Approximation Algorithms and Inapproximability Results for Art Gallery Problems

Registration seminar report submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

by

Pritam Bhattacharya
14CS92R07

under the supervision of

Prof. Sudebkumar Prasant Pal
& Prof. Subir Kumar Ghosh

Department of Computer Science and Engineering,
Indian Institute of Technology Kharagpur,
Kharagpur, India

July 2016

1 Introduction and Objective

The art gallery problem is a classic problem in the domain of computational geometry. It originates from the real-world problem of finding the minimum number of guards (or surveillance cameras) required to have an art gallery (represented by a polygon) under complete observation. Discovering good algorithms for solving these problems leads to several practical applications, particularly in the areas of surveillance and robot-motion planning. Unfortunately however, most standard variations of the art gallery problem have been established to be NP-hard, implying that exact solutions for them cannot be computed efficiently. Therefore, for multiple variants of the art gallery problem, my research focuses on designing efficient algorithms that output only an approximate solution, but provide a relative performance guarantee. In other words, the design of these algorithms ensures that the value of the computed solution lies within some (preferably constant) factor of the optimal value. This factor is typically referred to as the *approximation ratio*. While trying to improve the best known approximation ratios till date, we also make parallel efforts to try and establish *inapproximability bounds* on these problems. An inapproximability bound basically provides us with the best possible approximation ratio that we can realistically hope to achieve, by establishing the fact that no efficient algorithm for that problem can exist which has a better approximation ratio, unless P=NP. It is fairly obvious from the discussion above that designing approximation algorithms and proving inapproximability bounds are two endeavours that complement each other quite well. Thus, we also seek to pursue both these research directions in parallel, and apply them in the context of the art gallery problem and its variants.

2 Literature Survey

2.1 The art gallery problem

The art gallery problem enquires about the least number of guards that are sufficient to ensure that an art gallery (represented by a polygon P) is fully guarded, assuming that a guard's field of view covers $360°$ as well as an unbounded distance. This problem was first posed by Victor Klee at a conference in 1973, and over course of time, it has turned into one of the most well-studied problems in computational geometry.

A *polygon* P is defined to be a closed region in the plane bounded by a finite set of line segments, called edges of P, such that, between any two points of P, there exists a path which does not intersect any edge of P. If the boundary of a polygon P consists of two or more cycles, then P is called a *polygon with holes* (see Figure 1). Otherwise, P is called a *simple polygon* or a *polygon without holes* (see Figure 2).

An art gallery can be viewed as an n-sided polygon P (with or without holes) and guards as points inside P. Any point $z \in P$ is said to be *visible* from a guard g if the line segment zg does not intersect the exterior of P (see Figure 1 and Figure 2). In general, guards may be placed anywhere inside P. If the guards are allowed to be placed only on vertices of P, they are called *vertex guards*. If there is no such restriction, guards are called *point guards*. Point and vertex guards together are also referred to as *stationary guards*. If guards are allowed to patrol along a line segment inside P, they are called *mobile guards*. In particular, if they are allowed to patrol only along the edges of P, they are called *edge guards*. [16, 27]

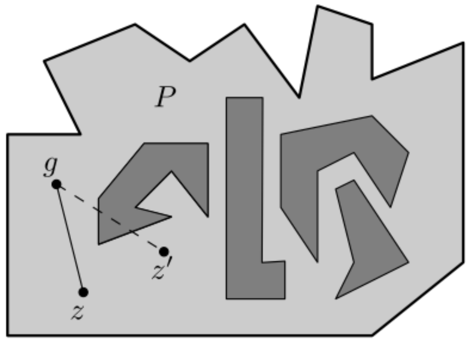

Figure 1: Polygon with holes

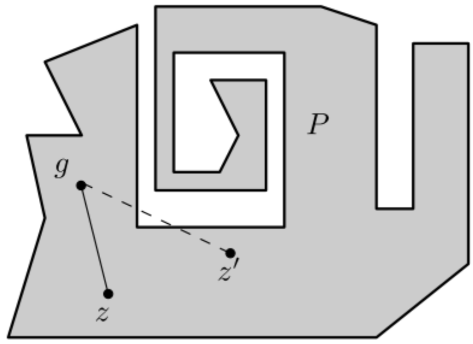

Figure 2: Polygon without holes

In 1975, Chvátal [6] showed that $\lfloor \frac{n}{3} \rfloor$ stationary guards are sufficient and sometimes necessary (see Figure 3) for guarding a simple polygon. In 1978, Fisk [14] presented a simpler and more elegant proof of this result. For a simple orthogonal polygon, whose edges are either horizontal or vertical, Kahn et al. [19] and also O'Rourke [26] showed that $\lfloor \frac{n}{4} \rfloor$ stationary guards are sufficient and sometimes necessary (see Figure 4).

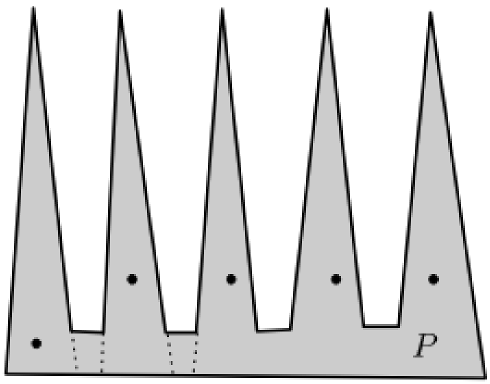

Figure 3: A polygon where $\lfloor \frac{n}{3} \rfloor$ stationary guards are necessary.

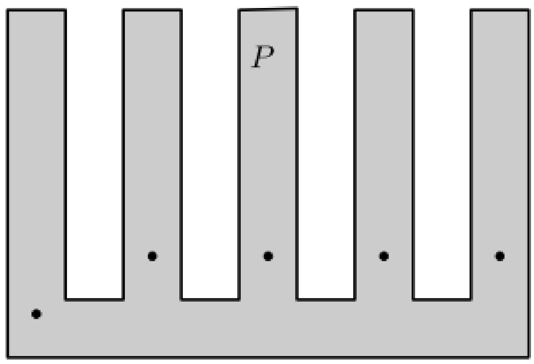

Figure 4: A polygon where $\lfloor \frac{n}{4} \rfloor$ stationary guards are necessary.

2.2 Related hardness and approximation results

The decision version of the art gallery problem is to determine, given a polygon P and a number k as input, whether the polygon P can be guarded with k or fewer guards. The problem was first proved to be NP-complete for polygons with holes by O'Rourke and Supowit [28]. For guarding simple polygons, it was proved to be NP-complete for vertex guards by Lee and Lin [24], and their proof was generalized to work for point guards by Aggarwal [1]. The problem is NP-hard even for simple orthogonal polygons as shown by Katz and Roisman [20] and Schuchardt and Hecker [29]. Each one of these hardness results hold irrespective of whether we are dealing with vertex guards, edge guards, or point guards.

In 1987, Ghosh [15, 17] provided a deterministic $\mathcal{O}(\log n)$-approximation algorithm for the case of vertex and edge guards by discretizing the input polygon P and treating it as an instance of the Set Cover problem. In fact, applying methods for the Set Cover problem developed after Ghosh's algorithm, the approximation ratio of this algorithm becomes $\mathcal{O}(\log OPT)$ for vertex guarding simple polygons and $\mathcal{O}(\log h \log OPT)$ for vertex guarding a polygon with h holes, where OPT denotes the size of the smallest guard set for P. Deshpande et al. [8] obtained an approximation factor of $\mathcal{O}(\log OPT)$ for point guards or perimeter guards by developing a sophisticated discretization method that runs in pseudopolynomial time. Efrat and Har-Peled

[10] provided a randomized algorithm with the same approximation ratio that runs in fully polynomial expected time. For guarding simple polygons using vertex and perimeter guards, King and Kirkpatrick [21] designed a deterministic $\mathcal{O}(\log\log OPT)$-approximation algorithm in 2011.

In 1998, Eidenbenz, Stamm and Widmayer [11, 12] proved that the problem is APX-complete, implying that an approximation ratio better than a fixed constant cannot be achieved unless NP = P. They also proved that if the input polygon is allowed to contain holes, then there cannot exist a polynomial time algorithm for the problem with an approximation ratio better than $((1-\epsilon)/12)\ln n$ for any $\epsilon > 0$, unless NP $\subseteq$ TIME($n^{\mathcal{O}(\log\log n)}$). Extending their method, Bhattacharya, Ghosh and Roy [3] recently proved that, even for the special subclass of polygons with holes that are weakly visible from an edge, there cannot exist a polynomial time algorithm for the problem with an approximation ratio better than $((1-\epsilon)/12)\ln n$ for any $\epsilon > 0$, unless NP = P. These inapproximability results establish that the approximation ratio of $\mathcal{O}(\log n)$ obtained by Ghosh in 1987 is in fact the best possible for the case of polygons with holes. But for simple polygons, the existence of a constant factor approximation algorithm for vertex and edge guards is still possible, as was conjectured by Ghosh [15, 18] in 1987.

Ghosh's conjecture has been proved to be true for two special subclasses of simple polygons, viz. monotone polygons and polygons weakly visible from an edge. In 2012, Krohn and Nilsson [22] presented an approximation algorithm that computes in polynomial time a guard set for a monotone polygon P, such that the size of the guard set is at most $30 \times OPT$. In 2015, Bhattacharya, Ghosh and Roy [3] presented a 6-approximation algorithm that runs in $\mathcal{O}(n^2)$ time for polygons that are weakly visible from an edge.

3 Details of Work Done Till Date

The research work pursued so far has led to the following outcomes till date:

- We obtained a 6-approximation algorithm, which has running time $\mathcal{O}(n^2)$, for vertex guarding polygons that are weakly visible from an edge and contain no holes. This result can be viewed as a step forward towards solving Ghosh's conjecture, since it settles the conjecture for a special class of polygons.
- We proved that the above approximation ratio can be improved to 3 for the special class of polygons without holes that are orthogonal as well as weakly visible from an edge.
- Through a reduction from the Set Cover problem, we proved that, for the special class of polygons containing holes that are weakly visible from an edge, there cannot exist a polynomial time algorithm for the vertex guard problem with an approximation ratio better than $((1-\epsilon)/12)\ln n$ for any $\epsilon > 0$, unless NP = P.
- We proved that the point guard problem for weak visibility polygons is NP-hard by showing a reduction from the decision version of the minimum line cover problem.

Throughout the rest of this section, we discuss each of these results in some detail.

3.1 Preliminary definitions

Let P be a simple polygon. Assume that the vertices of P are labelled $v_1, v_2, \ldots, v_n$ in clockwise order. Let $bd_c(p,q)$ (or $bd_{cc}(p,q)$) denote the clockwise (respectively, counterclockwise) boundary of P from a vertex p to another vertex q. Note that, by definition, $bd_c(p,q) = bd_{cc}(q,p)$. Also, we denote the entire boundary of P by $bd(P)$. So, $bd(P) = bd_c(p,p) = bd_{cc}(p,p)$ for any chosen vertex p belonging to P.

The *visibility polygon* of P from a point z, denoted as $\mathcal{VP}(z)$, is defined to be the set of all points of P that are visible from z. In other words, $\mathcal{VP}(z) = \{q \in P : q \text{ is visible from } z\}$. Observe that the boundary of $\mathcal{VP}(z)$ consists of polygonal edges and non-polygonal edges. We refer to the non-polygonal edges as *constructed edges*. Note that one point of a constructed edge is a vertex (say, v_i) of P, while the other point (say, u_i) lies on $bd(P)$. Moreover, the points z, v_i and u_i are collinear (see Figure 5).

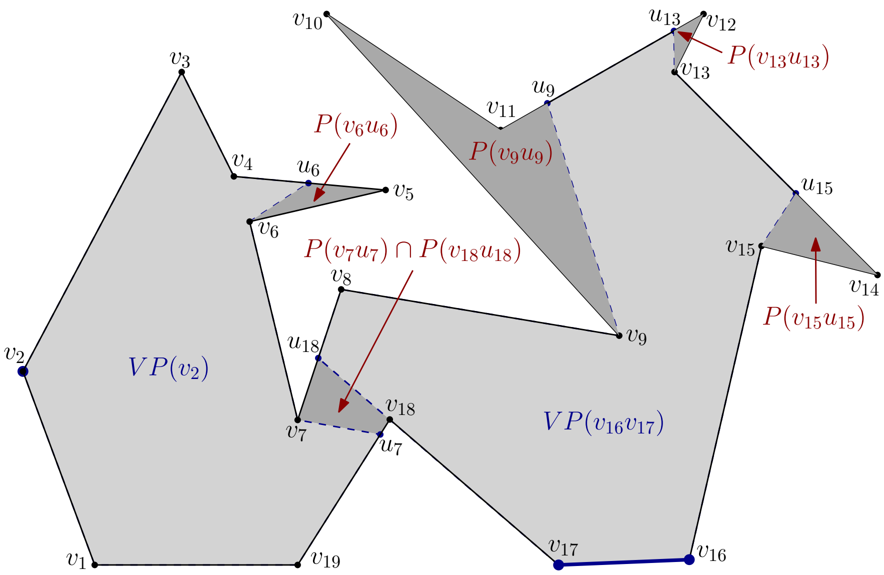

Figure 5: Figure showing visibility polygon $\mathcal{VP}(v_2)$ and weak visibility polygon $\mathcal{VP}(v_{16}v_{17})$, along with several pockets created by constructed edges belonging to both.

Let bc be an internal chord or an edge of P. A point q of P is said to be *weakly visible* from bc if there exists a point $z \in bc$ such that q is visible from z. The set of all such points of P is said to be the *weak visibility polygon* of P from bc, and denoted as $\mathcal{VP}(bc)$. If $\mathcal{VP}(v_iv_{i+1}) = P$ for a polygonal edge v_iv_{i+1}, then P is called a *weakly visible polygon*. Like $\mathcal{VP}(z)$, the boundary of $\mathcal{VP}(bc)$ also consists of polygonal edges and constructed edges v_iu_i (see Figure 5). If v_1 does not belong to $bd_c(v_iu_i)$, then v_iu_i is called a *left constructed edge*. Otherwise, v_iu_i is called a *right constructed edge*.

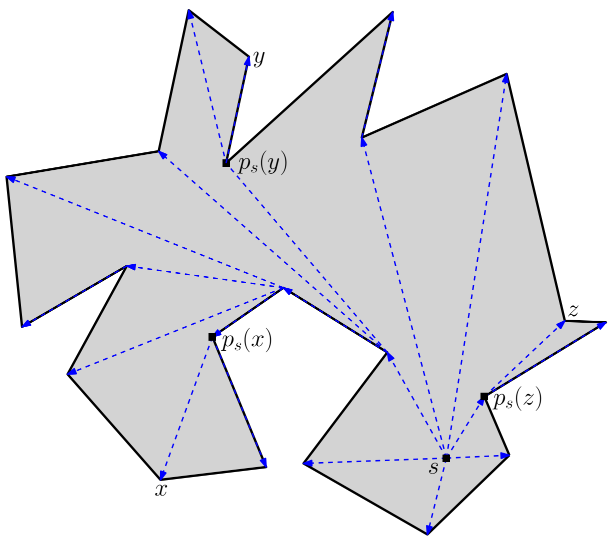

Figure 6: Euclidean shortest path tree rooted at s. The parents of vertices x, y and z in $SPT(s)$ are marked as $p_s(x)$, $p_s(y)$ and $p_s(z)$ respectively.

The *shortest path tree* of P rooted at a vertex s of P, denoted by $SPT(s)$, is the union of Euclidean shortest paths from s to all the vertices of P (see Figure 6. This union of paths is a planar tree, rooted at r, which has n nodes, namely the vertices of P. For every vertex x of P, let $p_u(x)$ and $p_v(x)$ denote the parent of x in $SPT(u)$ and $SPT(v)$ respectively. In the same way, for every interior point y of P, let $p_u(y)$ and $p_v(y)$ denote the vertex of P next to y in the Euclidean shortest path to y from u and v respectively.

3.2 A 6-Approximation Algorithm for Placing Vertex Guards in Weakly Visible Polygons

Let P be a simple polygon which is weakly visible from its edge uv. Suppose a guard is placed on every non-leaf vertex of $SPT(u)$ and $SPT(v)$. It is obvious that these guards see all points of P. However, the number of guards required may be very large compared to the size of an optimal guarding set. In order to reduce the number of guards, placing guards on every non-leaf vertex should be avoided. Let A be a subset of vertices of P. Let S_A denote the set which consists of the parents $p_u(z)$ and $p_v(z)$ of every vertex $z \in A$. Then, A should be chosen such that all vertices of P are visible from guards placed at vertices of S_A. We present a method for choosing A and S_A as follows:-

Algorithm 3.1 An $\mathcal{O}(n^2)$-algorithm for computing a guard set S_A for all vertices of P

Compute $SPT(u)$ and $SPT(v)$
Initialize all the vertices of P as unmarked
Initialize $A \leftarrow \emptyset$, $S_A \leftarrow \emptyset$ and $z \leftarrow u$
while $z \neq v$ **do**
  $z \leftarrow$ the vertex next to z clockwise on $bd_c(u, v)$
  if z is unmarked **then**
    $A \leftarrow A \cup \{z\}$ and $S_A \leftarrow S_A \cup \{p_u(z), p_v(z)\}$
    Place guards on $p_u(z)$ and $p_v(z)$
    Mark all vertices of P visible from $p_u(z)$ or $p_v(z)$
  end if
end while
return the guard set S_A

Henceforth, let S_{opt} denote an optimal set of vertex guards.

Lemma 1. *Any guard $g \in S_{opt}$ that sees vertex z of P must lie on $bd_c(p_u(z), p_v(z))$.*

Lemma 2. *Let z be a vertex of P such that all vertices of $bd_c(p_u(z), p_v(z))$ are visible from $p_u(z)$ or $p_v(z)$. For every vertex x lying on $bd_c(p_u(z), p_v(z))$, if x sees a vertex q of P, then q must also be visible from $p_u(z)$ or $p_v(z)$.*

Lemma 3. *If every vertex $z \in A$ is such that every vertex of $bd_c(p_u(z), p_v(z))$ is visible from $p_u(z)$ or $p_v(z)$, then $|A| \leq |S_{opt}|$.*

Theorem 4. *If every vertex $z \in A$ is such that all vertices of $bd_c(p_u(z), p_v(z))$ are visible from $p_u(z)$ or $p_v(z)$, then $|S_A| \leq 2|S_{opt}|$.*

Proof. We have $|S_A| = 2|A|$. By Lemma 3, $|A| \leq |S_{opt}|$. So, $|S_A| = 2|A| \leq 2|S_{opt}|$. □

However, the above bound does not hold if there exists $z \in A$ such that some vertices of $bd_c(p_u(z), p_v(z))$ are not visible from $p_u(z)$ or $p_v(z)$. Consider Figure 7. For each $i \in \{1, 2, \ldots, k-1\}$, z_{i+1} is not visible from $p_u(z_i)$ or $p_v(z_i)$, which forces Algorithm 3.1 to place guards at $p_u(z_{i+1})$ and $p_v(z_{i+1})$. Therefore, Algorithm 3.1 includes $z_1, z_2, z_3, \ldots, z_k$ in A and ends up placing a total of $2k$ guards at vertices $u, p_{v1}, p_{u2}, p_{v2}, \ldots, p_{uk}, p_{vk}$. However, all vertices of P are visible from just two guards placed at u and g. Hence, $|S_A| = 2k$ whereas $|S_{opt}| = 2$. Since the construction in Figure 7 can be extended for any arbitrary integer k, $|S_A|$ can be arbitrarily large compared to $|S_{opt}|$. So we will now present a new algorithm which gives us a 4-approximation.

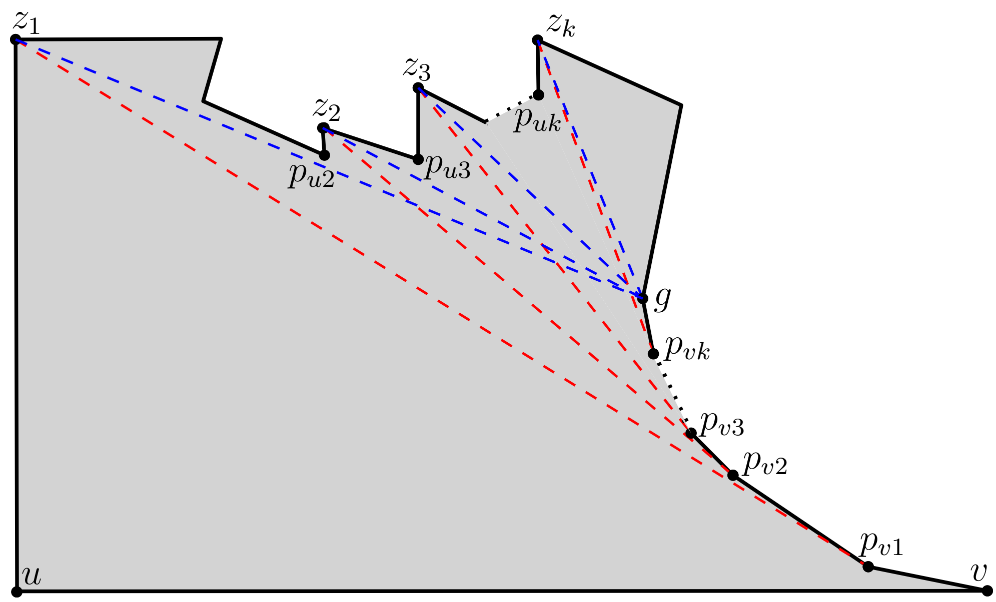

Figure 7: Instance where guard set S_A computed by Algorithm 3.1 is arbitrarily large compared to S_{opt}.

In the new algorithm, described in pseudocode as Algorithm 3.2, $bd_c(u, v)$ is scanned to identify a set of unmarked vertices, denoted as B, such that all vertices of P are visible from guards in $S_B = \{p_u(z)|z \in B\} \cup \{p_v(z)|z \in B\}$. However, unlike the previous algorithm (see Algorithm 3.1), the new algorithm does not blindly include in B every next unmarked vertex that it encounters during the scan. During the scan, if z denotes the current unmarked vertex under consideration, then it may either choose to include z in B or skip ahead to the next unmarked vertex along the scan depending on certain properties of z. At the end of each iteration of the outer while-loop (running from line 4 to line 23) maintains the invariant that, for every unmarked vertex y of $bd_c(u, z)$ (excluding z), $p_u(y)$ and $p_v(y)$ see all unmarked vertices of $bd_c(p_u(y), y)$. Let z' denote the next unmarked vertex of $bd_c(z, p_v(z))$ in clockwise order from z such that z' is not visible from either $p_u(z)$ or $p_v(z)$. Note that, depending on the current vertex z, z' may or may not exist. However, one of the following four mutually exclusive scenarios must be true.

- (A) Every vertex of $bd_c(z, p_v(z))$ is already marked due to guards currently included in S_B.
- (B) Every unmarked vertex of $bd_c(z, p_v(z))$ is visible from $p_u(z)$ or $p_v(z)$.
- (C) Not every unmarked vertex of $bd_c(p_u(z'), z')$ is visible from $p_u(z')$ or $p_v(z')$.
- (D) Every unmarked vertex of $bd_c(p_u(z'), z')$ is visible from $p_u(z')$ or $p_v(z')$.

If z satisfies property (A) or (B), then z is included in B and the first unmarked vertex of $bd_c(p_v(z), v)$ in clockwise order from $p_v(z)$ becomes the new z (see lines 6 to 9). If z satisfies property (C), then z is included in B and z' becomes the new z. If z satisfies property (D), then z' becomes the new z (see lines 11 to 15). Whenever z is included in B, $p_u(z)$ and $p_v(z)$ are included in S_B and all unmarked vertices that become visible from $p_u(z)$ or $p_v(z)$ are marked. After doing so, if there remain unmarked vertices on $bd_{cc}(z, u)$, then $bd_{cc}(z, u)$ is scanned from z in counterclockwise order and more guards are included in S_B according to the following strategy (see lines 18 to 23).

(i) $y \leftarrow z$
(ii) Scan $bd_{cc}(p_u(y), u)$ from y in counterclockwise till an unmarked vertex x is located.
(iii) $y \leftarrow x$
(iv) Add y to B. Add $p_u(y)$ and $p_v(y)$ to S_B.
(v) Mark every vertex visible from $p_u(y)$ or $p_v(y)$.
(vi) Repeat steps (ii)-(v) until all vertices of $bd_c(u, z, u)$ are marked.

Initially, z is chosen to be u itself (see line 3). Then, for each z under consideration along the clockwise scan of $bd_c(u, v)$, the appropriate action is performed corresponding to the property of z. Then, z is updated and the process is repeated till v is reached. The set of vertices S_B is returned by the algorithm (see line 26) as a guard set.

Algorithm 3.2 An $\mathcal{O}(n^2)$-algorithm for computing a guard set S for all vertices of P

1: Compute $SPT(u)$ and $SPT(v)$
2: Initialize all the vertices of P as unmarked
3: Initialize $B \leftarrow \emptyset$, $S_B \leftarrow \emptyset$ and $z \leftarrow u$
4: **while** there exists an unmarked vertex in P **do**
5: $z \leftarrow$ the first unmarked vertex on $bd_c(u, v)$ from z
6: **if** every unmarked vertex of $bd_c(z, p_v(z))$ is visible from $p_u(z)$ or $p_v(z)$ **then**
7: $B \leftarrow B \cup \{z\}$ & $S_B \leftarrow S_B \cup \{p_u(z), p_v(z)\}$
8: Mark all vertices visible from $p_u(z)$ or $p_v(z)$
9: $z \leftarrow p_v(z)$
10: **else**
11: $z' \leftarrow$ the first unmarked vertex on $bd_c(z, v)$
12: **while** every unmarked vertex of $bd_c(p_u(z'), z')$ is visible from $p_u(z')$ or $p_v(z')$ **do**
13: $z \leftarrow z'$
14: $z' \leftarrow$ the first unmarked vertex on $bd_c(z', v)$
15: **end while**
16: $B \leftarrow B \cup \{z\}$ & $S_B \leftarrow S_B \cup \{p_u(z), p_v(z)\}$
17: Mark all vertices visible from $p_u(z)$ or $p_v(z)$
18: $y \leftarrow z$
19: **while** $\exists$ an unmarked vertex on $bd_c(u, z)$ **do**
20: $y \leftarrow$ first unmarked vertex on $bd_{cc}(p_u(y), u)$
21: $B \leftarrow B \cup \{y\}$ & $S_B \leftarrow S_B \cup \{p_u(y), p_v(y)\}$
22: Mark all vertices visible from $p_u(y)$ or $p_v(y)$
23: **end while**
24: **end if**
25: **end while**
26: **return** the guard set S_B

By a rigorous analysis of Algorithm 3.2, the following lemma can be shown to be true.

Lemma 5. $|B| \leq 2|S_{opt}|$.

Theorem 6. $|S_B| \leq 4|S_{opt}|$.

Proof. We have $|S_B| = 2|B|$. Also, by Lemma 5, $|B| \leq 2|S_{opt}|$. So, $|S_B| = 2|B| \leq 4|S_{opt}|$. □

While the guard set S_B is guaranteed to see all vertices of P, it may not always be true that all interior points of P are also visible from guards in S_B. Consider the scenario shown in Figure 8. While scanning $bd_c(u, v)$, Algorithm 3.2 places guards at $p_u(z)$ and $p_v(z)$ as all vertices of $bd_c(p_u(z), p_v(z))$ become visible from $p_u(z)$ or $p_v(z)$. Observe that in fact all vertices of P

become visible from these two guards. But, $VP(p_u(z))$ has several left pockets and $VP(p_v(z))$ has several right pockets which intersect pairwise to create multiple invisible cells. In order to guard these invisible cells, a set S' of additional guards need to be placed.

Theorem 7. *There exists an algorithm with running time $\mathcal{O}(n^2)$ that returns a guard set S' for guarding all interior points of P invisible from guards in S_B such that $|S'| \leq 2|S_{opt}|$.*

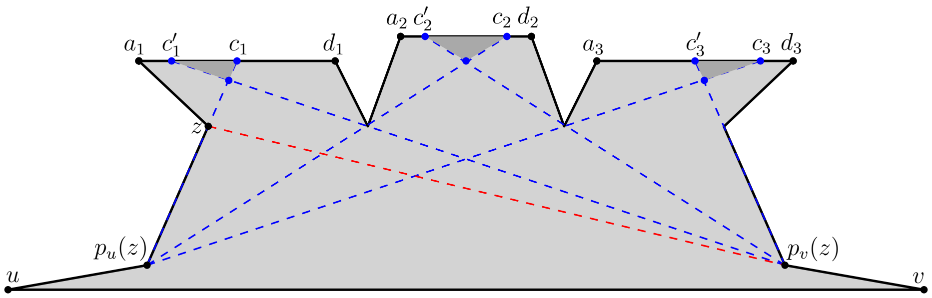

Figure 8: Multiple invisible cells exist within the polygon that are not visible from the guards placed at $p_u(z)$ and $p_v(z)$.

Theorem 8. *There exists an algorithm with running time $\mathcal{O}(n^2)$ that returns a guard set S for guarding all interior points of P such that $|S| \leq 6|S_{opt}|$.*

3.3 A 3-Approximation Algorithm for Placing Vertex Guards in Orthogonal Weak Visibility Polygons

The class of orthogonal polygons weakly visible from an edge has been previously studied by Carlsson, Nilsson and Ntafos [5] under the name of Manhattan skyline or histogram polygons, and they showed that there exists a linear time greedy algorithm to optimally guard these polygons with point guards. Let us also consider a polygon P belonging to this class, i.e. P is an orthogonal polygon weakly visible from an edge uv. In this section, we present an algorithm for vertex guarding P with an approximation factor of 3, which is a clear improvement over the factor 6 which we obtained for the more general class of weak visibility polygons.

First, we present an algorithm for computing a guard set S_A covering only the vertices of P, described below in pseudocode as Algorithm 3.3.

Algorithm 3.3 An $\mathcal{O}(n^2)$-algorithm for computing a guard set S_A for all vertices of P

Compute $SPT(u)$ and $SPT(v)$
Initialize all the vertices of P as unmarked
Initialize $A \leftarrow \emptyset$ and $S_A \leftarrow \emptyset$
while there exist unmarked vertices in P **do**
    $z \leftarrow u$
    while $z \neq v$ **do**
        $z \leftarrow$ the vertex next to z in clockwise order on $bd_c(u, v)$
        if z is unmarked and $bd_c(p_u(z), p_v(z))$ are visible from $p_u(z)$ or $p_v(z)$ **then**
            $A \leftarrow A \cup \{z\}$ and $S_A \leftarrow S_A \cup \{p_u(z), p_v(z)\}$
            Place guards on $p_u(z)$ and $p_v(z)$
            Mark all vertices of P that become visible from $p_u(z)$ or $p_v(z)$
        end if
    end while
end while
return the guard set S_A

Lemma 9. *Let S_{opt} be an optimal guard set. Then, $|S_A| \leq 2|S_{opt}|$.*

All interior points of P are not guaranteed to be visible from guards in the set S_A computed by Algorithm 3.3. Consider the polygon shown in Figure ??. While scanning $bd_c(u, v)$, our algorithm places guards at $p_u(z)$ and $p_v(z)$ as all vertices of $bd_c(p_u(z), p_v(z))$ become visible from $p_u(z)$ or $p_v(z)$. Observe that in fact all vertices of P become visible from these two guards. However, the triangular region $P \setminus (VP(p_u(z)) \cup VP(p_v(z)))$, bounded by the segments x_1x_2, x_2x_3 and x_3x_1, is not visible from $p_u(z)$ or $p_v(z)$. Also, one of the sides x_1x_2 of the triangle $x_1x_2x_3$ is a part of a polygonal edge. In fact, for any such region invisible from guards in S_A, one of the sides must always be a part of a polygonal edge. As mentioned previously in Section 3.2, any such region invisible from guards in S is referred to as an *invisible cell*, and the polygonal edge which contributes as a side to the invisible cell is referred to as its corresponding *partially invisible edge*. Also, we define lid points and lid vertices as before. Next, we present an algorithm for computing an additional set of guards S'_A whose placement ensures that all interior points of P are also guarded.

Algorithm 3.4 An $\mathcal{O}(n^2)$-algorithm for computing a guard set $S_A \cup S'_A$ for guarding P entirely

```
Compute SPT(u) and SPT(v)
Compute the set of guards S_A using Algorithm 3.3.
Initialize C ← ∅, S'_A ← ∅ and z ← u
while there exists an edge in P that is partially visible from guards in S_A ∪ S'_A do
    z' ← the vertex next to z in clockwise order on on bd_c(u, v)
    if if the edge zz' is partially visible from guards in S ∪ S'_A then
        c_i ← the lid point of the left pocket on zz'
        C ← C ∪ {c_i} and S'_A ← S'_A ∪ {p_u(c_i)}
    end if
    z ← z'
end while
return the guard set S_A ∪ S'_A
```

Theorem 10. *The running time of Algorithm 3.4 is $\mathcal{O}(n^2)$.*

Lemma 11. $|C| = |S'_A| \leq |S_{opt}|$.

Theorem 12. $|S_A \cup S'_A| \leq 3|S_{opt}|$.

Proof. By Lemma 9 and Lemma 11, $|S_A \cup S'_A| \leq |S_A| + |S'_A| \leq 2|S_{opt}| + |S_{opt}| \leq 3|S_{opt}|$. □

Therefore, Algorithm 3.4 is a 3-approximation algorithm for solving the problem of guarding orthogonal polygons that are weakly visible from an edge with minimum number of vertex guards.

3.4 An Inapproximability Result

As mentioned in Section 2.2 already, Eidenbenz, Stamm and Widmayer [11, 12] proved that, for polygons with holes, there cannot exist a polynomial time algorithm for the art gallery problem with an approximation ratio better than $((1 - \epsilon)/12) \ln n$ for any $\epsilon > 0$, unless NP $\subseteq$ TIME($n^{\mathcal{O}(\log \log n)}$). Using a modification of the same technique, and taking into consideration a very recent result by Dinur and Steurer [9] that allows us to show that approximating Set Cover to within a factor of $(1 - \epsilon) \ln n$ is NP-hard for every $\epsilon > 0$ (thus strengthening Feige's quasi-NP-hardness [13]), we obtained the following theorem.

Theorem 13. *For weak visibility polygons with holes, there cannot exist a polynomial time algorithm for the Vertex Guard problem with an approximation ratio better than $((1-\epsilon)/12) \ln n$ for every $\epsilon > 0$, unless NP = P.*

3.5 NP-Hardness for Point Guarding Polygons Weakly Visible from an Edge

We prove that the Point Guard problem in polygons weakly visible from an edge is NP-hard by showing a reduction from the decision version of the minimum line cover problem (MLCP), which is defined as follows. Let $\mathcal{L} = \{l_1, \ldots, l_n\}$ be a set of n lines in the plane. Find a set P of points, such that for each line $l \in \mathcal{L}$ there is a point in P that lies on l, and P is as small as possible. Let DLCP denote the corresponding decision problem, that is, given $\mathcal{L}$ and an integer $k > 0$, decide whether there exists a line cover of size k. DLCP is known to be NP-hard [25]. Moreover, MLCP was shown to be APX-hard [4, 23].

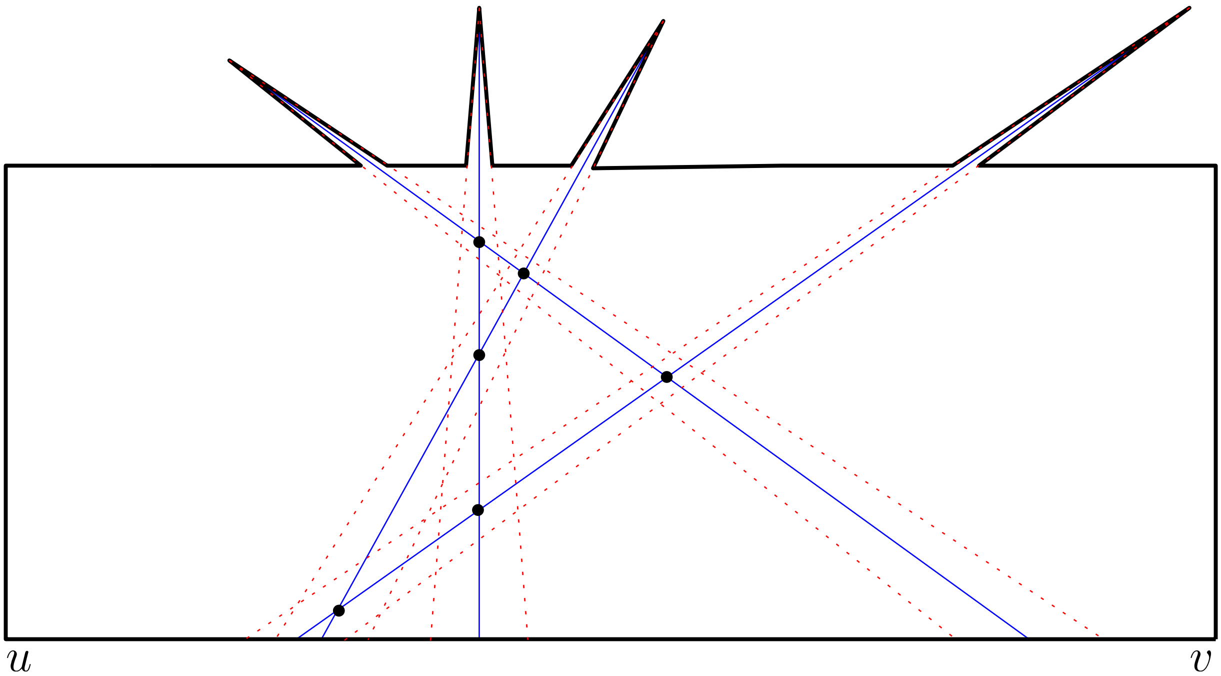

Figure 9: NP-hardness reduction from DLCP for point guarding polygons weakly visible from an edge

The reduction (see Figure 9) has the following steps. First, an axis-parallel rectangle R is drawn on the plane such that it contains all points of pairwise intersection of lines in $\mathcal{L}$. For each line $l \in \mathcal{L}$, consider the closed segment l' that lies within this rectangle. Then, for each such segment l', the end-point with the higher y co-ordinate is extended beyond the boundaries of R and a very narrow spike is added to the boundary of R at this point. Note that, under this construction, the lower horizontal edge uv of R does not have any spikes added to it. In fact, the bounding rectangle along with the added spikes gives a polygon P which is weakly visible from the edge uv. Let the tip of each spike be henceforth referred to as a *distinguished point*. By making the spikes narrow enough, if it is ensured that the visibility polygons of no three distinguished points intersect, then the weak visibility polygon P can be guarded using k point guards if and only if the set of lines $\mathcal{L}$ has a cover of size k. One obvious way to achieve this correspondence is to restrict the placement of potential point guards to only the points of pairwise intersection of lines in $\mathcal{L}$. However, observe that instead of being placed exactly at the point of intersection of two lines $l_i, l_j \in \mathcal{L}$, a point guard can be placed (without losing any visibility) at any point within the intersection region of the visibility polygons of the distinguished points corresponding to the spikes generated by extending l'_i and l'_j.

Theorem 14. *The Point Guard problem is NP-hard for polygons weakly visible from an edge.*

4 Roadmap for Future Work

Our current focus is on designing a constant factor approximation algorithm for vertex guarding all simple polygons. If we succeed in our endeavour, then we will end up proving that the decades-old conjecture of Ghosh is in fact true. Through the amalgamation of several new ideas over the recent past, we have already designed a new algorithm for vertex guarding polygons, which we strongly believe to be having a constant approximation factor. However, we are yet to prove it conclusively, and working out the proof is our primary goal at this point.

Once we manage to obtain the proof for the approximation ratio of our algorithm, we plan to implement it using the CGAL [30] library in C++, and then perform extensive benchmark testing using our implementation. This should help us accumulate practical evidence regarding how closely it approximates the minimum number of vertex guards required to guard a simple polygon. In fact, the theoretically established bound may not be too tight, and we are hopeful that running our implementation against known benchmarks will demonstrate that our algorithm actually approximates the optimal guard sets much more closely in practice than what is suggested by the theoretically proven approximation ratio. In order to compute the exact optimal (or at least a lower bound on its value), so that we can then compare it with the solution computed by our algorithm, we intend to use ideas similar to the practical iterative algorithm by Couto et al. [7] for obtaining exact solutions to the vertex guard problem.

We also plan to investigate the problem of vertex guarding in a setting where the guards are allowed to see points within the polygon directly as well as via a single diffuse reflection along one of the edges, which act as mirrors [2]. Moreover, we feel it may be interesting to explore natural variations of the problem where certain restrictions are imposed on the guard sets themselves. For example, a guard set may be considered to be valid only when it is say a hidden set, or perhaps a clique in the visibility graph of the polygon. In all these parallel threads of exploration, our objective would be to come up with an approximation algorithm with a reasonable approximation ratio, and to show the optimality of our algorithm by establishing corresponding inapproximability bounds.

Disseminations

1. Pritam Bhattacharya, Subir Kumar Ghosh, Bodhayan Roy
   Vertex Guarding in Weak Visibility Polygons
   In Proceedings of the 1st International Conference on Algorithms and Discrete Applied Mathematics, CALDAM 2015, LNCS 8959, pages 45–57, Springer, 2015
   (Winner of best student paper award at CALDAM 2015)
2. Pritam Bhattacharya, Subir Kumar Ghosh, Bodhayan Roy
   Approximability of Guarding Weak Visibility Polygons
   To appear in CALDAM 2015 Special Issue of Discrete Applied Mathematics

References

[1] Alok Aggarwal. *The art gallery theorem: its variations, applications and algorithmic aspects.* PhD thesis, The Johns Hopkins University, Baltimore, Maryland, 1984.

[2] Boris Aronov, Alan R Davis, Tamal K Dey, Sudebkumar Prasant Pal, and D Chithra Prasad. Visibility with Reflection. *Symposium on Computational Geometry*, pages 316–325, 1995.

[3] Pritam Bhattacharya, Subir Kumar Ghosh, and Bodhayan Roy. Vertex Guarding in Weak Visibility Polygons. In *Proceedings of the 1st International Conference on Algorithms and Discrete Applied Mathematics (CALDAM 2015)*, volume 8959 of *Lecture Notes in Computer Science (LNCS)*, pages 45–57. Springer, 2015.

[4] Björn Brodén, Mikael Hammar, and Bengt J. Nilsson. Guarding lines and 2-link polygons is APX-hard. In *Proceedings of the 13th Canadian Conference on Computational Geometry, University of Waterloo, Ontario, Canada, August 13-15, 2001*, pages 45–48, 2001.

[5] Svante Carlsson, Bengt J. Nilsson, and Simeon C. Ntafos. Optimum guard covers and m-watchmen routes for restricted polygons. *International Journal of Computational Geometry & Applications*, 3(1):85–105, 1993.

[6] Václav Chvátal. A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory, Series B*, 18(1):39–41, 1975.

[7] Marcelo C. Couto, Pedro J. de Rezende, and Cid C. de Souza. An exact algorithm for minimizing vertex guards on art galleries. *ITOR*, 18(4):425–448, 2011.

[8] Ajay Deshpande, Taejung Kim, Erik D. Demaine, and Sanjay E. Sarma. A pseudopolynomial time $o(\log n)$-approximation algorithm for art gallery problems. In *WADS*, pages 163–174, 2007.

[9] Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 624–633. ACM, 2014.

[10] Alon Efrat and Sariel Har-Peled. Guarding galleries and terrains. *Information Processing Letters*, 100(6):238–245, 2006.

[11] Stephan Eidenbenz, Christoph Stamm, and Peter Widmayer. Inapproximability of some art gallery problems. *Canadian Conference on Computational Geometry*, pages 1–11, 1998.

[12] Stephan Eidenbenz, Christoph Stamm, and Peter Widmayer. Inapproximability results for guarding polygons and terrains. *Algorithmica*, 31(1):79–113, 2001.

[13] Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

[14] Steve Fisk. A short proof of Chvátal's watchman theorem. *Journal of Combinatorial Theory, Series B*, 24(3):374, 1978.

[15] Subir Kumar Ghosh. Approximation algorithms for art gallery problems. In *Proceedings of Canadian Information Processing Society Congress*, page 429–434. Canadian Information Processing Society, 1987.

[16] Subir Kumar Ghosh. *Visibility Algorithms in the Plane*. Cambridge University Press, 2007.

[17] Subir Kumar Ghosh. Approximation algorithms for art gallery problems in polygons. *Discrete Applied Mathematics*, 158(6):718–722, 2010.

[18] Subir Kumar Ghosh. Approximation algorithms for art gallery problems in polygons and terrains. *WALCOM: Algorithms and Computation*, pages 21–34, 2010.

[19] J. Kahn, M. Klawe, and D. Kleitman. Traditional galleries require fewer watchmen. *SIAM Journal of Algebraic and Discrete Methods*, 4(2):194–206, 1983.

[20] Matthew J. Katz and Gabriel S. Roisman. On guarding the vertices of rectilinear domains. *Computational Geometry*, 39(3):219–228, 2008.

[21] James King and David G. Kirkpatrick. Improved approximation for guarding simple galleries from the perimeter. *Discrete & Computational Geometry*, 46(2):252–269, 2011.

[22] Erik A Krohn and Bengt J Nilsson. Approximate Guarding of Monotone and Rectilinear Polygons. *Algorithmica*, 66:564–594, 2013.

[23] V. S. Anil Kumar, Sunil Arya, and H. Ramesh. Hardness of set cover with intersection 1. In *Automata, Languages and Programming, 27th International Colloquium, ICALP 2000, Geneva, Switzerland, July 9-15, 2000, Proceedings*, pages 624–635, 2000.

[24] D.T. Lee and A. Lin. Computational complexity of art gallery problems. *IEEE Transactions on Information Theory*, 32(2):276–282, 1986.

[25] Nimrod Megiddo and Arie Tamir. On the complexity of locating linear facilities in the plane. *Oper. Res. Lett.*, 1(5):194–197, November 1982.

[26] Joseph O'Rourke. An alternative proof of the rectilinear art gallery theorem. *Journal of Geometry*, 211:118–130, 1983.

[27] Joseph O'Rourke. *Art gallery theorems and algorithms.* Oxford University Press, London, 1987.

[28] Joseph O'Rourke and Kenneth J. Supowit. Some NP-hard polygon decomposition problems. *IEEE Transactions on Information Theory*, 29(2):181–189, 1983.

[29] Dietmar Schuchardt and Hans-Dietrich Hecker. Two NP-Hard Art-Gallery Problems for Ortho-Polygons. *Mathematical Logic Quarterly*, 41:261–267, 1995.

[30] The CGAL Project. *CGAL User and Reference Manual.* CGAL Editorial Board, 4.8.1 edition, 2016.