Guarding Monotone Polygons with Vertex Half-Guards is NP-Hard*

Matt Gibson[1], Erik Krohn[2], and Matthew Rayford[2]

1 University of Texas - San Antonio, gibson@cs.utsa.edu
2 University of Wisconsin - Oshkosh, {krohne, rayfom16}@uwosh.edu

Abstract

We consider a variant of the art gallery problem where all guards are limited to seeing to the right inside a monotone polygon. We show that the problem is NP-hard if guards are restricted to be at the vertices of the polygon.

1 Introduction

An instance of the art gallery problem takes as input a simple polygon P. If these edges do not intersect other than at the vertices in V, then P is called a simple polygon. The edges of a simple polygon give us two disjoint regions: the interior and exterior of the polygon. For any two points $p, q \in P$, we say that p *sees* q if the line segment $\overline{pq}$ does intersect the exterior of P. The art gallery problem seeks to find a set of points $G \subseteq P$ such that every point $p \in P$ is seen by a point in G. We call this set G a guarding set. In the point guarding problem, guards can be placed anywhere in the interior of P. In the vertex guarding problem, guards are only allowed to be placed at V. The optimization problem is thus defined as finding the smallest such G.

Art gallery problems are motivated by applications such as line of-sight transmission networks in terrains, signal communications and cellular telephony systems and other telecommunication technologies as well as placement of motion detectors and security cameras.

1.1 Previous Work

The question of whether guarding simple polygons is NP-hard was independently confirmed by Aggarwal [2] and Lee and Lin [15]. They showed that the problem is NP-hard for both vertex guarding and point guarding. Along with being NP-complete, Brodén et al. [6] and Eidenbenz [8] independently proved that point guarding simple polygons is APX-hard. This means that there exists a constant $\epsilon > 0$ such that no polynomial-time algorithm can guarantee an approximation ratio of $(1+\epsilon)$ unless P=NP. Ghosh provides a $O(\log n)$-approximation for the problem of vertex guarding an n-vertex simple polygon in [10]. This result can be improved for simple polygons using randomization, giving an algorithm with expected running time $O(nOPT^2 \log_4 n)$ that produces a vertex guard cover with approximation factor $O(\log OPT)$ with high probability, where OPT is the smallest vertex guard cover for the polygon [7]. Bhattacharya et. al claim a constant factor approximation for guarding simple polygons using vertex guards in [4]. Assuming integer coordinates and a specific general position, Bonnet and Miltzow present an algorithm for finding a point guard cover with approximation factor $O(\log OPT)$ in [5]. King and Kirkpatrick provide a $O(\log \log OPT)$-approximation algorithm for the problem of guarding a simple polygon with guards on the perimeter in [12].

* Partially supported by a faculty development grant from UW-Oshkosh

34th European Workshop on Computational Geometry, Berlin, Germany, March 21–23, 2018.

Additional Polygon Structure. Due to the inherent difficulty in fully understanding the art gallery problem for simple polygons, there has been some work done guarding polygons with some additional structure. A simple polygon P is x-monotone (or simply monotone) if any vertical line intersects the boundary of P in at most two points. Let l and r denote the leftmost and rightmost point of P respectively. Consider the "top half" of the boundary of P by walking along the boundary clockwise from l to r. We call this the *ceiling* of P. Similarly we obtain the *floor* of P by walking clockwise along the boundary from r to l. Notice that both the ceiling and the floor are x-monotone polygonal chains, that is a vertical line intersects it in at most one point. Krohn and Nilsson [14] give a polynomial-time constant factor approximation algorithm for point guarding monotone polygons. They also proved point guarding and vertex guarding a monotone polygon is NP-hard [13, 14].

α-Floodlights. Motivated by the fact that many cameras and other sensors generally are not able to sense in 360 degrees, previous works have considered the problem when guards have a fixed sensing angle α for some $0 < \alpha \leq 360$. This problem is often referred to as the *α-floodlight problem*. 180°-floodlights are sometimes referred to as *half-guards*. Some of the work on this problem has involved proving necessary and sufficient bounds on the number of α-floodlights required to guard (or illuminate) an n vertex simple polygon P, where floodlights are anchored at vertices in P and no vertex is assigned more than one floodlight, see for example [17, 9, 16]. From an approximation complexity standpoint, it is known that computing a minimum cardinality set of α-floodlights to illuminate a simple polygon P is APX-hard for both the point guard and vertex guard variants [1, 3]. Other works in this area include considering the problem where $\alpha < 180°$.

1.2 Our Contribution

In this paper, we consider guarding monotone polygons with half-guards that can see in one direction, namely to the right. Let $p.x$ denote the x-coordinate of a point p. We modify the definition of *sees* to be the following: a point p sees a point q if the line segment $\overline{pq}$ does not intersect the exterior of P and $p.x \leq q.x$. A constant factor approximation for this problem was given in [11].

Our main result is to show that vertex guarding a monotone polygon with half-guards is NP-hard. Krohn and Nilsson [14] obtained a similar NP-hardness result using full guards, but guards were required to see in all directions. The reduction could not be trivially tweaked to show the half-guard problem is NP-hard.

In Section 2, we provide a high level overview that vertex guarding a monotone polygon with half-guards is NP-hard. Section 3 provides the details of the proof.

2 NP-Hardness for Vertex Guards

The reduction is from *3SAT*. A 3SAT instance (X, C) contains a set of Boolean variables, $X = \{x_1, x_2, \dots, x_n\}$ and a set of clauses, $C = \{c_1, c_2, \dots, c_m\}$. Each clause contains three literals, $c_i = (x_j \vee x_k \vee x_l)$. A 3SAT instance is satisfiable if a satisfying truth assignment for X exists such that all clauses c_i are true. We show that any 3SAT instance is polynomially transformable to an instance of vertex guarding a monotone polygon using half-guards. We construct a monotone polygon P from the 3SAT instance such that P is guardable by $K = (2 + m)n + 1$ or fewer guards if and only if the 3SAT instance is satisfiable.

The high level overview of the reduction is that certain vertices represent the truth values of the variables in the 3SAT instance. All starting patterns are placed on the ceiling on the left side of the polygon, see Figure 1. We assume that all guards can see only to the right. In these starting patterns, one must choose one of two guardset locations in order to guard distinguished vertices for that particular pattern. A *distinguished vertex* is a vertex that is seen only by a small number of specific vertices. In each variable pattern, similar to a starting pattern, certain vertices will represent a truth assignment of true and certain vertices will represent a truth assignment of false for some variable. This information is then “mirrored rightward” going from the ceiling, to the floor and then back to the ceiling such that there is a consistent choice of the x_j vertices or the $\overline{x_j}$ vertices for each variable. This differs from previous results where variable information was mirrored from the “left side” of the polygon to the “right side” of the polygon and then back to the left side. A distinguished clause vertex is placed to the right of the variable patterns such that only the vertices representing the literals in the specific clause can see the clause distinguished vertex. A high level example of the entire reduction is shown in Figure 1.

▶ **Theorem 1.** *Finding the smallest vertex guard cover for a monotone polygon using half guards is NP-hard.*

3 Hardness Details

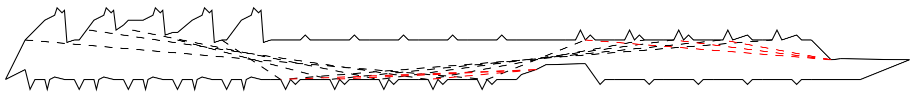

Figure 1 A high level overview of the reduction.

Starting Pattern: This pattern appears along the left side of the monotone polygon a total of n times, one corresponding to each variable, see Figure 2. In each pattern, there are 3 distinguished vertices: $\{h_4, h_{10}, h_{13}\}$. These vertices are seen by a specific subset of vertices in each starting pattern. It is important to note that no other vertex outside of this starting pattern sees these distinguished points. Let $v_l(p)$ be the set of vertices that see p. Note that all vertices in $v_l(p)$ lie to the left of p or on the vertical line that contains p.

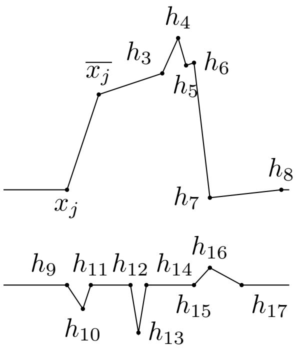

Figure 2 A starting pattern.

Let's assume we are considering the starting pattern for variable x_j. $v_l(h_{10}) = \{h_{10}, h_9, x_j\}$, $v_l(h_{13}) = \{h_{12}, h_{13}, \overline{x_j}\}, v_l(h_4) = \{h_3, h_4, h_9, h_{11}, h_{12}, h_{14}\}$. One should note that one guard does not see all of the distinguished points. Two guards are necessary and sufficient. The only possible combinations of vertex guards that see each distinguished vertex are: $\{x_j, h_{12}\}, \{\overline{x_j}, h_9\}$. If the second option is chosen, then it appears that the x_j vertex is unseen. However, the polygon is drawn in such a way such that the leftmost point in the polygon sees x_j for all j, see Figure 1.

Variable Pattern: On the floor of the polygon to the right of the n starting patterns are the first n variable patterns, one for each variable, that verify and propagate the assigned truth value of each variable. The variables are in reverse order from the initial starting pattern. The variables are ordered from $x_1, x_2, \ldots, x_n$ in the starting patterns from left to right. However, the variables are ordered from $x_n, x_{n-1}, \ldots, x_1$ in the first grouping of variable patterns from left to right. When the variables are “mirrored” rightward again to the ceiling, the ordering will again reverse. See Figure 1 for a high level overview.

A single variable pattern is shown in Figure 4. Similar to the starting pattern, there are 3 distinguished vertices located at $\{v_2, v_5, v_7\}$. The visibility for these points within this pattern are as follows: $v_l(v_2) = \{v_1, v_2, x_j, v_6, \overline{x_j}, v_8\}, v_l(v_5) = \{v_5, x_j\}, v_l(v_7) = \{v_7, \overline{x_j}\}$. It should be noted that v_2 is not seen by another vertex outside of this pattern. One guard within this pattern is necessary to guard this distinguished vertex. Along with these visibilities, v_5 is seen by the $\overline{x_j}$ vertex in the starting pattern representing x_j. v_5 does not see the x_j vertex from the starting pattern because it is angled in such a way that its line of sight is "above" the x_j vertex in the starting pattern. v_7 is seen by the x_j vertex in the starting pattern representing x_j. The reason it is not seen by $\overline{x_j}$ is because the $\overline{x_j}$ vertex in the starting pattern is being blocked by h_7 in the starting pattern. Figure 3 shows how the starting patterns are connected to variable patterns.

Variable patterns are connected to other variable patterns in a similar fashion. Consider a set of n variable patterns on the floor representing one mirroring of the variables. At the far right of these patterns is a vertex called c_2. This vertex will block our x_j and $\overline{x_j}$ vertices from seeing too far to the right. Consider a single variable x_j being mirrored from the floor to the ceiling. In Figure 5, the ceiling variable pattern is simply an inverted floor variable pattern. The v_5 vertex in the ceiling variable pattern sees the $\overline{x_j}$ vertex in the floor variable pattern and not the x_j vertex in the floor variable pattern because the angle of the polygon blocks it. v_7 in the ceiling variable pattern sees the x_j vertex in the floor variable pattern but not the $\overline{x_j}$ vertex in the floor variable pattern because it is being blocked by c_2.

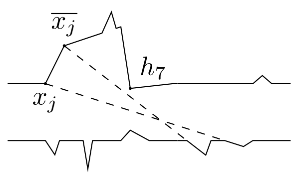

Figure 3 Starting pattern interacting with first variable gadget.

Different variable patterns that represent different variables will not affect each other. For example, take the starting pattern for an arbitrary x_i and call the vertices that see the distinguished vertices in that starting pattern the X_i set. Now consider the variable pattern for x_i and look at the variable patterns to the left of x_i on the floor. None of X_i can see the distinguished vertices of variable patterns to the left of the variable pattern for x_i because the distinguished vertices in those variable patterns are angled too far to the "right." None of X_i can see distinguished vertices of variable patterns to the right of the variable pattern for x_i because h_7 or h_{16} is blocking them from seeing too far right, see Figure 2.

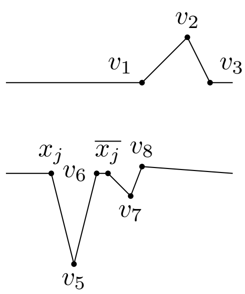

Figure 4 Variable pattern x_j.

In a similar fashion, variable patterns will not affect other variable patterns when mirroring, see Figure 6. The variable pattern on the ceiling for x_i will not be seen by the previous variable pattern on the floor for x_{i+1} because the angle of the polygon in the variable pattern for x_i on the ceiling is too steep. In other words, the distinguished vertices for x_i on the ceiling will not be able to be seen from that far left. The vertices in the variable pattern on the ceiling for x_i will not be seen by the previous variable pattern on the ceiling for x_{i-1} because the c_2 vertex will block them.

We allow one guard to be placed in a single variable pattern. No single guard is able to see all of the distinguished points. Therefore, one must rely on previously placed guards to help see at least 1 of the distinguished points in the variable pattern. If we choose x_j in the starting pattern or in some previous variable pattern, we see the v_7 vertex in the subsequent variable pattern. The only guard in the variable pattern that sees v_2 and v_5 is x_j. If we choose $\overline{x_j}$ in the starting pattern or in some previous variable pattern, the distinguished points that are unseen are v_2 and v_7 in the subsequent variable patterns and the only guard in the variable pattern that sees them is $\overline{x_j}$. In this second case, x_j is seen by that previously placed guard that also sees v_5.

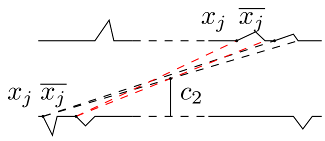

Figure 5 An example of a variable being mirrored.

Clauses: For each clause c in the boolean formula, there is a sequence of variable patterns $x_1, \ldots, x_n$ along either the ceiling or the floor of the polygon. Immediately to the right of the variable patterns exists a clause pattern. A clause pattern consists of one vertex such that the vertex is only seen by the variable patterns corresponding to the literals in the clause; see Figure 7. The distinguished vertex of the clause pattern is the c_3 vertex. This vertex is seen only by specific vertices in its respective sequence of variable patterns.

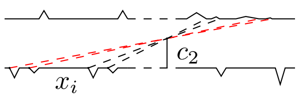

Figure 6 An example of multiple variables being mirrored.

To see how a clause is placed in the polygon, consider Figure 8 that represents the clause $x_1 \vee \overline{x_3} \vee x_5$. Initially, all x_i and $\overline{x_i}$ vertices in their respective variable patterns are blocked from seeing the c_3 clause point by their respective v_8 vertex. Consider the example clause of $x_1 \vee \overline{x_3} \vee x_5$. In the case of x_1 and x_5, their respective v_8 vertices have been lowered just enough such that the v_8 vertex is no longer blocking them from seeing c_3. However, v_8 is still blocking $\overline{x_j}$ from seeing c_3. In the case of $\overline{x_3}$, the v_8 guard is lowered enough such that $\overline{x_3}$ sees c_3. To keep x_3 from seeing c_3, we raise the v_6 vertex just enough so it blocks x_3 from c_3. It should be noted that these small tweaks do not affect the mirroring of variable truth values. None of the x_j or $\overline{x_j}$ vertices were moved. Their position with respect to the key blocker of c_2 is the same. Therefore, c_2 still blocks each respective vertex from seeing too far to the right.

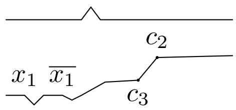

Figure 7 A clause gadget to the right of x_1.

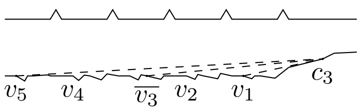

Figure 8 The clause $(v_1 \vee \overline{v_3} \vee v_5)$.

Putting it all together: We choose our truth value for each variable in the starting variable patterns. The truth values are then mirrored in turn between variable patterns on the ceiling and the floor. In the example of Figure 8 the 3SAT clause corresponds to $c = x_1 \vee \overline{x_3} \vee x_5$. Hence, a vertex guard placement that corresponds to a truth assignment that makes c_3 true, will have at least one guard on $x_1, \overline{x_3}$ or x_5 and can therefore see vertex c_3 without additional guards. We still have variables x_2 and x_4 on the polygon, however, none of them or their negations see the vertex c_3. They are simply there to transfer their truth values in case these variables are needed in later clauses.

The monotone polygon we construct consists of $17n + (9n + 3)m + 2$ vertices. Each starting variable pattern has 17 vertices, each variable pattern 9 vertices, the clause pattern has 3 vertices, plus 2 vertices for the leftmost and rightmost points of the polygon. Exactly $K = (2 + m)n + 1$ guards are required to guard the polygon. 2 guards are required to see the distinguished points of the starting patterns ($2n$) and 1 guard is required at every variable pattern, of which there are (mn) of them. Lastly, since a starting pattern cannot begin at the leftmost point, a guard is required at the leftmost vertex of the polygon. If the 3SAT instance is satisfiable, then guards are placed at vertices in accordance to whether the variable is true or false in each of the sequences of variable patterns. Each clause vertex is seen since one of the literals in the associated clause is true and the corresponding vertex has a guard.

References

1. Ahmed Abdelkader, Ahmed Saeed, Khaled A. Harras, and Amr Mohamed. The inapproximability of illuminating polygons by α-floodlights. In *Proceedings of the 27th Canadian Conference on Computational Geometry, CCCG 2015, Kingston, Ontario, Canada, August 10-12, 2015*, 2015.
2. Alok Aggarwal. *The art gallery theorem: its variations, applications and algorithmic aspects.* PhD thesis, 1984. AAI8501615.
3. Jay Bagga, Laxmi Gewali, and David Glasser. The complexity of illuminating polygons by alpha-flood-lights. In *Proceedings of the 8th Canadian Conference on Computational Geometry, Carleton University, Ottawa, Canada, August 12-15, 1996*, pages 337–342, 1996.
4. Pritam Bhattacharya, Subir Kumar Ghosh, and Sudebkumar Pal. Constant approximation algorithms for guarding simple polygons using vertex guards. `https://arxiv.org/abs/1712.05492v1`.
5. Édouard Bonnet and Tillmann Miltzow. An approximation algorithm for the art gallery problem. In *33rd International Symposium on Computational Geometry, SoCG 2017, July 4-7, 2017, Brisbane, Australia*, pages 20:1–20:15, 2017.
6. Björn Brodén, Mikael Hammar, and Bengt J. Nilsson. Guarding lines and 2-link polygons is apx-hard. In *CCCG*, pages 45–48, 2001.
7. Alon Efrat and Sariel Har-Peled. Guarding galleries and terrains. *Inf. Process. Lett.*, 100(6):238–245, 2006.
8. Stephan Eidenbenz. Inapproximability results for guarding polygons without holes. In Kyung-Yong Chwa and Oscar H. Ibarra, editors, *ISAAC*, volume 1533 of *Lecture Notes in Computer Science*, pages 427–436. Springer, 1998.
9. Vladimir Estivill-Castro, Joseph O'Rourke, Jorge Urrutia, and Dianna Xu. Illumination of polygons with vertex lights. *Information Processing Letters*, 56(1):9 – 13, 1995.
10. Subir Kumar Ghosh. On recognizing and characterizing visibility graphs of simple polygons. *Discrete & Computational Geometry*, 17(2):143–162, 1997.
11. Matt Gibson, Erik Krohn, and Matthew Rayford. Guarding monotone polygons with half-guards. In *Proceedings of the 29th Canadian Conference on Computational Geometry, July 26-28, 2017, Carleton University, Ottawa, Ontario, Canada*, pages 168–173, 2017.
12. James King and David G. Kirkpatrick. Improved approximation for guarding simple galleries from the perimeter. *Discrete & Computational Geometry*, 46(2):252–269, 2011.
13. Erik Krohn and Bengt J. Nilsson. The complexity of guarding monotone polygons. In *CCCG*, pages 167–172, 2012.
14. Erik A. Krohn and Bengt J. Nilsson. Approximate guarding of monotone and rectilinear polygons. *Algorithmica*, pages 1–31, 2012.
15. D. T. Lee and A. K. Lin. Computational complexity of art gallery problems. *IEEE Trans. Inform. Theory*, 32(2):276–282, March 1986.
16. Bettina Speckmann and Csaba D. Tóth. Allocating vertex π-guards in simple polygons via pseudo-triangulations. *Discrete & Computational Geometry*, 33(2):345–364, 2005.
17. Csaba D. Tóth. Art gallery problem with guards whose range of vision is 180°. *Computational Geometry*, 17(3):121 – 134, 2000.