav algorithm:

$$l(\text{E-Nav}) \le 7\pi \left(\frac{3}{2} + \log_2 \left(\frac{8\pi}{\sqrt{3}} \frac{l_{opt}^2}{D^2} \right) \right) \cdot \frac{l_{opt}^2}{D} + c_0 \tag{3}$$

where $c_0 = \frac{7\mathcal{A}_0}{D}\left(2 + \log_2\left(\frac{16\mathcal{A}_0}{\sqrt{3}D^2}\right)\right)$. Note that D and $\mathcal{A}_0$ are known to the robot.

5.1 Upper Bound on the Robot On-Line Path Length

The E-Nav algorithm guides the robot between decision nodes using a nearest neighbor approach. The nearest neighbor approach has been analyzed by Koenig [12] in the context of on-line graph exploration. The same technique is used here to obtain an upper bound on the E-Nav algorithm path length.

The robot's on-line path can be described by the decision nodes visited in the i^{th} search ellipse, $N_d^{(1)}(i) \to \cdots \to N_d^{(k)}(i)$, augmented with the initial decision node V_i. In every step the robot moves from $N_d^{(j)}(i)$ to the decision node $N_d^{(j+1)}(i)$ that is closest along the current PEMG in the i^{th} search ellipse. The current PEMG is constructed based on *conclusive knowledge* of the environment. Hence the online path from $N_d^{(j)}(i)$ to $N_d^{(j+1)}(i)$ would *not* change even if the entire final PEMG was apriori known to the robot, since any alternative path from $N_d^{(j)}(i)$ to $N_d^{(j+1)}(i)$ must pass through a decision node that is *farther away* from $N_d^{(j)}(i)$ than $N_d^{(j+1)}(i)$. Using the same reasoning, the path from $N_d^{(j)}(i)$ to any decision node $v \in \{N_d^{(j+2)}(i), \ldots, N_d^{(k)}(i)\}$ along the PEMG is *longer* than the path to $N_d^{(j+1)}(i)$. It follows that the on-line path is identical to the off-line path that starts at V_i and visits the nodes $N_d^{(1)}(i) \to \cdots \to N_d^{(k)}(i)$ under a nearest neighbor approach along the PEMG in the i^{th} search ellipse. The length of this off-line path is denoted $l_{offNN}(i)$. After reaching the last decision node, the robot travels to T along a path of length l_T on the final PEMG. Hence, $l(\text{E-Nav}) = \sum_{i=1}^{n} l_{offNN}(i) + l_T$.

To obtain an upper bound on $l_{offNN}(i)$, consider the auxiliary graph, PEMG', whose nodes are $\{V_i, N_d^{(1)}(i), \ldots, N_d^{(k)}(i)\}$ and whose edges are the *shortest paths* between the respective nodes along the PEMG in the i^{th} search ellipse. The off-line path of length $l_{offNN}(i)$ visits all nodes of the PEMG' using a nearest neighbor approach. The edge lengths of the PEMG' are symmetric and satisfy the triangle inequality. Rosenkrantz [19] provides the following *logarithmic* bound on the nearest-neighbor path in such graphs. Let l_{TSP} denote the shortest *traveling* *salesman path* (TSP) that visits all nodes $\{V_i, N_d^{(1)}(i), \ldots, N_d^{(k)}(i)\}$ of the PEMG' (a complete graph). The length of the nearest neighbor path, $l_{offNN}(i)$, satisfies

$$l_{offNN}(i) \leq (\tfrac{1}{2} + \tfrac{1}{2}\log_2 N_i) \cdot l_{TSP} \tag{4}$$

where N_i is the number of nodes in the PEMG'. The TSP path satisfies the inequality $l_{TSP} \leq 2 \cdot Weight_i(\text{PEMG})$, since a simple DFS traversal of the PEMG in the i^{th} search ellipse visits all nodes of the PEMG' while traversing each edge twice. It follows from Eq. (4) that $l_{offNN}(i) \leq (1 + \log_2 N_i) \cdot Weight_i(\text{PEMG})$. After reaching the last decision node, $N_d^{(k)}(n)$, the robot travels at most $Weight_n$ (PEMG) to T. The upper bound on the E-Nav algorithm path length is thus

$$l(\text{E-Nav}) \leq \sum_{i=1}^{n} \Big(1 + \log_2 N_i\Big) \cdot Weight_i(\text{PEMG}) + Weight_n(\text{PEMG}) \tag{5}$$

Maximum number of decision nodes: The maximum number of decision nodes, $|\mathcal{N}_d|$, visited by the robot is computed in three steps (see Ref. [22]). First, when the robot is located at a particular decision node, the minimum distance to any other decision node is found to be the robot radius $D/2$. Next, the plane is tiled with a triangular lattice of equilateral triangles whose edge-length equals $D/2$. It is then shown that for any bounded convex region in $I\!R^2$ that contains a rectangle of size $\frac{\sqrt{3}}{2}D \times \frac{3}{2}D$, the number of decision nodes inside this region is bounded by $\mathcal{A}/\Delta$, where $\mathcal{A}$ is the area of the convex region and Δ is the area of a single equilateral triangle in the triangular lattice. Since $\Delta = \frac{\sqrt{3}}{4}(\frac{D}{2})^2$, the upper bound on the number of decision nodes the robot visits follows. (Note that this is a non-trivial bound since E-Nav is *not* grid based.)

Lemma 5.1. *The number of decision nodes the robot visits during execution of the E-Nav algorithm, $|\mathcal{N}_d|$, in any bounded convex region that contains a rectangle of size $\frac{\sqrt{3}}{2}D \times \frac{3}{2}D$ is bounded by*

$$|\mathcal{N}_d| \leq \frac{16\mathcal{A}}{\sqrt{3}D^2} \tag{6}$$

where D is the robot size and $\mathcal{A}$ is the area of the convex region.

5.2 Competitive Bound on E-Nav

The bound obtained in Eq. (5) is now expressed in terms of the optimal evasive off-line path length, l_{opt}. During the i^{th} step of the E-Nav algorithm, the robot moves inside a search ellipse with focal points S and T and area that doubles in each step. The following lemma was proved in the original C-BUG paper [6]:

Lemma 5.2 [The 4-Lemma]. *Let a disc robot move in an environment populated by traceable obstacles $Q_1, \ldots, Q_m$. Let $\mathcal{A}_j$ be the total area swept by the robot diameter during circumnavigation of Q_j. Let c be any simple closed curve that surrounds the m regions swept by the robot. Then $\sum_{j=1}^{m} \mathcal{A}_j \leq 4\mathcal{A}(c)$, where $\mathcal{A}(c)$ is the area of the obstacle-free points enclosed by c.*

The proof of Lemma 5.2 in [6] uses only the definition of contact preserving curves and is therefore applicable to the E-Nav algorithm. By construction of the PEMG, adjacent transition lines are filtered out so that no transition lines intersect the transition area of adjacent transition lines. As shown in [22], this property implies that there *cannot* be an overlap of more than three transition areas associated with the remaining transition lines.

Lemma 5.3. *Let the PEMG contain transition-lines $l_{tr,1},\ldots,l_{tr,n}$. Let $\mathcal{A}_{tr,j}$ be the area swept by the robot diameter during motion along transition line $l_{tr,j}$. Let c be any closed convex curve that surrounds the n regions swept by the robot. Then $\sum_{j=1}^{n}\mathcal{A}_{tr,j}\leq 3\mathcal{A}\left(c\right)$ where $\mathcal{A}\left(c\right)$ is the area enclosed by c.*

The total area swept by the robot while moving along the PEMG can be bounded by Lemmas 5.2 and 5.3, as stated in the following proposition.

Proposition 5.4. *Let a disc robot move in an environment populated by traceable obstacles $Q_1,\ldots,Q_m$. Let $\mathcal{A}_i(PEMG)$ be the total area swept by the robot diameter during motion on the PEMG within the i^{th} search ellipse. Then*

$$\mathcal{A}_i(PEMG)\leq 3\mathcal{A}\left(i\right)+4\mathcal{A}\left(i\right)$$

where $\mathcal{A}\left(i\right)$ is the area of the i^{th} search ellipse.

The above bound is next used to obtain a bound on the robot's path length inside the i^{th} search ellipse.

Proposition 5.5. *The length l_i of the path traveled by the robot in the i^{th} search ellipse is bounded by $l_i\leq\left(1+\log_2\left(\frac{16\mathcal{A}(i)}{\sqrt{3}D^2}\right)\right)\cdot\frac{7\mathcal{A}(i)}{D}$, where $\mathcal{A}\left(i\right)$ is the i^{th} search ellipse area and D is the robot diameter.*

Proof Sketch: Substituting the bound $N_i\leq\frac{16\mathcal{A}(i)}{\sqrt{3}D^2}$ of Eq. (6) into Eq. (5), the length l_i is bounded by

$$l_i\leq\left(1+\log_2\left(\frac{16\mathcal{A}(i)}{\sqrt{3}D^2}\right)\right)\cdot Weight_i(\text{PEMG})$$

where $Weight_i\left(\text{PEMG}\right)$ is the total length of edges of the PEMG in the i^{th} search ellipse. For a size D robot, $Weight_i\left(\text{PEMG}\right)=\mathcal{A}_i(\text{PEMG})\,/D$, where $\mathcal{A}_i(\text{PEMG})$ is the total area swept by the robot diameter while traversing the PEMG in the i^{th} search ellipse. Since $\mathcal{A}_i(\text{PEMG})\leq 7\mathcal{A}\left(i\right)$ according to Proposition 5.4, the path length bound is $l_i\leq\left(1+\log_2\left(\frac{16\mathcal{A}(i)}{\sqrt{3}D^2}\right)\right)\cdot\frac{7\mathcal{A}(i)}{D}$. □

Any path of length l_{opt} from S to T lies inside an ellipse with focal points S and T and major axis of length $2a=l_{opt}$. The area of the latter ellipse is $\mathcal{A}_{min}=\frac{\pi}{4}l_{opt}\sqrt{l_{opt}^2-\|S-T\|^2}$. Let $\mathcal{A}\left(1\right),\ldots,\mathcal{A}\left(n\right)$ be the series of search ellipses used by the E-Nav algorithm. Since E-Nav finds an evasive path to T in the n^{th} ellipse, the shortest evasive path of length l_{opt} also lies in the n^{th} ellipse. The PEMG in the $(n{-}1)^{th}$ search ellipse does *not* contain an evasive path to T. Hence its area satisfies the inequality $\mathcal{A}\left(n{-}1\right)<\mathcal{A}_{min}$, or else the optimal path would have to lie inside it. As $\mathcal{A}\left(n\right)=2\mathcal{A}\left(n{-}1\right)$, $\mathcal{A}\left(n\right)\leq 2\mathcal{A}_{min}$. The bound on the total path length of the E-Nav algorithm follows.

Theorem 1 [E-NAV Bound]. *If the target T is reachable from S under the local exposure constraint, the E-Nav algorithm finds the target using a path whose length satisfies the inequality*

$$l\left(E\text{-}Nav\right)\leq 7\pi\left(\frac{3}{2}+\log_2\left(\frac{8\pi}{\sqrt{3}}\frac{l_{opt}^2}{D^2}\right)\right)\frac{l_{opt}^2}{D}+c_0$$

where $c_0=\frac{7\mathcal{A}_0}{D}\left(2+\log_2\left(\frac{16\mathcal{A}_0}{\sqrt{3}D^2}\right)\right)$. Here D is the robot diameter, $\mathcal{A}_0$ is the initial ellipse area, and l_{opt} is the length of the optimal evasive off-line path from S to T.

Proof: Suppose the search ellipse is expanded n times until the target is found. The total length of the path traveled by the robot is a sum of the paths taken in the individual search ellipses, $\sum_{i=1}^{n}l_i$, plus the path length from the last decision node to T. Using Proposition 5.5,

$$l\left(E\text{-}Nav\right)\leq\sum_{i=1}^{n}\left(1+\log_2\left(\frac{16\mathcal{A}(i)}{\sqrt{3}D^2}\right)\right)\cdot\frac{7\mathcal{A}(i)}{D}+\frac{7\mathcal{A}(n)}{D}.$$

Equivalently

$l\,(E\text{-}Nav) \leq \frac{7}{D}\left\{\left(1+\log_2\left(\frac{16}{\sqrt{3}D^2}\right)\right)\cdot\sum_{i=1}^{n}\mathcal{A}\,(i) + \sum_{i=1}^{n}\log_2\left(\mathcal{A}\,(i)\right)\cdot\mathcal{A}\,(i)+\mathcal{A}(n)\right\}.$

The search ellipse area doubles at each stage. Hence the first summand in this equation can be bounded by $\sum_{i=1}^{n}\mathcal{A}\,(i)<2^n\mathcal{A}_0=2\mathcal{A}\,(n)$. The second summand can be bounded by $\sum_{i=1}^{n}\log_2\left(\mathcal{A}\,(i)\right)\cdot\mathcal{A}\,(i)<\log_2\left(\mathcal{A}\,(n)\right)\sum_{i=1}^{n}\mathcal{A}\,(i)<\log_2\left(\mathcal{A}\,(n)\right)\cdot 2\mathcal{A}\,(n)$. The total length of the path traveled by the robot is thus

$$l\,(E\text{-}Nav) \leq 7\left\{\left(\frac{3}{2}+\log_2\left(\frac{16\mathcal{A}(n)}{\sqrt{3}D^2}\right)\right)\cdot\frac{2\mathcal{A}(n)}{D}\right\}.$$

Since $\mathcal{A}\,(n) \leq 2\mathcal{A}_{min} = \frac{\pi}{2}l_{opt}\sqrt{l_{opt}^2-\|S-T\|^2} \leq \frac{\pi}{2}l_{opt}^2$,

$$l\,(E\text{-}Nav) \leq 7\pi\left(\frac{3}{2}+\log_2\left(\frac{8\pi l_{opt}^2}{\sqrt{3}D^2}\right)\right)\cdot\frac{l_{opt}^2}{D}.$$

The constant additive term, c_0, bounds the path length traveled by the robot when the initial search ellipse already contains an evasive path from S to T. □

6 Conclusion

The E-Nav algorithm navigates a disc robot in a planar unknown environment along an evasive path that strives to minimize exposure of the robot to potential threats whose position is unknown to the robot. The algorithm guides the robot along a *partial evasive motion graph,* PEMG, constructed in the currently known environment. The PEMG consists of obstacle-hugging segments and locally shortest transition lines whose length is bounded by a specified *local exposure* constraint. As the robot navigates toward the target T using a nearest neighbor approach, the PEMG of the environment is monotonically enlarged until T becomes a node of the current PEMG, or until the robot determines that T is not accessible. Execution of a nearest neighbor approach within a series of search ellipses was shown to be *log-quadratic* in terms of the optimal evasive off-line path length.

The following two topics were not included in the paper. Ref. [22] describes an adversarial environment of narrow corridors that forces every evasive on-line algorithm to generate a path of length $l(alg) \geq l_{opt}^2/D$. The log-quadratic upper bound on E-Nav is thus close to the universal lower bound over all on-line evasive navigation algorithms. Ref. [22] also studies the paths generated by E-Nav in a simulated hostile compound and in an urban road crossing environment. The E-Nav paths were compared to those of the on-line TangentBug algorithm [10], that uses the same sensors and generates close to optimal *non-evasive* paths. The path generated by E-Nav in the compound environment is three time longer than the TangentBug path, and five time longer in the road-crossing environment. However, the paths generated by E-Nav are much safer. The maximum local exposure length for the compound environment is almost *seven times shorter* than the maximum exposure during execution of TangentBug, while it is more than two times shorter in the road-crossing environment.

The paper tacitly assumed that the robot's obstacle detection scanner can observe all features of the surrounding environment. In future research, we will study *evasive navigation* with a myopic scanner that detects obstacles only within a finite detection range, R_{max}, that is larger than the local exposure constraint, $R_{max} \geq L_{max}$. The partial evasive motion graph, PEMG, can be augmented with *temporary decision nodes* located on the boundary of the known environment measured with finite detection range. Similarly, by limiting the navigation to a series of expanding ellipses, competitive navigation will be ensured under finite detection range. However, one needs to verify that the robot can accumulate *conclusive knowledge* of the environment that is key for the efficiency of a nearest neighbor approach. That is, the PEMG augmented with temporary decision nodes must be monotonically increasing throughout the navigation process. This and other challenging problems will be addressed in future research.

References

1. N. Agmon. *Multi-Robot Patrolling and Other Multi-Robot Cooperative Tasks: An Algorithmic Approach.* PhD thesis, Bar Ilan University, 2009.
2. T. Bandyopadhyay, Y. Li, M. H. Ang, D. Hsu. Stealth tracking of an unpredictable target among obstacles. *Algorithmic Foundations of Robotics VI*, 17:43–58, 2005.
3. A. Blum, P. Raghavan, and B. Schieber. Navigating in unfamiliar terrain. In *STOC*, 494–504, 1991.
4. A. Blum, P. Raghavan, and B. Schieber. Navigating in unfamiliar geometric terrain. *SIAM J. of Comput.*, 26:110–137, 1997.
5. Y. Elmaliach, N. Agmon, and G. A. Kaminka. Multi-robot area patrol under frequency constraints. *Annals of Math and AI*, 57(3–4):293–320, 2010.
6. Y. Gabriely and E. Rimon. CBUG: a quadratically competitive mobile robot navigation algorithm. *IEEE Trans. on Robotics*, 6:1451–1457, 2008.
7. J. E. Goodman, J. O. Rourke, and C. D. Toth. *Handbook of Discrete and Computational Geometry, 3'rd Ed.* CRC Press, 2018.
8. V. Isler, S. Kannan, and S. Khann. Randomized pursuit-evasion with local visibility. *SIAM J. on Discrete Mathematics*, 1(20):26–41, 2006.
9. V. Isler, S. Kannan, and S. Khanna. Randomized pursuit-evasion in a polygonal environment. *IEEE Trans. on Robotics*, 5(2):864–875, 2005.
10. I. Kamon, E. Rimon, and E. Rivlin. Tangentbug: A range-sensor based navigation algorithm. *Int. J. of Robotics Research*, 17(9):934–953, 1998.
11. D. G. Kirkpatrick. Efficient computation of continuous skeletons. In *IEEE Symposium on Foundations of Computer Science, FOCS*, 18–27, 1979.
12. S. Koenig and Y. Smirnov. Graph learning with a nearest neighbor approach. In *Conference on Computational Learning Theory*, 19–28, 1996.
13. S. M. LaValle. *Planning Algorithms.* Cambridge University Publishers, 2006.
14. V. J. Lumelsky and A. Stepanov. Path planning strategies for point automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, 2:403–430,1987.
15. L. Matthies et al. A portable, autonomous, urban reconnaissance robot. *Robotics and Autonomous Systems*, 57(2-3):163–172, 2002.
16. F. Meddah and L. Dib. P*: a new path planning algorithm for autonomous robot in unknown environment. *Int. J. of Adv. in Comp. Science and Its App.*, 15–19, 2015.
17. J.-H. Park, J.-S. Choi, J. Kim, and B.-H. Lee. Roadmap-based stealth navigation for intercepting an invader. In *ICRA*, 442–447, 2009.
18. Ravela, Weiss, Draper, Pinette, Hanson, and Riseman. Stealth navigation: Planning and behaviors. *DARPA Image Understanding Workshop*, 1093–1100, 1994.
19. D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis. An analysis of several heuristics for the traveling salesman problem. *SIAM J. of Computing*, 6(3):563–581, 1977.
20. A. Sankaranarayanan and M. Vidyasagar. Path planning for moving a point object amidst unknown obstacles in a plane. In *ICRA*, 1734–1941, 1991.
21. N. M. Stiffler, A. Kolling, and J.M. O'Kane. Persistent pursuit-evasion: The case of the preoccupied pursuer. In *ICRA*, 5027–5034, 2017.
22. Y. Veksler and Rimon. Evasive navigation of an autonomous mobile robot in hostile unknown environments. https://robots.net.technion.ac.il/publications, Jan. 2020.