Approximation Algorithms for Anchored Multiwatchman Routes[*]

Joseph S. B. Mitchell
Department of Applied Mathematics and Statistics, Stony Brook University, USA

Linh Nguyen
Department of Applied Mathematics and Statistics, Stony Brook University, USA

Abstract

We study some variants of the k-Watchman Routes problem, the cooperative version of the classic Watchman Route problem in a simple polygon. The k watchmen may be required to see the whole polygon, or some pre-determined quota of area within the polygon, and we want to minimize the maximum length traveled by any watchman. While the single watchman ($k = 1$) version of the problem has received considerable attention and is relatively well understood, much less is known about the multiple watchmen ($k > 1$) variant(s).

We provide the first tight approximability results for the anchored k-Watchman Routes problem in a simple polygon, assuming k is fixed, by a fully-polynomial time approximation scheme. The basis for the FPTAS is provided by an exact dynamic programming algorithm. If k varies (i.e., it is part of the input), we give constant-factor approximations.

Keywords and phrases Watchman Route, k-Watchman Routes, Quota Watchman Route, Quota k-Watchman Routes

Digital Object Identifier 10.57717/cgt.v4i1.71

Funding This work is partially supported by the National Science Foundation (CCF-2007275)

1 Introduction

1.1 Prior and related work

The Art Gallery problem was introduced in 1973 by Victor Klee: given an art gallery with n walls (a polygon P), determine the minimum number of stationary guards needed within P so that every point inside the gallery is visible to at least one guard. Since then, the Art Gallery problem, along with its many variants, has been the focus of extensive research in computational geometry and algorithms.

When guards are mobile, a single guard can monitor any connected domain. Consequently, the objective shifts to finding routes for one or more guards that optimize certain aspects of their movement, such as path lengths or the number of turns. The problem of minimizing the distance a guard must travel to cover the entire polygon is known as the Watchman Route problem. Chin and Ntafos introduced the Watchman Route problem, proved that it is NP-hard in polygons with holes (later revisited in [7]), and provided a linear time algorithm for simple orthogonal polygons. For general simple polygons, there are exact polynomial-time algorithms, for example, [6, 20, 21, 22]. In a polygon with holes, Mitchell showed that the Watchman Route problem cannot be approximated better than an $O(\log n)$ factor, and gave an $O(\log^2 n)$-approximation.

In some scenarios, complete coverage may be impractical or unnecessary, so we are also interested in finding the shortest route that covers at least an area of $A \geq 0$ within P. This

[*] A preliminary version of this paper was entitled “Multirobot Watchman Routes in a Simple Polygon” and appears in the Proceedings of the 36th Canadian Conference on Computational Geometry, St. Catharines, Ontario, Canada, pp. 49-55, 2024.

variant is known as the Quota Watchman Route problem, introduced in [11]. In contrast to the tractable Watchman Route problem, the Quota Watchman Route problem is weakly NP-hard, but a fully polynomial-time approximation scheme (FPTAS) is known. Any results for the Quota Watchman Route problem can be applied to the Watchman Route problem by simply setting A to be the area of P.

We consider the generalization of both aforementioned problems to multiple agents, the k-Watchman Routes problem and the Quota k-Watchman Routes problem, with the objective of minimizing the length of the longest path traveled by any one watchman. Even in a simple polygon, when no starting points are specified (so, we are to determine the best starting locations), both problems are NP-hard to approximate within any multiplicative factor [17].

Our focus is thus on the anchored version, where a team of robots or searchers enter a domain P through a door on its boundary to search for a stationary target that may be randomly distributed within the domain. The goal is to plan an optimal collective effort to ensure at least a certain probability of detection (1 in the k-Watchman Routes problem and some $p \in [0, 1]$ in the Quota k-Watchman Routes problem).

Little is known about the k-Watchman Routes problem in its most general form. Very recently, Nilsson and Packer [16] proposed some constant-factor approximation algorithms for when $k = 2$, their methods however do not extend to larger values of k. Additionally, some studies have focused on developing and analyzing experimental heuristics [8, 17], but those approaches lack provable performance guarantees. It was clear that many algorithmic results for the Watchman Routes problem do not generalize to the k-Watchman Routes problem.

1.2 Main results

This paper is an extended journal version of our earlier work that appeared in preliminary form in [14], and includes some new findings. The main results can be summarized as follows:

- Assuming that k is fixed, we give a pseudopolynomial-time exact algorithm to solve the anchored k-Watchman Routes problem in a simple orthogonal (integral coordinate) polygon P under L_1 distance. This is a reasonable assumption as in most practical situations, it is not feasible to employ an arbitrarily large number of watchmen or robots.
- For any fixed k, we give an FPTAS for the anchored k-Watchman Routes problem in an orthogonal simple polygon (with orthogonal watchmen movements) and a general simple polygon. We also argue that the anchored k-Watchman Routes problem is weakly NP-hard, thus our results show tight approximability.
- When k is part of the input, we give constant-factor approximation algorithms, for the anchored k-Watchman Routes problem as well as the anchored Quota k-Watchman Routes problem. The algorithms given are simple and have low approximation factors (3 or $(2 + \varepsilon)$).

While we restrict ourselves to the anchored version, we achieve better approximation factors for any k than those proposed by Nilsson and Packer for the case where $k = 2$ [16] (5.969 and 11.939 when no starting points are given, 6.922 when two starting points, one for each watchman, are given).

2 Preliminaries

Our domain is a simple polygon P whose boundary, ∂P, is a simple, non-self-intersecting polygonal chain, consisting of n vertices. A vertex is *reflex* (resp. *convex*) if its internal angle is greater than (resp. less than) 180 degrees. An *orthogonal simple polygon* is a simple polygon where all internal angles are 90 or 270 degrees.

We say that $x \in P$ and $y \in P$ *see* each other if the segment xy is contained inside P. The *visibility region* of x, denoted by $V(x)$, is defined as the set of all points that x sees, i.e. $V(x) = \{y \in P : xy \subseteq P\}$. The visibility region of a segment or route (or any subset) $X \subseteq P$ is the union of the visibility regions of all points of X; or equivalently the set of all points that see some point of X. Efficient algorithms for computing visibility regions of points and segments are known, see [9, 23].

The *geodesic shortest path* from $x \in P$ to $y \in P$, which we denote by $\pi(x, y)$ is a shortest path between x and y that stays within P. If P is a simple polygon, then $\pi(x, y)$ is necessarily unique and polygonal. The *geodesic distance* between x and y is the length of $\pi(x, y)$ measured in the L_2 metric, also called the Euclidean metric. The geodesic shortest path from x to a set $Y \subseteq P$ is $\pi(x, y)$ such that the length of $\pi(x, y)$ is the minimum among all $y \in Y$. Throughout this paper, if we use the term "geodesic shortest path", we mean the Euclidean geodesic shortest path. The *geodesic convex hull* or *relative convex hull* of $X \subseteq P$ is the minimal subset of P that contains X and is closed under taking shortest paths; one important property of the relative convex hull is that it is the minimum-perimeter superset of X within P. We also consider another metric, namely the L_1 metric, also known as the Manhattan metric. A polygonal path with all edges parallel to an axis is called a rectilinear path. A rectilinear path of minimum length between two points that stays within P is called a *geodesic L_1 shortest path*. Unlike the Euclidean metric, there are, in general, multiple geodesic L_1 shortest paths between two points, all of the same length. We denote by $\pi^{\perp}(x, y)$ a geodesic L_1 shortest paths from x to y . Refer to the survey [12] for more details about geodesic shortest paths and relative convex hulls.

The Watchman Route problem asks for a minimum-length tour that sees all of P. A *visibility cut* c_i with respect to the starting point s is a chord obtained from extending the edge e incident on a reflex vertex, v_i, where e is such that its extension creates a convex vertex at v_i in the subpolygon containing s. The other subpolygon (not containing s) is the *pocket* induced by c_i. Furthermore, c_i is an *essential cut* if its pocket does not fully contain any other pocket. The shortest visibility covering tour must visit all essential cuts in order around ∂P [5, 6], thus by treating the Watchman Route problem as an instance of the "touring polygons" problem, we can get a polynomial-time algorithm [6].

The Quota Watchman Route problem seeks a minimum-length tour that sees at least a given amount of area within P. The QWRP is (weakly) NP-hard, and the touring polygons approach does not apply since we do not have the optimality structure implied by having to see all of P, which yields a sequence of essential cuts (the watchman may only need to look "partially" around some corners to achieve the quota) for the tour to visit. By exploiting the following geometric property: an optimal (complete or partial) visibility covering tour must be relatively convex (the tour as well as its interior are closed under taking geodesic shortest paths), we can separate the visibility region of different intervals of the tour. This allows for a simple dynamic programming algorithm on an appropriate discretization of the domain, leading to an FPTAS [11].

We consider the generalization of both of the above problems to multiple watchmen. Denote by $|.|$ the measure of geometric objects (such as length or area). Given a simple polygon P and a starting point s. The k-Watchman Routes problem is that of computing a collection of k tours $\{\gamma_i\}_{i=1,\dots,k}$ such that $\bigcup\limits_{i=1,\dots,k} V(\gamma_i) = P$ and $\max\limits_{i=1,\dots,k} |\gamma_i|$ is minimized. Note that each γ_i is necessarily a polygonal tour (Corollary 2).

In any multi-agent routing problem, the two commonly studied objective functions are min-max and min-sum cost, however, in the context of the k-Watchman Routes problem with a common starting point for all watchmen, the min-sum version is trivially solvable: simply take the relative convex hull of $\{\gamma_i\}_{i=1,\ldots,k}$, its perimeter is no greater than $\sum_{i=1}^{k}|\gamma_i|$ since all routes are connected (through s), thus we can let one watchman travel the perimeter of said relative convex hull to see all of P while all others stand still. In other words, the optimal solution for the min-sum objective is a shortest route for a single-watchman and $k-1$ degenerate routes of length 0. If the perimeter of the relative convex hull of $\{\gamma_i\}_{i=1,\ldots,k}$ does not pass through s, instead of said relative convex hull, we can take the ("nearly relatively convex") minimum-perimeter set enclosing $\{\gamma_i\}_{i=1,\ldots,k}$ and containing s as a vertex, which is computable if given s and the set $\{\gamma_i\}_{i=1,\ldots,k}$, and proceed with a similar argument. With multiple starting points given, one for each route, the min-sum k-Watchman Routes problem is open but conjectured to be polynomial for fixed k. When no starting points are given, both the min-sum and min-max versions get significantly harder (no multiplicative approximation factors possible [17]).

The min-max version remains NP-hard, even with a common depot. In [15], the authors showed that even for $k=2$, the min-max k-Watchman Routes without starting points in a simple polygon is weakly NP-hard via a simple reduction from the Partition problem: given a set of positive integers $\{x_1, x_2, \ldots, x_n\}$, partition it into two subsets of equal sum. We utilize the same construction in [15] and simply add a starting point s at the intersection of all narrow corridors (Figure 1). We see that the anchored version is also NP-hard, since an optimal solution will have to divide the corridors between the two routes (which might properly intersect many times) so that the total length of the corridors traversed by each route is as close to $\frac{1}{2}(x_1 + x_2 + \ldots + x_n)$ as possible.

The next problem we study, the Quota k-Watchmen Routes problem has the same min-max objective function, but with a generalized constraint of $\left|\bigcup_{i=1,\ldots,k} V(\gamma_i)\right| \geq A$ for some given $0 \leq A \leq |P|$. The fraction of area seen, $\frac{A}{|P|}$, can be interpreted as the probability that the watchmen detect a target at an unknown location uniformly distributed in P.

Throughout this paper, we assume a real RAM model of computation [18].

3 Multiwatchman routes with a constant number of watchmen

First, we characterize the structure of an optimal collection of k-watchman routes $\{\gamma_i\}_{i=1,\ldots,k}$.

▶ **Lemma 1.** *$\bigcup_{i=1,\ldots,k} V(\gamma_i) = P$ if and only if $\{\gamma_i\}_{i=1,\ldots,k}$ collectively visit all essential cuts of P.*

Proof. See Figure 2. Since all routes pass through s, their union is connected. The proof follows simply from the well-known fact: A connected set sees all of P if and only if it visits all essential cuts [2, 5, 6]. ◀

Note that when we have multiple starting points, or no starting points specified at all, Lemma 1 does not necessarily hold.

For $1 \leq i \leq k$, denote by C_i the set of essential cuts visited by γ_i. It is worth pointing out that, due to Lemma 1, each γ_i behaves like a solution to the touring polygons problem on a subset of essential cuts.

▶ **Corollary 2.** *There exists an optimal solution $\{\gamma_i\}_{i=1,\ldots,k}$ such that for any i, γ_i is the shortest route to visit all cuts in C_i and s in the order in which they appear around ∂P (more precisely, the order of the reflex edges whose extensions constitute the essential cuts).*

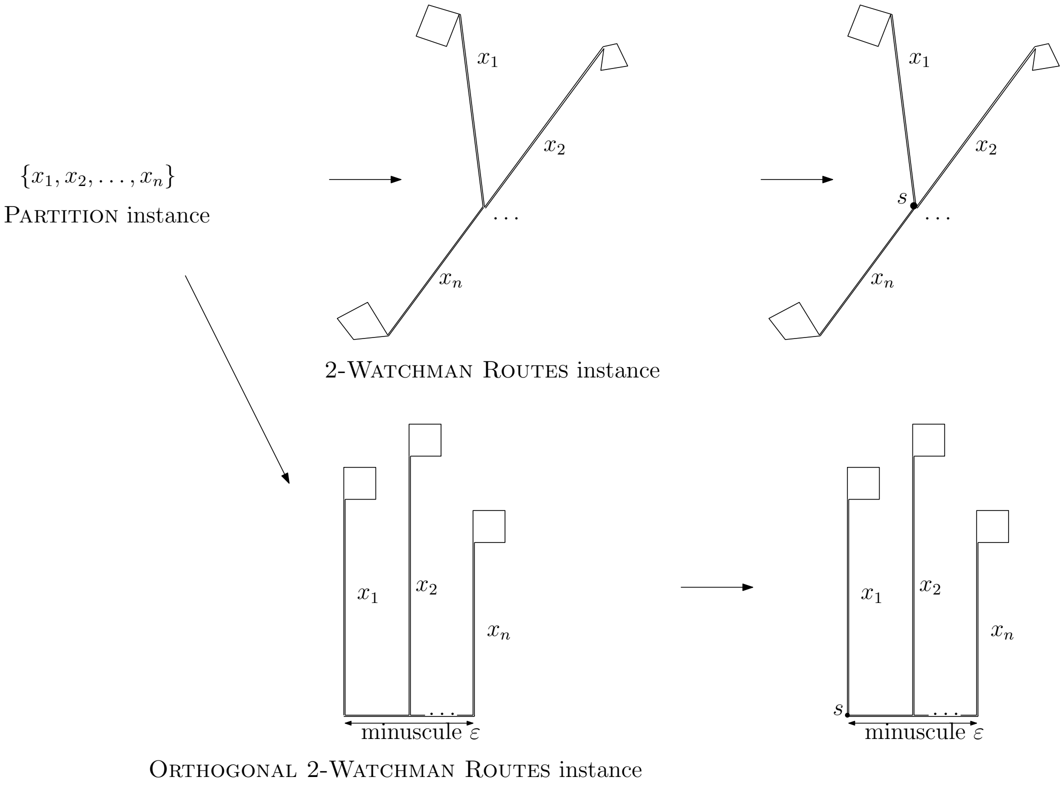
$\{x_1, x_2, \ldots, x_n\}$
Partition instance
x_1
x_2
$\ldots$
x_n
s
2-Watchman Routes instance
minuscule ε
Orthogonal 2-Watchman Routes instance

Figure 1 The construction used in [15] to show NP-hardness of the k-Watchman Routes problem for fixed k, which we modify slightly to show that the anchored version is also NP-hard.

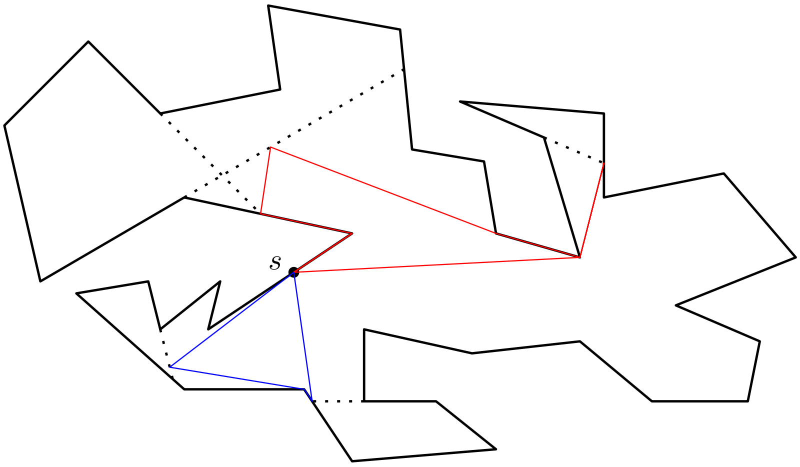

Figure 2 A simple polygon and its essential cuts (dashed). The 2-watchman routes (red and blue) visit all essential cuts between them and see the whole polygon.

One immediate observation is that each γ_i is necessarily a closed polygonal curve, with at most $2n$ vertices [6, 13]. It is easy to show that if $i \neq j$, then $C_i \cap C_j = \varnothing$. Suppose to the contrary that, there exists $c \in C_i \cap C_j$, then we can replace γ_i by the shortest tour that visits all of $C_i \setminus c$, which is no longer than the shortest tour that visits all of C_i.

3.1 Orthogonal simple polygon, rectilinear movement

We first consider the special case where P is a simple, orthogonal polygon. Assume that the watchmen all start from a common starting point $s \in \partial P$ and can only move along axis-aligned segments. It is straightforward to see from the construction given in Figure 1 that this special case is also NP-hard. Further, assume all vertices of P have integer coordinates (this assumption helps to achieve a pseudopolynomial-time algorithm).

3.1.1 Dynamic programming exact algorithm

We decompose P into rectangular cells using maximal (within P) extensions of all edges, as well as a horizontal and vertical line through s. This is known as the Hanan grid [10] (Figure 3). We refer to the set of vertices of the rectangular cells, including vertices of P as grid points.

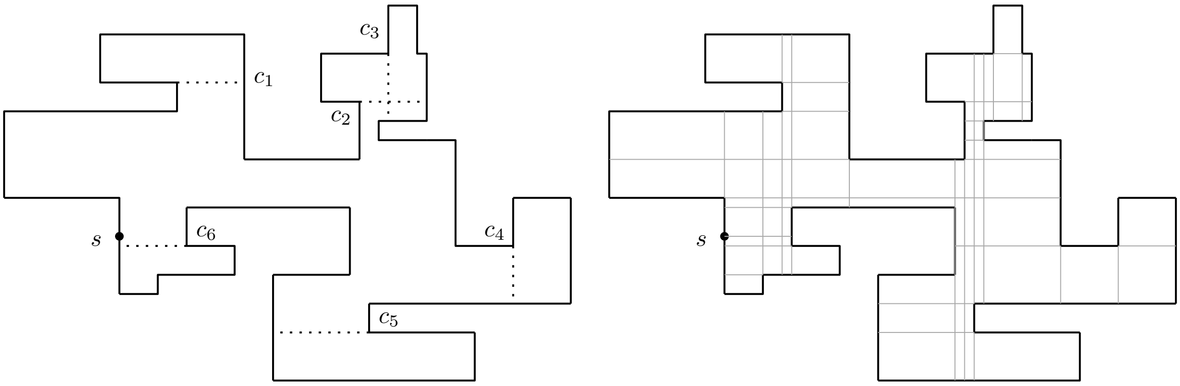

Figure 3 *Left*: The essential cuts of P. *Right*: The Hanan grid formed by extensions of all edges in P.

We show that, given the assumptions of an orthogonal P and rectilinear movement, the Hanan grid (or in particular, the portion of the Hanan grid on the essential cuts) suffices to discretize the problem.

▶ **Lemma 3.** *There exists an optimal solution $\{\gamma_i\}_{i=1,\ldots,k}$ that lies within the edges of the Hanan grid.*

Proof. Given an optimal solution $\{\gamma_i\}_{i=1,\ldots,k}$, for $1 \leq i \leq k$, let $C_i = \{c_{i1}, \ldots, c_{ij}\}$ (in order around ∂P). Also, let $p_{i1}, \ldots, p_{ij}$ be the point where γ_i first intersects with $c_{i1}, \ldots, c_{ij}$.

First, note that for every i, we may replace γ_i with a concatenation of geodesic L_1 shortest paths, namely $\gamma_i := \pi^{\perp}(s, p_{i1}) \cup \pi^{\perp}(p_{i1}, p_{i2}) \cup \ldots \cup \pi^{\perp}(p_{ij}, s)$ without increasing $\max_{i=1,\ldots,k} |\gamma_i|$ while maintaining complete visibility coverage of P.

Now, we argue that $\pi^{\perp}(s, p_{i1})$ is a geodesic L_1 shortest path from s to c_{i1}. Suppose to the contrary, that geodesic L_1 shortest paths from s to c_{i1} make contact with c_{i1} at $p'_{i1} \neq p_{i1}$ (all geodesic L_1 shortest paths from a point to a segment have the same endpoint). Due to orthogonality $|\pi^{\perp}(s, p'_{i1})| + |p'_{i1}p_{i1}| = |\pi^{\perp}(s, p_{i1})|$, which means $|\pi^{\perp}(s, p_{i1})| + |\pi^{\perp}(p_{i1}, p_{i2})| = |\pi^{\perp}(s, p'_{i1})| + |p'_{i1}p_{i1}| + |\pi^{\perp}(p_{i1}, p_{i2})| \geq |\pi^{\perp}(s, p'_{i1})| + |\pi^{\perp}(p'_{i1}, p_{i2})|$. This implies γ_i should take a geodesic L_1 shortest path from s to c_{i1}, and it suffices to find such a path in the Hanan grid. By a straightforward inductive argument, we can show the same for any portion of γ_i between any two essential cuts. ◀

Lemma 3 allows us to treat the orthogonal k-watchman problem as that of finding a set of grid points on the essential cuts (one per cut) that each route γ_i visits so as to minimize the maximum length of any route. Note that we need not consider any grid point that lies beyond an essential cut.

For the purpose of describing our algorithm, it may be more convenient to think of each route γ_i in terms of two components: the path from s to p_{ij}, which we denote by Γ_i (in particular, $\Gamma_i := \pi^{\perp}(s, p_{i1}) \cup \pi^{\perp}(p_{i1}, p_{i2}) \cup \ldots \cup \pi^{\perp}(p_{i(j-1)}, p_{ij})$), and the return trip $\pi^{\perp}(p_{ij}, s)$.

Our algorithm is based on dynamic programming. Let $\{c_1, c_2, \ldots, c_m\}$ be the set of essential cuts in order around ∂P (s lies between c_1 and c_m). Each subproblem $(c_j, p_1, l_1, \ldots, p_k, l_k)$ is defined by a $(2k + 1)$-tuple which consists of

- an essential cut c_j,
- k Hanan grid points $p_1, \ldots, p_k$, each on one different essential cut among $s, c_1, \ldots, c_j$. One of the grid points $p_1, \ldots, p_k$ is on c_j.
- k integers $l_1, \ldots, l_k$.

We abuse notation slightly, for the sake of brevity, by using the tuple to also denote the value of the subproblem in our dynamic programming algorithm. Subproblem $(c_j, p_1, l_1, \ldots, p_k, l_k) =$ TRUE if and only if there exists a collection of k paths $\Gamma_1, \ldots, \Gamma_k$ collectively visiting all essential cuts from c_1 up to c_j such that

- Γ_i starts at s, ends at p_i, and consists of geodesic L_1 shortest paths between grid points on essential cuts,
- $|\Gamma_i| = l_i$.

We call $\{\Gamma_i\}_{i=1,\ldots,k}$ the paths associated with subproblem $(c_j, p_1, l_1, \ldots, p_k, l_k)$. See Figure 4 for an illustration.

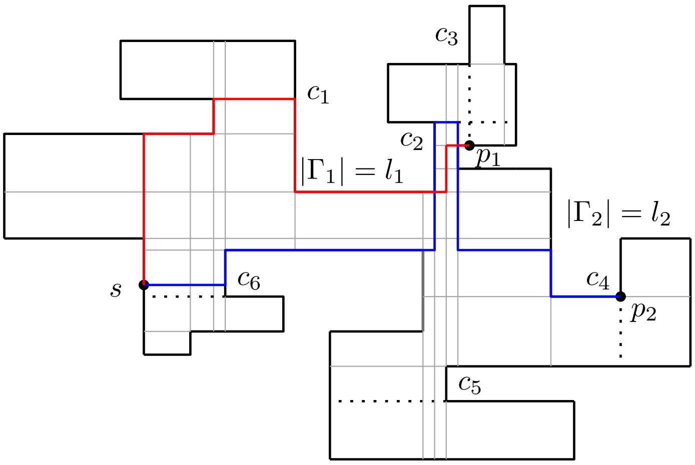

Figure 4 An example subproblem $(c_4, p_1, l_1, p_2, l_2) =$ TRUE, Γ_1 (resp. Γ_2) is drawn in red (resp. blue).

For each Hanan grid point $p \in c_j$ and $i = 1, \ldots, k$, the value of the subproblem $(c_j, p_1, l_1, \ldots, p_i := p, l_i, \ldots, p_k, l_k)$ is computed according to the following disjunction of valid subproblems

$$(c_j, p_1, l_1, \ldots, p_i := p, l_i, \ldots, p_k, l_k) = \bigvee_{p'} (c_{j-1}, p_1, l_1, \ldots, p_i := p', l_i - |\pi^{\perp}(p, p')|, \ldots, p_k, l_k) \tag{1}$$

where p' is taken from the set of all Hanan grid points on the cuts $c_1, \ldots, c_{j-1}$ such that geodesic L_1 shortest paths from p' to c_j make contact with c_j at p (Lemma 3). See Figure 5 for a concrete example of the above recursion.

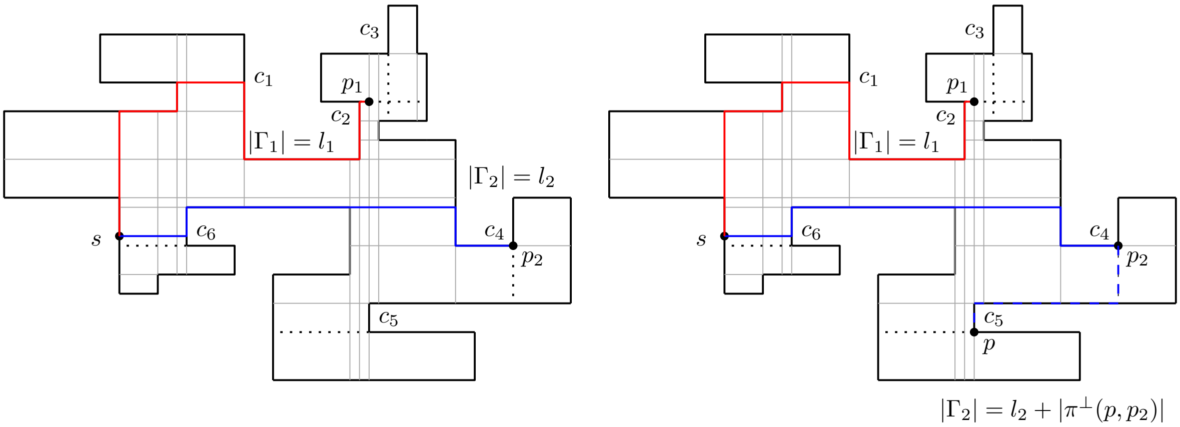

Figure 5 If $(c_4, p_1, l_1, p_2, l_2)$ = TRUE (right), then $(c_5, p_1, l_1, p, l_2 + |\pi^\perp(p, p_2)|)$ = TRUE (left).

The base case is simply $(s, s, 0, \ldots, s, 0)$ = TRUE. From there, we tabulate all possible subproblems, take the subproblem $(c_m, p_1, l_1, \ldots, p_k, l_k)$ whose value is TRUE with the minimum $\max\limits_{i=1,\ldots,k} \{l_i + |\pi^\perp(p_i, s)|\}$ and return the associated paths, each concatenated with the corresponding return trip to the starting point, specifically $\{\gamma_i := \Gamma_i \cup \pi^\perp(p_i, s)\}_{i=1,\ldots,k}$. This gives us an exact optimal solution to the k-Watchman Routes problem in an orthogonal polygon, where the watchmen are restricted to rectilinear movement.

Our proof of correctness relies on two arguments:

- If the paths associated with subproblem $(c_{j-1}, p_1, l_1, \ldots, p_i := p', l_i - |\pi^\perp(p, p')|, \ldots, p_k, l_k)$ visit $s, c_1, \ldots, c_{j-1}$, then those associated with $(c_j, p_1, l_1, \ldots, p_i := p, l_i, \ldots, p_k, l_k)$ visit all essential cuts from s to c_j since $p \in c_j$. By induction, the tours $\{\gamma_i\}_{i=1,\ldots,k}$ returned by the algorithm hence visit all essential cuts.
- γ_i consists of geodesic L_1 shortest paths between contact points with essential cuts (proof of Lemma 3). If we identify two consecutive contact points on γ_i, say p' and p in that order, then the length of the portion of γ_i from s to p is l_i if and only if the length of the portion of γ_i from s to p' is $l_i - |\pi^\perp(p, p')|$.

3.1.2 Analysis of running time

First we bound the number of subproblems in the dynamic program. There are $O(n)$ essential cuts, $O(n)$ Hanan grid points on each cut. The length of each tour γ_i can be trivially upper bounded by the perimeter of P, $|\partial P|$. Hence, in total there are $O\left(n \cdot n^{2k} \cdot |\partial P|^k\right) = O\left(n^{2k+1}|\partial P|^k\right)$ subproblems.

We pre-compute and store geodesic L_1 shortest paths between Hanan grid points, as well as between Hanan grid points and essential cuts by solving the All Pairs Shortest Path problem in the embedded graph of the Hanan grid; the running time of this step is clearly dominated by the dynamic program. Then, we can solve each subproblem by iterating through $O(n^2)$ previously solved subproblems, using recursion (1). The overall running time of the dynamic program is thus $O(n^{2k+3}|\partial P|^k)$, which is pseudopolynomial for any fixed k. This is in congruence with the weak NP-hardness from Partition, for which there exists an algorithm polynomial in the number of input integers and the largest input integer.

We can also achieve a tighter, however straightforward, time bound, which is useful for deriving the FPTAS. Let L be the length of a shortest single orthogonal watchman route of P, which is computable in $O(n)$ time if P is simple and orthogonal (simply use the L_1 shortest path in the algorithm given in [4] instead of the usual L_2 shortest path). Clearly $L \leq |\partial P|$ and $\min \max_{i=1,\ldots,k} |\gamma_i| \leq L$ (one shortest single-watchman route and $k-1$ routes of length 0 is a feasible solution to the k-Watchman Routes problem). Thus, we get a time bound of $O(n^{2k+3}L^k)$.

3.1.3 Fully polynomial-time approximation scheme

The running time of the dynamic programming algorithm depends on the maximum possible length of the routes. To achieve fully polynomial running time for fixed k, we bound the number of subproblems by a "bucketing" technique.

Recall in Section 2, we showed that the min-sum version with a common depot is readily solvable in polynomial time by arguing $L \leq \sum_{i=1}^{k} |\gamma_i|$. Thus

$$\frac{L}{k} \leq \max_{i=1,\ldots,k} |\gamma_i| \leq L. \tag{2}$$

Given any $\varepsilon > 0$, we divide L into $\lceil \frac{nk}{\varepsilon} \rceil$ uniform intervals, each must be no longer than $\frac{\varepsilon L}{nk}$. The length of any geodesic L_1 shortest path we take into consideration for recursion (1) must fall into one of the intervals (otherwise, the upper bound $\max_{i=1,\ldots,k} |\gamma_i| \leq L$ is violated). We round the length of the aforementioned path down to the nearest interval endpoint. Then, apply the dynamic programming algorithm to the new instance with a slight modification: let the intervals' endpoints define the subproblems instead of the k integers $l_1, l_2, \ldots, l_k$. Denote by $\{\gamma'_i\}_{i=1,\ldots,k}$ the solution returned. For clarity, we denote by $d(.)$ distance/length in the "rounded down" instance. We have the following chain of inequalities

$$\max_{i=1,\ldots,k} |\gamma_i| \geq \max_{i=1,\ldots,k} d(\gamma_i) \geq \max_{i=1,\ldots,k} d(\gamma'_i). \tag{3}$$

The first inequality follows simply from the fact that we round down any distance from the original instance. The second inequality is by definition, since $\{\gamma'_i\}_{i=1,\ldots,k}$ is an optimal solution of the new instance.

Now, any route in $\{\gamma'_i\}_{i=1,\ldots,k}$ consists of at most n geodesic L_1 shortest paths between Hanan grid points on essential cuts, the length of each differs by no more than $\frac{\varepsilon L}{nk}$ between the original instance and the "rounded down" instance. Thus, for any i

$$|\gamma'_i| - d(\gamma'_i) \leq n \cdot \frac{\varepsilon L}{nk}$$

and therefore

$$\max_{i=1,\ldots,k} d(\gamma'_i) + \frac{\varepsilon L}{k} \geq \max_{i=1,\ldots,k} |\gamma'_i|. \tag{4}$$

Combining (2), (3), (4), we get

$$(1+\varepsilon) \max_{i=1,\ldots,k} |\gamma_i| \geq \max_{i=1,\ldots,k} |\gamma'_i|.$$

Thus, we achieve an FPTAS with a running time of $O\left(n^{2k+3} \cdot \left(\frac{nk}{\varepsilon}\right)^k\right) = O\left(\frac{n^{3k+3}}{\varepsilon^k}\right)$.

If the watchmen are not constrained to rectilinear movement, then the dynamic programming algorithm and the FPTAS above give a $\sqrt{2}$-approximation and a $(\sqrt{2}+\varepsilon)$-approximation, respectively, for unconstrained movement (L_2 metric).

▶ **Theorem 4.** *For any fixed k, the anchored k-*Watchman Routes *problem in a simple orthogonal polygon has an FPTAS for the L_1 metric and a polynomial-time $(\sqrt{2}+\varepsilon)$-approximation for the L_2 metric.*

3.2 General simple polygon, unconstrained movement

In our dynamic programming algorithm for the orthogonal anchored k-Watchman Routes problem, by restricting to rectilinear movement and axis-aligned essential cuts, we have a discrete set of points at which the routes can come into contact with the essential cuts. We extend that idea to the case in which P is a general simple polygon and the watchmen can travel in any directions with an appropriate approach to discretization, to achieve an FPTAS. Naively discretizing the whole of P into a fine grid can be far too computationally expensive. We overcome this challenge by showing how to localize the problem.

3.2.1 Discretization of P

Denote by $GD_s(r)$ the geodesic disk of radius r centered at the given starting point s, i.e. the set of all points with geodesic distance (length of the geodesic shortest path) no greater r from s. Let $r = r_{\min}$, where $r_{\min}$ is the smallest value of r such that $V(GD_s(r)) = P$, that is, $GD_s(r)$ sees everything in P or equivalently, $\partial GD_s(r)$ is a visibility covering tour of P. To determine $r_{\min}$, simply compute the largest geodesic distance from s to any of the $O(n)$ essential cuts, then $GD_s(r_{\min})$ touches every essential cut and sees all of P. Consider an optimal collection of tours $\{\gamma_i\}_{i=1,\ldots,k}$ with the minimum $\max\limits_{i=1,\ldots,k} |\gamma_i|$.

▶ **Lemma 5.** $r_{\min} \leq \max\limits_{i=1,\ldots,k} |\gamma_i| \leq 9nr_{\min}$.

Proof. The geodesic disk $GD_s\left(\frac{\max\limits_{i=1,\ldots,k}|\gamma_i|}{2}\right)$ contains $\{\gamma_i\}_{i=1,\ldots,k}$, and therefore must see all of P (since P is simple and hole-free). By definition $r_{\min}$ is the smallest radius such that $GD_s(r_{\min})$ sees all of P, so $r_{\min} \leq \frac{\max\limits_{i=1,\ldots,k}|\gamma_i|}{2}$.

In this proof, let L be the length of the shortest single-watchman route of P through s. Clearly, L is an upper bound on $\max\limits_{i=1,\ldots,k} |\gamma_i|$. We show that $L \leq 9nr_{\min}$. Indeed, since $\partial GD_s(r_{\min})$ sees all of P, $L \leq |\partial GD_s(r_{\min})| + 2r_{\min}$ (the right-hand side is the most cost incurred by traveling around $\partial GD_s(r_{\min})$ and the shortest path from s to $\partial GD_s(r_{\min})$ once in each direction).

Finally, observe that $\partial GD_s(r_{\min})$ consists of polygonal chains that are portions of ∂P and at most n circular arcs; the circular arcs have total length no greater than $2n\pi r_{\min}$. Now, each segment in the polygonal parts of $\partial GD_s(r_{\min})$ has its length bounded by the sum of geodesic distances from its endpoints to s (triangle inequality), which is no more than $2r_{\min}$. There are at most n segments in the polygonal portions of $\partial GD_s(r_{\min})$, therefore their total length is no greater than $2nr_{\min}$, implying $|\partial GD_s(r_{\min})| + 2r_{\min} \leq 2nr_{\min} + 2n\pi r_{\min} + 2r_{\min} \leq 9nr_{\min}$. ◀

Starting from $r := r_{\min}$, if we repeatedly multiple r by 2, at some point we must have $r \leq \max\limits_{i=1,\ldots,k} |\gamma_i| \leq 2r$, suppose we have reached this point. Consider a regular square grid of pixels of side lengths δ (we specify δ below) within an axis-aligned closed square box B of size $2r$-by-$2r$ centered on s. The restriction of B within P, $P \cap B$ contains an optimal $\{\gamma_i\}_{i=1,\ldots,k}$, note that this means B necessarily intersects all essential cuts. We triangulate P (using any of the numerous known algorithms, for example, the linear-time algorithm in [3]), including s as a vertex of the triangulation. The arrangement of the square grid, the triangulation and the essential cuts decomposes $P \cap B$ into convex cells, each of perimeter no greater than 4δ (Figure 6). The set of vertices of this decomposition is composed of intersections between chords of the triangulation, the essential cuts and the axis-aligned grid lines, as well as vertices of P and the starting point s.

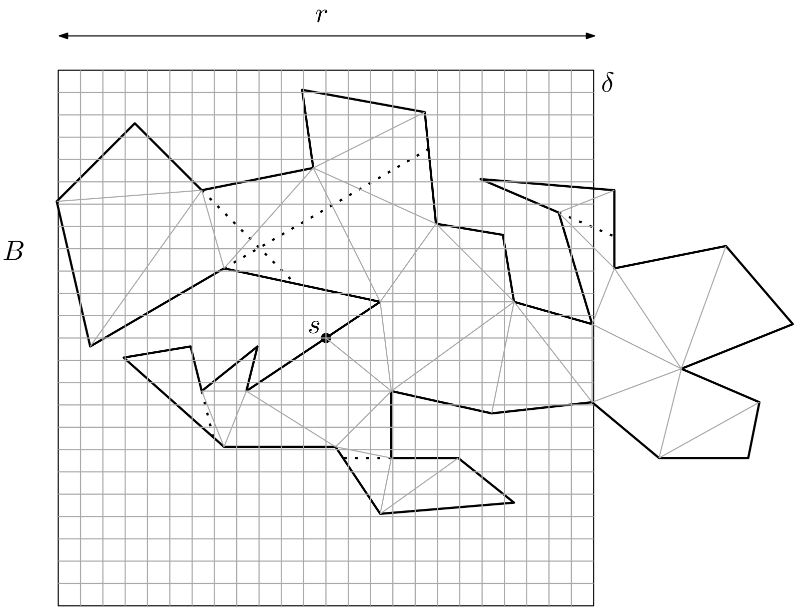

Figure 6 Discretization and localization of the general anchored k-Watchman Routes problem.

We argue that, for purposes of approximation, it suffices to restrict ourselves to grid-rounded solutions, that is, tours that have vertices among vertices of the decomposition. For an arbitrary $i = 1, \ldots, k$, let $\sigma_1, \ldots, \sigma_m$ ($m \leq 2n$) be the (closed) cells of the decomposition containing the vertices of γ_i. Some vertices of γ_i may lie on the common boundary of multiple cells, in which case we can choose any of those cells. The relative convex hull of $\gamma_i \cup \sigma_1 \cup \ldots \cup \sigma_m$ is the minimum-perimeter subset of P that contains it, and must thus have perimeter no greater than $|\gamma_i| + |\partial\sigma_1| + \ldots + |\partial\sigma_m| \leq |\gamma_i| + 8n\delta$. Moreover, since $\sigma_1, \ldots, \sigma_m$ are convex and enclose vertices of γ_i, the vertices of the relative convex hull $\gamma_i \cup \sigma_1 \cup \ldots \cup \sigma_m$ are vertices of $\sigma_1, \ldots, \sigma_m$.

Hence, for any $\varepsilon > 0$, if we let $\delta = \frac{\varepsilon L}{8nk}$, then there exists a grid-rounded tour γ'_i such that $|\gamma'_i| \leq (1+\varepsilon)|\gamma_i|$. It is straightforward to argue that γ'_i consists of consecutive geodesic shortest paths between grid points on some essential cuts (proof of Lemma 3).

3.2.2 Fully polynomial-time approximation scheme

The very same dynamic programming algorithm for the anchored orthogonal k-Watchman Routes works to compute (approximately) $\{\gamma'_i\}_{i=1,\ldots,k}$, though we require some modifications, which we have actually introduced. In that algorithm, we have integers defining subproblems since Hanan grid points have integer coordinates, thus the length of any rectilinear path we consider is integral. In the general case with the L_2 metric, lengths and distances are potentially irrational even with the assumption that the vertices of P have integer coordinates. Consequently, there is no natural increment we can set between consecutive values.

We compute the shortest single-watchman route, let its length be L. Recall that we argued the following chain of inequalities for rectilinear routes, but the same can be said for the general case since orthogonality did not play a part in the argument:

$$\frac{L}{k} \leq \max_{i=1,\ldots,k} |\gamma_i| \leq L.$$

We divide L into $\lceil \frac{nk}{\varepsilon} \rceil$ uniform intervals and round the length of any geodesic shortest path we consider in the algorithm down to the nearest interval endpoint. Using the same analysis of the approximation factor as in the rectilinear routes case, the dynamic program with subproblems defined by the intervals' endpoints returns a $(1+\varepsilon)$-approximation to $\{\gamma'_i\}_{i=1,\ldots,k}$, which is a $(1+\varepsilon)^2$-approximation to $\{\gamma_i\}_{i=1,\ldots,k}$.

It remains to show that the dynamic programming algorithm with the above discretization of P has fully polynomial running time (for fixed k). The number of grid lines within $P \cap B$ is $O\left(\frac{r}{\delta}\right) = O\left(\frac{nk}{\varepsilon}\right)$. Therefore, the number of grid points on the essential cuts (intersections of essential cuts with triangulation chords and grid lines) is $O(n) \cdot O(n) + O(n) \cdot O\left(\frac{nk}{\varepsilon}\right) = O\left(\frac{n^2k}{\varepsilon}\right)$. There are in total $O\left(n \cdot \left(\frac{n^2k}{\varepsilon}\right)^k \cdot \left(\frac{nk}{\varepsilon}\right)^k\right) = O\left(\frac{n^{3k+1}}{\varepsilon^{2k}}\right)$ subproblems, each of which requires iterating through $O\left(\frac{n^2k}{\varepsilon}\right)$ previously solved subproblems. This results in an overall $O\left(\frac{n^{3k+3}}{\varepsilon^{2k+1}}\right)$ running time for the dynamic program, which computes a $(1+\varepsilon)^2$-approximation. Note that if we let $2\varepsilon + \varepsilon^2 = \varepsilon'$, then $\frac{1}{\varepsilon} = \Theta\left(\frac{1}{\varepsilon'}\right)$ as ε and ε' approach 0, thus the running time is in the same order when written in terms of ε'.

We summarize the description of our algorithm as follows:

Step 1: Compute L, the length of the the shortest single-watchman route for P.

Step 2: Triangulate P.

Step 3: Compute $r_{\min}$, the smallest value for which $GD_s(r_{\min})$ sees all of P. Set $r := r_{\min}$.

Step 4: Overlay a square grid of δ-size pixels within a bounding box B of size $2r$-by-$2r$ onto the triangulation of P and the essential cuts, where $\delta = \frac{\varepsilon L}{8nk}$.

Step 5: If B intersects all essential cuts, execute the dynamic program, output the collection of routes $\{\gamma'_i\}_{i=1,\ldots,k}$ and store it.

Step 6: If $r \leq 9nr_{\min}$, then set $r := 2r$, then repeat from Step 4. Else, terminate the algorithm.

We return the collection $\{\gamma'_i\}_{i=1,\ldots,k}$ that minimizes $\max\limits_{i=1,\ldots,k} |\gamma'_i|$ out of all collections from all values of r in the doubling search (henceforth, we use doubling search to refer to recursively multiplying a quantity by 2 until it is greater than an upper bound). Since the doubling search for r takes $\log(9n) = O(\log n)$ iterations, our algorithm runs in time $O\left(\frac{n^{3k+3}}{\varepsilon^{2k+1}} \log n\right)$.

▶ **Theorem 6.** *For any fixed k, the anchored k-Watchman Routes problem in a simple polygon has an FPTAS.*

4 Multiwatchman routes with the number of watchmen as part of the input

4.1 Seeing all of P

The idea of localizing an optimal set of routes $\{\gamma_i\}_{i=1,\ldots,k}$ with a geodesic disk whose radius is no larger than $O\left(\max\limits_{i=1,\ldots,k} |\gamma_i|\right)$ gives us a simple constant-factor approximation. Similar to the previous section, suppose r is such that $\frac{r}{2} \leq \frac{\max_{i=1,\ldots,k} |\gamma_i|}{2} \leq r$, then the geodesic disk of radius r centered on s, $GD_s(r)$ sees all of P. Let $\gamma \subset GD_s(r)$ be the shortest route that sees all of P within $GD_s(r)$. Note that γ is not necessarily the overall shortest watchman route of P, but we can still compute γ using the polygon touring algorithm [6, 22]. Any essential cut c must intersect $GD_s(r)$, a relatively convex set, in a contiguous segment. By applying the polygon touring algorithm on the segments that are intersection of the essential cuts with $GD_s(r)$, we obtain γ. Divide γ into k subpaths of equal length, each of which is bounded by two endpoints $a_i, a_{i+1} \in \gamma$ ($a_1 \equiv s \equiv a_{k+1}$) and denoted by $\gamma_{a_i a_{i+1}}$. Then, return the collection of routes $\{\gamma'_i\}_{i=1,\dots,k}$ where $\gamma'_i = \pi(s, a_i) \cup \gamma_{a_i a_{i+1}} \cup \pi(a_{i+1}, s)$.

The approximation algorithm can be summarized in the following steps:

Step 1: Compute $r_{\min}$, the smallest value for which $GD_s(r_{\min})$ sees all of P. Set $r := r_{\min}$.

Step 2: If $GD_s(r)$ intersects all essential cuts, compute γ, the shortest route within $GD_s(r)$ that sees all of P using the polygon touring algorithm.

Step 3: Divide γ into k subpaths of equal length, each bounded by $a_i, a_{i+1} \in \gamma$ ($a_1 \equiv s \equiv a_{k+1}$) and denoted by $\gamma_{a_i a_{i+1}}$.

Step 4: For each i, we obtain γ'_i by joining $\pi(s, a_i), \gamma_{a_i a_{i+1}}$ and $\pi(a_{i+1}, s)$.

Step 5: Set $r := 2r$, then repeat from Step 2, until $r > 9nr_{min}$.

Finally, we choose the collection of routes $\{\gamma'_i\}_{i=1,\dots,k}$ that minimizes $\max\limits_{i=1,\dots,k} |\gamma'_i|$ out of all collections from all values of r in the doubling search.

▶ **Theorem 7.** *The above algorithm is a polynomial-time 3-approximation.*

Proof. First, note that $\frac{|\gamma|}{k} \leq \max\limits_{i=1,\dots,k} |\gamma_i| \leq |\gamma|$.

On the other hand, the geodesic distance from any point on γ to s is no longer than r. Thus, when $r \leq \max\limits_{i=1,\dots,k} |\gamma_i| \leq 2r$, any γ'_i returned by the algorithm is no longer than $\frac{|\gamma|}{k} + 2r \leq 3 \max\limits_{i=1,\dots,k} |\gamma_i|$.

We can determine $r_{\min}$ by computing the geodesic distance from s to any of the $O(n)$ essential cuts in $O(n^2)$ time. For each value of r, we execute the $O(n^3)$ polygon touring algorithm [22] and compute the k-collection $\{\gamma'_i\}_{i=1,\dots,k}$ in $O(nk)$ time. The doubling search for r takes $O(\log n)$ iterations (Lemma 5), thus the algorithm incurs a total running time of $O\left((n^3 + nk)\log n\right)$ ◀

. The running time can be made faster, if in Step 2, instead of computing an exact route γ, we compute a $\sqrt{2}$-approximate route by using the linear time approximation algorithm given by Tan [19]. The approximation factor in that case would be $3\sqrt{2} \approx 4.24$ and the running time would be $O\left((n^2 + nk)\log n\right)$.

4.1.1 Improving the approximation factor

In the approximation algorithm earlier, we gradually expand $GD_s(r)$ until $GD_s(r)$ contains an optimal $\{\gamma_i\}_{i=1,\dots,k}$. If in each iteration, we instead multiply r by a smaller positive number, such as $(1+\varepsilon)$, then at some point $\frac{r}{(1+\varepsilon)} \leq \frac{\max\limits_{i=1,\dots,k} |\gamma_i|}{2} \leq r$. The distance from each point on γ to s is then no greater than $(1+\varepsilon)\frac{\max\limits_{i=1,\dots,k} |\gamma_i|}{2}$. Hence, the length of any of the k routes returned by the approximation algorithm is bounded by $\frac{|\gamma|}{k} + 2(1+\varepsilon)\frac{\max\limits_{i=1,\dots,k} |\gamma_i|}{2} \leq (2+\varepsilon) \max\limits_{i=1,\dots,k} |\gamma_i|$.

There is however, a trade-off between the approximation factor and the number of iterations of the multiplicative search for r. If we multiply r by $(1+\varepsilon)$ each time, the search requires $O(\log_{1+\varepsilon} n)$ iterations. Note that

$$\log_{1+\varepsilon} n = \log n \frac{\ln 2}{\ln(1+\varepsilon)} = \log n O\left(\frac{1}{\varepsilon}\right).$$

In summary, we can achieve an approximation ratio of $(2+\varepsilon)$ with a running time of $O\left(\frac{(n^3+nk)\log n}{\varepsilon}\right)$, or $(2\sqrt{2}+\varepsilon)$ with a running time of $O\left(\frac{(n^2+nk)\log n}{\varepsilon}\right)$ (by using Tan's linear-time approximation algorithm rather than the exact algorithm).

4.2 Seeing a quota of area within P

Given an area quota A, let $\{\gamma_i\}_{i=1,\ldots,k}$ be such that $\left|\bigcup_{i=1,\ldots,k} V(\gamma_i)\right| \geq A$ and $\max_{i=1,\ldots,k} |\gamma_i|$ is minimum. To discretize P and localize $\{\gamma_i\}_{i=1,\ldots,k}$, we employ a technique similar as before, only more generalized. Instead of computing the smallest geodesic disk centered at s that sees all of P, we compute the smallest disk that sees A, the given quota of area. Let $r_{\min}$ be the smallest value of r such that $|V(GD_s(r)| = A$. We compute $r_{\min}$ using the "visibility wave" method in [1]. Consider the visibility graph of P, whose nodes are vertices of P and two nodes are adjacent if they see one another. We sort the geodesic distance from s to every visibility edge in increasing order in $O(n^2 \log n)$ time. Then, propagate a "wavefront" through the sequences of visibility edges hit for the first time in the process of increasing r, stop when $|V(GD_s(r)| = A$.

▶ **Lemma 8.** $r_{\min} \leq \max_{i=1,\ldots,k} |\gamma_i| = O(nr_{\min})$.

Proof. Identical to that of Lemma 5. ◀

Lemma 8 allows us to find r such that $r \leq \max_{i=1,\ldots,k} |\gamma_i| \leq 2r$ by a doubling search from $r = r_{\min}$ in $O(\log n)$ iterations. Suppose $\max_{i=1,\ldots,k} |\gamma_i| \leq 2r$, then $GD_s(r)$ contains $\{\gamma_i\}_{i=1,\ldots,k}$. We compute (approximately) a shortest tour within $GD_s(r)$ that sees at least an area of A, which we denote by γ. Such a route must exist, since $\{\gamma_i\}_{i=1,\ldots,k}$ collectively see an area no smaller than A.

▶ **Lemma 9.** *[11, Section 3] Given a budget $B \geq 0$ and any $\varepsilon > 0$, there exists an $O\left(\frac{n^5}{\varepsilon^6}\right)$ algorithm that computes a route of length at most $(1+\varepsilon)B$ seeing as much area as any route of length B within $GD_s(r)$.*

We briefly describe the algorithm for completeness, and refer the readers to [11] for more details. First, triangulate P, including s as a vertex of the triangulation. Then, overlay onto the triangulation a regular square grid of side lengths $\delta = O\left(\frac{\varepsilon B}{n}\right)$ within an axis aligned square of size B-by-B centered at s.

We consider the set of (convex) cells of the decomposition that overlap (both fully and partially) with $GD_s(r)$ and their vertices, $S_{\delta,r}$. Let γ_B be the B-length route within $GD_s(r)$ that achieves the most area of visibility. Then, there exists a route of length at most $(1+\varepsilon)B$ with vertices coming from $S_{\delta,r}$ enclosing γ_B, i.e. the boundary of the relative convex hull of the cells containing vertices of γ_B, thus seeing at least as much area as γ_B (Figure 7).

The dynamic programming algorithm for the Budgeted Watchman Route problem can compute a relatively convex closed curve that sees the most with vertices among $S_{\delta,r}$. Sort $S_{\delta,r} = \{s_1, s_2, \ldots\}$ in (geodesic) angular order around s to acquire a topological ordering of the subproblems. A subproblem $(s_j, \overline{B})$ is defined by a point in $S_{\delta,r}$ and a length. Let

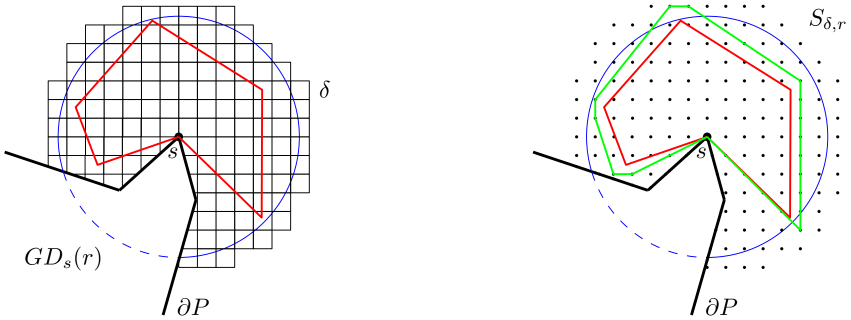

Figure 7 Left: γ_B (red) is a tour no longer than B within $C_g(r)$ (blue) that sees the most area. Right: enclosing γ_B with a tour whose vertices are in $S_{\delta,r}$ seeing everything γ_B sees (green).

$a(s_j,\overline{B})$ be the maximum area of visibility that a relatively convex polygonal chain from s to s_j of length no greater than $\overline{B}$ can achieve, then

$$a(s_j,\overline{B}) = \max_{i\leq j} a(s_i,\overline{B}-|s_is_j|) + |V(s_is_j) \setminus V(\pi(s,s_i))|$$

(recall that $\pi(s,s_i)$ is the geodesic shortest path between s and s_i).

Thus, if $|\gamma| \leq B \leq \alpha|\gamma|$ for some $\alpha \geq 1$, we can compute a route γ' of length no longer than $\alpha(1+\varepsilon)|\gamma|$ with vertices in $S_{\delta,r}$ that sees the most area, which must be no smaller than $|V(\gamma_B)| \geq |V(\gamma)| \geq A$. We show how to acquire a polynomial-sized set of values, from which we can "guess" B so that B is close to $|\gamma|$ (in particular, $|\gamma| \leq B \leq (1+\varepsilon)|\gamma|$).

▶ **Lemma 10.** $r_{\min} \leq |\gamma| \leq 9nr_{\min}$.

Proof. Identical to that of Lemma 5. ◀

We divide $9nr_{min}$ into $\lceil \frac{9n}{\varepsilon} \rceil$ uniform intervals so that each is no longer than εr_{min}: the smallest interval endpoint that is no smaller than $|\gamma|$ must also be no larger than $(1+\varepsilon)|\gamma|$, and hence is the value of B that we desire. We perform a binary search on the values $\left\{0, \frac{9nr_{min}}{\lceil \frac{9n}{\varepsilon} \rceil}, \ldots, 9nr_{min}\right\}$ as the input budget for the algorithm in Lemma 9, and seek out the smallest value for which the output route γ' sees an area no smaller than A. Clearly, $|\gamma'| \leq (1+\varepsilon)^2|\gamma|$.

We are now ready to describe the approximation algorithm for the Quota k-Watchmen Routes problem as follows:

Step 1: Set $r := r_{\min}$.

Step 2: Compute γ', a $(1+\varepsilon)^2$-approximation to γ.

Step 3: Divide γ' into k subpaths of equal length, each bounded by $a_i, a_{i+1} \in \gamma'$ ($a_1 \equiv s \equiv a_{k+1}$) and denoted by $\gamma'_{a_ia_{i+1}}$.

Step 4: For each i, we obtain γ'_i by joining $\pi(s,a_i), \gamma'_{a_ia_{i+1}}$ and $\pi(a_{i+1},s)$.

Step 5: Set $r := 2r$, then repeat from Step 2, until $r > 9nr_{min}$.

Return the collection of routes $\{\gamma'_i\}_{i=1,\ldots,k}$ that minimizes $\max\limits_{i=1,\ldots,k} |\gamma'_i|$.

We argue that the algorithm described above has an approximation factor of $3+\varepsilon$, where ε is arbitrarily close to 0. First. we once again note that $\frac{|\gamma|}{k} \leq \max\limits_{i=1,\ldots,k} |\gamma'_i| \leq |\gamma|$.

Now, since all our choices for B are no larger than $9nr_{\min}$, we can choose an appropriate $\delta = O\left(\frac{\varepsilon B}{n}\right)$ so that the geodesic distance from any point on γ' to s is no longer than $r + \varepsilon r$.

Thus, when $\frac{r}{2} \leq \frac{\max\limits_{i=1,\ldots,k} |\gamma_i|}{2} \leq r$, any one of the k routes returned by the algorithm is no longer than $\frac{|\gamma'|}{k} + 2r + 2\varepsilon r \leq [(1+\varepsilon)^2 + 2 + 2\varepsilon] \max\limits_{i=1,\ldots,k} |\gamma_i| = (3+\varepsilon') \max\limits_{i=1,\ldots,k} |\gamma_i|$, where $\varepsilon' = 4\varepsilon + \varepsilon^2$. Since $\frac{1}{\varepsilon} = \Theta\left(\frac{1}{\varepsilon'}\right)$ as ε and ε' approach 0, the running time is in the same order when written in terms of either ε or ε'.

For each choice of B, we execute the $O\left(\frac{n^5}{\varepsilon^6}\right)$ algorithm, thus computing an approximation to γ' for each value of r takes $O\left(\frac{n^5}{\varepsilon^6} \log\left(\frac{n}{\varepsilon}\right)\right)$ time. Deriving the collection $\{\gamma'_i\}_{i=1,\ldots,k}$ takes $O(nk)$ time. Since there are $O(\log n)$ iterations of the doubling search for r, the overall running time is $O\left(\left(\frac{n^5}{\varepsilon^6} \log\left(\frac{n}{\varepsilon}\right) + nk\right) \log n\right)$.

Similar to the k-Watchman Routes problem, we can improve the approximation factor, in particular, to $(2+\varepsilon)$ by multiplying r with $(1+\varepsilon)$ instead of 2 in the multiplicative search. The running time in that case, is $O\left(\left(\frac{n^5}{\varepsilon^6} \log\left(\frac{n}{\varepsilon}\right) + nk\right) \frac{\log n}{\varepsilon}\right)$.

▶ **Theorem 11.** *Both the anchored k-Watchman Routes problem and the anchored Quota k-Watchman Routes problem in a simple polygon have a polynomial-time $(2+\varepsilon)$-approximation.*

5 Conclusion and future work

The Watchman Route problem is a classic problem in computational geometry, with many practical implications. Even though the single-watchman version is well studied, the multiple watchmen version is much less so. We give tight approximability results when the number of watchmen is fixed and a common depot is specified, strengthening our theoretical understanding of the problem. The main source of intractability and approximation hardness of the problem seems to stem from having multiple depots (or none at all), in which case the notion of essential cuts is not well-defined, and we do not have a good characterization of when a set of routes see everything.

Even in the anchored setting, many questions remain open. Can we get a PTAS when k, the number of watchmen, is part of the input? Is there a fixed-parameter tractable algorithm? What about in a polygon with holes? We suspect new ideas and algorithmic techniques are needed to answer these questions.

References

1 Esther M. Arkin, Alon Efrat, Christian Knauer, Joseph S. B. Mitchell, Valentin Polishchuk, Günter Rote, Lena Schlipf, and Topi Talvitie. Shortest path to a segment and quickest visibility queries. *Journal of Computational Geometry*, 7(2):77–100, 2016.

2 Svante Carlsson, Håkan Jonsson, and Bengt J. Nilsson. Finding the shortest watchman route in a simple polygon. *Discrete & Computational Geometry*, 22:377–402, 1999.

3 Bernard Chazelle. Triangulating a simple polygon in linear time. *Discrete & Computational Geometry*, 6(3):485–524, 1991.

4 Wei-Pang Chin and Simeon Ntafos. Optimum watchman routes. In *Proceedings of the 2nd Annual Symposium on Computational Geometry*, pages 24–33, 1986.

5 Wei-Pang Chin and Simeon Ntafos. Shortest watchman routes in simple polygons. *Discrete & Computational Geometry*, 6(1):9–31, 1991.

6 Moshe Dror, Alon Efrat, Anna Lubiw, and Joseph S. B. Mitchell. Touring a sequence of polygons. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 473–482, 2003.

7 Adrian Dumitrescu and Csaba D. Tóth. Watchman tours for polygons with holes. *Computational Geometry*, 45(7):326–333, 2012.

8 Jan Faigl. Approximate solution of the multiple watchman routes problem with restricted visibility range. *IEEE Transactions on Neural Networks*, 21(10):1668–1679, 2010.

9 Leonidas Guibas, John Hershberger, Daniel Leven, Micha Sharir, and Robert Tarjan. Linear time algorithms for visibility and shortest path problems inside simple polygons. In *Proceedings of the 2nd Annual Symposium on Computational Geometry*, pages 1–13, 1986.

10 Maurice Hanan. On Steiner's problem with rectilinear distance. *SIAM Journal on Applied mathematics*, 14(2):255–265, 1966.

11 Kien C. Huynh, Joseph S. B. Mitchell, Linh Nguyen, and Valentin Polishchuk. Optimizing visibility-based search in polygonal domains. In *Proceedings of the 19th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2024)*, volume 294 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:16, 2024.

12 Joseph S. B. Mitchell. Geometric shortest paths and network optimization. *Handbook of Computational Geometry*, 334:633–702, 2000.

13 Joseph S. B. Mitchell. Approximating watchman routes. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 844–855. SIAM, 2013.

14 Joseph S. B. Mitchell and Linh Nguyen. Multirobot watchman routes in a simple polygon. In *Proceedings of the 36th Canadian Conference on Computational Geometry*, pages 49–55, 2024.

15 Joseph S. B. Mitchell and Erik L. Wynters. Watchman routes for multiple guards. In *Proceedings of the 3rd Canadian Conference on Computational Geometry*, pages 126–129, 1991.

16 Bengt J. Nilsson and Eli Packer. Approximation algorithms for the two-watchman route in a simple polygon. *Algorithmica*, pages 1–40, 2024.

17 Eli Packer. Computing multiple watchman routes. In *Experimental Algorithms: 7th International Workshop, WEA 2008 Provincetown, MA, USA, May 30-June 1, 2008 Proceedings 7*, pages 114–128. Springer, 2008.

18 Michael I. Shamos. *Computational Geometry.* Yale University, 1978.

19 Xuehou Tan. Approximation algorithms for the watchman route and zookeeper's problems. *Discrete Applied Mathematics*, 136(2-3):363–376, 2004.

20 Xuehou Tan, Tomio Hirata, and Yasuyoshi Inagaki. An incremental algorithm for constructing shortest watchman routes. *International Journal of Computational Geometry & Applications*, 3(04):351–365, 1993.

21 Xuehou Tan, Tomio Hirata, and Yasuyoshi Inagaki. Corrigendum to "an incremental algorithm for constructing shortest watchman routes". *International Journal of Computational Geometry & Applications*, 9(03):319–323, 1999.

22 Xuehou Tan and Bo Jiang. The touring polygons problem revisited. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 101(5):772–777, 2018.

23 Csaba D. Tóth, Joseph O'Rourke, and Jacob E. Goodman. *Handbook of Discrete and Computational Geometry.* CRC press, 2017.