

Coverage Path Planning For Minimizing Expected Time to Search For an Object With Continuous Sensing

Linh Nguyen^[0009–0009–3518–929X]

Stony Brook University, Stony Brook NY 11794, USA
linh.nguyen.1@stonybrook.edu

Abstract. We present several results of both theoretical as well as practical interests in optimal search theory. First, we propose the quota lawn mowing problem, an extension of the classic lawn mowing problem in computational geometry, as follows: given a quota of coverage, compute the shortest lawn mowing route to achieve said quota. We give constant-factor approximations for the quota lawn mowing problem.

Second, we investigate the expected detection time minimization problem in geometric coverage path planning with local, continuous sensory information. We provide the first approximation algorithm with provable error bounds that runs in pseudopolynomial time. Our ideas also extend to another search mechanism, namely visibility-based search, which is related to the watchman route problem. We complement our theoretical analysis with some simple but effective heuristics for finding an object in minimum expected time, on which we provide simulation results.

Keywords: Quota lawn mowing problem · Expected detection time .

1 Introduction and Related Work

In this work, we study search and path planning problems in *geometric domains*. Consider, for example, a remote-controlled camera inspecting a manufactured component for porosities, or a submarine using radar/sonar to search for a missing object, etc. We model this search, where the agent needs to be in proximity to the target for the sensor to detect it, by the *lawn mowing problem*, a classic computational geometry problem posed by Arkin, Fekete, and Mitchell in [2]. Given a planar region, compute a minimum-length route along which a circle or square (“mower”) can be slid so that every point in the region is “mowed”. We investigate a natural generalization of the lawn mowing problem, in which we are given a *quota* of coverage, and we are to compute the shortest route to achieve said quota. This is relevant in time-sensitive or resource-constrained scenarios, where complete coverage may not be feasible or necessary. Another related problem, which shares a similar detection mechanism with the lawn mowing problem is the honey-pot search problem [11], in which a searcher must maximize the probability of finding a target within the given time budget.

The equivalence of the lawn mowing problem for visibility-based search is the watchman route problem, introduced by Chin and Ntafos [9], where the agent needs to see the target: “seeing” means establishing a line of vision that lies entirely within the domain. Given a polygon, the watchman route problem asks for a minimum-length route that sees the whole polygon.

Both the lawn mowing problem and the watchman route problem, when stated as search problems, are concerned with the *worst-case* guarantee, as are many previous studies in coverage path planning, see the surveys [8, 32]. We study the *average-case*, i.e., we seek to optimize *expected* duration of the search, which is more beneficial in the long run if the search is to be carried out on a regular basis. Several studies have looked at minimizing expected detection time in a graph. In [6], Bernman, Ianovsky and Krass proposed an exponential-time dynamic programming algorithm for finding a uniformly distributed target, which Teller, Zofi and Kaspi later extended to non-uniform distributions while also presenting a branch-and-bound algorithm along with several heuristics [33]. Sarmiento, Murrieta and Hutchinson were the first to investigate the expected detection time in visibility-based search [26, 27]. They proposed a heuristic that selects a set of points that collectively see the whole polygon (known as “guards”) and visits those points in an ordering resulting from minimizing a utility function.

The problem of minimizing expected detection time is also related to the minimum latency problem (also referred to in the literature as the traveling repairman problem). Given some nodes in a graph or some points in the plane, compute a route that minimizes the average delay of all the nodes. Several approximation algorithms are known [1, 4, 15], including a polynomial-time approximation scheme [30]. Our problem is a continuous minimum-latency type problem.

Most search problems involving path planning in a geometric domain, or even a discrete graph theoretic domain are known to be NP-hard [20, 34]. Optimal search is an overarching field of research that arises in numerous practical applications, including but not limited to: surveillance, robotics, search-and-rescue, etc. The theoretical basis for search theory was introduced in the seminal work of Koopman [21–23]. Depending on the task, perception, and target modeling, along with the objective function to be optimized, many different variations have been proposed and studied. We refer the reader to the surveys [10, 19, 25, 28, 36] for a comprehensive review.

We present several novel results on optimal search in a geometric setting:

1. For the quota lawn mowing problem, we provide theoretical approximation algorithms, the running time of which depends on the approximation algorithm for quota-TSP/MST involved.
2. We prove that minimizing the expected detection time is NP-hard in a simple polygon for both search mechanisms, lawn mowing and visibility-based search. Previously, only hardness in polygons with holes was known [26].
3. We provide the first pseudopolynomial-time approximation algorithm with provable error bounds for minimizing the expected detection time for both search mechanisms.

2 Preliminaries

We model the area to be searched by a closed planar region R , a subset of \mathbb{R}^2 . If R is connected and the boundary of R , ∂R , consists of a finite number of non-crossing line segments, then R is a *polygonal region*, or synonymously, a *polygon with holes*. If R has no holes, i.e., R is simply connected, then we call R a *simple polygon*. We assume R is a polygon with holes with n vertices, though our results can be generalized to more general regions (with regular curved arcs on the boundary or multiple connected components).

We are given a pointwise *robot/cutter*. Throughout this paper, we shall use the term “cutter” when discussing the lawn mowing problem and the term “robot” when discussing searching. Assume that the operational area of the robot/cutter, \mathcal{A} , is either an axis-aligned square or a circle centered on it. Without loss of generality, we can scale R so that \mathcal{A} is a unit square (side length 1) or a unit circle (radius 1). If the robot/cutter traverses a trajectory γ , the *area of coverage*, denoted $C(\gamma)$, is the Minkowski sum of γ with the operational area: $C(\gamma) = (\gamma + \mathcal{A}) \setminus R$. Note that, in the lawn mowing problem, γ need not stay within R . See Figure 1 for an illustration.

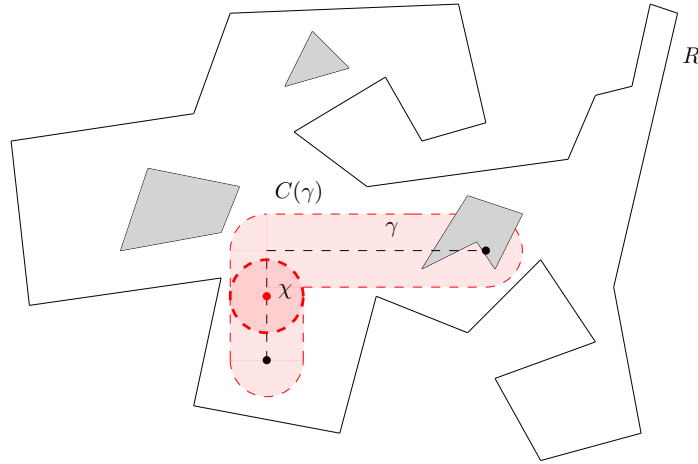


Fig. 1. The area of coverage $C(\gamma)$ (red) by a robot with circular operational area traveling along γ .

We use $|j|$ to denote Euclidean measure of geometric objects (e.g., length or area), as well as cardinality of finite sets, the meaning should be clear from the context. First, we consider the following problem:

The quota lawn mowing problem Given an area quota $0 < A < |R|$, compute the shortest tour (cycle) γ such that $|C(\gamma)| = A$.

This can be seen as a generalization of the lawn mowing problem, which is NP-hard even if the input region is a simple polygon [2]. Moreover, the lawn mowing problem is algebraically hard [12]. The same hardness for the quota lawn mowing problem follows naturally.

We consider how the lawn mower analogy can model some geometric optimal search problems. Suppose there is a *target* located within R . We assume that the location of the target is uniformly distributed, though our results can be easily adapted to non-uniform distributions, that is, when we have prior knowledge of some regions more likely to contain the target, by simply replacing area with probability measure. A robot, given a starting point S and equipped with a sensor, looks for the target. Think of a camera scanning a manufactured part for a small dent, or a submarine using radar/sonar to look for a missing ship/airplane on the ocean floor.

If the sensor can detect the target within a circle or square centered around the robot, we call it the *lawn mowing search mechanism*. If the sensor is visibility-based with infinite range, we have the classic watchman route problem (in which the geometry of the region R not only limits visibility but also movements), which we refer to as *visibility-based search mechanism*.

The lawn mowing problem and the watchman route problem are concerned with the worst-case scenario of the search process, where the objective is to minimize the distance traveled to guarantee detection of the target, i.e. every point in the domain must be covered. The quota lawn mowing problem introduced earlier, can be interpreted as that of computing a minimum route to guarantee a certain probability of detection. In many applications, we are also interested in an average-case analysis, thus we study the *expected detection time* as the objective function to optimize. Suppose the robot follows a trajectory γ such $C(\gamma) = R$. We define the random variable T to be the time at which the target is detected by the robot. We assume that the robot travels at a constant speed of 1, without loss of generality, so that distance and time are equivalent. Denote by $C(\gamma; t)$ the partial area of coverage accumulated by the robot by time t as it travels along γ (note that $C(\gamma; j) = C(\gamma) = R$). Then the probability of detection at time t is equal to the fraction of area covered: $\Pr_\gamma(T \leq t) = \frac{jC(\gamma; t)j}{jRj}$. Thus, since $T > 0$, the expected value of T can be written as

$$E[T] = \int_0^\infty \Pr_\gamma(T > t) dt = \int_0^\infty \left(1 - \frac{jC(\gamma; t)j}{jRj}\right) dt:$$

The expected detection time minimization problem Compute a route that minimizes $E[T]$.

3 Lawn Mowing Problem with a Quota

We begin by assuming the operational area of the cutter, R , is a closed unit square. As the cutter moves, R translates but does not rotate.

To achieve a quota of coverage, the cutter must sweep over every point in a subset of the domain R . The uncountably infinite number of points that the cutter must cover makes for a computational challenge. We overcome this challenge (at the cost of an approximation factor) by discretizing R . We denote by P the set of pixels whose vertices are integer grid points that have a non-empty intersection with R . If the cutter is given a starting point s , the grid inducing P is shifted if necessary so that s is at the center of a pixel. Denote by G the *dual grid graph* of P . Each node in G corresponds to the center point of a pixel in P , two nodes are adjacent if their pixels share a side edge (Figure 2). Let $N = |P|$, then we can identify G and P in $O(N + n \log n)$ from a description of R as an n -vertex polygonal domain.

We consider the input complexity to depend on the *magnitude* (the bit complexity) of the coordinates of the vertices of R . This allows for algorithms whose running time is *pseudopolynomial* (depending on the input size as well as magnitude of the input). This is because even for a relatively combinatorially simple region, a quota lawn mowing tour can have a very large number of turns. This was first observed in [2].

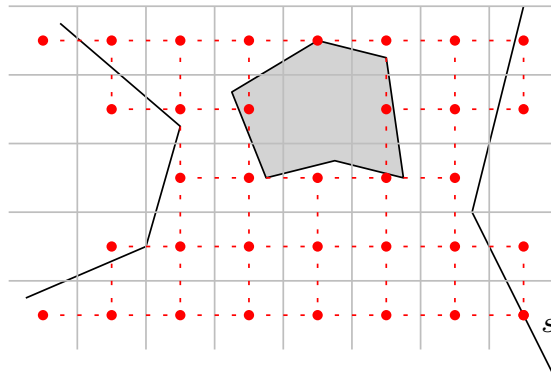


Fig. 2. The pixels P (grey) and the dual grid graph G (red).

Let τ^* denote an optimal lawn mowing tour for a given quota of coverage A . To approximate τ^* , we require the following lemma.

Lemma 1. *For a unit square cutter, there exists a tour τ_g whose vertices are a subset of the vertices of G such that $|\tau_g| = O(|\tau^*|)$ and $C(\tau_g) \leq C(\tau^*)$.*

Proof. Notice that τ^* is necessarily the shortest (fully covering) lawn mowing tour of $C(\tau^*)$, a subregion of R . According to [2, Theorem 3]

- (i) If the cutter is limited to rectilinear motion (τ^* consists of only axis-aligned segments), there exists τ_g within G of length no greater than $3|\tau^*|$ covering $C(\tau^*)$.

- (ii) If the cutter can translate arbitrarily, we augment G with diagonals: join two center points with a diagonal (45 degrees slope) if they are separated by distance $\frac{c}{2}$. There exists a \mathcal{C} within the diagonal-augmented grid of length no greater than $\frac{6}{2+\sqrt{2}}j^*$ covering C^* .

When j^* is small, we can employ the strategy for short covering tours in [2], the details of which we omit in this version.

An algorithm that computes an optimal (quota) lawn mowing tour will generally need to select turning points from a continuum rather than a discrete, finite set. This presents significant computational challenges (recall the algebraic hardness of the problem). Lemma 1 establishes that, for purposes of constant-factor approximations, it suffices to discretize the problem into a grid graph.

For each node p of G , the center point of the pixel dual to p , let $r(p)$ be the area of the pixel that lies inside R , i.e., $r(p) = |p \cap R|$. This allows us to treat the lawn mowing problem with a quota as an instance of the quota TSP (a special case of the Prize Collecting TSP [5]):

The quota TSP Find the shortest tour through a subset of pixel center points such that the total reward of the tour is no smaller than A .

By Lemma 1, a c -approximation to the quota TSP yields a $3c$ -approximation in the case of rectilinear motion and a $\frac{6c}{2+\sqrt{2}}$ -approximation in the case of arbitrary motion, to the quota lawn mowing problem.

We can approximate the quota TSP problem with a factor of 5 using the methods in [5] with polynomial (in N) running time. Further, we can achieve a factor of $(1 + \epsilon)$ for any $\epsilon > 0$ by adapting the methods for geometric k -TSP (Arora [3] or Mitchell [24]), a special case of quota TSP where the reward of every point is unary and the quota is k as follows:

- (1) Scale the rewards and quota to be integers. Specifically, let $M = 10^D$, where D is the maximum number of decimal digits used to represent the fraction part of any $r(p)$ or A . Then $M = D \log_2(10) = O(B)$, where B is the bit complexity of the problem instance. We set $\bar{A} := MA$ and $\tau(p) := Mr(p)$ for every $p \in P$.
- (2) For each p , create $\tau(p)$ -many duplicates of p at the same location as p (distance 0 away from p). This creates an $O(NM) = O(NB)$ instance of k -TSP.
- (3) Approximate the optimal k -TSP tour for the instance created with $k = \bar{A}$.

The running time of this approach is $(NB)^{O(1/\epsilon)}$.

Similarly, we can use the same transformation to the k -MST problem: Find the minimum cost tree spanning at least k vertices. Doubling an ϵ -approximate tree (traversing each edge twice in opposite directions) gives a 2ϵ -approximate tour. Several approximations with small factors are known for the k -MST problem, the current best factor is 2 for general graphs [13] and $(1 + \epsilon)$ for points in the plane [3, 24].

Circular cutter If the cutter is a circle, rather than square pixels, we consider hexagonal pixels, each of diameter 2. We join two center points by an edge if their respective hexagons share part of their boundary, creating a triangular grid graph (Figure 3).

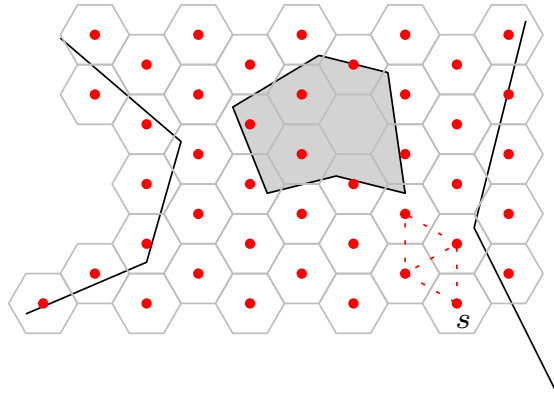


Fig. 3. The hexagonal pixels for the case of a circular cutter.

A constant-factor approximation to an optimal quota lawn mowing tour γ^* , can once again be found within the triangular grid graph. Specifically, one no longer than $2\sqrt{3} \cdot j^*$ (see [2, Theorem 3]). The remainder of the approximation method follows through like the case with square cutter.

4 Minimizing Expected Time to Search For an Object

We turn our attention to the problem of minimizing expected search time, which has numerous practical implications. The findings on the quota lawn mowing problem while are of great independent interests, also allow us to find approximation algorithms with provable guarantees for the expected search time minimization problem.

4.1 Lawn mowing search mechanism

First, we rewrite the formula for the expected value of T , the amount of time (or distance, interchangeably, because we assume the robot travels at a fixed speed of 1) before the target is detected by the robot as follows:

$$E[T_j] = \int_0^\infty \left(1 - \frac{jC(\gamma; t)j}{jR_j}\right) dt = \frac{1}{jR_j} \int_0^\infty \left(\int_{x \notin C(\gamma, t)} dx \right) dt.$$

For the purpose of optimizing expected detection time in a given region R , we can drop the denominator $|R|$. Notice that for each infinitesimal area element $dx \in R$, the integral above is taken over the region $dx \in R \cap C(\gamma; t)$. Hence, we can reverse the order of integration to be

$$\int_{x \in R} \left(\int_0^{t: x \notin C(\gamma, t)} dt \right) dx = \int_{x \in R} t_\gamma(x) dx:$$

where $t_\gamma(x)$ is the time when the accumulated area of coverage of the robot first contains x when traveling along γ . It may be more intuitive to consider this formula in the discrete case, where the location of the target is given as a uniform distribution on m discrete points x_1, \dots, x_m . Suppose γ visits those points in that order, then

$$E[T_j] = \sum_{i=1}^m t_\gamma(x_i) \Pr(\text{target is found when } \gamma \text{ reaches } x_i) = \frac{1}{m} \sum_{i=1}^m t_\gamma(x_i):$$

In fact, $t_\gamma(x_i)$ is known as the *latency* of x_i .

Theorem 1. *Minimizing $E[T_j]$ is NP-hard, even in a simple polygon.*

Proof. Our reduction is from the Minimum Latency problem: Given n points p_1, \dots, p_n on the Euclidean plane, compute a tour γ that minimizes $\sum_{i=1}^n t_\gamma(p_i)$, where $t_\gamma(p_i)$ is the distance traveled before first visiting p_i . The Minimum Latency problem is known to be as hard as the Euclidean TSP [7]. Our reduction works for both rectilinear and unrestricted movement, square and circular searcher. Given a Minimum Latency problem instance, scale the instance so that the side length (or radius) of the area of coverage of the robot is some ϵ infinitesimally small. We construct a simple polygon P as shown in Figure 4. First, interconnect the points by a tree (e.g., the minimum spanning tree). Surround each point with a small square room of side-length ϵ . Each edge of the tree is a narrow hallway of width ϵ^2 .

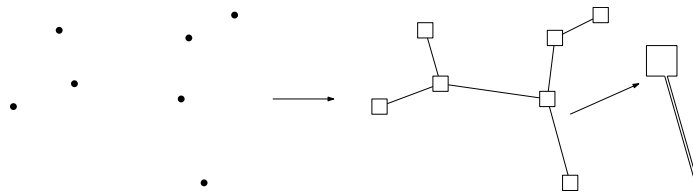


Fig. 4. The construction used in showing NP-hardness of minimizing $E[T_j]$.

Recall that in the lawn mowing search mechanism, the movement of the robot is not constrained to stay within P . Consequently, an expected detection time

minimizing tour prioritizes the rooms before attempting to cover all the area of the hallways. Such a tour must visit the rooms in the order that a latency-minimizing tour visits the corresponding points.

Approximation algorithm We give an approximation algorithm for minimizing $E[T_j]$, utilizing the solution to the quota lawn mowing problem as a subroutine. In the reduction from the quota lawn mowing problem to the k -TSP, we multiplied the area of each pixel with $M = 10^D$, where D is the decimal precision, thus effectively multiplied the area of R by M . Recall that $M = O(B)$, where B is the bit complexity.

For any $A \in [1; 2; \dots; M/R]g$, let $L(A)$ be length of the shortest tour that covers an area of A , which means a c -approximation algorithm would give a tour no longer than $cL(A)$ for area quota A . For $j = 1; \dots; O(\log N)$ (suppose $N > 1$, else the problem is trivial), denote by A_j the maximum value such that $L(A_j) \leq 2^j$. For $j = 1; \dots; O(\log N)$, we can find \overline{A}_j , the maximum area quota for which the c -approximation returns a tour no longer than $c2^j$, by a binary search on $[1; 2; \dots; M/R]g$ as the input quota to the c -approximation. Clearly, $A_j \leq \overline{A}_j$.

We state our approximation algorithm for finding a route that minimizes expected detection time as follows: For $j = 1; \dots; O(\log N)$, follow the tour (returned by the c -approximation) associated with area quota \overline{A}_j , return to the starting point before each increment of j . We denote this tour by γ .

Running time The running time of the above approximation algorithm is the running time of the c -approximation to the quota lawn mowing problem (which depends on the quota TSP/MST or k -TSP/MST heuristic employed) multiplied by $O(\log N)O(\log M/R) = O(\log N(B + \log N))$.

Analyzing the approximation factor Let t^* be an expected detection time optimal tour, and let $\Pr_\gamma(T > t) = f_\gamma(t)$. Specifically, $f_\gamma(t)$ is the uncovered fraction of area after the robot follows along t^* for a time/distance of t . Note that $f_\gamma(t)$ is continuous and monotone (decreasing).

We show another way to compute $E[T_j^*]$, which will be helpful in our analysis of the approximation factor. Draw the graph of $f_\gamma(t)$ on the Cartesian plane, we wish to compute the area bounded by its curve and the 2 axes. If we switch the 2 axes and consider t a function of ρ , the area under the curve stays the same. For any $0 \leq \rho \leq 1$, define the function $f'_\gamma(\rho) = t$, where t is the minimum latency along t^* so that the uncovered fraction of area is no smaller than ρ . Then

$$\int_0^{\lceil \gamma \rceil} f_\gamma(t) dt = \int_0^1 f'_\gamma(\rho) d\rho$$

See Figure 5 for an illustration.

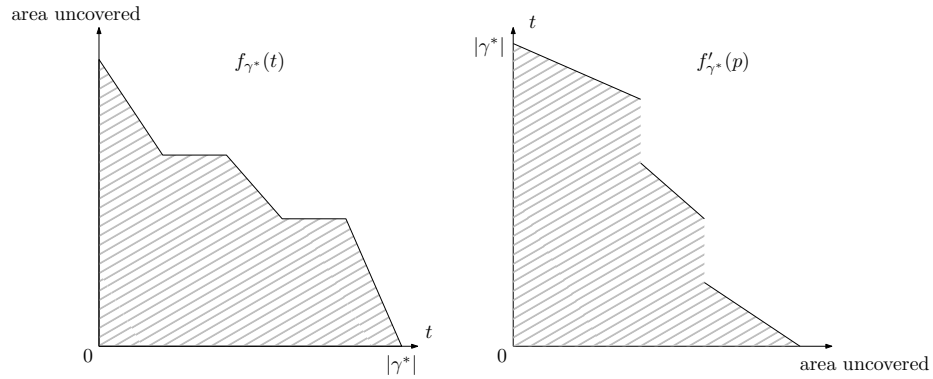


Fig. 5. The graph of $f(t)$ (left) and the graph of $f'(p)$ (right), the latter is acquired from switching the two axes, effectively “flipping” $f(t)$ and removing some vertical segments. Both envelope the same amount of area of the Cartesian plane (shaded).

Remark We can also use a standard change of variable in Lebesgue integration, however given that the focus of the paper is not on theoretical analysis, we opt for a more elementary and visual approach.

For some $0 < \rho < 1$, suppose $j \geq 2$ is such $A_{j-1} < (1 - \rho)jRj < A_j$. Consider the subpath of γ^* from the starting point until an area of $(1 - \rho)jRj$ is covered, the length of which is $f'_\gamma(\rho)$ by definition. Since doubling that subpath gives a tour that covers an area of $(1 - \rho)jRj$, we have $2^{j-1} \leq 2f'_\gamma(\rho)$.

On the other hand, γ^* is a concatenation of tours of length no longer than $c2^j$ covering an area of $A_j - A_j$ for $j = 1; 2; \dots$. Thus, the distance traveled along γ^* by the robot before covering an area of ρ is no longer than $\sum_{k \leq j} c2^k \leq 8c2^{j-2}$.

This implies, $f_{\gamma^*}(\rho) \leq 8cf_\gamma(\rho)$ for all $(1 - \rho) > \frac{A_1}{|R|}$. Consider that

$$\int_0^1 f'_{\gamma^*}(p) dp = \int_0^{1 - \frac{A_1}{jRj}} f'_{\gamma^*}(p) dp + \int_{1 - \frac{A_1}{jRj}}^1 f'_{\gamma^*}(p) dp:$$

We have

$$\int_0^{1 - \frac{A_1}{jRj}} f'_{\gamma^*}(p) dp \leq 8c \int_0^{1 - \frac{A_1}{jRj}} f'_\gamma(p) dp \leq 8c \int_0^1 f'_\gamma(p) dp: \tag{1}$$

When the fraction of uncovered area is between $1 - \frac{A_1}{|R|}$ and 1 (meaning the covered area is between 0 and A_1), $f'_{\gamma^*}(p)$ is no greater than $2c$, since following γ^* by a distance of $2c$ covers an area of at least A_1 . As a result

$$\int_{1 - \frac{A_1}{jRj}}^1 f'_{\gamma^*}(p) dp \leq \int_{1 - \frac{A_1}{jRj}}^1 2c dp = 2c \frac{A_1}{jRj} \leq 2c: \tag{2}$$

Combining the two inequalities (1) and (2) gives

$$E[T_{\gamma^*}] \leq 8cE[T_\gamma] + 2c:$$

Theorem 2. *Given a c -approximation to the quota lawn mowing problem. Let j^* be an optimal route for minimizing the expected detection time in a polygonal region R under the lawn mowing search mechanism, in pseudopolynomial time, we can compute a route j satisfying $E[T(j)] \leq 8cE[T(j^*)] + 2c$ in pseudopolynomial time.*

Remark The approximation bound we give includes both a multiplicative and an additive error term. We leave open the problem of removing the additive term. In practice, for a sufficiently large region R which requires a long tour to cover a large area, $\frac{A_1}{|R|}$ should be close to 0.

4.2 Visibility-based search mechanism

We show that the idea of recursively doubling the search perimeter in the previous subsection extends also to the visibility-based search mechanism, which models a robot entering and navigating through a room with obstacles (e.g., pieces of furniture). The robot is equipped with an omnidirectional camera for sensing, and it detects the stationary target when vision of the target is established. One difference between the visibility-based search mechanism and the lawn mowing search mechanism is that the robot cannot go through obstacles (the notion of visibility is not well-defined from within obstacles), nevertheless this does not prevent us from employing the same idea as before, thanks to a known result on partial visibility-based search.

In a polygonal region R , two points x and y see each other if the line segment xy does not intersect the exterior of R . The *visibility region* of x , denoted $V(x)$, is the set of all points that x sees. We also define the visibility region of a subset $S \subseteq R$ (for example, a route), which we denote by $V(S)$, to be the union of the visibility regions of all points in S (refer to Figure 6). Efficient algorithms for computing the visibility region of points and segments in polygonal domains are known [16, 18].

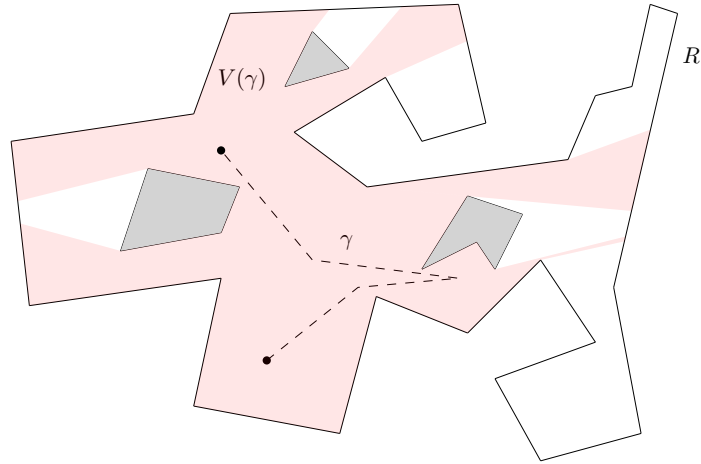


Fig. 6. The area of coverage $V(\gamma)$ (red) by a robot with visibility-based detection.

Given a polygonal domain R . We want to compute a route γ that minimizes the expected amount of time before the robot sees the target, which we also denote $E[T_j]$ and define similarly to that of the lawn mowing mechanism. For visibility-based detection, it has been known that minimizing the expected time until detection is NP-hard in a polygon with holes [26]. We strengthen the hardness result in the following theorem.

Theorem 3. *For visibility-based search, minimizing $E[T_j]$ is NP-hard, even in a simple polygon.*

Proof. We once again show NP-hardness by a reduction from the minimum latency problem, but for a weighted tree, which is known to be NP-hard [29]. Given a weighted tree, create a straight-line embedding of the tree using any graph drawing algorithm (see [31] for a comprehensive survey about graph drawings) in polynomial time. Then, construct a simple polygon from the embedding, similar to the construction used in the proof of Theorem 1. A tour within the polygon minimizing expected detection time yields a minimum latency tour.

The budgeted watchman route problem The budgeted watchman route problem was first introduced in [20]. Given a polygon P and a budget $B > 0$, compute a route γ such that $|\gamma| \leq B$ and the area of the visibility region of γ , $|V(\gamma)|$ is maximized.

The budgeted watchman route problem is (weakly) NP-hard, the following approximation algorithm is known [20], which we require as a subroutine.

Lemma 2. [20, Theorem 11] *Let P be a simple polygon with n vertices. Given a starting point S , in fully polynomial time (in n and $\frac{1}{\epsilon}$), we can compute a*

route of length no longer than $(1 + \epsilon)B$ that passes through s and sees an area no smaller than any route of length B can see.

Theorem 4. *Let σ^* be an optimal route for minimizing the expected detection time in a simple polygon P under the visibility-based search mechanism. For $B = 2^j$ where $j = 1, \dots, O(\log N)$ (recall that N is the number of unit-sized integer pixels intersecting P), run the approximation algorithm for the budgeted watchman route problem in Lemma 2 and concatenate the approximate tours into one, which we denote by σ^- . Then $E[T(\sigma^-)] \leq (8 + 8\epsilon)E[T(\sigma^*)] + (2 + 2\epsilon)B$ for any $\epsilon > 0$.*

Proof. The analysis follows through the same way as for the lawn mowing search mechanism. The running time is equal to that of the approximation algorithm to the budgeted watchman problem multiplied by $O(\log N)$.

5 Simulation Results

Algorithms developed to guarantee theoretical bounds in quota TSP-type problems are usually far too complicated and incur impractical running times. Thus, we complement our theoretical analysis with some experimental heuristics that are relatively straightforward to implement. Figure 7 shows our test polygons with holes in which we want to search for a target with a square scanner and rectilinear movement.

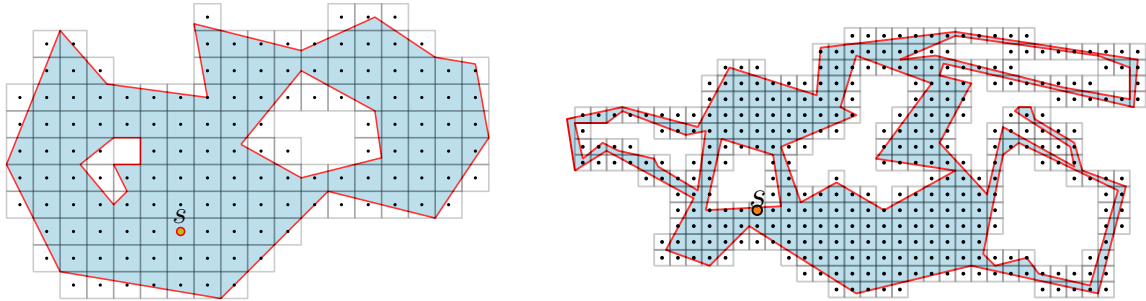


Fig. 7. The test polygonal regions, with the starting point s .

Our code is written in Python [35], we use the following packages: Shapely [14] for processing and handling the geometry of the domain and NetworkX [17] for handling the graph-theoretical components of the heuristic. Given below is a high-level description of what we call the exponential tree heuristic.

Algorithm Exponential Tree Heuristic

```

j = 0
visited = {}
while j < N
  Initialize the tree T = [starting center point]
  while j < min(2^j, N)
    T = T ∪ {v} where v is adjacent to a center point in T and r(v) ≤ max
    Traverse the TSP tour on the nodes in T ∪ {v} v ∉ starting point; v ∉ visited
    visited = visited ∪ {v}
  j = j + 1

```

The main idea is to greedily pick a subset of center points that maximize the total reward, while ensuring the length of the shortest tour through them is no larger than 2^j for $j = 0, 1, \dots$. The center points should induce a connected graph in G , and their spanning tree should have cost no greater than 2^j , making the TSP tour on them no longer than 2^{j+1} . By discarding the center points already visited (by previous tours of smaller maximum sizes) and computing only the TSP tour on the remaining ones, the running time as well as the detection time is improved by a margin.

It is quite infeasible to find an optimal minimum latency tour by brute force, even for small instances, due to the factorial number of possible tours. We thus compare the mean detection time of the route generated by our heuristic with that of a heuristic designed to minimize the total latency tour on all the center points. First, note that it suffices, for purposes of approximating the minimum latency tour within a factor of $(1 + \epsilon)$, to concatenate a number of TSP paths with a geometrically decreasing sequence of number of nodes [4]. Thus, we find the overall TSP tour on N center points, replace the subpath on the first $\lfloor \frac{N}{1 + \epsilon} \rfloor$ center points with the TSP path on those points, then repeat with the next $\lfloor \frac{N}{(1 + \epsilon)^2} \rfloor, \lfloor \frac{N}{(1 + \epsilon)^3} \rfloor, \dots$ points. We set $\epsilon = 0.01$ to get close to the minimum latency tour.

We perform our experiment with 1000 different random generations of the location of the target, the results are given in Table 1.

The exponential tree heuristic seems to yield average detection time within a constant factor of that of the minimum latency heuristic, but has faster running time. This demonstrates that gradually expanding the search area is an effective strategy for finding a target within a reasonable expected time. The search routes generated by the exponential tree heuristic are given in Figures 8 and 9.

Acknowledgments. This work is partially supported by the National Science Foundation (CCF-2007275). The author would like to express his gratitude to his advisor, Joseph S. B. Mitchell for incredibly helpful and encouraging discussions on the paper.

Disclosure of Interests. The author has no competing interests to declare that are relevant to the content of this article.

Table 1. Detection time comparisons between the exponential tree heuristic and the minimum latency heuristic.

1st test polygonal region	mean	standard deviation	running time
Exponential Tree Heuristic	112.081	70.333	1.308
Minimum Latency Heuristic	69.577	42.764	8.980

2nd test polygonal region	mean	standard deviation	running time
Exponential Tree Heuristic	233.745	151.886	4.843
Minimum Latency Heuristic	183.167	119.555	52.360

References

1. Archer, A., Levin, A., Williamson, D.P.: A faster, better approximation algorithm for the minimum latency problem. *SIAM Journal on Computing* **37**(5), 1472–1498 (2008)
2. Arkin, E.M., Fekete, S.P., Mitchell, J.S.B.: Approximation algorithms for lawn mowing and milling. *Computational Geometry* **17**(1-2), 25–50 (2000)
3. Arora, S.: Polynomial time approximation schemes for Euclidean TSP and other geometric problems. In: *Proceedings of 37th Conference on Foundations of Computer Science*. pp. 2–11. IEEE (1996)
4. Arora, S., Karakostas, G.: Approximation schemes for minimum latency problems. *SIAM Journal on Computing* **32**(5), 1317–1337 (2003)
5. Ausiello, G., Bonifaci, V., Leonardi, S., Marchetti-Spaccamela, A.: Prize collecting traveling salesman and related problems. In: *Handbook of Approximation Algorithms and Metaheuristics*, pp. 611–628. Chapman and Hall/CRC (2018)
6. Berman, O., Iarovsky, E., Krass, D.: Optimal search path for service in the presence of disruptions. *Computers & Operations Research* **38**(11), 1562–1571 (2011)
7. Blum, A., Chalasani, P., Coppersmith, D., Pulleyblank, B., Raghavan, P., Sudan, M.: The minimum latency problem. In: *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*. pp. 163–171 (1994)
8. Cabreira, T.M., Brisolara, L.B., Paulo R., F.J.: Survey on coverage path planning with unmanned aerial vehicles. *Drones* **3**(1), 4 (2019)
9. Chin, W.p., Ntafos, S.: Optimum watchman routes. In: *Proceedings of the 2nd Annual Symposium on Computational Geometry*. pp. 24–33 (1986)
10. Chung, T.H., Hollinger, G.A., Isler, V.: Search and pursuit-evasion in mobile robotics: A survey. *Autonomous Robots* **31**, 299–316 (2011)
11. DasGupta, B., Hespanha, J.P., Riehl, J., Sontag, E.: Honey-pot constrained searching with local sensory information. *Nonlinear Analysis: Theory, Methods & Applications* **65**(9), 1773–1793 (2006)
12. Fekete, S.P., Krupke, D., Perk, M., Rieck, C., Scheffer, C.: Circling a square: The lawn mowing problem is algebraically hard. In: *European Workshop on Computational Geometry* (2023)
13. Garg, N.: Saving an epsilon: a 2-approximation for the k -MST problem in graphs. In: *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*. pp. 396–402 (2005)
14. Gillies, S., et al.: Shapely: manipulation and analysis of geometric objects (2007–), <https://github.com/Toblerity/Shapely>

15. Goemans, M., Kleinberg, J.: An improved approximation ratio for the minimum latency problem. *Mathematical Programming* **82**(1), 111–124 (1998)
16. Guibas, L., Hershberger, J., Leven, D., Sharir, M., Tarjan, R.: Linear time algorithms for visibility and shortest path problems inside simple polygons. In: *Proceedings of the 2nd Annual Symposium on Computational Geometry*. pp. 1–13 (1986)
17. Hagberg, A., Swart, P., S Chult, D.: Exploring network structure, dynamics, and function using networkx. Tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States) (2008)
18. Heffernan, P.J., Mitchell, J.S.B.: An optimal algorithm for computing visibility in the plane. *SIAM Journal on Computing* **24**(1), 184–201 (1995)
19. Hohzaki, R.: Search games: Literature and survey. *Journal of the Operations Research Society of Japan* **59**(1), 1–34 (2016)
20. Huynh, K.C., Mitchell, J.S.B., Nguyen, L., Polishchuk, V.: Optimizing visibility-based search in polygonal domains. In: *Proceedings of the 19th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2024)*. Leibniz International Proceedings in Informatics (LIPIcs), vol. 294, pp. 27:1–27:16 (2024)
21. Koopman, B.O.: The theory of search. i. kinematic bases. *Operations Research* **4**(3), 324–346 (1956)
22. Koopman, B.O.: The theory of search. ii. target detection. *Operations Research* **4**(5), 503–531 (1956)
23. Koopman, B.O.: The theory of search: Iii. the optimum distribution of searching effort. *Operations Research* **5**(5), 613–626 (1957)
24. Mitchell, J.S.B.: Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k -MST, and related problems. *SIAM Journal on Computing* **28**(4), 1298–1309 (1999)
25. Raap, M., Preuß, M., Meyer-Nieberg, S.: Moving target search optimization—a literature review. *Computers & Operations Research* **105**, 132–140 (2019)
26. Sarmiento, A., Murrieta, R., Hutchinson, S.A.: An efficient strategy for rapidly finding an object in a polygonal world. In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*(Cat. No. 03CH37453). vol. 2, pp. 1153–1158. IEEE (2003)
27. Sarmiento, A., Murrieta-Cid, R., Hutchinson, S.: Planning expected-time optimal paths for searching known environments. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. vol. 1, pp. 872–878. IEEE (2004)
28. Sato, H.: Path optimization for single and multiple searchers: models and algorithms. Ph.D. thesis, Naval Postgraduate School (2008)
29. Sitters, R.: The minimum latency problem is NP-hard for weighted trees. In: *Integer Programming and Combinatorial Optimization: 9th International IPCO Conference* Cambridge, MA, USA, May 27–29, 2002 *Proceedings* 9. pp. 230–239. Springer (2002)
30. Sitters, R.: Polynomial time approximation schemes for the traveling repairman and other minimum latency problems. In: *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*. pp. 604–616. SIAM (2014)
31. Tamassia, R.: *Handbook of graph drawing and visualization*. CRC press (2013)
32. Tan, C.S., Mohd-Mokhtar, R., Arshad, M.R.: A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms. *IEEE Access* **9**, 119310–119342 (2021)
33. Teller, R., Zofi, M., Kaspi, M.: Minimizing the average searching time for an object within a graph. *Computational Optimization and Applications* **74**, 517–545 (2019)

34. Trummel, K.E., Weisinger, J.R.: The complexity of the optimal searcher path problem. *Operations Research* **34**(2), 324–327 (1986)
35. Van Rossum, G., Drake Jr, F.L.: Python tutorial (1995)
36. Yu, L., Han, Q., Tuo, X., Tian, W.: A survey of probabilistic search based on Bayesian framework. In: 2019 4th International Conference on Mechanical, Control and Computer Engineering (ICMCCE). pp. 930–9305. IEEE (2019)

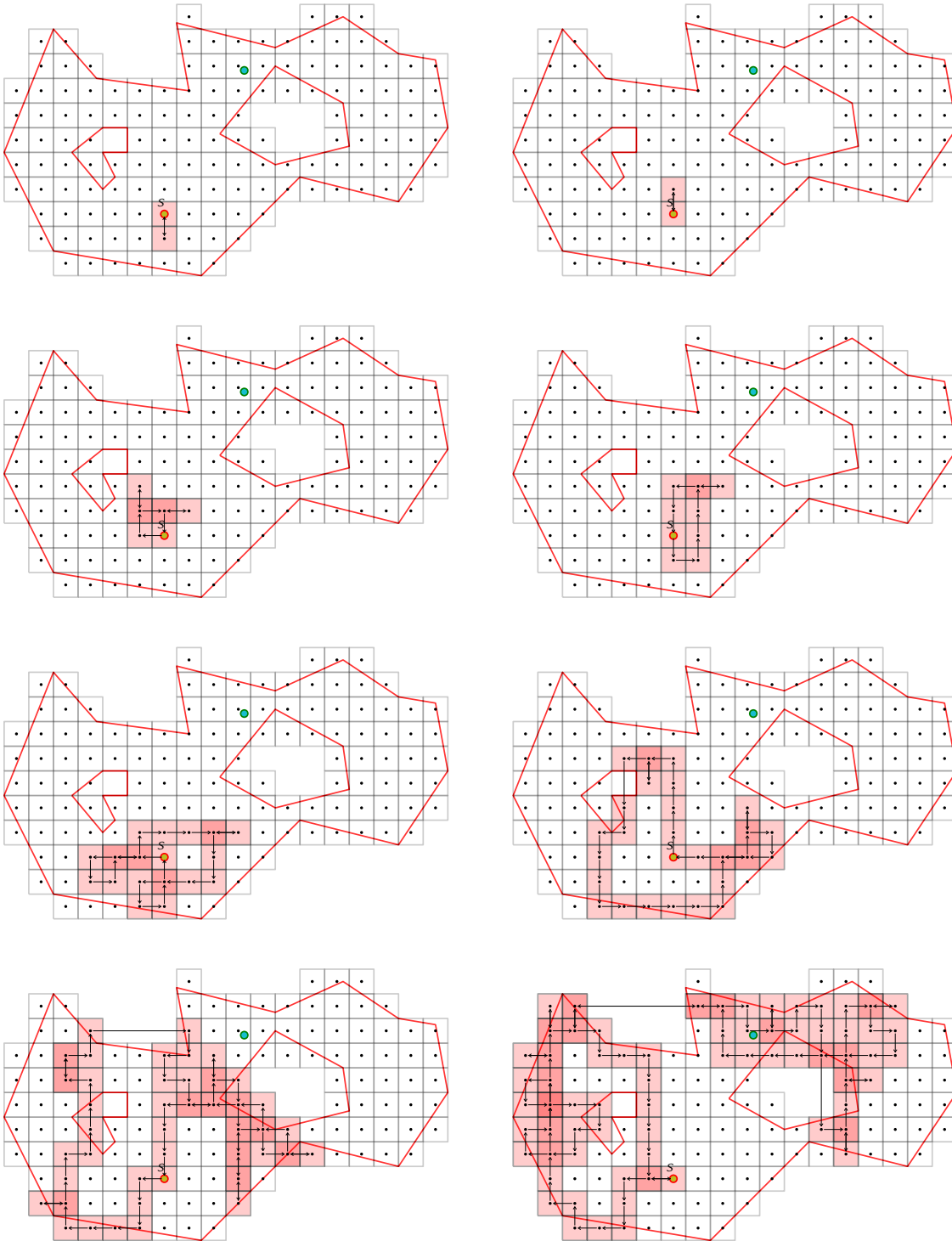


Fig. 8. The search route, which consists of tours of increasing lengths, generated for the first test polygonal region by the exponential tree heuristic. The target is drawn in green.

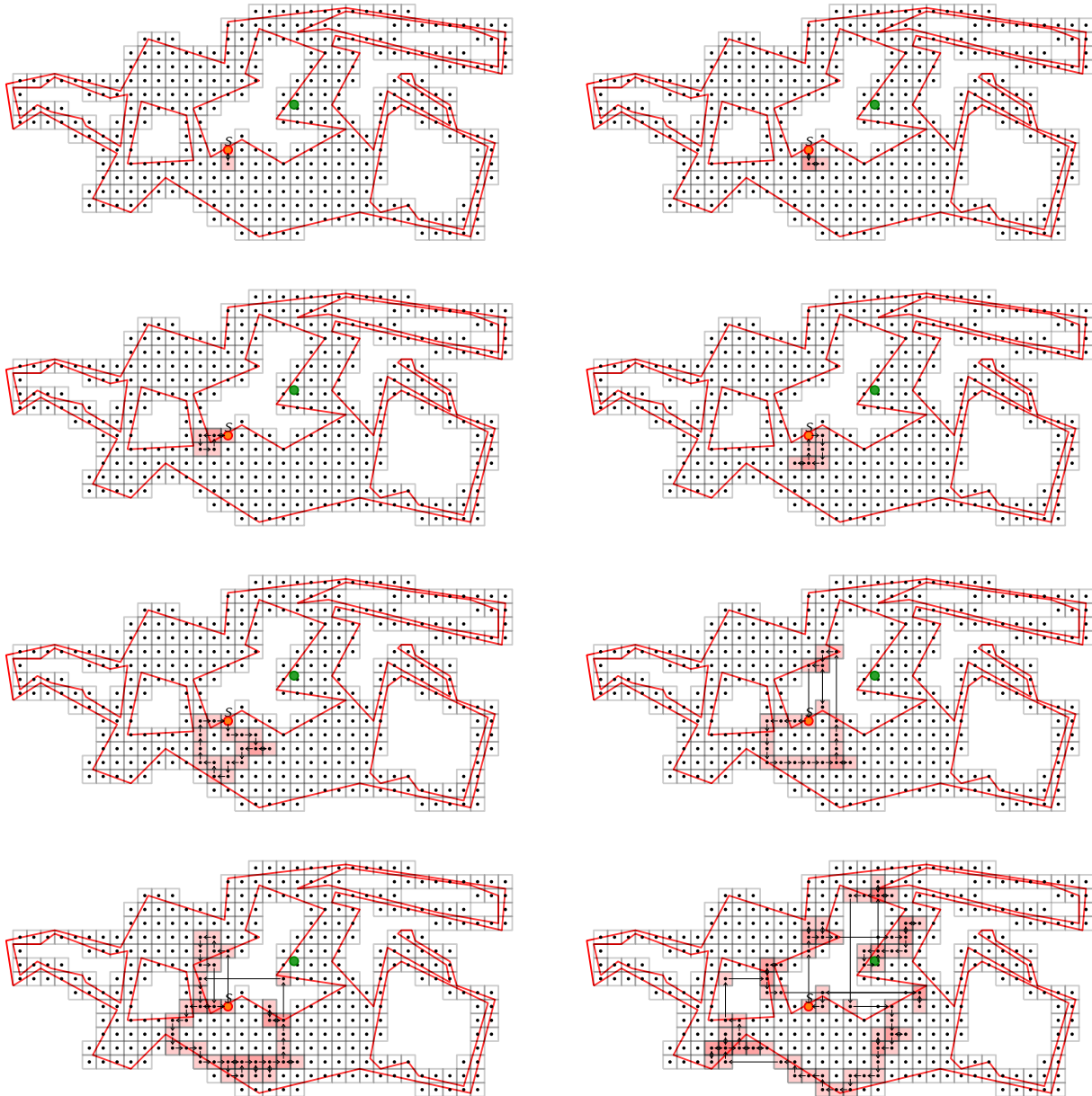


Fig. 9. The search route, which consists of tours of increasing lengths, generated for the second test polygonal region by the exponential tree heuristic. The target is drawn in green.