Constant Approximation Algorithms for Guarding Simple Polygons using Vertex Guards

PRITAM BHATTACHARYA
lord.pritomose@gmail.com
Department of Computer Science and Engineering,
Indian Institute of Technology Kharagpur,
Kharagpur, West Bengal - 721302, India.

***Abstract*—The art gallery problem enquires about the least number of guards sufficient to ensure that an art gallery, represented by a simple polygon P, is fully guarded. Most standard versions of this problem are known to be NP-hard. In 1987, Ghosh provided a deterministic $\mathcal{O}(\log n)$-approximation algorithm for the case of vertex guards and edge guards in simple polygons. In the same paper, Ghosh also conjectured the existence of constant factor approximation algorithms for these problems. We present (https://arxiv.org/abs/1712.05492) three polynomial-time algorithms with a constant approximation ratio for guarding an n-sided simple polygon P using vertex guards.**
(i) The first algorithm, that has an approximation ratio of 18, guards all vertices of P in $\mathcal{O}(n^4)$ time.
(ii) The second algorithm, that has the same approximation ratio of 18, guards the entire boundary of P in $\mathcal{O}(n^5)$ time.
(iii) The third algorithm, that has an approximation ratio of 27, guards all interior and boundary points of P in $\mathcal{O}(n^5)$ time. Further, these algorithms can be modified to obtain similar approximation ratios while using edge guards.

The significance of our results lies in the fact that these results settle the conjecture by Ghosh regarding the existence of constant-factor approximation algorithms for this problem, which has been open since 1987 despite several attempts by researchers. Our approximation algorithms exploit several deep visibility structures of simple polygons which are interesting in their own right.

***Index Terms*—Art gallery problem, approximation algorithm, simple polygons, visibility, vertex guards, weakly visible, shortest path, minimum link path**

I. Introduction

The art gallery problem enquires about the least number of guards sufficient to ensure that an art gallery (represented by a polygon P) is fully guarded, assuming that a guards field of view covers 360° as well as unbounded distance. This problem was first posed by Victor Klee in a conference in 1973, and has become a well investigated problem 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. 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, then they are called *point guards*. The point guards that are constrained to lie on the boundary of P, but not necessarily at the vertices, are referred to as *perimeter guards*. Point, vertex and perimeter 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*. If they are allowed to patrol only along the edges of P, they are called *edge guards* [1], [2].

In 1975, Chvtal [3] 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 [4] 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. [5] and also ORourke [6] showed that $\lfloor \frac{n}{4} \rfloor$ stationary guards are sufficient and sometimes necessary (see Figure 4).

A. 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 point guards. This problem was first shown to be NP-hard for polygons with holes by ORourke and Supowit [7]. This problem was also shown to be NP-hard for simple polygons for guarding using only vertex guards by Lee and Lin [8]. Their proof was generalized to work for point guards by Aggarwal [9]. The problem was shown to be NP-hard even for simple orthogonal polygons by Katz and Roisman [10] and Schuchardt and Hecker [11]. Abrahamsen, Adamaszek and Miltzow [12] have recently shown that the art gallery problem for point guards is ETR-complete.

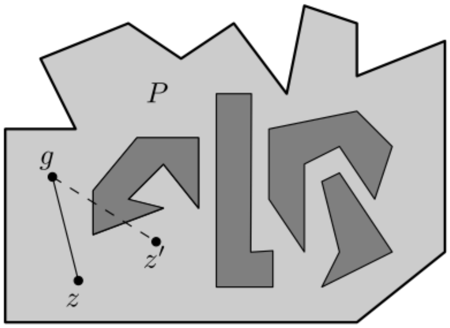

Fig. 1. Polygon with holes

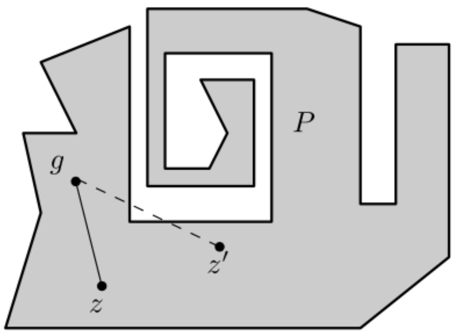

Fig. 2. Polygon without holes

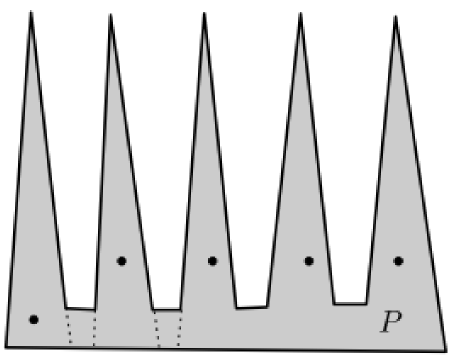

Fig. 3. A polygon where $\lfloor \frac{n}{3} \rfloor$ stationary guards are necessary.

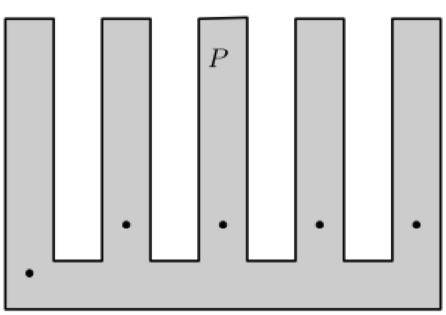

Fig. 4. A polygon where $\lfloor \frac{n}{4} \rfloor$ stationary guards are necessary.

In 1987, Ghosh [13], [14] 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. As pointed out by King and Kirkpatrick [15], newer methods for improving the approximation ratio of the Set Cover problem itself have been developed in the time after Ghoshs algorithm was published. By applying these methods, the approximation ratio of Ghosh's algorithm becomes $\mathcal{O}(\log OPT)$ for guarding simple polygons and $\mathcal{O}(\log h \log OPT)$ for guarding a polygon with h holes, where OPT denotes the size of the smallest guard set for P. Deshpande et al. [16] obtained an approximation factor of $\mathcal{O}(\log OPT)$ for point guards or perimeter guards by developing a sophisticated discretization method that runs in pseudo-polynomial time. Efrat and Har-Peled [17] provided a randomized algorithm with the same approximation ratio that runs in fully polynomial expected time. Bonnet and Miltzow [18] obtained an approximation factor of $\mathcal{O}(\log OPT)$ for the point guard problem assuming integer coordinates and a specific general position. For guarding simple polygons using perimeter guards, King and Kirkpatrick [15] designed a deterministic $\mathcal{O}(\log \log OPT)$-approximation algorithm in 2011. The analysis of this result was simplified by Kirkpatrick [19].

In 1998, Eidenbenz, Stamm and Widmayer [20], [21] 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 [22] 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. However, for simple polygons, the existence of a constant factor approximation algorithm for vertex and edge guards was conjectured by Ghosh [13], [23] in 1987.

Ghosh's conjecture has been shown to be true for vertex guarding in two special sub-classes of simple polygons, viz. monotone polygons and polygons weakly visible from an edge. In 2012, Krohn and Nilsson [24] 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 the optimal guard set. Bhattacharya, Ghosh and Roy [22], [25] presented a 6-approximation algorithm that runs in $\mathcal{O}(n^2)$ time for vertex guarding simple polygons that are weakly visible from an edge. For vertex guarding this subclass of simple polygons that are weakly visible from an edge, a PTAS has recently been proposed by Katz [26].

B. Our contributions

In this paper, we present three polynomial-time algorithms with a constant approximation ratio for guarding an n-sided simple polygon P using vertex guards. The first algorithm, that has an approximation ratio of 18, guards all vertices of P in $\mathcal{O}(n^4)$ time. The second algorithm, that has the same approximation ratio of 18, guards the entire boundary of P in $\mathcal{O}(n^5)$ time. The third algorithm, that has an approximation ratio of 27, guards all interior and boundary points of P in $\mathcal{O}(n^5)$ time. As an extension we show, using similar techniques, constant-factor approximation can also be achieved for guarding P we also present identical algorithms, maintaining both the approximation bounds as well as the running times, can be obtained using edge guards. In particular, we show that the same approximation ratios of 18, 18 and 27 hold for guarding all vertices, the entire boundary, and the interior of P, with time complexities $\mathcal{O}(n^4)$, $\mathcal{O}(n^5)$ and $\mathcal{O}(n^5)$ respectively. The significance of our results lies in the fact that these results settle the long-standing conjecture by Ghosh [13] regarding the existence of constant-factor approximation algorithms for these problem, which has been open since 1987 despite several attempts by researchers.

In each of our algorithms, P is first partitioned into a hierarchy of *weak visibility polygons* according to the *link distance* from a starting vertex (see Figure 6). This partitioning is very similar to the *window partitioning* given by Suri [27], [28] in the context of computing minimum link paths. Then, starting with the farthest level in the hierarchy (i.e. the set of weak visibility polygons that are at the maximum link distance from the starting vertex), the entire hierarchy is traversed backward level by level, and at each level, vertex guards (of two types, viz. *inside* and *outside*) are placed for guarding every weak visibility polygon at that level of P. At every level, a novel procedure is used that has been developed for placing guards in (i) a simple polygon that is weakly visible from an internal chord, or (ii) a union of overlapping polygons that are weakly visible from multiple disjoint internal chords. Note that these chords are actually the constructed edges introduced during the hierarchical partitioning of P.

Due to partitioning according to link distances, guards can only see points within the adjacent weak visibility polygons in the hierarchical partitioning of P. This property locally restricts the visibility of the chosen guards, and thereby ensures that the approximation bound on the number of vertex guards placed by our algorithms at any level leads directly to overall approximation bounds for guarding P. Thus, a constant factor approximation bound on the overall number of guards placed by our algorithms is a direct consequence of choosing vertex guards in a judicious manner for guarding each collection of overlapping weak visibility polygons obtained from the hierarchical partitioning of P. Our algorithms exploit several deep visibility structures of simple polygons which are interesting in their own right.

II. Preliminary definitions and notations

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 $\mathcal{V}(P)$ denote the set of all vertices. 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).

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 $\mathcal{VP}(bc) = P$, then P is said to be *weakly visible from bc*. 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 z (or bc) does not belong to $bd_c(v_iu_i)$, then v_iu_i is called a *left constructed edge* of $\mathcal{VP}(z)$ (respectively, $\mathcal{VP}(bc)$). Otherwise, v_iu_i is called a *right constructed edge*. For any constructed edge v_iu_i of $\mathcal{VP}(bc)$ (or $\mathcal{VP}(z)$), observe that v_iu_i divides P into two subpolygons. One of the subpolygons is bounded by $bd_c(v_i,u_i)$ and v_iu_i, whereas the other one is bounded by $bd_{cc}(v_i,u_i)$ and v_iu_i. Out of these two, the subpolygon that does not contain bc (respectively, z) is referred to as the *pocket* of v_iu_i, and is denoted by $P(v_iu_i)$ (see Figure 5). If v_iu_i is a left (or right) constructed edge, then $P(v_iu_i)$ is called a *left pocket* (or *right pocket*).

A *link path* between two points s and t in P is a path inside P that connects s and t by a chain of line segments called *links*. A *minimum link path* between s and t is a link path connecting s and t that has the minimum number of links. Observe that there may be several different minimum link paths between s and t. The *link distance* between any two points of P is defined to be the number of links in a minimum link path between them.

III. Partitioning a simple polygon into weak visibility polygons

Our partitioning algorithm partitions P into regions according to their link distance from v_1. The algorithm starts by computing $\mathcal{VP}(v_1)$, which is the set of all points of P whose link distance from v_1 is 1. Let us denote $\mathcal{VP}(v_1)$ as $V_{1,1}$. Then the algorithm computes the weak visibility polygons from every constructed edge of $V_{1,1}$. Let $v_{k(1)}u_{k(1)}, v_{k(2)}u_{k(2)}, \ldots, v_{k(c)}u_{k(c)}$ denote the constructed edges of $V_{1,1}$ along $bd(P)$ in clockwise order from v_1, where c is the number of constructed edges in $V_{1,1}$. Then the algorithm removes $V_{1,1}$ from P. It can be seen that the remaining polygon $P \setminus V_{1,1}$ consists of c disjoint polygons $P(v_{k(1)}u_{k(1)}), P(v_{k(2)}u_{k(2)}), \ldots, P(v_{k(c)}u_{k(c)})$. For each $j \in \{1, 2, \ldots, c\}$, the weak visibility polygon $\mathcal{VP}(v_{k(j)}u_{k(j)})$ is computed inside the pocket $P(v_{k(j)}u_{k(j)})$, and it is denoted as $V_{2,j}$, i.e. $V_{2,j} = \mathcal{VP}(v_{k(j)}u_{k(j)}) \cap P(v_{k(j)}u_{k(j)})$. Let $W_1 = \{V_{1,1}\}$ and $W_2 = \bigcup_{j=1}^{c} \{V_{2,j}\}$. Observe that W_2 is the set of all the disjoint regions of P, such that every point of each disjoint region in W_2 is at link distance two from v_1.

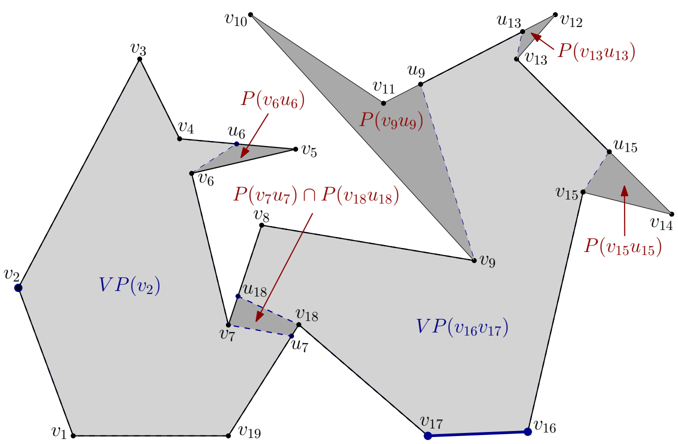

Fig. 5. Figure showing visibility polygon $\mathcal{VP}(v_2)$ and weak visibility polygon $\mathcal{VP}(v_{16}v_{17})$, along with several pockets created by constructed edges belonging to both.

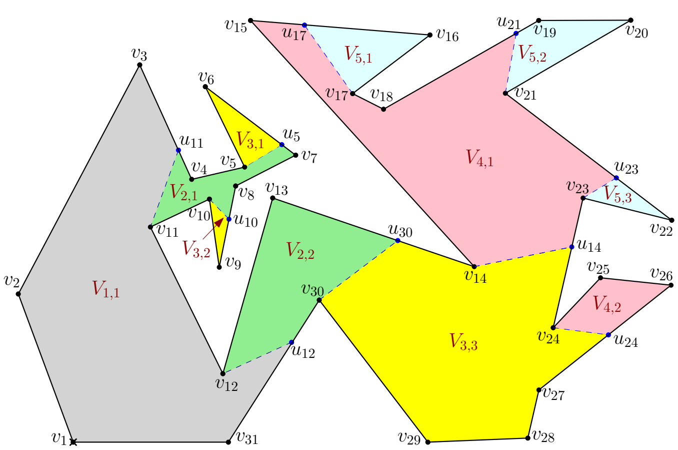

Fig. 6. Figure showing the partitioning of a simple polygon into visibility windows.

Repeating the same process, the algorithm computes $W_3, W_4, \ldots, W_d$, where d denotes the maximum link distance of any point of P from v_1. Note that it is not possible for any visibility polygon belonging to $W_d = \bigcup_{j=1}^{c} V_{d,j}$ to have any constructed edge. Therefore, no further visibility polygon is computed. Hence, $P = W_1 \cup W_2 \cup \ldots W_d = V_{1,1} \cup V_{2,1} \cup V_{2,2} \cup \ldots \cup V_{d,1} \cup V_{d,2} \cup \ldots$. Thus, the algorithm returns the set $W = \bigcup_{i=1}^{d} W_i$, which is a partition of P. We present the pseudocode for the entire partitioning algorithm below as Algorithm III.1.

Algorithm III.1 An algorithm for partitioning P

```
Compute VP(v_1)
V_{1,1} ← VP(v_1), W_1 ← {V_{1,1}}
C ← ⋃_{s∈W_1} (constructed edges of s), c ← |C|
W ← W_1, i ← 1
while c > 0 do
    i ← i + 1, W_i ← ∅
    for j = 1 to c do
        V_{i,j} ← VP(v_{k(j)}u_{k(j)}) ∩ P(v_{k(j)}u_{k(j)})
        W_i ← W_i ∪ {V_{i,j}}
    end for
    W ← W ∪ W_i
    C ← ⋃_{s∈W_i} (constructed edges of s), c ← |C|
end while
return W =0
```

Figure 6 shows the outcome of running Algorithm III.1 on a simple polygon P having 31 vertices, where the maximum link distance of any point of P from v_1 is 5. The algorithm returns the partition $W = \{V_{1,1}, V_{2,1}, V_{2,2}, V_{3,1}, V_{3,2}, V_{3,3,}, V_{4,1}, V_{4,2}, V_{5,1}, V_{5,2}, V_{5,3}\}$.

It can be seen that Algorithm III.1, as stated above, requires $\mathcal{O}(n^2)$ time, since the visibility polygons are computed separately. However, the running time can be improved to $\mathcal{O}(n)$ by using the partitioning method given by Suri [27], [28] in the context of computing minimum link paths. Using the algorithm of Hershberger [29] for computing visibility graphs of P, Suri's algorithm computes weak visibility polygons from selected constructed edges. The same method can be used to compute weak visibility polygons from all constructed edges of visibility polygons in W in $\mathcal{O}(n)$ time. The *visibility graph* of P is a graph which has a node corresponding to every vertex of P and there is an edge between a pair of nodes if and only if the corresponding pair of vertices are visible from each other in P. We summarize the result as follows.

Theorem 1: A simple polygon P can be partitioned into visibility polygons according to their link distance from any vertex in $\mathcal{O}(n)$ time.

IV. Traversing the Hierarchy of Visibility Polygons

Our algorithm for placement of vertex guards uses the hierarchy of visibility polygons W, as computed in Section III. Let $S_d, S_{d-1}, \ldots, S_2, S_1$ be the set of vertex guards chosen for guarding vertices of visibility polygons in $W_d, W_{d-1}, \ldots, W_2, W_1$ respectively. Since $W_1 = \{V_{1,1}\}$ and $V_{1,1} = \mathcal{VP}(v_1)$, we have $S_1 = \{v_1\}$. So the algorithm essentially has to decide guards in $S_d, S_{d-1}, \dots, S_2$. We have the following observation.

Lemma 2: For $2 \leq i < d$, every vertex guard in S_i belongs to some visibility polygon in $W_{i+1} \cup W_i \cup W_{i-1}$, whereas every vertex guard in S_d belongs to some visibility polygon in $W_d \cup W_{d-1}$.

As can be seen from Lemma 2, the placement of guards is locally restricted to visibility polygons belonging to adjacent levels in the partition hierarchy W. We formalize this intuition by introducing the notion of the *partition tree* of P, which is a *dual graph* denoted by T. Each visibility polygon $V_{i,j} \in W$ is represented as a vertex of T (also denoted by $V_{i,j}$), and two vertices of T are connected by an edge in T if and only if the corresponding visibility polygons share a constructed edge. Treating $V_{1,1}$ as the root of T, the standard parent-child-sibling relationships can be imposed between the visibility polygons in W.

Our algorithm starts off by guarding all vertices belonging to the visibility polygons in $W_d = \{V_{d,1}, V_{d,2}, \dots\}$, which are effectively the nodes of T furthest from the root $V_{1,1}$. The algorithm scans $V_{d,1}$,$V_{d,2}$,... separately for identifying the respective guards in S_d. We know from Lemma 2 that every vertex guard in S_d belongs to some visibility polygon in $W_d \cup W_{d-1}$. Consider a particular $V_{d,k} \in W_d$, and let $V_{d-1,j} \in W_{d-1}$ be the parent of $V_{d,k}$ in T. Consider the constructed edge $v_k u_k$ between $V_{d,k}$ and $V_{d-1,j}$. For guarding the vertices of $V_{d,k} = \mathcal{VP}(v_k u_k) \setminus V_{d-1,j}$, it is enough to focus on the subpolygon Q consisting of $V_{d,k}$ itself and the portion of $V_{d-1,j}$ that is weakly visible from $v_k u_k$. So, the subproblem of guarding $V_{d,k}$ (or any other visibility polygon belonging to W_d) essentially reduces to placing vertex guards in a polygon containing a weak visibility chord vu (corresponding to $v_k u_k$ in the original subproblem) in order to guard only the vertices lying on one side of uv; however, vertex guards can be chosen freely from either side of the chord uv.

Instead of guarding each weak visibility polygon Q separately, common vertex guards can be placed by traversing the boundary of overlapping weak visibility polygons. Let us explain by considering any $V_{d-1,j} \in W_{d-1}$. Let us denote the constructed edges that are shared between $V_{d-1,j}$ and the m children of $V_{d-1,j}$ as $v_{j(1)}u_{j(1)}, v_{j(2)}u_{j(2)}, \dots, v_{j(m)}u_{j(m)}$ respectively. Using all these constructed edges, let us construct the weak visibility polygons $\mathcal{VP}(v_{j(1)}u_{j(1)})$, $\mathcal{VP}(v_{j(2)}u_{j(2)})$, $\dots$, $\mathcal{VP}(v_{j(m)}u_{j(m)})$. Observe that each such weak visibility polygon is divided into two portions by the corresponding constructed edge; one of the portions forms a child of $V_{d-1,j}$ belonging to W_d, whereas the other portion is a subregion of $V_{d-1,j}$ itself. Moreover, for several of the weak visibility polygons among $\mathcal{VP}(v_{j(1)}u_{j(1)}), \mathcal{VP}(v_{j(2)}u_{j(2)}), \dots, \mathcal{VP}(v_{j(m)}u_{j(m)})$, the second portions may have overlapping subregions in $V_{d-1,j}$. Thus, there may exist vertex guards in these overlapping subregions that can see portions of several of the children of $V_{d-1,j}$. Therefore, for guarding vertices of polygons from W_d, let us extend the definition of Q to be the union of all the overlapping weak visibility polygons defined by the constructed edges corresponding to the children of each $V_{d-1,j}$. For instance, consider the constructed edges $v_{17}u_{17}$, $v_{21}u_{21}$ and $v_{23}u_{23}$ on the boundary of $V_{4,1}$ in Figure 6; for guarding the corresponding children $V_{5,1}$, $V_{5,2}$ and $V_{5,3}$ respectively, we define Q as $\mathcal{VP}(v_{17}u_{17}) \cup \mathcal{VP}(v_{21}u_{21}) \cup \mathcal{VP}(v_{23}u_{23})$ and traverse Q.

After having successively computed S_d for guarding vertices belonging to visibility polygons in $W_d = \{V_{d,1}, V_{d,2}, \dots\}$, the algorithm next computes S_{d-1} for guarding vertices belonging to visibility polygons in $W_{d-1} = \{V_{d-1,1}, V_{d-1,2}, \dots\}$. Since all vertices belonging to visibility polygons in W_d are already marked by guards chosen belonging to S_d, all remaining unmarked vertices of P can have link distance at most $d-1$ from v_1. So, any weak visibility polygon $V_{d-1,k} \in W_{d-1}$ can now be treated as a weak visibility polygon that is the farthest link distance from v_1. Therefore, the guards of S_{d-1} are chosen in a similar way as those of S_d. It can be seen that this same method can be used for computing S_i for every $i < d$. Thus, in successive phases, our algorithm computes the guard sets $S_d, S_{d-1}, S_{d-2}, \dots, S_2$ for guarding vertices belonging to visibility polygons in $W_d, W_{d-1}, W_{d-2}, \dots, W_2$ respectively, until it finally terminates after placing a single guard at v_1 for guarding vertices of $V_{1,1} \in W_1$. The final guard set $S = S_d \cup S_{d-1} \cup S_{d-2} \cup \dots \cup S_2 \cup S_1$ returned by the algorithm guards all vertices of P. The pseudocode for the entire algorithmic framework is provided below.

Algorithm IV.1 Algorithm for computing a guard set S from the partition tree T rooted at v_1

```
Initialize all vertices of P as unmarked
d ← number of levels in the partition tree T
for each i ∈ {d − 1, ..., 3, 2, 1} do
    S_{i+1} ← ∅
    c_i ← |W_i| {c_i denotes the number of nodes at the ith level of T}
    for each j ∈ {1, 2, ..., c_i} do
        Place new guards in S_{i+1} for guarding every unmarked vertex of all children of V_{i,j}
        Mark all vertices of P that are visible from the new guards added to S_{i+1}
    end for
end for
S_1 ← {v_1}
return S = S_d ∪ S_{d−1} ∪ S_{d−2} ∪ ··· ∪ S_2 ∪ S_1 =0
```

V. Conclusions

Using the algorithmic framework described above, we obtained three approximation algorithms for guarding a simple n-sided polygon P using vertex guards, which we have summarized in the following theorems [30].

Theorem 3: A set G of vertex guards for guarding all vertices of P can be computed in $\mathcal{O}(n^4)$ time, such that $|G| \leq 18 \times |G_{opt}|$, where G_{opt} is a an optimal vertex guard set for guarding all vertices of P. [30]

Theorem 4: A set G of vertex guards for guarding the entire boundary of P can be computed in $\mathcal{O}(n^5)$ time, such that $|G| \leq 18 \times |G_{opt}|$, where G_{opt} is a an optimal vertex guard set for guarding the entire boundary of P. [30]

Theorem 5: A set G of edge guards for guarding the entire interior and boundary of P can be computed in $\mathcal{O}(n^5)$ time, such that $|G| \leq 27 \times |G_{opt}|$, where G_{opt} is a an optimal vertex guard set for guarding the entire interior of P. [30]

These algorithms can be easily modified to obtain the same approximation bounds while using edge guards as well. We also believe that suitable modifications of our algorithm may lead to constant-factor approximations for the version of the problem that uses perimeter guards, which is still open. Though the approximation ratios for our algorithms are slightly on the higher side, they do successfully settle the long-standing conjecture by Ghosh by providing constant-factor approximation algorithms for these problems. We feel that, in practice, our algorithms will provide guard sets that are much closer in size to an optimal solution. Our algorithms exploit several deep visibility structures of simple polygons which are interesting in their own right.

Acknowledgment

The author would like to thank Tata Consultancy Services for their continued financial support since July 2015.

References

[1] S. K. Ghosh, *Visibility Algorithms in the Plane*. Cambridge University Press, 2007.

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

[3] V. Chvtal, "A combinatorial theorem in plane geometry," *Journal of Combinatorial Theory, Series B*, vol. 18, no. 1, pp. 39–41, 1975.

[4] S. Fisk, "A short proof of Chvátal's watchman theorem," *Journal of Combinatorial Theory, Series B*, vol. 24, no. 3, p. 374, 1978.

[5] J. Kahn, M. Klawe, and D. Kleitman, "Traditional galleries require fewer watchmen," *SIAM Journal of Algebraic and Discrete Methods*, vol. 4, no. 2, pp. 194–206, 1983.

[6] J. O'Rourke, "An alternative proof of the rectilinear art gallery theorem," *Journal of Geometry*, vol. 211, pp. 118–130, 1983.

[7] J. O'Rourke and K. J. Supowit, "Some NP-hard polygon decomposition problems," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 181–189, 1983.

[8] D. Lee and A. Lin, "Computational complexity of art gallery problems," *IEEE Transactions on Information Theory*, vol. 32, no. 2, pp. 276–282, 1986.

[9] A. Aggarwal, "The art gallery theorem: its variations, applications and algorithmic aspects," Ph.D. dissertation, The Johns Hopkins University, Baltimore, Maryland, 1984.

[10] M. J. Katz and G. S. Roisman, "On guarding the vertices of rectilinear domains," *Computational Geometry*, vol. 39, no. 3, pp. 219–228, 2008.

[11] D. Schuchardt and H.-D. Hecker, "Two NP-Hard Art-Gallery Problems for Ortho-Polygons," *Mathematical Logic Quarterly*, vol. 41, pp. 261–267, 1995.

[12] M. Abrahamsen, A. Adamaszek, and T. Miltzow, "The art gallery problem is ∃ R-complete," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, 2018, pp. 65–73.

[13] S. K. Ghosh, "Approximation algorithms for art gallery problems," in *Proceedings of Canadian Information Processing Society Congress*. Canadian Information Processing Society, 1987, pp. 429–434.

[14] ——, "Approximation algorithms for art gallery problems in polygons," *Discrete Applied Mathematics*, vol. 158, no. 6, pp. 718–722, 2010.

[15] J. King and D. G. Kirkpatrick, "Improved approximation for guarding simple galleries from the perimeter," *Discrete & Computational Geometry*, vol. 46, no. 2, pp. 252–269, 2011.

[16] A. Deshpande, T. Kim, E. D. Demaine, and S. E. Sarma, "A pseudopolynomial time *o*(log *n*)-approximation algorithm for art gallery problems," in *WADS*, 2007, pp. 163–174.

[17] A. Efrat and S. Har-Peled, "Guarding galleries and terrains," *Information Processing Letters*, vol. 100, no. 6, pp. 238–245, 2006.

[18] É. Bonnet and T. Miltzow, "An Approximation Algorithm for the Art Gallery Problem," in *33rd International Symposium on Computational Geometry (SoCG 2017)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), B. Aronov and M. J. Katz, Eds., vol. 77. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017, pp. 20:1–20:15.

[19] D. G. Kirkpatrick, "An o(lg lg opt)-approximation algorithm for multi-guarding galleries," *Discrete & Computational Geometry*, vol. 53, no. 2, pp. 327–343, 2015.

[20] S. Eidenbenz, C. Stamm, and P. Widmayer, "Inapproximability of some art gallery problems," *Canadian Conference on Computational Geometry*, pp. 1–11, 1998.

[21] ——, "Inapproximability results for guarding polygons and terrains," *Algorithmica*, vol. 31, no. 1, pp. 79–113, 2001.

[22] P. Bhattacharya, S. K. Ghosh, and B. Roy, "Vertex Guarding in Weak Visibility Polygons," in *Proceedings of the 1st International Conference on Algorithms and Discrete Applied Mathematics (CALDAM 2015)*, ser. Lecture Notes in Computer Science (LNCS), vol. 8959. Springer, 2015, pp. 45–57.

[23] S. K. Ghosh, "Approximation algorithms for art gallery problems in polygons and terrains," *WALCOM: Algorithms and Computation*, pp. 21–34, 2010.

[24] E. A. Krohn and B. J. Nilsson, "Approximate Guarding of Monotone and Rectilinear Polygons," *Algorithmica*, vol. 66, pp. 564–594, 2013.

[25] P. Bhattacharya, S. K. Ghosh, and B. Roy, "Approximability of guarding weak visibility polygons," *Discrete Applied Mathematics*, vol. 228, pp. 109 – 129, 2017.

[26] M. J. Katz, "A PTAS for vertex guarding weakly-visible polygons," *arXiv preprint arXiv:1803.02160*, 2018.

[27] S. Suri, "A linear time algorithm with minimum link paths inside a simple polygon," *Comput. Vision Graph. Image Process.*, vol. 35, no. 1, pp. 99–110, Jul. 1986.

[28] ——, "Minimum link paths in polygons and related problems," Ph.D. dissertation, The Johns Hopkins University, Baltimore, Maryland, 1987.

[29] J. Hershberger, "An optimal visibility graph algorithm for triangulated simple polygons," *Algorithmica*, vol. 4, no. 1, pp. 141–155, 1989.

[30] P. Bhattacharya, S. K. Ghosh, and S. P. Pal, "Constant approximation algorithms for guarding simple polygons using vertex guards," *CoRR*, vol. abs/1712.05492, 2017. [Online]. Available: http://arxiv.org/abs/1712.05492