Generalized Watchman Route Problem with Discrete View Cost

Pengpeng Wang [*] , Ramesh Krishnamurti [†], and Kamal Gupta [*]

Abstract

In this paper, we introduce a generalized version of the Watchman Route Problem (WRP) where the objective is to plan a continuous closed route in a polygon (possibly with holes) and a set of discrete viewpoints on the planned route. Each planned viewpoint has some associated cost. The total cost to minimize is a weighted sum of the view cost, proportional to the number of viewpoints, and the travel cost, the total length of the route. We call this problem the Generalized Watchman Route Problem or the GWRP in short. We tackle a restricted nontrivial (it remains NP-hard and log-inapproximable) version of GWRP where each polygon edge is entirely visible from at least one planned viewpoint. We call it *Whole Edge Covering GWRP*. Our algorithm proposed first constructs a graph that connects $O(n^{12})$ number of sample points in the polygon, where n is the number of polygon vertices; and then solves the corresponding *View Planning Problem with Combined View and Traveling Cost*, using an LP-relaxation based algorithm we introduced in [WKG06]. We show that our algorithm has an approximation ratio in the order of either the view frequency, defined as the maximum number of sample points that cover a polygon edge, or a polynomial of $\log n$, whichever is smaller.

1 Introduction

The watchman route problem (WRP) refers to planning a closed curve, called a *watchman route*, in a polygon (possibly with holes), with the shortest distance such that every point on the polygon boundary is visible from at least one point on the route. Here we consider the anchored version where a point on the route, called the *start position*, denoted by S, is given [LL86]. WRP combines elements of two NP-hard problems, namely the *Art Gallery Problem with Point Guards* [LL86], denoted by *Point AGP*, and the *Euclidean Traveling Salesman Problem* [Pap77], denoted by *Euclidean TSP*. While for polygon with holes WRP is NP-hard [CN88], for simple polygons (i.e., without holes), it is solvable in polynomial time [CN91, Tan04].

However, WRP makes impractical assumptions that the watchman senses continuously along the route (taking infinite number of viewpoints) and that the sensing actions do not incur any cost. For instance, in an environment inspection task by an autonomous robot-sensor system, each sensing action incurs a large overhead, corresponding to image acquisition, feature extraction from the image, and integration with existing environment model [SRR03]. In addition, often for better sensing qualities, the robot has to stop its movements during image acquisitions. We relax this continuous sensing assumption of WRP, and introduce a view cost, in addition to the existing path cost. We call the resulting problem, generalized watchman route with discrete view cost, or GWRP in short. It refers to planning both a route and a number of discrete points on it, called viewpoints, such that every point on the polygon boundary is visible from at least one planned viewpoint; while the cost incurred is minimized. The cost is a weighted sum of both view cost, proportional to the number of viewpoints planned, and the traveling cost, the total length of the route planned. Note that GWRP is not a simple extension to the WRP. First, for cases where traveling cost is negligible, GWRP is reduced to Point AGP. So unlike WRP, which is in **P** for simple polygons, the GWRP is **NP**-hard. Second, as noticed in [GB04], the optimal WRP solution may incur an unbounded cost for the corresponding GWRP solution, i.e., infinite number of viewpoints are needed on the route to cover the whole polygon boundary.

In this paper, we consider a nontrivial restricted version of the GWRP, called the *Whole Edge Covering GWRP*, or *WEC-GWRP* in short, in which any polygon edge is required to be entirely visible from at least one planned viewpoint. The restriction arises naturally in inspection tasks in robotic applications,

[*] School of Engineering Science, Simon Fraser University, Burnaby, B.C., Canada. Email: {pwangf, kamal}@cs.sfu.ca

[†] School of Computing Science, Simon Fraser University, Burnaby, BC, Canada. Email: ramesh@cs.sfu.ca

where the "map" given to the robot is often a discretized boundary representation and during inspection tasks each small discretized boundary piece is considered as inspected via one planned viewpoint if and only if all the points on it are visible. Thus, by regarding each piece as a polygon edge, we have a whole edge covering instance. The same restriction is also used in the terrain guarding problem [Eid02]. WEC-GWRP has the same NP-hardness and inapproximability as GWRP. The reductions used for establishing the NP-hardness for Point AGP construct *whole edge covering* Point AGP instances, for polygons without holes [LL86] and polygons with holes [OS83] respectively, from an arbitrary 3-Satisfiability instance. In addition, the inapproximability result for Point AGP, i.e., Point AGP is log-inapproximable for polygons with holes, follows from the reduction from an arbitrary Set Covering Problem instance to a *whole edge covering* Point AGP instance.

Although a natural and nontrivial generalization to both the AGP and the WRP, to the best of our knowledge, there are few related works for the GWRP or WEC-GWRP. In [GB97a, GB97b, CN99], the authors consider a simpler problem of choosing a set of discrete viewpoints on a *given* route, while maintaining the visible polygon boundary the same as that by the route. Since such a route is not available to (WEC-) GWRP, this approach cannot be applied here.

In [WKG06], we consider a related problem, the problem of view planning with combined view and travel cost, or the *Traveling VPP* in short. Given a number of discrete viewpoints connected via a graph, Traveling VPP asks for a subset of the viewpoints and a route connecting them such that the boundary edges of a given object (can be the whole polygon boundary as in this paper) are all covered, while minimizing the cost, a weighted sum of both the view cost, proportional to the viewpoints planned, and the travel cost, the distance traveled to realize the planned viewpoints. We gave an LP-relaxation based rounding algorithm called *Round and Connect* that first takes the LP optimal solution and choose the viewpoints greedily according to their LP solution values, and then solves the Steiner tree problem [Vaz01] to connect the chosen viewpoints. We show that the approximation ratio of Round and Connect is in the order of view frequency, defined as the maximum number of viewpoints that cover a single boundary edge. In addition, we gave a reduction that, given an arbitrary Traveling VPP instance, constructs a Group Steiner Tree problem (GST) instance [GKR00] in polynomial time. By calling the poly-log approximation algorithm for GST in [GKR00] after the reduction, we can approximate the optimal solution of Traveling VPP within a poly-log ratio. Thus, Traveling VPP admits an approximation algorithm with the approximation ratio of either the order of view frequency or a poly-log function.

A key distinction between the Traveling VPP and the WEC-GWRP is that while the former has a discrete viewpoint set given in advance, for the latter, we have to deal with the continuous polygon interior and an infinite number of possible viewpoints. One could either discretize up to a certain resolution or randomly sample the polygon interior, and then call the Traveling VPP solver. For example, in [EHP02], the authors assume that a dense grid laid on the polygon is available and viewpoints are restricted to be grid vertices. In [GBL01], the authors propose to randomly sample the polygon to get discrete viewpoints. These approaches have an immediate drawback. If the polygon has a small kernel (the set of viewpoints that sees the whole polygon boundary), in which case a single point in this kernel becomes the optimal solution, both methods are likely to fail: the discretization method has to make the resolution very fine; and to sample a point in the kernel becomes a rare event for the randomized sampling method.

In this paper, we propose a novel sampling algorithm that computes a finite number ($O(n^{12})$, n being the number of polygon vertices) of discrete viewpoints in the polygon. We show that if we restrict the problem to choose planned viewpoints only from these sample points, the optimal cost of such solutions is at most a constant times the cost of the true optimal WEC-GWRP solution. We then construct a Traveling VPP instance using the sample viewpoints and call the approximation algorithm in [WKG06] to solve for a solution. This implies that the resulting solution cost is at most the cost of the optimal solution to WEC-GWRP times either the order of the view frequency or a polynomial of $\log n$, whichever is smaller.

The sampling algorithm works in two steps: first it reduces the viewpoint space from the polygon (2D) to a bounded number of line segments (1D), and then from these line segments (1D) to a bounded number of points. In the first step, we decompose the polygon into *visibility cells*, computed via a partition such that the same polygon edges are entirely visible from all points in each cell. We then restrict the planned viewpoints to be on the visibility cell edges. The reason is as follows. First, we can easily check if all the polygon edges are entirely visible from the start position S of the watchman route. If so, S is the optimal WEC-GWRP solution. If it is not the case, for any feasible WEC-GWRP solution, any other planned viewpoint X cannot belong to the same visibility cell as S, and the route connecting X and S must cross some edge of the visibility cell that X belongs to. After replacing X with the crossing point, we have a feasible WEC-GWRP solution with the same cost and all planned viewpoints are on the visibility cell edges.

Note that if traveling cost is ignored, it suffices to sample one viewpoint arbitrarily on each visibility cell edge. However, due to the view and travel tradeoff, we do not know where on each cell edge the optimal WEC-GWRP solution may choose as the viewpoint. This motivates us to utilize the metric structure in the problem to guide our sampling from 1D to points. We define a local region of each visibility cell edge, called *domain*, and compute a bounded number of viewpoints inside the domains such that the optimal WEC-GWRP solution can be approximated (within a constant ratio) locally using these sample points. For sampling inside each domain, intuitively, we would like to impose an "ordering" on the cell edges, which lets us exploit the weak "metric" between them. This is achieved via dividing domains into *strips* using the visibility cell vertices such that the cell edge ordering remains the same within a strip. We also show the optimal WEC-GWRP solution as a whole can be approximated within a constant ratio once all the local approximations are chained together.

The rest of the paper is organized as follows. First, we give notations and formulate the WEC-GWRP. Second, we give the sampling algorithm. Third, we analyze the approximation ratio of proposed algorithm. Last, we conclude and discuss potential applications of the proposed sampling algorithm.

2 Problem definition

We now formally state the WEC-GWRP. Let $\mathcal{P}$ denote the given polygon (with or without holes), a closed set. Let $\partial\mathcal{P}$ denote its boundary, including the boundary of the holes. Let $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$ and $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ denote the set of polygon vertices and the set of polygon edges, respectively. Let $\mathcal{A}_r$ denote the set of reflex vertices of $\mathcal{P}$ (internal angle > 180 degrees). We use $\overline{X_1X_2}$ to denote the closed line segment between two points X_1 and X_2. Under the line-of-sight assumption, the *visibility* relation between two points X_1 and X_2, i.e., "X_1 is visible from X_2" or "X_2 is visible from X_1", denoted by $X_1 \diamond X_2$, is defined as:

$$X_1 \diamond X_2 \Leftrightarrow \overline{X_1X_2} \subseteq \mathcal{P}.$$

If X_1 is a viewpoint, we say "X_1 *covers* X_2". We also say viewpoint X covers a polygon edge e, denoted by $X \diamond e$, if all points of e are visible from X, i.e.,

$$X \diamond e \Leftrightarrow \forall X' \in e : X \diamond X'.$$

Let $S \in \mathcal{P}$ denote the start position of the watchman. Let $\mathcal{V}'$ denote a subset of viewpoints, i.e., $\mathcal{V}' = \{X : X \in \mathcal{P}\}$ and $route(\mathcal{V}')$ denote a route connecting the viewpoints in $\mathcal{V}'$ and S. Let w_v and w_p denote the weights for the view and traveling costs, respectively. Let $|\mathcal{B}|$ denote the cardinality of a discrete set $\mathcal{B}$, and let $\|\phi\|$ denote the length of route ϕ.

The WEC-GWRP is defined as follows:

$$\begin{array}{rl} \min & w_v|\mathcal{V}'| + w_p\|route(\mathcal{V}')\| \\ \text{Subject to} & \forall e \in \mathcal{E}, \exists X \in \mathcal{V}' : \quad X \diamond e \end{array} \qquad (1)$$

3 Sampling Algorithm

The sampling algorithm consists of two steps. It first constructs the *visibility cell* decomposition and restricts the planned viewpoints to be on the cell edges. It then samples in the vicinity of each visibility cell edge. We define a local (w.r.t. a visibility cell edge) region called *domain* to quantify this vicinity concept. The shape of the domains is so designed that the behavior (both view and travel) of any watchman route is locally approximated within a constant ratio. In the following, we give details of each step.

3.1 Visibility cell decomposition

First, we give some definitions and observations to clarify the visibility cell decomposition. Our decomposition is a "finer" version than that given in [ZG05], i.e., each cell defined here is completely contained in a single cell defined in [ZG05]. This implies that the properties of the cells defined in [ZG05] are preserved here. Similar terminologies (not by exactly the same names) and results can also be found in [BLM92, GMR97].

Let $VP(X)$ denote the *visibility polygon* of a point $X \in \mathcal{P}$, i.e., the set of points in $\mathcal{P}$ that is visible from X. $VP(X)$ is a *star-shaped* simple polygon, whose edges are either those contained in $\partial\mathcal{P}$ or are *constructed edges* incident on reflex vertices. We call these constructed edges the *windows* of point X. We further extend each window in the direction from X to the incident reflex vertex until it hits the polygon boundary for the last time, and call it the *extended window*. An extended window is a single line segment that may contain parts outside the polygon $\mathcal{P}$. For example, in Fig. 1, the visibility polygon of vertex A_1 consists of a window $\overline{A_5X_1}$, and the corresponding extended window is $\overline{A_5X_3}$. We use $\mathcal{W}(X)$ to denote the set of extended windows of point X. The extended windows of the polygon vertices are of particular interest here. We call them the *critical extended windows*, the set of which is denoted by $\mathcal{CW}(\mathcal{P})$, i.e., $\mathcal{CW}(\mathcal{P}) = \cup_{A_i \in \mathcal{A}} \mathcal{W}(A_i)$. It is easy to see $|\mathcal{CW}(\mathcal{P})| \leq |\mathcal{A}_r||\mathcal{A}| = O(n^2)$.

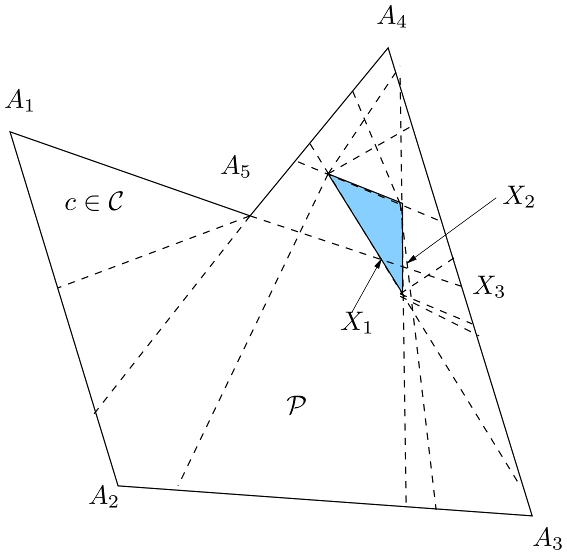


Figure 1: Visibility cell decomposition of polygonal $\mathcal{P}$. The shaded region is a hole.

We use critical *extended* windows to partition the polygon into *visibility cells*. See Fig. 1. It is easy to see this partition is finer than the one used in [ZG05]. We use $\mathcal{C}$ to denote the set of all visibility cells and use $\mathcal{L}$ to denote the set of all visibility cell edges. This *visibility cell decomposition* is efficiently computed by first computing the extended critical windows, computing the arrangement of these windows and the polygon edges, and then excluding part of the arrangement that are outside the polygon. We refer to [O'R98] for efficient arrangement algorithms. And by the *Zone Theorem* [O'R98], the number of visibility cells, $|\mathcal{C}|$, and the number of visibility cell edges, $|\mathcal{L}|$, are bounded by $O(n^4)$.

Our visibility cell decomposition preserves the property shown in [ZG05], and is stated as Lemma 1.

Lemma 1 *All points in the same visibility cell have the same polygon edges entirely visible from them.*

3.2 Sampling visibility cell edge domain

For a visibility cell edge l, as shown in Fig. 2, we draw a diamond shape consisting of two isosceles triangles with l as the common base. The sides of each triangle form an angle of $\alpha < 90$ degrees with the base. We will subsequently show how to determine α in Section 4. We define the domain of the cell edge, denoted by $Dom(l)$, as the set of all points of polygon $\mathcal{P}$ inside the diamond (including the diamond boundary edges). In Fig. 2, $Dom(l)$ is the set of points in the diamond shape excluding the shaded area.

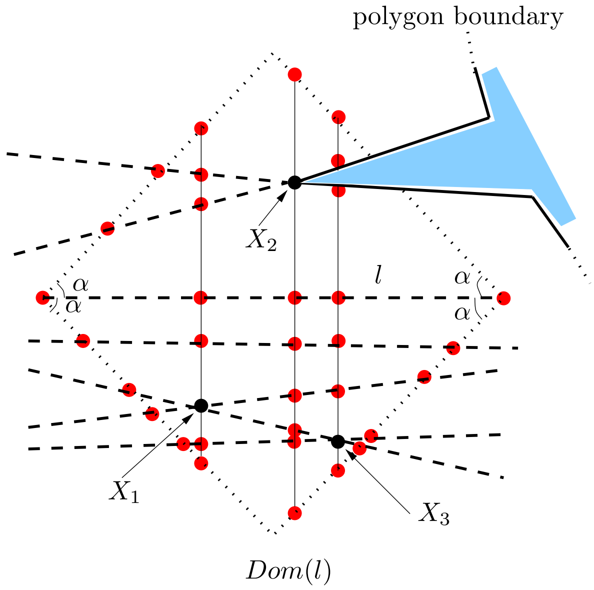


Figure 2: Domain of cell edge l, $Dom(l)$. Inside $Dom(l)$, we use vertical line segments from each visibility cell vertex, the dots X_1, X_2, and X_3. The intersection points between these vertical line segments, other visible cell edges, the polygon boundaries, and the domain boundaries are included in the viewpoint sample set.

Lemma 2 states a simple, but crucial observation:

Lemma 2 *For a visibility cell edge l, the slope (w.r.t. l) of any other visibility cell edge that has intersections with $Dom(l)$ is less than α, i.e.,*

$$\forall l' : l' \cap Dom(l) \neq \emptyset, |\angle l'l| \leq \alpha$$

Proof. Otherwise, the extended critical window collinear with l' intersects l and splits it into two edges. This contradicts that l is a single visibility cell edge. ■

See Fig. 2. Inside each visibility cell edge domain $Dom(l)$, we draw vertical (w.r.t. l) lines from all the vertices of visibility cells. The segments of these vertical lines contained in $Dom(l)$, the other visibility cell edges, the polygon boundaries, and the boundaries of $Dom(l)$ intersect each other. We call these intersection points *sample points* and denote the set of sample points for all domains by Γ. The number of sample points in each domain is the number of vertices in the arrangements of the line segments described above, and is bounded by $(|\mathcal{L}| + |\mathcal{L}| + n + 4)^2 = O(n^8)$, according to Zone Theorem [O'R98]. (The terms in the brackets are the bounds on the number of vertical line segments in each domain (bounded by the number of visibility cell vertices), the number of other visibility cell edges, the number of polygon edges, the number of domain boundaries, respectively.) Thus, Γ is bounded:

$$|\Gamma| \leq |\mathcal{L}| \cdot O(n^8) = O(n^{12}).$$

We construct the complete graph $\mathcal{G}$ on Γ where the edge cost between two sample points is the shortest path distance between them in $\mathcal{P}$. This can be done by constructing first the *visibility graph* of Γ; and then the shortest path graph on the visibility graph. Note that all the reflex vertices are included in Γ. Now we have an *induced Traveling VPP* instance, with the set of viewpoints and traveling graph given as Γ and $\mathcal{G}$ respectively.

4 Sampling Algorithm Analysis

In this section, we show that the cost of the optimal solution to the induced Traveling VPP is at most a constant times that of the optimal solution to WEC-GWRP.

The idea is as follows. Assume we have the optimal solution to the WEC-GWRP, we will find a solution to the induced Traveling VPP, by first partitioning the optimal route into pieces, then replacing each piece with a route passing through sample points while keeping endpoints of the piece fixed, and then moving the endpoints to sample points after the pieces are chained together. The partition process guarantees that the visibility cell edges that each piece passes through are ordered. The slope lemma, Lemma 2, then helps bound the length of the replacing piece w.r.t. that of the original piece on the optimal route.

For the optimal WEC-GWRP solution, let $\mathcal{V}^*$ denote the set of planned viewpoints, and let ϕ^* denote the shortest route connecting $\mathcal{V}^*$. As discussed in Section 1, we can assume all planned viewpoints are on the visibility cell edges. We denote the subset of visibility cell edges where the planned viewpoints are located by $\mathcal{L}^*$, i.e., $\mathcal{L}^* = \{l \in \mathcal{L} : \exists X \in \mathcal{V}^*, X \in L\}$.

4.1 Partition of ϕ^*

In the following, we partition ϕ^* in two steps. First, we partition it according to some visibility cell edge domains. Then, inside each domain, we further partition it using the vertical line segments incident on the cell vertices.

4.1.1 Partition according to domains

To help define the partition, we first introduce a labeling of some (not all) edges in $\mathcal{L}^*$. Note that all the edges in $\mathcal{L}^*$ are naturally ordered by the order in which ϕ^* intersects them. We start with the first cell edge and label it l_1. Skip the following cell edges whose corresponding planned viewpoints ($\mathcal{V}^*$) belong to the currently labeled domain, $Dom(l_1)$. We then label the next cell edge after ϕ^* exits $Dom(l_1)$ by l_2, and continue in this fashion. Please see Fig. 3 for an example.

We partition ϕ^* according to the labeled visibility cell edges. See Fig. 3. Let $\phi_k^*, k = 1, \ldots, K$ denote these parts. Thus, the start- and end-points of ϕ_k^* are S_k and S_{k+1}, respectively. Also, $S_k \in l_k \subset Dom(l_k)$ and $S_{k+1} \notin Dom(l_k)$ by definition.

4.1.2 Partition inside a domain

See Fig. 4. The vertical line segments in a domain partition a domain into vertical *strips.* The strips are bounded by three types of boundaries: the polygon boundary; the *strip boundary* that separates two

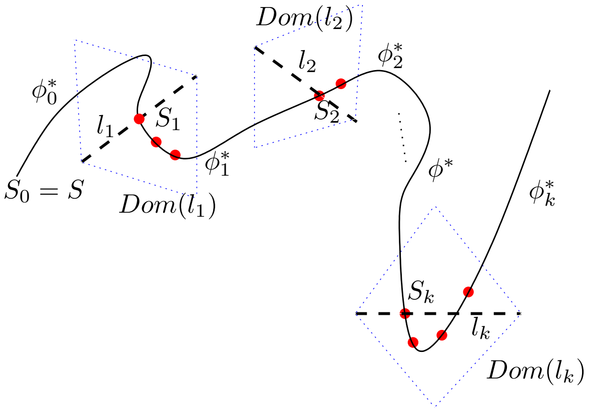


Figure 3: Partition of the optimal WEC-GWRP solution according to the visibility cell edges and corresponding domains it crosses. Note that although shown disjointed, the labeled domains may intersect each other and planned viewpoints on ϕ_k^* may be contained in previous labeled domains.

neighboring strips; and the *domain boundary*, the boundary of the diamond shape defining a domain.

Note that all the visibility cell edges are ordered inside a strip, since otherwise a vertical line at the intersection point where the order changes would have divided the strip into smaller strips. We order the visibility cell edges in a strip according to their intersection point with the left strip or domain boundary. The position relation between cell edges, *above/below*, is thus well defined inside a strip.

We further partition each part of the optimal route, ϕ_k^*, inside the domain $Dom(l_k)$ by its strips. For a strip st, we denote the partitioned piece by $\phi_k^*(st) = \phi_k^* \cap st$. We denote the start- and end-points $\phi_k^*(st)$ by L_{st} and R_{st} respectively, also called the entry and exit points respectively in the following. We denote the highest and lowest visibility cell edges that $\phi_k^*(st)$ crosses (if applicable) by l_{high} and l_{low} respectively.

4.1.3 Categories and properties of $\phi_k^*(st)$

$\phi_k^*(st)$ can be categorized into five cases, according to whether its entry point is on the labeled visibility cell edge or on the strip boundary and whether its exit point is on the strip boundary or on the domain boundary. $\phi_k^*(st)$ cannot enter or exit from the polygon boundary. See Fig. 5 for illustrations. For cases (Ia) and (Ib), $\phi_k^*(st)$ enters on l_k (at point S_k, i.e., $L_{st} = S_k$), and exits either through the strip boundary, Case (Ia), or through the domain boundary, Case (Ib). For cases (IIa), (IIb) and (IIc), $\phi_k^*(st)$ enters st through a strip boundary and exits through either the other strip boundary, Case (IIa), or the same strip boundary, Case (IIb), or the domain boundary, Case (IIb).

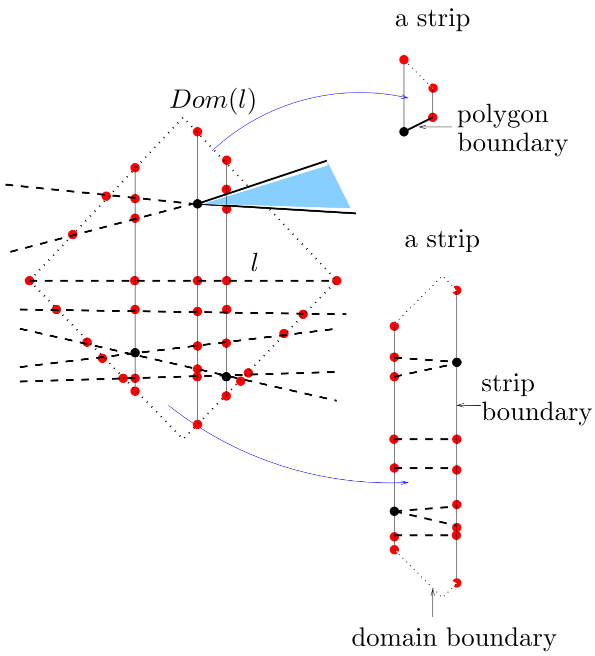


Figure 4: The vertical line segments from visibility cell vertices partition a domain into strips, bounded by domain boundary and/or polygon and strip boundary.

We now give a simple observation that $\phi_k^*(st)$ consists of at most three straight line segments:

Lemma 3 *For the optimal WEC-GWRP solution ϕ^*, $\phi_k^*(st)$, consists of at most three consecutive line segments. If $\phi_k^*(st)$ consists of three segments, the exit point must lie between the highest and lowest visibility cell edges $\phi_k^*(st)$ touches, l_{high} and l_{low}.*

Proof. The proof is based on the following observation. If we can replace $\phi_k^*(st)$ by a strictly shorter route in st with the same entry and exit points, which crosses the same visibility cell edges as $\phi_k^*(st)$, then ϕ^* cannot be optimal. This is because we can replace $\phi_k^*(st)$ by this shorter route and choose viewpoints on the visibility cell edges that are collinear with the viewpoints taken on $\phi_k^*(st)$ and reduce the solution cost.

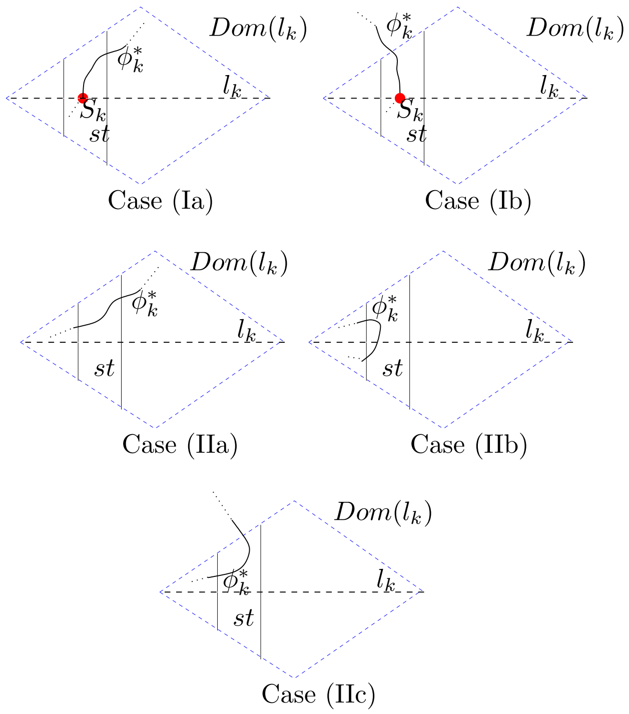


Figure 5: Five cases of $\phi_k^*(st)$. Case (Ia): $\phi_k^*(st)$ starts at S_k and exits from the strip boundary. Case (Ib): $\phi_k^*(st)$ starts at S_k and exits through the domain boundary. Case (IIa): $\phi_k^*(st)$ enters and exits through different strip boundaries. Case (IIb): $\phi_k^*(st)$ enters and exits through the same strip boundaries. Case (IIc): $\phi_k^*(st)$ enters through the strip boundary and exits through the domain boundary.

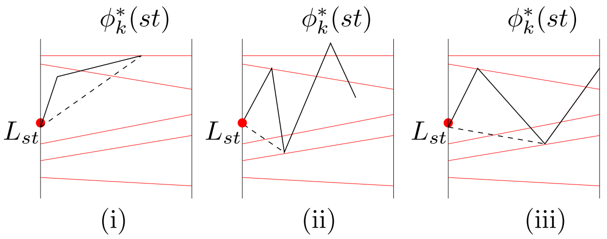


Figure 6: ϕ_{St}^* consists of at most 3 line segments.

Wlog, suppose $\phi_k^*(st)$ goes upwards first after it enters st at point L_{st}. When $\phi_k^*(st)$ changes directions, it has to go downwards. Otherwise, as shown in Fig. 6(i), we can simply replace the two upwards segments (solid line shown) by a single line segments (dashed line shown) and the route is shorter. Since this replacing single line segment visits all the edges the two segments visit, it has exactly the same set of entirely visible polygon edges. Similarly, $\phi_k^*(st)$ cannot change direction when going downwards. We can also easily show that if afterwards $\phi_k^*(st)$ goes upwards again, it has to exit st. Otherwise, as shown in Fig. 6(ii), we can replace it with a shorter route.

See Fig. 6(iii). If $\phi_k^*(st)$ consists of three segments, and it exits the strip by intersecting the highest (lowest) visibility cell edge, we can replace the first (last) two segments by a single segment and reduce solution cost. So the exit point must lie between l_{high} and l_{low}.

Although Fig. 6 shows L_{st} is on the strip boundary, it generalizes trivially to cases where $L_{st} = S_k$. ■

4.2 Approximating $\phi_k^*(st)$ using a route connecting sample points

We show, case by case, how to approximate $\phi_k^*(st)$ using a route connecting sample points in Γ and the start- and end-points of $\phi_k^*(st)$, denoted by $\phi'_k(st)$. For lack of space, we only give the proof for Case (Ia).

4.2.1 Approximation for Case (Ia)

This is the case where smaller α gives us better approximation ratio. See Fig. 7. We replace $\phi_k^*(st)$ by $\phi'_k(st)$, the straight line segment between S_k and R_{st} and two detours on the strip boundary from R_{st}, one going upwards to l_{high} and back to R_{st}, and the other going downwards to l_{low} and back to R_{st}.

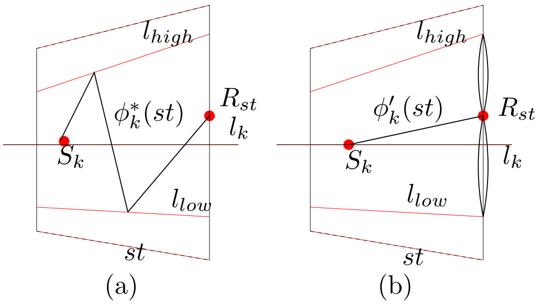


Figure 7: Approximate $\phi_k^*(st)$ for case (Ia). (a) $\phi_k^*(st)$ reaches l_{high} and l_{low} and exits at point R_{st}. (b) $\phi'_k(st)$ goes straight to R_{st} and detours to l_{high} and l_{low} at R_{st}.

We now show that the alternative route can approximate the corresponding part of the optimal WEC-GWRP solution within a constant ratio.

Lemma 4 *For case (Ia), $\phi'_k(st)$ can approximate $\phi^*_k(st)$ within the constant ratio of* $1 + \frac{4}{\cos\alpha}$.

Proof. There are two possible sub-cases: first, $\phi^*_k(st)$ consists of two segments; second, $\phi^*_k(st)$ consists of three segments. One-segment case is true trivially. For lack of space, here we give the proof of the first sub-case.

See Fig. 8. $\phi^*_k(st)$ consists of two segments, a and b; $\phi'_k(st)$ consists of the segment c and a detour consisting of segments d_1 and d_2.

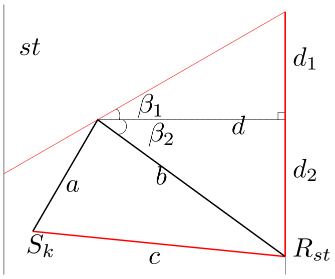


Figure 8: Approximate $\phi^*_k(st)$ for case (Ia) when $\phi^*_k(st)$ consists of two segments.

We have the following relations:

$$\begin{aligned}
&\|c\| \leq \|a\| + \|b\| \\
&\|d_1\| = \|d\| \cdot \tan\beta_1 = \|b\| \cos\beta_2 \tan\beta_1 \\
&\|d_2\| = \|b\| \sin\beta_2
\end{aligned}$$

Since $\beta_1 \leq \alpha < \pi/2$, $\cos\beta_1 \neq 0$, we have,

$$\begin{aligned}
\|d_1\| + \|d_2\| &= \|b\|(\cos\beta_2 \tan\beta_1 + \sin\beta_2) \\
&= \frac{\|b\|}{\cos\beta_1} \sin(\beta_1 + \beta_2) \\
&\leq \frac{\|b\|}{\cos\beta_1} \leq \frac{\|a\| + \|b\|}{\cos\beta_1} \leq \frac{\|a\| + \|b\|}{\cos\alpha}.
\end{aligned}$$

So we have,

$$\begin{aligned}
\|\phi'_k(st)\| &= \|c\| + 2(\|d_1\| + \|d_2\|) \\
&\leq (\|a\| + \|b\|)(1 + \frac{2}{\cos\alpha}) \\
&= \|\phi^*_k(st)\|(1 + \frac{2}{\cos\alpha}),
\end{aligned}$$

i.e., $\frac{\|\phi'_k(st)\|}{\|\phi^*_k(st)\|} \leq (1 + \frac{2}{\cos\alpha})$.

The case for the equality occurs when $\beta_1 = \alpha$ and $\beta_2 = \pi/2 - \alpha$, shown in Fig. 9.

For the second sub-case where $\phi^*_k(st)$ consists of three segments, we can similarly show $\|\phi'_k(st)\|$ is at most $\|\phi^*_k(st)\|$ times $1 + \frac{4}{\cos\alpha}$.

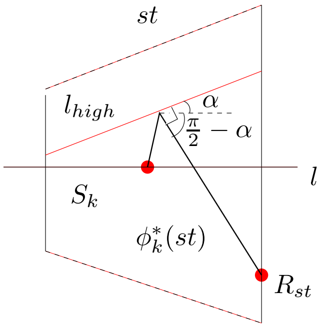


Figure 9: The worst case of the approximation of case (Ia). S_k is arbitrarily close to l_{high}; l_{high} is parallel to the domain boundary, i.e. its slope is α; and the line segment connecting R_{st} is $\frac{\pi}{2} - \alpha$, i.e, it is perpendicular to l_{high}.

To summarize, for both sub-cases where $\phi'_k(st)$ consists of two or three segments respectively, $\|\phi'_k(st)\|$ is at most $1 + \frac{4}{\cos\alpha}$ times $\|\phi^*_k(st)\|$. ■

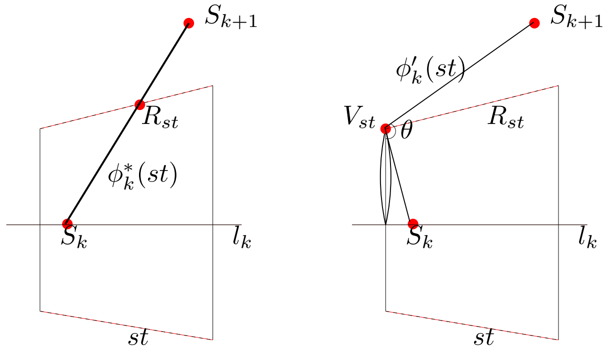
4.2.2 Approximation for Case (Ib)

This is the case where bigger α gives us better approximation ratio. Wlog, here we only consider the case where $\phi^*_k(st)$ exits from the top left domain boundary. First, by Lemma 3, if $\phi^*_k(st)$ has three segments, the exit point must lie between the highest and the lowest visibility cell edges $\phi^*_k(st)$ touches. Since we know the exit point is on the domain boundary, higher than l_{high}, $\phi^*_k(st)$ consists of at most two segments. Here we illustrate only the single segment sub-case.

See Fig. 10. We approximate $\phi^*_k(st)$ using the straight line segment connecting S_k and the top-left strip boundary vertex, V_{st}, and a detour on the strip boundary, if necessary, from V_{st} to reach l_{low} and back to V_{st}. In Lemma 5, we show the approximation ratio using the left domain vertex on strip st. For lack of space, the proof of Lemma 5 is skipped. Intuitively, since the angle θ is at least α, the length of $route(S_k V_{st} S_{k+1})$ is at most a constant times that of $\overline{S_k S_{k+1}}$.

Lemma 5 *For case (Ib), $\phi'_k(st)$ can approximate $\phi^*_k(st)$ within the constant ratio of* $\frac{1}{\sin\frac{\alpha}{2}} + \frac{2}{\cos\alpha}$.

Similarly, for cases (IIa), (IIb), (IIc), we can find a $\phi'_k(st)$ whose length is at most that of $\phi^*_k(st)$ times $1 + \frac{4}{\cos\alpha}$ or $\frac{1}{\sin\frac{\alpha}{2}} + \frac{2}{\cos\alpha}$.



(a) (b)

Figure 10: Approximate $\phi_k^*(st)$ for case (Ib): one-segment sub-case. (a) $\phi_k^*(st)$ consists of a line segment $\overline{S_kS_{k+1}}$. (b) $\phi'_k(st)$ consists of two segments $\overline{S_kV_{st}}$ and $\overline{V_{st}S_{k+1}}$ and a detour from V_{st} to reach l and back.

4.3 Chaining together the partitions

By its construction, $\phi'_k(st)$'s chained together form a continuous route, a sequence of line segments with loops at their endpoints, Fig. 11. However, the start- and end-points of $\phi'_k(st)$ may not belong to the sample point set Γ. We construct a solution $\hat{\phi}$ from ϕ' as follows. For endpoints inside their loops, e.g. point V in Fig. 11, we move them to the nearest sample point *inside* the loops. By the triangular inequality, we loose at most an additional constant (2) approximation ratio. For S_k and endpoints without detours, e.g. point W and S_k in Fig. 11, we can simply ignore it, since it is taken care of by the endpoints of its previous and next strips. For example, in Fig. 11, let U and V denote the planned viewpoints that are immediately before and immediate after S_k on ϕ' respectively. We can bypass S_k and connect directly U and V, (the dotted line segment).

It is clear that $\hat{\phi}$ is a solution to the induced Traveling VPP, i.e., all the planned viewpoints are from the sample point set. Using this solution, we can bound the cost of the optimal solution to the induced Traveling VPP w.r.t. the true optimal to the WEC-GWRP.

Theorem 1 *The cost of the optimal solution to the induced Traveling VPP is at most* 11.657 *times that of the optimal solution to the WEC-GWRP.*

Proof. By summarizing cases (Ia), (Ib), (IIa), (IIb) and (IIc), the length of $\phi'_k(st)$ is at most that of $\phi_k^*(st)$ times $\max(1+\frac{4}{\cos\alpha}, \frac{1}{\sin\frac{\alpha}{2}}+\frac{2}{\cos\alpha})$. After chaining and moving the entry and exit points, the length of $\hat{\phi}$ is at most twice that of $\phi'_k(st)$. Since $\hat{\phi}$ is a feasible solution to the induced Traveling VPP, its cost is a lower bound on the optimal solution to the induced Traveling VPP. Thus, the cost of the optimal solution to the induced Traveling VPP is at most $2\max(1+\frac{4}{\cos\alpha}, \frac{1}{\sin\frac{\alpha}{2}}+\frac{2}{\cos\alpha})$ times the cost of the true optimal solution to the WEC-GWRP.

Now we choose the α value to minimize this ratio, i.e., to solve $\min_{\alpha} 2\max(1+\frac{4}{\cos\alpha}, \frac{1}{\sin\frac{\alpha}{2}}+\frac{2}{\cos\alpha})$. And the solution is approximately 11.657, when $\alpha \approx 34^o$ (via numerical minimization using Matlab). ■

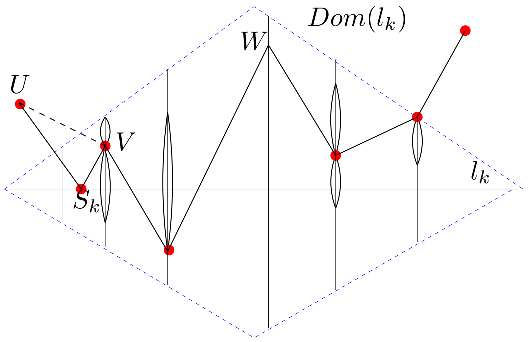


Figure 11: The approximation solution within a domain $Dom(l_k)$, after chaining together $\phi'_k(st)$ for the strips ϕ^* crosses.

After constructing the induced Traveling VPP using the sample points, we call the approximation algorithm in [WKG06] to get a solution. The approximation ratio result of this Traveling VPP solver [WKG06] and Theorem 1 implies that the cost of the solution is at most the cost of the optimal WEC-GWRP solution times either the order of the view frequency or a polynomial of $\log n$, whichever is smaller.

5 Conclusion

In this paper, we introduce the generalized watchman route problem of planning a continuous route and a set of discrete viewpoints on the route such that every polygon edge is entirely visible from at least one planned viewpoint, while minimizing a (weighted) sum of view cost and travel cost. The problem is NP-hard and log-inapproximable. We propose an approximation algorithm that consists of two steps. The first step is a novel sampling method that, for any given WEC-GWRP instance, produces a polynomial number of discrete viewpoints in the polygon. We then restrict the planned viewpoints to be from these sample points and construct a corresponding Traveling VPP instance. The second step uses the approximation algorithm in [WKG06] to solve the constructed Traveling VPP instance. We show that the optimal solution cost to the Traveling VPP instance is at most a constant times the optimal solution cost of the WEC-GWRP instance. As a result, the approximation ratio of our algorithm is in the order of the view frequency or a poly-log function of the input size, whichever is smaller.

We believe that the sampling algorithm proposed here is a general technique and can also be used for other shortest route problems where one would like to get an approximation algorithm by first reducing the infinite input space to a discrete sample point set, and then solving the resulting discrete problem. For example, the algorithm can be applied to a generalized version of the 2.5D terrain guarding problem [Eid02] with additional travel cost in the objective function. We can then first apply the cell decomposition to reduce the input space to a set of line segments, the cell edges, (same decomposition was used in [Eid02]), then use the sampling algorithm proposed in this paper to reduce it to a Traveling VPP instance, and then call the Traveling VPP solver [WKG06]. The resulting algorithm has an approximation ratio of the order of the view frequency or a poly-log function of the input size, whichever is smaller.

References

[BLM92] P. Bose, A. Lubiw, and J. Munro. Efficient visibility queries in simple polygons. In *Proc. of Fourth Canadian Conference on Computational Geometry*, pages 23–28, 1992.

[CN88] W. Chin and S. Ntafos. Optimum watchman routes. *Information Processing Letters*, 28:39–44, 1988.

[CN91] W. Chin and S. Ntafos. Watchman routes in simple polygons. *Discrete and Computational Geometry*, 6(1):9–31, 1991.

[CN99] S. Carlsson and B. Nilsson. Computing vision points in polygons. *Algorithmica*, 24(1):50–75, 1999.

[EHP02] A. Efrat and S. Har-Peled. Guarding galleries and terrains. In *Proc. of the IFIP Seventeenth World Computer Congress - TC1 Stream / 2nd IFIP International Conference on Theoretical Computer Science*, pages 181–192, 2002.

[Eid02] S. Eidenbenz. Approximation algorithms for terrain guarding. *Information Processing Letters*, 82:99–105, 2002.

[GB97a] S. Ghosh and J. Burdick. An on-line algorithm for exploring an known polygonal environment by a point robot. In *Proc. of the Ninth Canadian Conference on Computational Geometry*, pages 100–105, 1997.

[GB97b] S. Ghosh and J. Burdick. Understanding discrete visibility and related approximation algorithms. In *Proc. of the Ninth Canadian Conference on Computational Geometry*, pages 106–111, 1997.

[GB04] S. Ghosh and J. Burdick. Exploring an unknown polygonal environment with discrete visibility. Unpublished Manuscript, Tata Institute of Fundamental Research, Mumbai, India, 2004.

[GBL01] H. Gonzalez-Banos and J. Latombe. A randomized art-gallery algorithm for sensor placement. In *Proc. of Seventeenth ACM Symposium on Computational Geometry*, pages 232 – 240, 2001.

[GKR00] N. Garg, G. Konjevod, and R. Ravi. A polylogorithmic approximation algorithm for the group steiner tree problem. *Journal of Algorithms*, 37:66–84, 2000.

[GMR97] L. Guibas, R. Motwani, and P. Raghavan. The robot localization problem. *SIAM Journal on Computing*, 26(4):1120–1138, 1997.

[LL86] D. Lee and A. Lin. Computational complexity of art gallery problems. *IEEE Transactions on Information Theory*, 32:276–282, 1986.

[O'R98] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, second edition, 1998.

[OS83] J. O'Rourke and K. Supowit. Some NP-hard polygon decomposition problems. *IEEE Transactions on Information Theory*, 29(2):181–190, 1983.

[Pap77] C. Papadimitriou. Euclidean TSP is NP-complete. *Theoretical Computer Science*, 4:237–244, 1977.

[SRR03] W. Scott, G. Roth, and J. Rivest. View planning for automated three-dimensional object reconstruction and inspection. *ACM Computing Surveys*, 35(1):64–96, March 2003.

[Tan04] X. Tan. Approximation algorithm for the watchman route and zookeeper's problems. *Discrete Applied Mathematics*, 136(2-3):363–376, 2004.

[Vaz01] V. Vazirani. *Approximation algorithms*. Spinger, 2001.

[WKG06] P. Wang, R. Krishnamurti, and K. Gupta. View planning problem with combined view and traveling costs: Problem formulation, hardness of approximation, and approximation algorithms. Technical Report TR2006-17, Simon Fraser University, Burnaby, B.C., Canada, May 2006. A version also submitted to IEEE International Conference on Robotics and Automation 2007.

[ZG05] A. Zarei and M. Ghodsi. Efficient computation of query point visibility in polygons with holes. In *Proc. of Twenty-first Annual Symposium on Computational Geometry*, pages 314–320, 2005.