The Complexity of Guarding Monotone Polygons

Erik Krohn[*] Bengt J. Nilsson[†]

Abstract

A polygon P is x-monotone if any line orthogonal to the x-axis has a simply connected intersection with P. A set G of points inside P or on the boundary of P is said to guard the polygon if every point inside P or on the boundary of P is seen by a point in G.

An interior guard can lie anywhere inside or on the boundary of the polygon. Using a reduction from Monotone 3SAT, we prove that interior guarding a monotone polygon is NP-hard. Because interior guards can be placed anywhere inside the polygon, a clever gadget is introduced that forces interior guards to be placed at very specific locations.

1 Introduction

The art gallery problem is perhaps one of the best known problems in computational geometry. It asks for the minimum number of guards to guard a space. An instance of the art gallery problem takes as input a polygon P. The polygon P is defined by a set of points $V = \{v_1, v_2, ..., v_n\}$. There are edges connecting (v_i, v_{i+1}) where $i = 1, 2, ..., n-1$. There is an edge connecting (v_n, v_1). These edges give us two disjoint regions: inside the polygon and outside the polygon. For any two points $p, q \in P$, we say that p sees q if the line segment $\overline{pq}$ does not go outside of P. We wish to find a set of points $G \subseteq P$ such that every point $p \in P$ is seen by a guard in G. We call this set G a guarding set. 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 broadcasting, 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 settled by Aggarwal [1] and Lee and Lin [14] independently. They showed that the problem is NP-hard for both vertex guards and interior guards. Along with being NP-complete, Brodén *et al.* and Eidenbenz [2, 7] independently prove that interior 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. Further results have shown that guarding a restricted subclass of polygons is still NP-hard [2, 15]. O'Rourke and Supowit show that several polygon decomposition problems are NP-hard [17].

Ghosh provides a $O(\log\ n)$-approximation for the problem of vertex guarding an n-vertex simple polygon in [11]. This result can be improved for simple polygons using randomization, giving an algorithm with expected running time $O(nOPT_v^2 \log^4 n)$ that produces a vertex guard cover with approximation factor $O(\log OPT_v)$ with high probability, where OPT_v is the smallest vertex guard cover for the polygon [6]. Whether a constant factor approximation can be obtained for vertex guarding a simple polygon is a longstanding and well-known open problem. Deshpande *et al.* [5] present a pseudopolynomial randomized algorithm for finding a point guard cover with approximation factor $O(\log OPT)$. King and Kirkpatrick provide an O(log log OPT)-approximation algorithm for the problem of guarding a simple polygon with guards on the perimeter in [12]. The point guarding problem seems to be much more difficult and precious little is known about it [5]. A constant factor approximation is given by Nilsson for the special case of the problem when the polygon is x-monotone [16]. Based on his result, Nilsson gives an $O(OPT^2)$-approximation algorithm for rectilinear polygons.

The approximation complexity of guarding polygons has been studied by Eidenbenz and others. Eidenbenz [8] shows that polygons with holes cannot be efficiently guarded by fewer than $\Omega(\log n)$ times the optimal number of interior or vertex guards, unless P=NP, where n is the number of vertices of the polygon.

Tight bounds for the number of guards necessary and sufficient were found by Chvátal [4]. It is sometimes necessary to place $\frac{n}{3}$ guards to guard the entire polygon. Fisk provided a simpler proof of [4] in [9] that broke up any polygon into a set of triangles and showed that this set of triangles can be 3-colored which implies that $\frac{n}{3}$ guards are sufficient for guarding a simple polygon.

[*] Department of Computer Science, University of Wisconsin - Oshkosh, Oshkosh, WI, USA. email: krohne@uwosh.edu

[†] Computer Science, Malmö University College, SE-205 06 Malmö, Sweden. email: Bengt.Nilsson.TS@mah.se

1.2 Our Contribution

Chen *et al.* [3] claim that vertex guarding a monotone polygon is NP-hard, however the details of their proof were omitted and still to be verified. Krohn and Nilsson [13] show that vertex guarding a monotone polygon is NP-hard. However, a proof showing NP-hardness for interior guards does not immediately follow from that claim. Since guards can be placed anywhere inside the polygon for interior guarding, moving a guard too far away from a vertex causes the reduction to fail. This is because too much of the polygon is seen by this guard.

Guarding a monotone polygon is very similar to the terrain guarding problem. The question of whether or not terrain guarding was NP-hard was an open problem for many years. Recently, the terrain guarding problem was shown to be NP-hard by King and Krohn [12]. Despite the similarities of guarding terrains and monotone polygons, the NP-hardness result for terrain guarding does not imply interior guarding a monotone polygon is NP-hard. In order to obtain a hardness result for interior guarding a monotone polygon, additional observations had to be made about the properties of monotone polygons. In doing so, we have developed a different reduction from Monotone 3SAT. Despite the very simple structure of a monotone polygon, we were able to create a new, intricate gadget that allows us to force guards to be placed at very specific locations.

The remainder of this paper is organized as follows. Section 2 contains the relevant section from [13] which shows vertex guarding a monotone polygon is NP-hard. Section 3 describes how to modify the reduction from Section 2 to show NP-hardness for interior guarding a monotone polygon. Section 4 provides a conclusion and future work.

2 Vertex Guarding is NP-hard

In this section, we will show that vertex guarding a monotone polygon is NP-hard. The reduction is from *Monotone 3SAT* (M3SAT) [10, page 259 (problem L02)]. An M3SAT instance $(\mathcal{X}, \mathcal{C})$ contains a set of Boolean variables, $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ and a set of clauses, $\mathcal{C} = \{c_1, c_2, ..., c_m\}$. Each clause contains three literals, $c_i = x_j \vee x_k \vee x_l$, a *positive clause*, or $c_i = \bar{x}_j \vee \bar{x}_k \vee \bar{x}_l$, a *negative clause*, for $1 \le j, k, l \le n$. An M3SAT instance is satisfiable if a satisfying truth assignment for $\mathcal{C}$ exists such that all clauses c_i are true.

We show that any M3SAT instance is polynomially transformable to an instance of vertex guarding a monotone polygon. We construct a monotone polygon P from the M3SAT instance such that P is guardable by K or fewer guards if and only if the M3SAT instance is satisfiable. We first present some basic gadgets to show how the polygon is constructed. We then connect these gadgets together to create a polygon.

Starting Pattern: The lower boundary of the polygon is divided into two parts, the left and the right sides. The first gadgets on the left side are the *starting patterns*. The starting patterns are shown to the left in Figure 1. *In each pattern, the bottom of the downward spike $b(x)$ is the distinguished vertex of the pattern.* This area is only seen by x and $\bar{x}$ and must be guarded by one of these two vertices. This pattern appears along the left side of the lower boundary of the monotone polygon a total of n times, one corresponding to each variable.

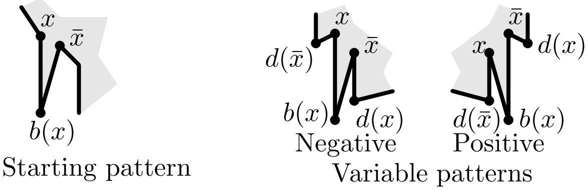

Figure 1: The different types of variable patterns.

Variable Pattern: On the left and the right side of the lower boundary we have *variable patterns* that verify the assigned truth value of each variable. This pattern is shown to the right in Figure 1. Once again, the bottom of the spike at $b(x)$ must be guarded by either x or $\bar{x}$. The pattern has additional distinguished vertices that we call *ledges* $d(x)$ and $d(\bar{x})$ that must both be seen and this is what forces the choice of guard placement at either x or $\bar{x}$.

Figure 2 shows how the starting patterns are connected to variable patterns. If we choose x_j in the starting pattern, we are forced to continuing to choose x_j in each of subsequent variable patterns. If we at some variable pattern would choose $\bar{x}_j$ instead of x_j, the ledge $d(\bar{x}_j)$ is not seen. Similarly, if we in the starting pattern choose $\bar{x}_j$, we are, by the same argument, forced to continuing to choose $\bar{x}_j$ in each of subsequent variable patterns.

Clauses: For each clause c in the boolean formula, there is a sequence of variable patterns $x_1, \ldots, x_n$ along either the left or the right side of the lower boundary and a clause pattern along the upper boundary of the polygon. On the left side of the lower boundary the variable pattern sequence corresponds to negative clauses, on the right side to positive clauses.

The clause pattern on the upper boundary consists of three vertices in an upward spike such that the top vertex of the spike is only seen by the variable patterns corresponding to the literals in the clause; see Figure 3. We denote the top vertex of the spike by c to correspond to the clause.

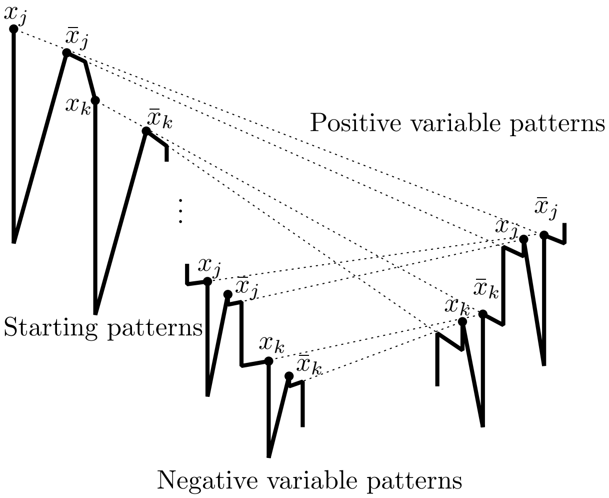

Figure 2: Variable patterns transferring logical values.

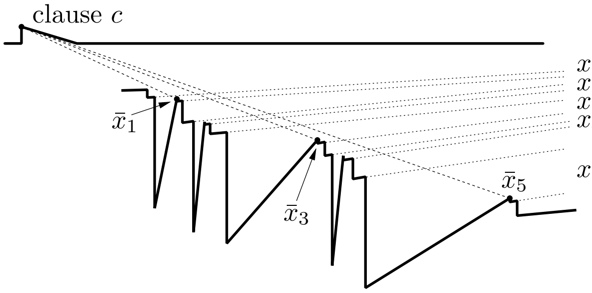

Figure 3: A variable pattern sequence with its clause spike.

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 right side, corresponding to positive clauses, and variable patterns on the left side, corresponding to negative clauses, of the lower boundary. Truth values do not change in the mirroring process since a variable x_j in clause c_i only sees the ledge $d(x_j)$ in the next variable pattern and none of the other ledges. Similarly $\bar{x}_j$ only sees ledge $d(\bar{x}_j)$ in the next variable pattern; see Figure 2.

In the example of Figure 3 the M3SAT clause corresponds to $c = \bar{x}_1 \vee \bar{x}_3 \vee \bar{x}_5$. Hence, a vertex guard placement that corresponds to a truth assignment that makes c true, will have at least one guard on $\bar{x}_1$, $\bar{x}_3$ or $\bar{x}_5$ and can therefore see vertex c without additional guards.

We still have variables x_2 and x_4 in the clause, however none of them or their negations see the vertex c. 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 $4n + (6n + 4)m + 2$ vertices. Each starting variable pattern having four vertices, each variable pattern six vertices, the clause spike consists of three vertices plus one blocking vertex at the start of each clause sequence on the lower boundary and the two leftmost and rightmost points of the polygon.

Consider an M3SAT instance $(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee \bar{x}_5) \wedge (x_3 \vee x_4 \vee x_5)$. Figure 4 shows how this instance is transformed into a monotone polygon and a placement of guards corresponding to the satisfying truth assignment $x_1 = x_2 = x_4 = x_5 = \mathit{false}$, $x_3 = \mathit{true}$.

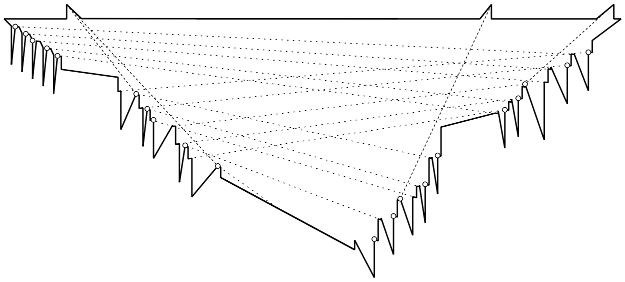

Figure 4: Example reduction of $(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee \bar{x}_5) \wedge (x_3 \vee x_4 \vee x_5)$. Points with white centers mark the guards.

Exactly $K = n(m + 1)$ guards are required to guard the polygon since there are K bottom vertices $b(x_j)$ at downward spikes and no vertex in the polygon can see more than one such $b(x_j)$ vertex.

If the M3SAT instance is satisfiable, then we place guards 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.

Suppose we have a vertex guard cover of size exactly K. Since each bottom spike $b(x_j)$ is guarded there is a guard at one of x_j, $\bar{x}_j$, or $b(x_j)$ itself. They together make up K guards so there can be no other guards. Since each clause vertex c_i is also seen, we can establish which of the guards see this vertex and deduce a satisfying truth assignment from this guard placement. We have proved the following theorem.

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

3 Interior Guarding is NP-hard

The hardness result for vertex guarding a monotone polygon does not immediately generalize to interior guards. For example, a guard placed slightly above a variable pattern can ruin the mirroring of truth assignments because this guard would see too many ledges. The next section introduces a modified variable pattern which forces the potential guard locations to be very close to the guard locations from Section 2.

3.1 Modified Variable Pattern

The following definition is used in this section: let *VP(p)* denote the visibility polygon of P from the point p, i.e., the set of points in P that can be connected with a line segment to p without intersecting the outside of P.

Modified Variable Pattern: Similar to the variable pattern introduced in Section 2, this pattern is used to verify the assigned truth value of each variable. It is important to note that this modified pattern does not move the x or $\bar{x}$ vertices. This pattern replaces the distinguished vertex $b(x)$ at the bottom of the spike with two distinguished vertices, namely $b(x)$ and $b(\bar{x})$; see Figure 5. This pattern also introduces six new distinguished vertices which are placed directly above the original variable pattern on the top of the polygon. We will call these eight new vertices *variable distinguished vertices*. We will call the six new vertices on the top of the polygon *upper distinguished vertices*. Figure 5 shows the complete modified variable pattern. Each of the upper distinguished vertices can see at most two guards from the following set: $\{x, \bar{x}, b(x), b(\bar{x})\}$.

Lemma 2 *Two guards are necessary and sufficient to see all of the variable distinguished vertices in a modified variable pattern.*

Proof. At least two guards are needed to see all of the distinguished vertices of this modified variable pattern. $VP(1) \cap VP(6) = \emptyset$; see Figure 5. Therefore, to see all upper distinguished vertices, two guards are necessary.

Let us assume that the ledge $d(x)$ is seen from a previous variable pattern. Two guards are sufficient for seeing all of the following variable distinguished vertices: $\{1, 2, 3, 4, 5, 6, b(x), b(\bar{x})\}$ and the unseen ledge $d(\bar{x})$. A possible solution would be to place a guard at x and also at $b(x)$. x would see $\{1, 2, b(\bar{x}), d(\bar{x})\}$ and $b(x)$ would see $\{3, 4, 5, 6\}$. If we assume that $d(\bar{x})$ was seen from a previous variable pattern, then a possible solution would be to place a guard at $\bar{x}$ and $b(\bar{x})$. □

Corollary 3 *We need at least* $K = 2n(m + 1)$ *guards to see all of the variable distinguished vertices in* P.

All potential guard locations for the upper distinguished vertices are located inside this vertical "strip" as shown in Figure 5. In other words, no guard placed to the left of *VP(1)* and no guard placed to the right of *VP(6)* can see any of the upper distinguished vertices. Said another way, *no guard can see upper distinguished vertices in more than one modified variable pattern.*

We will now show that if all of the variable distinguished vertices are seen, 2 guards must be placed at either $(x, b(x))$ or at $(\bar{x}, b(\bar{x}))$. Consider the horizontal line L drawn in Figure 6. L is split up into several

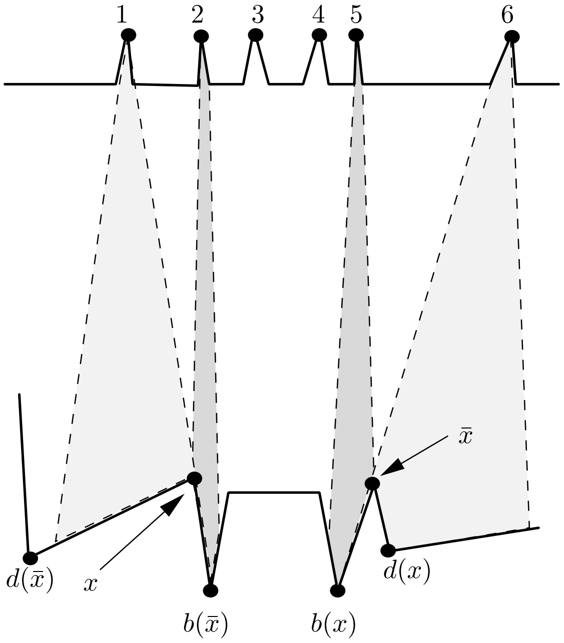

Figure 5: A complete modified variable pattern. Point 1 sees $\{x, b(\bar{x})\}$. Point 2 sees $\{x, b(\bar{x})\}$. Point 3 sees $\{b(\bar{x}), b(x)\}$. Point 4 sees $\{b(\bar{x}), b(x)\}$. Point 5 sees $\{b(x), \bar{x}\}$. Point 6 sees $\{b(x), \bar{x}\}$. The visibility polygons for points 1, 2, 5 and 6 are displayed.

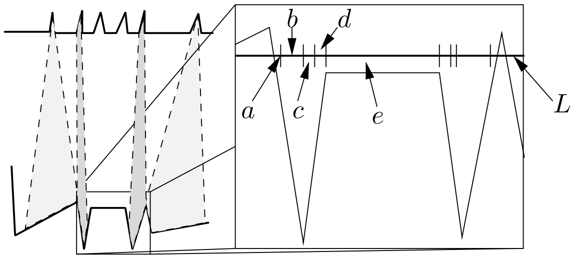

Figure 6: A horizontal line L such that no guard placed on or above L sees more than two upper variable distinguished points.

segments. The endpoints of these segments are where the edges of the visibility polygons of $1, 2, \ldots, 6$ hit L. Any guard placed on or above L will see at most 2 upper distinguished vertices. A guard placed on segment a will see vertices $\{1, 2\}$. A guard placed on segment b will see only the vertex $\{2\}$. A guard placed on segment c will see vertices $\{2, 3\}$. A guard placed on segment d will see only the vertex $\{3\}$. A guard placed on segment e will see vertices $\{3, 4\}$. The remaining segments are not named but these are the upper distinguished vertices they see in order from left to right: $\{\{4\}, \{4, 5\}, \{5\}, \{5, 6\}\}$. No guards placed above L will be able to see more upper distinguished vertices than a guard placed on L because of the monotonicity of the polygon. Since there are no obstacles, such a guard could be moved down to L without losing visibility of any of the upper distinguished vertices. It is important to note that no guard placed on L sees any of the ledges $d(x)$ or $d(\bar{x})$. A guard must be placed above L in order for the unseen ledge to be seen.

Lemma 4 *No one guard can see more than 4 upper distinguished vertices.*

Proof. We compare the visibility polygons of 1 and 2 with the visibility polygons of 5 and 6; see Figure 5. $(VP(1) \cup VP(2)) \cap (VP(5) \cup VP(6)) = \emptyset$. Any guard that sees 1 or 2 cannot see 5 or 6. For a guard to see more than 4 upper distinguished vertices, such a guard must see all but 1 of the upper distinguished vertices. This is not possible. □

Lemma 5 *For all variable distinguished vertices and one ledge from $\{d(x), d(\bar{x})\}$ of a modified variable pattern to be seen, guards must be placed at $(x, b(x))$ or $(\bar{x}, b(\bar{x}))$.*

Proof. Referring to Figure 5, let us assume that $d(\bar{x})$ is seen by a guard placed in a previous variable pattern. A guard must be placed somewhere in P to see $d(x)$. However, any guard that sees $d(x)$ must be placed above L and therefore can see at most 2 upper distinguished vertices. Because of Lemma 2, if we want to see all of the upper distinguished vertices by using only 2 guards, we are forced to place a second guard below L. To determine where such a guard must be placed, consider the $VP(d(x))$. No point in the visibility polygon of $d(x)$ sees upper variable distinguished points 1 or 2. From Lemma 4, a guard that sees 1 or 2 cannot see 5 or 6. Therefore, if we are only allowed to place 2 guards, a guard that sees $d(x)$ must see 5 and 6. This guard location must be placed at $\bar{x}$. Consider a vertical line through $\bar{x}$. A guard placed just slightly to the left of this vertical line will not see 6. A guard placed slightly to the right of this vertical line will not see 5. Draw a horizontal line through $\bar{x}$. If the guard is moved slightly above this line, neither 5 nor 6 will be seen. Therefore a guard must be placed at $\bar{x}$. Placing a guard at $\bar{x}$ leaves the following upper distinguished vertices unseen: $\{1, 2, 3, 4\}$. For these vertices to be seen, a guard must be placed below the line L. Such a guard placement will not affect the mirroring of variable truth values. The only region that sees the remaining upper variable distinguished points is $b(\bar{x})$. Similar arguments can be made when $d(x)$ has already been seen and the only solution is to place guards at x and $b(x)$. □

The introduction of this modified pattern allows us to force guards to be in certain positions. The forced positions are the same guard locations from the construction in Section 2. The original construction remains unchanged with 2 exceptions. The variable pattern is replaced with a modified variable pattern. A modified starting pattern replaces the original starting pattern. A modified starting pattern is identical to a modified variable pattern without the $d(x)$ and $d(\bar{x})$ ledges. Because there are no ledges, either $(x, b(x))$ or $(\bar{x}, b(\bar{x}))$ can be chosen. This starting choice will then affect guard locations for all future modified variable patterns.

3.2 Entire Polygon is Seen

The previous subsection showed that all distinguished vertices are seen but it does not immediately follow that the entire polygon is seen. We will make a few observations to show that the entire polygon is seen. We will break the polygon into smaller pieces and show that each of those pieces is seen by some subset of the guards already placed. It can easily be seen that every point in the interior of the polygon must fit into at least one of these categories and therefore must be seen. The numbered boxes in Figure 7 correspond to the area we are discussing in the list below.

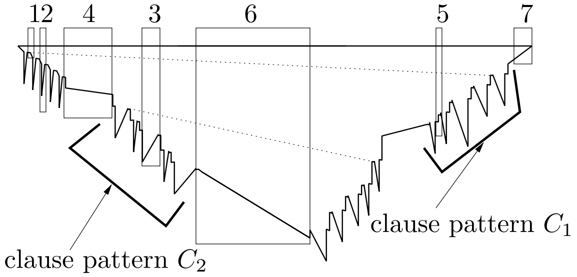

Figure 7: A simplified diagram showing different areas of the polygon.

1. All of the polygon in the vertical strip between a modified starting pattern for x_i and x_{i+1} is seen by a guard placed at one of $\{x_i, \bar{x}_i\} \in C_1$.
2. All of the polygon in a vertical strip containing a modified starting pattern for x_i is seen by either guards placed at $(x_i \in C_0, b(x_i) \in C_0, x_i \in C_1)$ or $(\bar{x}_i, b(\bar{x}_i))$.
3. A *clause pattern* is a grouping of modified variable patterns that determine whether a clause is satisfiable or not. A clause pattern always contains n modified variable patterns. Let us number clause patterns from top to bottom in the order shown in Figure 7. Consider any variable x_i in any clause C_j where $j > 0$ and j is even. The vertical strip containing the modified variable pattern is seen by either guards placed at $(x_i \in C_j, b(x_i) \in C_j, x_i \in C_{j-1})$ or $(\bar{x}_i \in C_j, b(\bar{x}_i) \in C_j, \bar{x}_i \in C_{j-1})$. Similar arguments can be made in the cases where j is odd. One should consider the initial grouping of modified starting patterns as C_0 when thinking about clause pattern C_1.

4. Consider 3 consecutive clause patterns C_{i-1}, C_i, C_{i+1}. The area of the polygon located in a vertical strip between C_{i-1} and C_{i+1} can be seen by a guard placed at either $(x_1, \bar{x}_1) \in C_i$.

5. Consider 2 consecutive modified variable patterns x_i, x_{i+1} in some clause C_i where i is odd. The vertical strip between them is seen by a guard placed at either of $(x_{i+1} \in C_{i-1}, \bar{x}_{i+1} \in C_i)$. Similar arguments can be made if i is even.

6. Consider the vertical strip between the modified variable pattern for $x_n \in C_{m-1}$ and the modified variable pattern for $x_n \in C_m$. In other words, this is the vertical strip in the "middle" of the polygon. This strip is seen by either $\bar{x}_n \in C_{m-1}$ or $x_n \in C_m$. Similar arguments can be made if C_m is to the left of C_{m-1}.

7. Lastly, consider the upper corners of the polygon. A guard placed at either $(x_1, \bar{x_1}) \in C_0$ will see both of these areas.

Using the observations in this section that show the entire polygon is seen and the modified patterns which force guards to be in specific locations along with the hardness result for vertex guarding from [13] with the new $K = 2n(m+1)$ given in Corollary 3, we have proved the following theorem.

Theorem 6 *Finding the smallest interior guard cover for a monotone polygon is NP-hard.*

4 Conclusions and Future Work

We have proved that interior guarding a monotone polygon is NP-hard. Open problems include improving the approximation bounds for monotone polygons. Since a PTAS has not yet been found for guarding a monotone polygon, an interesting open question is whether or not one exists. If a PTAS cannot be found, can guarding a monotone polygon be shown to be APX-hard? Other open problems include finding approximation algorithms for other classes of polygons and ultimately finding better approximations for guarding a simple polygon in general.

Acknowledgments

The authors would like to thank Kasturi Varadarajan and Matt Gibson for their valuable comments, discussions and suggestions.

References

[1] A. Aggarwal. *The art gallery theorem: its variations, applications and algorithmic aspects.* PhD thesis, 1984.

[2] B. Brodén, M. Hammar, and B. J. Nilsson. Guarding lines and 2-link polygons is APX-hard. In *In 13th Canadian Conf. on Computational Geometry*, pages 45–48, 2001.

[3] D. Z. Chen, V. Estivill-Castro, and J. Urrutia. Optimal guarding of polygons and monotone chains. In *Proceedings of the 7th Canadian Conference on Computational Geometry*, pages 133–138, 1995.

[4] V. Chvátal. A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory Series B*, 18:39–41, 1975.

[5] A. Deshpande, T. Kim, E. D. Demaine, and S. E. Sarma. A pseudopolynomial time O(log n)-approximation algorithm for art gallery problems. In F. K. H. A. Dehne, J.-R. Sack, and N. Zeh, editors, *WADS*, volume 4619 of *Lecture Notes in Computer Science*, pages 163–174. Springer, 2007.

[6] A. Efrat and S. Har-Peled. Guarding galleries and terrains. *Inf. Process. Lett.*, 100(6):238–245, 2006.

[7] S. Eidenbenz. Inapproximability results for guarding polygons without holes. In *Lecture Notes in Computer Science*, pages 427–436. Springer, 1998.

[8] S. Eidenbenz. *Inapproximability of Visibility Problems on Polygons and Terrains.* PhD thesis, 2000.

[9] S. Fisk. A short proof of Chvátal's watchman theorem. *Journal of Combinatorial Theory Series B*, 24:374+, 1978.

[10] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[11] S. Ghosh. Approximation algorithms for art gallery problems. In *Proc. Canadian Information Processing Society Congress*, 1987.

[12] J. King and D. Kirkpatrick. Improved approximation for guarding simple galleries from the perimeter. *Discrete Comput. Geom.*, 46(2):252–269, Sept. 2011.

[13] E. Krohn and B. J. Nilsson. Approximate guarding of monotone and rectilinear polygons, 2012. To Appear in Algorithmica.

[14] D. T. Lee and A. K. Lin. Computational complexity of art gallery problems. *IEEE Trans. Inf. Theor.*, 32(2):276–282, March 1986.

[15] B. Nilsson. *Guarding Art Galleries — Methods for Mobile Guards.* PhD thesis, 1995.

[16] B. J. Nilsson. Approximate guarding of monotone and rectilinear polygons. In L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *ICALP*, volume 3580 of *Lecture Notes in Computer Science*, pages 1362–1373. Springer, 2005.

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