Generalized Watchman Route Problem with Discrete View Cost

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

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 the *start position* is given [5]. Although seemingly very related to two well-known NP-hard problems, namely the *Art Gallery Problem with Point Guards* [13] (*Point AGP*) and the *Euclidean Traveling Salesman Problem* [15] (*Euclidean TSP*), WRP is solvable in polynomial time for simple polygons. It is still NP-hard for polygons with holes [4].

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 a robot-sensor system, each sensing action incurs a large overhead, corresponding to image acquisition, processing, and integration [17]. In addition, often for better sensing qualities, the robot has to stop its movements during image acquisitions. We introduce the problem of generalized watchman route with discrete view cost, or GWRP in short , to relax the continuous sensing assumption of WRP. It refers to planning both a route and a number of discrete viewpoints on it, such that every point on the polygon boundary is visible from at least one planned viewpoint; while the cost 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. 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 [10], 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 [3, 9], the authors consider the problem of choosing a set of discrete viewpoints on a given route, while maintaining the same visible polygon boundary. However, it is not always successful and some routes may need infinite number of viewpoints. Their algorithm stops once the approximate viewpoints are too close to each other.

In this paper, we consider a nontrivial restricted version of the GWRP, called the *Whole Edge Covering GWRP* (WEC-GWRP), in which any polygon edge is required to be entirely visible from at least one planned viewpoint. The restriction arises naturally in robot inspection tasks, where the “map” given 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 [6]. WEC-GWRP has the same NP-hardness and inapproximability as GWRP.

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 [7, 12], the authors considered a local version of the robot exploration problem, “to look around a corner”, i.e., to detect an object hidden behind a corner while minimizing the sum of the robot traveling distance and the sensor scan time. The problem is considerably simpler since the goal is local, i.e., the objective is not to cover all the object surfaces.

In [19], we considered the problem of view planning with combined view and travel cost (Traveling VPP), which, given a number of discrete viewpoints connected via a graph, asks for a subset of the viewpoints connected by a route such that the boundary edges of a given object are all covered, while minimizing a weighted sum of the number of planned viewpoints and the length of the route. We gave an LP-based rounding algorithm *Round and Connect* that chooses the viewpoints greedily according to their LP optimal solution values, and then solves the Steiner tree problem [18] to connect the chosen viewpoints. We showed that the approximation ratio of Round and Connect is in the order of *view frequency*, the maximum number of viewpoints that cover a single boundary edge. We also gave a reduction from Traveling VPP to the Group Steiner Tree problem (GST) instance [8] in polynomial time. By calling the poly-log approximation algorithm for GST [8] after the reduction, we can approximate the optimal solution of Traveling VPP within a poly-log ra-

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

tio. The result is summarized in the following lemma:

Lemma 1 *The optimal solution to Traveling VPP can be approximated within the ratio of either the order of view frequency or a poly-log function.*

To use the approximation algorithm for the Traveling VPP, we propose a novel sampling algorithm that computes a bounded number ($O(n^{12})$, where n is the number of polygon vertices) of discrete viewpoints in the polygon, to reduce the GWRP to Traveling VPP. We emphasize here that the number of computed viewpoints does not depend on any geometric parameter of the polygon as opposed to [16] and [1]. We show that if we restrict the problem to choose planned viewpoints only from these sample viewpoints, the cost of the optimal solution to the problem 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 [19] for a solution. This implies that the cost of the resulting solution is at most the cost of the optimal solution to WEC-GWRP times the smaller of the order of the view frequency and a polynomial of $\log n$.

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. 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 detail the two steps of the sampling algorithm. Last, we discuss potential applications of the proposed sampling algorithm. We refer to the extended abstract version of this paper for the approximation ratio analysis of the proposed algorithm.

2 Problem definition

We now formally state the WEC-GWRP. Let $\mathcal{P}$ denote the given polygon (with or without holes). 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). Point X_1 is *visible* from point X_2, if the closed line segment $\overline{X_1X_2}$ is contained in $\mathcal{P}$ (including $\partial\mathcal{P}$). Edge e is visible from point X, if every point of e is visible from 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:

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

3 Sampling Algorithm

3.1 Visibility cell decomposition

Our decomposition is a "finer" version than that given in [20], i.e., each cell defined here is completely contained in a single cell defined in [20]. This implies that the properties of the cells defined in [20] are preserved here. Similar terminologies (not by exactly the same names) and results can also be found in [2, 11].

The *visibility polygon* of a point $X \in \mathcal{P}$ is the set of points in $\mathcal{P}$ that is visible from X. Its edges are either those contained in $\partial\mathcal{P}$ or the *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 call the extended windows of the polygon vertices the *critical extended windows*, the number of which is $O(n^2)$.

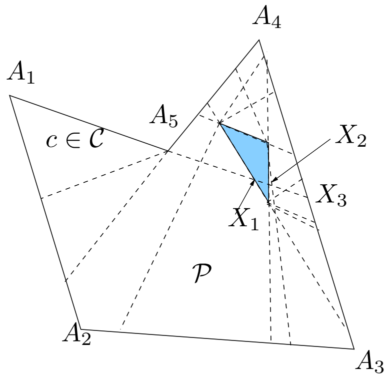

Figure 1: Visibility cell decomposition of polygonal $\mathcal{P}$. The shaded region is a hole.

Critical *extended* windows partition the polygon into *visibility cells*. This process is called *visibility cell decomposition*. See Fig. 1. (The decomposition given in [20] uses *critical windows*.) Let $\mathcal{C}$ denote the set of all visibility cells and $\mathcal{L}$ denote the set of all visibility cell edges. By the *Zone Theorem* [14], 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 following property.

Lemma 2 *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.

See Fig. 2. Inside each visibility cell edge domain $Dom(l)$, we draw orthogonal (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

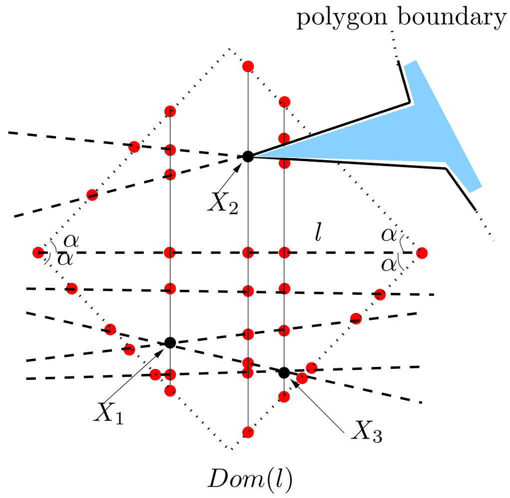

Figure 2: Illustration of the sampling algorithm. See text for details.

$(|\mathcal{L}| + |\mathcal{L}| + n + 4)^2 = O(n^8)$, according to Zone Theorem [14]. (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: $|\Gamma| \le |\mathcal{L}| \cdot O(n^8) = O(n^{12})$.

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 is done by constructing first the *visibility graph* of Γ; and then the shortest path graph on the visibility graph. Now we have an *induced Traveling VPP* instance, with the set of viewpoints and traveling graph being Γ and $\mathcal{G}$ respectively.

4 Sampling Algorithm Analysis

For lack of space, we give the idea of the algorithm analysis, to show that the cost of the optimal solution to the induced Traveling VPP is at most a constant times that of the optimal solution cost to WEC-GWRP. Please see the electronic version of this paper for details.

Assume we have the optimal solution to the WEC-GWRP. We construct 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 is done using *strips* defined for each domain, and guarantees that the visibility cell edges that each piece passes through are ordered. We then bound the length of the replacing piece w.r.t. that of the original piece on the optimal route. To bound the length, we further categorize the ways the optimal route crosses the strips, for example one category corresponds to that the route starts from the visibility cell edge that defines the domain and escapes from the domain boundary. Different categories have different bounds as functions of the parameter α. By optimizing over the value of α (this corresponds to solving a min-max problem), we can have the tightest bound. The main result is summarized in the following theorem:

Theorem 3 *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.*

5 Conclusion

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 [6] 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 [6]), then use the sampling algorithm proposed in this paper to reduce it to a Traveling VPP instance, and then call the Traveling VPP solver. 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

[1] L. Aleksandrov, A. Maheshwari, and J.-R. Sack. Determining approximate shortest paths on weighted polyhedral surfaces. *Journal of the ACM*, 52(1):25–53, January 2005.

[2] P. Bose, A. Lubiw, and J. Munro. Efficient visibility queries in simple polygons. In *Proc. CCCG*, pages 23–28, 1992.

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

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

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

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

[7] S. Fekete, R. Klein, and A. Nuchter. Online searching with an autonomous robot. In *Proc. Workshop on Algorithmic Foundation of Robotics*, pages 350–365, 2004.

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

[9] S. Ghosh and J. Burdick. Understanding discrete visibility and related approximation algorithms. In *Proc. CCCG*, pages 106–111, 1997.

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

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

[12] V. Isler, S. Kannan, and K. Daniilidis. Local exploration: online algorithms and a probabilistic framework. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1913 – 1920, 2003.

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

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

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

[16] C. Papadimitriou. An algorithm for shortest-path motion three dimensions. *Information Processing Letters*, 20:259–263, 1985.

[17] 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.

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

[19] P. Wang, R. Krishnamurti, and K. Gupta. View planning problem with combined view and traveling cost. In *Proc. IEEE ICRA*, pages 711 – 716, 2007.

[20] A. Zarei and M. Ghodsi. Efficient computation of query point visibility in polygons with holes. In *Proc. SOCG*, pages 314–320, 2005.