The Inapproximability of Illuminating Polygons by α-Floodlights

Ahmed Abdelkader* Ahmed Saeed † Khaled Harras‡ Amr Mohamed§

Abstract

We consider variants of the art gallery problem where guard visibility is limited to a certain angular aperture α. We show that the problem is NP-hard even when guards can be located in the interior of the polygon. We then proceed to prove that both this problem and its vertex variant, where guard placement is restricted to the vertices of the polygon, are APX-hard.

We observe that earlier constructions for such results in art gallery problems with 360° guards, usually required them to cover few specific elements. We exploit this by carefully updating the construction to replace 360° guards with α-floodlights. Similar transformations may be applicable to other constructions in traditional art gallery theorems, which is of independent interest.

1 Introduction

The study of art gallery problems is a rich area in geometry with a variety of combinatorial bounds, algorithms and hardness results [20, 21, 24]. While we are only concerned with floodlight illumination, we build upon the construction of Lee and Lin [18] through the work of Eidenbenz, Stamm and Widmayer [11]. This construction was used in [18] to show that deciding the minimum number of guards in a polygon without holes is NP-hard. The construction was refined in [11] to further show that there exists a constant $\epsilon > 0$, such that no polynomial time algorithm can guarantee an approximation ratio of $1 + \epsilon$ unless $P = NP$. In other words, the problem is APX-hard, as was obtained independently in [5]. Exact [10], approximate [17, 4] and heuristic [1] solutions have been developed.

Most of the aforementioned work focused on omnidirectional guards, i.e., guards with 360° range of vision. However, many recent applications in sensor networks and smart surveillance are more concerned with sensors that have limited sensing ranges. This leads us to study the α-floodlight illumination variant of the art gallery problem.

The first documented floodlight illumination problem is perhaps the stage illumination problem (SIP), presented in 1992 by Urrutia [7, 3]. Given a line segment, i.e., the stage, together with a set of floodlights, of known origins and angles, decide whether the floodlights may be rotated to illuminate the stage. The original SIP remained unsolved for more than ten years [7] and was later shown to be NP-complete [16], even under two different restrictions. Variants of the SIP and other problems related to floodlights include [23, 22, 6, 13, 9].

Estivill-Castro and Urrutia [14] asked whether computing the minimum set of covering α-floodlights is NP-hard. Indeed, Bagga, Gewali and Glasser [2] showed that the vertex Floodlight Illumination Problem (FIP) is NP-hard, for $0 < \alpha \leq 360°$. The status of the point variant, where floodlights can be placed anywhere inside the polygon, remains open.

The renewed interest in this classical problem is motivated by several coverage problems in visual and directional sensor networks. α-floodlights, which restrict visibility to a certain angular aperture α, are particularly appealing as a better model for sensors with a limited sensing range, e.g., cameras.

We define α-floodlights and the two polygon illumination problems at hand. We also define distinguished arrows [11], which will be used in some of our arguments.

Definition 1 *An α-floodlight at point p, with orientation θ, is the infinite wedge $W(p, \alpha, \theta)$ bounded between the two rays $\overrightarrow{v_l}$ and $\overrightarrow{v_r}$ starting at p with angles $\theta \pm \frac{\alpha}{2}$. In a polygon P, a point q belongs to the α-floodlight if $\overline{pq}$ lies entirely in both P and $W(p, \alpha, \theta)$.*

Definition 2 *A distinguished arrow (DA) is an infinitesimal ray along an edge of the polygon such that any α-floodlight that covers it must be placed in a prespecified region, i.e., the interior of a gadget or a cone.*

Definition 3 *The Vertex Floodlight Illumination Problem (FIP) [2] Given a simple polygon P with n sides, a positive integer m and angular aperture α, determine if P can be illuminated by at most m α-floodlights placed only on the vertices of P with at most one α-floodlight per vertex.*

Definition 4 *The Point Floodlight Illumination Problem (PFIP) Given a simple polygon P with n sides, a positive integer m and angular aperture α, determine if P can be illuminated by at most m α-floodlights placed in the interior of P.*

*Department of Computer Science, University of Maryland, College Park, akader@cs.umd.edu

†School of Computer Science, Georgia Institute of Technology, ahmed.saeed@gatech.edu

‡School of Computer Science, Carnegie Mellon University, kharras@cs.cmu.edu

§Department of Computer Science and Engineering, Qatar University, amrm@ieee.org

In both FIP and PFIP, floodlights can be oriented in any direction as long as P is illuminated. However, to verify a given solution in polynomial time, we cannot deal with arbitrary orientations. To remedy this, [2] introduced a *flushing* restriction which brings FIP into NP. As our main result uses a gap-preserving reduction from 5-OCCURRENCE-MAX-3-SAT (FOM-3SAT), which we define below, a similar restriction will be necessary. When the restriction is in effect, we prefix the problem name with the letter R. We define flushing as follows:

Definition 5 *An α-floodlight is flush with the vertices of the polygon P if at least one of $\overrightarrow{v_l}$ or $\overrightarrow{v_r}$ passes through some vertex of P, different from p, such that θ is determined implicitly.*

Definition 6 *(FOM-3SAT) Given a boolean formula Φ in conjunctive normal form, with m clauses and n variables, 3 literals at most per clause, and 5 literals at most per variable, find an assignment of the variables that satisfies as many clauses as possible.*

We develop a construction for point α-floodlights and outline how to adapt it for vertex α-floodlights. This allows us to obtain the following.

Theorem 7 *PFIP is NP-hard.*

Theorem 8 *RPFIP is NP-complete.*

Theorem 9 *RFIP is APX-hard.*

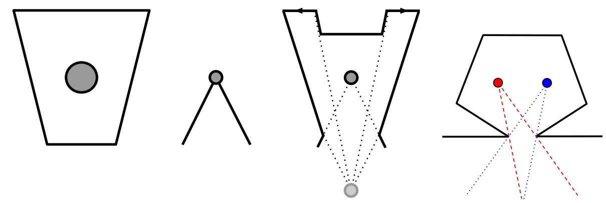
Theorem 10 *RPFIP is APX-hard.*

The construction in [2] utilizes beam machine gadgets [8] to control the visibility of the α-floodlight guards in FIP. In Section 2, we develop beam machines for point α-floodlights in addition to the *Point α-Floodlight Gadget (PFG)* to have corresponding tools in PFIP. This immediately yields Theorems 7 and 8 by plugging the new gadgets in the construction from [2].

In Section 3, we start by examining the construction of [11] and describe how 360° guards can be replaced with α-floodlights without changing the essence of the construction. The main observation is that while guards can see in all directions, the construction only requires them to guard few specific elements or regions. We exploit this to carry over the construction of [11] from the 360° guard setting to the α-floodlight setting, and carry along the result obtained in the former to get Theorems 9 and 10.

2 Point α-floodlights

We develop the *Point α-Floodlight Gadget (PFG)* and use it to create a Point α-Floodlight Beam Machine (BM). Then, we discuss the extension of [2] using the new BM to obtain the first proof of Theorem 7.



(a) Trapezoid, *vertex*-PFG, PFG. (b) ABM.

Figure 1: PFGs and Abstract Beam Machine (ABM).

2.1 PFG

The building block of our construction is the Point Floodlight Gadget (PFG) in Figure 1a. The PFG is attached to the polygon through its mouth and extrudes outside forming a cavity. The cavity is the union of two overlapping wedges. Both wedges share the same axis with one outward wedge looking into the cavity and one inward wedge extending into the interior of the polygon. The extrusion includes two ears which require an α-floodlight guard at the apex of the outward wedge to cover their pockets. Depending on how the PFG is used in a larger gadget, the PFG can be configured such that a second α-floodlight at the apex of the inward wedge is either optional or obligatory. Note that both α-floodlights would satisfy the flushing condition.

When using vertex α-floodlights, a PFG equivalent is just an ear vertex. We refer to both as PFGs and use a trapezoidal symbol in our schematic diagrams as a placeholder for the appropriate PFG. Figure 1a demonstrates the correspondence.

2.2 Beam Machines

Beam machine gadgets were introduced in [8] which showed the hardness of finding a minimum convex cover for a given polygon. The beam machine (BM) is a butterfly shaped extrusion that attaches to the polygon through a mouth. The internal design of the BM requires 4 convex polygons to cover the BM itself and allows one of two slim polygons, i.e., *beams*, to *shoot* into the interior of the polygon in two different directions. The construction needed such shooting beams to cover other parts of the polygon, i.e., *dents*, which corresponds to the satisfaction of boolean clauses by the assignment of their literals. This enabled a reduction from 3SAT to show the problem is NP-hard.

BMs were reused in [2] to force the inclusion of one of two vertex α-floodlights in a construction similar to the one in [8]. The BMs in [2] required 3 vertex α-floodlights to cover their interior and could shoot *light beams* to illuminate their dents. Again, this enabled a reduction from 3SAT to show that FIP is NP-hard.

A BM can be stretched and skewed to control the beams, which need not be symmetric. We abstract BMs as an extrusion with two potential points for α-floodlight placement, as in Figure 1b. We identify `True` and `False` with the red and blue colors, respectively.

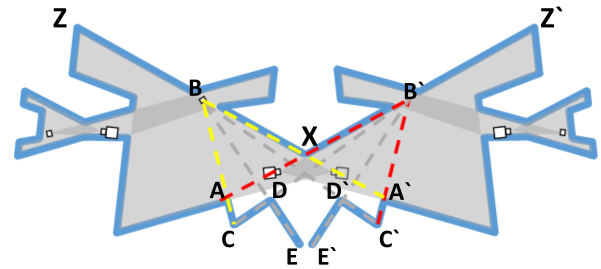
We can now develop a BM for point α-floodlights. The BM is basically one big PFG to create the two wings of the butterfly plus one PFG on each side to extrude two cavities on the upper sides of the wings. All 3 PFGs require 2 floodlights each, e.g. the big PFG needs one guard for the edge denoted Z and another for Z' as in Figure 2. The mouth is designed to require one α-floodlight at one of the two cavities denoted B and B', which results in the two BM configurations. We identify the red and blue points of the ABM with B and B', respectively. The BM requires 7 α-floodlights which all satisfy the flushing condition by construction.

2.3 Updating the reduction by Bagga et al. [2]

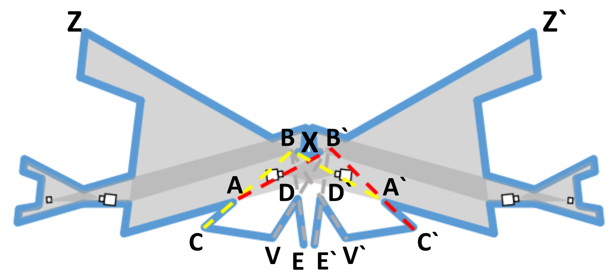
Using the point α-floodlight BM and PFG, it is straightforward exercise to update the construction in [2]. The Background of Variable Generator requires 4 PFGs at vertices $\{v_4, v_{11}, v_{13}, v_{20}\}$ where the inward wedge of the PFGs at either v_4 or v_{20} is used to specify an assignment for the variable, for a total of 7 point α-floodlights. Each literal is represented by a BM and the final polygon requires a single PFG contributing 2 additional α-floodlights. Given a `3SAT` instance with m clauses and n variables, the `PFIP` instance output by the reduction can be covered using $21m+7n+2$ point α-floodlights iff the `3SAT` instance is satisfiable. This yields Theorem 7. As all our gadgets satisfy the flushing condition, Theorem 8 follows as well. These two theorems also follow from the construction presented in the next section.

3 Reusing the construction of Eidenbenz et al. [11]

[18] showed that determining the minimum number of guards to cover an art gallery is NP-hard. They presented a construction for vertex guards and showed how it can be modified to yield similar results for the edge and point variants. [11] followed the lines of the reduction in [18] to describe a gap-preserving reduction from the MAXSNP-complete `FOM-3SAT`, which shows these problems are APX-hard. In doing so, [11] gives a detailed construction for all gadgets to guarantee certain properties necessary for the gap-preserving reduction. A similar approach was applied to the construction in [8] for the problem of finding a minimum convex cover to show it is APX-hard as well [12]. Later on, [15] assigned weights to the edges of the construction of [11] to show that maximizing the guarded boundary of an art gallery is APX-hard. For that problem, a constant-factor approximation was developed earlier [19], so the problem is actually APX-complete.



(a) $\alpha \leq 90°$.



(b) $\alpha > 90°$.

Figure 2: BMs for different values of α.

We briefly recall the construction of [11] before we list our observations and the modifications we apply.

3.1 Recalling the gadgets of [11]

Literal pattern for literal l is a triangular extrusion with a spike that requires one *literal guard* at one of two locations called $T^{lit}(l)$ and $F^{lit}(l)$.

Clause pattern for clause c_i uses 3 literal patterns $l_j(c_i)$ such that it can be covered iff at least one literal is assigned a guard at its $T^{lit}(l_j(c_i))$.

Variable pattern for variable x_k has two quadrilateral extrusions called *legs* and a *tail* that requires one *variable guard* at one of two locations called $T^{var}(x_k)$ and $F^{var}(x_k)$.

Ear pattern is a cavity at the top-left corner of the final polygon which hosts one *ear guard* w that covers the ear itself plus the background quadrilateral supporting the gadgets which define the polygon and all left and right legs of the variable patterns.

Spike pattern for a literal is a tiny extrusion in the legs of its variable pattern to ensure consistent truth assignments. The spike pattern is a cone that, in a *canonical solution*, must be covered by either the variable guard of the leg containing it or the literal guard tied to it. Positive and negative literal guards are tied to their variable by a spike in the appropriate leg.

3.2 Observations and modifications

Spike patterns are only a subset of the guard's visibility polygon. A guard can typically see a much larger area containing the spike pattern. When using α-floodlights, located in a BM, it is only necessary that the spike extrusion is covered by the beam the floodlight shoots through the BM's mouth.

T^{lit} and F^{lit}. The only functions these two locations may serve are: (1) Cover the interior of the literal gadget. (2) Cover the corresponding spikes in the variable gadget. (3) Satisfy the clause. When using BMs, (1) will be taken care of by the design of the BM. (2) and (3) turn out to be difficult to achieve using a single α-floodlight. To remedy this, we use two *coupled* BMs per literal to collectively support two configurations corresponding to the assignment of the literal's variable. Figure 3 illustrates the coupling technique. Basically, we copy the `TRUE` signal communicated through the spike in the variable pattern by introducing a dent. A literal can satisfy the clause iff the BM at the top is allowed to shoot its left beam. This would only work if the dent is covered by the BM to the right which only happens iff this BM is allowed to shoot its `TRUE` beam. In addition, we ensure that no single floodlight can cover two such dents.

This allows us to redesign the clause pattern as in Figure 4. Satisfying a clause corresponds to illuminating the dent containing the DA denoted by 2. This dent is adjusted such that it may not be illuminated by a floodlight in any of the spike patterns of the 3 literals of that clause. A single PFG at the top left corner covers the background quadrilateral of the clause pattern and DA-1, which only leaves uncovered the interiors of the BMs, their dents and DA-2.

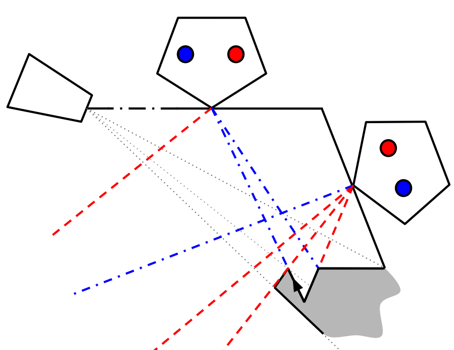


Figure 3: Coupled BMs. Dent must be covered.

Locating T^{lit} and F^{lit}. These two vertices of the literal pattern are at a distance controlled by two arbitrary constants [11]: (1) Distance between T^{lit} and s_6. (2) Distance between s_6 and the vertical line v'. They can be made arbitrarily close as required by the BM to enable shooting the beams to illuminate the corresponding spikes. Finally, we move these locations along the lines defined by the spike patterns to place the BMs on an oblique edge in the clause pattern to give it more flexibility to adjust all BMs and beams to cover their assigned targets. Note that we only generate a restricted class of the spike patterns constructed by the algorithm in [11], but otherwise we do not move them. This preserves the property that no 3 spike patterns of 3 different legs intersect in a common point per Lemma 1 in [11].

Switching T^{lit} and F^{lit}. The roles played by either of these two locations is determined by the spikes they are tied to, which depends on the literal being positive or negative. In addition, T^{lit} can satisfy the clause while F^{lit} cannot. To avoid changing the construction in [11] by much, we effectively exchange the roles of the guards at T^{lit} and F^{lit} such that F^{lit} is the location that can satisfy the clause. While this would not work for the literal pattern in [11], we will be replacing it anyway with a BM.

Moving F^{lit}. Due to the modifications we apply to the variable pattern, we identify F^{lit} with s_4 instead of s_5. We then move it along to find its location in the BM attached to the oblique edge. Again, while this does not make sense in the construction of [11], we are only interested in the coordinates produced for these vertices. In particular, we only need to make sure the spike patterns in the construction of [11] include the locations of the α-floodlights inside their literal BMs.

Limiting the required aperture T^{var} and F^{var}. Each of these two vertex guards is required to cover the variable pattern's tail in addition to the literal spikes in its leg. This implies the effective range of vision is bounded by the variable tail and the *lowest spike* in the leg. To make sure a single α-floodlight can cover both the variable tail and all the spikes in its leg, we require that literal patterns are far enough to the right from all variable patterns such that the lowest spike in any leg does not require an aperture larger than α. Adjusting the variable tail accordingly can be achieved by stretching the variable pattern as shown in Figure 5.

T^{var} and F^{var} for vertex α-floodlights. As we only assign one guard to either of these two locations, the cavity of the unassigned vertex-PFG, as in Figure 5, will need to be covered. This can be achieved by cutting off the left supporting edge of the vertex-PFG such that the cavity is covered by the ear guard. Note that the right supporting edge is still sufficient for the variable guards to satisfy the flushing condition.

The ear pattern and the final polygon. We replace the ear pattern with a PFG and stretch the polygon to include the background quadrilateral and the legs of all variable patterns in the PFG's inward wedge.

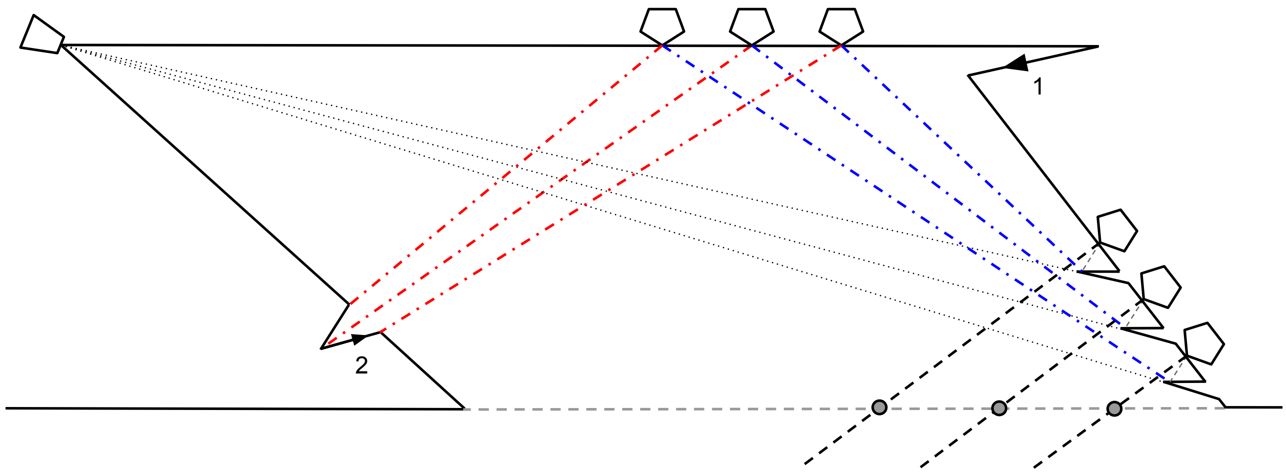


Figure 4: Clause Gadget. Circles highlight the neighborhoods of $T^{lit}(l_j(c_i))$ computed in [11].

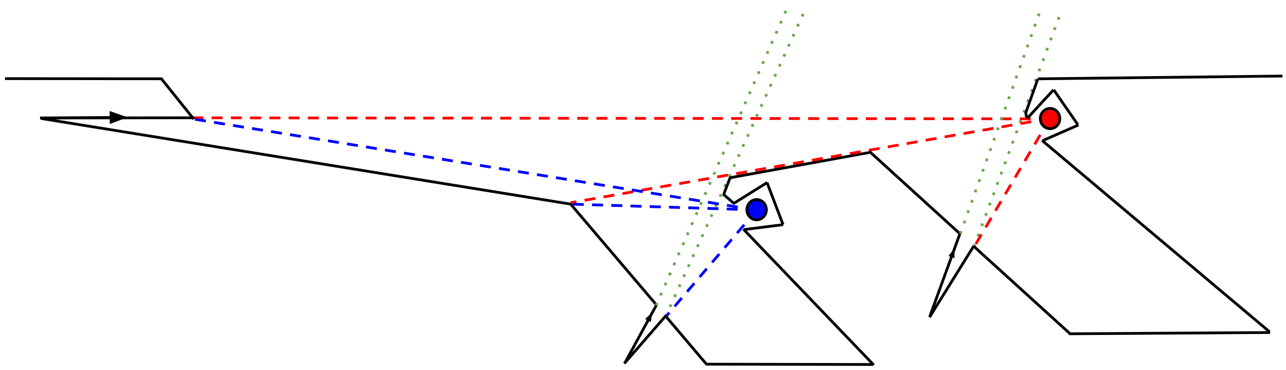


Figure 5: Variable Gadget. The spike to the left and the lowest spike in each leg must fit in the wedges of the PFGs.

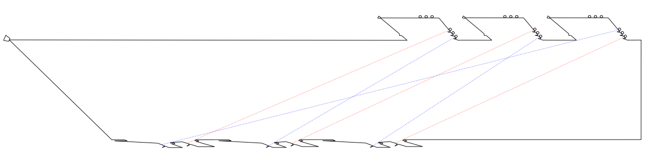


Figure 6: Rough sketch of the final construction.

4 Inapproximability results

Using this construction for PFIP, we get that the PFIP instance can be covered by $44m + 3n + 2$ point α-floodlights iff the FOM-3SAT instance is satisfiable. For FIP, the number is $19m + n + 1$. This provides an alternative proof that both problems are NP-hard. The *if* part is a straightforward mapping from Lemma 2 in [11], observing the number of α-floodlights required for each gadget. The *only if* part is obtained by observing that all variable patterns will have exactly one α-floodlight in such solutions, which yields a satisfying assignment.

Updating the construction of [11], per 3.2, preserves all its relevant properties. In particular, at most two spike patterns belonging to two different legs intersect.

Now, we may find an ϵ-approximate solution S to a given FOM-3SAT instance I by reducing it to an RFIP instance I', computing an ϵ'-approximate solution S' of I' and then transforming S' into S. We develop a transformation process similar to the one described in [11], which we could not fit here due to space constraints. This amounts to a gap-preserving reduction from FOM-3SAT to RFIP. Since the former is MAXSNP-complete, this shows RFIP is APX-hard.

As we managed to stay close to the construction in [11], we carry over a close equivalent of their Lemma 3 and Theorem 1. With that, unless $P = NP$, no polynomial time approximation algorithm for RPFIP can achieve an approximation ratio of

$$\frac{44m + 3n + 2 + \epsilon m}{44m + 3n + 2} = 1 + \frac{\epsilon m}{44m + 3n + 2} \geq 1 + \frac{\epsilon}{54}.$$

This yields Theorem 10. As pointed out in [11], since there will be no floodlights added in the transformation of a given solution of RFIP, we would get a slightly bigger constant for the inapproximability of RFIP than the constant of RPFIP and Theorem 9 follows.

5 Conclusion

In this paper, we resolved the hardness and inapproximability of two classical α-floodlight illumination problems for both vertex and point floodlights. We observed that many earlier constructions for 360° guards, only required guards to cover specific regions in the construction. We exploit this to present a structured update of such constructs to work for guards with limited angle of view. We gave two examples of this process by presenting APX-hardness proofs for vertex and point α-floodlight polygon illumination problems for simple polygons. A flushing restriction is introduced to avoid dealing with arbitrary orientations of floodlights and allow polynomial-time verification and gap-preserving reduction. We believe that similar approaches can be used to carry over more results for 360° guards to α-floodlights which can greatly help the ongoing work in sensor networks and smart surveillance.

Acknowledgement

This work was made possible by NPRP grant # 4-463-2-172 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

[1] Y. Amit, J. S. Mitchell, and E. Packer. Locating guards for visibility coverage of polygons. *International Journal of Computational Geometry & Applications*, 20(05):601–630, 2010.

[2] J. Bagga, L. Gewali, and D. Glasser. The complexity of illuminating polygons by alpha-flood-lights. In *CCCG*, pages 337–342. Carleton University Press, 1996.

[3] P. Bose, L. Guibas, A. Lubiw, M. Overmars, D. Souvaine, and J. Urrutia. The floodlight problem. *International Journal of Computational Geometry & Applications*, 7(01n02):153–163, 1997.

[4] A. Bottino and A. Laurentini. A nearly optimal algorithm for covering the interior of an art gallery. volume 44, pages 1048–1056. Elsevier, 2011.

[5] B. Brodén, M. Hammar, and B. J. Nilsson. Guarding Lines and 2-Link Polygons is APX-hard. In *CCCG*, 2001.

[6] M. Cary, A. Rudra, A. Sabharwal, and E. Vee. Floodlight illumination of infinite wedges. *Computational Geometry*, 43(1):23–34, 2010.

[7] F. Contreras, J. Czyzowicz, E. Rivera-Campo, and J. Urrutia. Optimal floodlight illumination of stages. In *Proceedings of the Fourteenth Annual Symposium on Computational Geometry*, SCG '98, pages 409–410, New York, NY, USA, 1998. ACM.

[8] J. C. Culberson and R. A. Reckhow. Covering polygons is hard. In *FOCS*, pages 601–611, 1988.

[9] J. Czyzowicz, S. Dobrev, B. Joeris, E. Kranakis, D. Krizanc, J. Maňuch, O. Morales-Ponce, J. Opatrny, L. Stacho, and J. Urrutia. Monitoring the plane with rotating radars. *Graphs and Combinatorics*, 31(2):393–405, 2015.

[10] A. Efrat and S. Har-Peled. Guarding galleries and terrains. *Information processing letters*, 100(6):238–245, 2006.

[11] S. Eidenbenz, C. Stamm, and P. Widmayer. Inapproximability results for guarding polygons and terrains. *Algorithmica*, 31(1):79–113, 2001.

[12] S. J. Eidenbenz and P. Widmayer. An approximation algorithm for minimum convex cover with logarithmic performance guarantee. *SIAM Journal on Computing*, 32(3):654–670, 2003.

[13] D. Eppstein, M. T. Goodrich, and N. Sitchinava. Guard placement for efficient point-in-polygon proofs. In *Proceedings of the twenty-third annual Symposium on Computational geometry*, SCG '07, pages 27–36. ACM, 2007.

[14] V. Estivill-Castro and J. Urrutia. Optimal floodlight illumination of orthogonal art galleries. In *CCCG*, pages 81–86, 1994.

[15] C. Fragoudakis, E. Markou, and S. Zachos. Maximizing the guarded boundary of an art gallery is APX-complete. *Computational Geometry*, 38(3):170–180, 2007.

[16] H. Ito, H. Uehara, and M. Yokoyama. Np-completeness of stage illumination problems. In *Discrete and Computational Geometry*, pages 158–165. Springer, 2000.

[17] J. King. Fast vertex guarding for polygons with and without holes. *Computational Geometry*, 46(3):219 – 231, 2013.

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

[19] E. Markou, C. Fragoudakis, and S. Zachos. Approximating visibility problems within a constant. In *3rd Workshop on Approximation and Randomization Algorithms in Communication Networks, Rome*, pages 91–103, 2002.

[20] J. O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, Inc., New York, NY, USA, 1987.

[21] T. Shermer. Recent results in art galleries [geometry]. *Proceedings of the IEEE*, 80(9):1384–1399, Sep 1992.

[22] A. Spillner and H.-D. Hecker. Minimizing the size of vertexlights in simple polygons. *Mathematical Logic Quarterly*, 48(3):447–458, 2002.

[23] W. Steiger and I. Streinu. Illumination by floodlights. *Computational Geometry*, 10(1):57–70, 1998.

[24] J. Urrutia et al. Art gallery and illumination problems. *Handbook of computational geometry*, pages 973–1027, 2000.

Appendix A: Additional Figures

We include additional figures to aid our description of the gadgets we created.

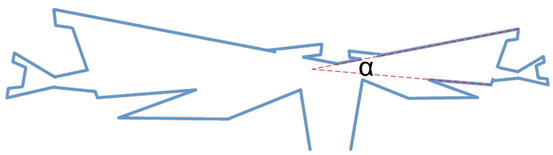


Figure 7: Demonstration of the flexibility of the BM.

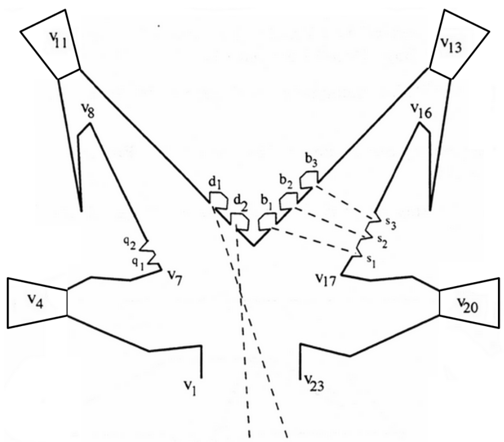


Figure 8: The BVG of [2] updated with PFGs.

Appendix B: α-Floodlights with $\alpha \geq \pi$

(Subsection 3.2) To motivate the intuition for the modifications we applied to the construction of [11], we informally discuss the case when $\alpha \geq \pi$.

Define $\texttt{RFIP}^{\geq\pi}$ and $\texttt{RPFIP}^{\geq\pi}$ to be the restriction of these α-floodlight illumination problems with $\alpha \geq \pi$. We can easily extend the construction in [11] to show similar inapproximability results for these two problems. The key idea is to ensure that all elements required to be covered by a given guard location lies in a half-plane defined by a line passing through this location.

We can ensure the ear guard only looks *down* by attaching its ear to the left edge, instead of the top one. For T^{lit}, we smooth out the right pockets of the clause pattern and introduce a second ear at the bottom side of the polygon, right below the first ear, that looks *up* so it can cover the right side of all clause patterns. F^{lit} can be moved to the same edge of the literal pattern as T^{lit} by introducing a little bend to it to create a new *convex* vertex, such that F^{lit} can still cover the entire literal pattern and its spike, but not satisfy the clause. Finally, T^{var} and F^{var} need only cover the half-plane below the line connecting them to the point w, which requires no change.

Appendix C: How we computed the numbers

The construction we created in Section 3, by modifying the one given in [11] uses the following gadgets:

1. Ear gadget: 1 PFG.
2. Literal gadget: 2 BMs.
3. Clause: 3 literal gadgets + 1 PFG = 6 BMs + 1 PFG.
4. Variable gadget: 2 PFGs.

Note that the PFGs in the variable gadget need not be *activated*, i.e., receive a floodlight at their inward wedge is optional. All other PFGs must be activated. For `FIP`, we use 3 vertex α-floodlights per BM and 1 α-floodlight for PFGs. The number of vertex α-floodlights required to *operate* the gadgets is

$$1 + (6 \times 3 + 1)m + 1 \times n = 19m + n + 1. \qquad (1)$$

For `PFIP`, we use 7 α-floodlights per BM, 2 α-floodlights per *active* PFG and 1 α-floodlight per *inactive* PFG. The number of α-floodlights required to *operate* the gadgets is

$$2 + (6 \times 7 + 2)m + 3 \times n = 44m + 3n + 2. \qquad (2)$$

Appendix D: Transformation of a feasible solution

Following the lines of the transformation process in [11] we move α-floodlights in such a way that the set of DAs that a floodlight sees changes in only one of two ways: either more arrows are included or it is ensured that another floodlight, possibly added to the solution, covers any arrows removed from this set.

With that, the α-floodlights in a given solution S' computed for the `RFIP` instance I' are moved as follows:

1. Determine the, at least, 2 floodlights that cover the ear PFG and move them to the standard PFG configuration. In addition to the PFG itself, this also ensures that the legs of all variable patterns are covered.
2. For each clause pattern, determine the, at least, 2 floodlights that cover its PFG and move them to the standard PFG configuration. In addition to the PFG itself, this also ensures that the clause pattern, except for the dent denoted by arrow 2 in Figure 4, is entirely covered.
3. For each BM, there will be at least 7 floodlights inside it. We start with the, at least 6, floodlights that do not illuminate any part of the mouth. We move these 6 to illuminate the interior of the BM, except for the mouth, by the configurations shown in Figure 2. Any remaining floodlights that do not illuminate any part of the mouth are moved to the red configuration, i.e., such that they illuminate the entire mouth, the associated dent and spike, if one is associated with the BM at hand, corresponding to setting the literal to `TRUE`. For floodlights that illuminate parts of the mouth, there will be three cases:
   (a) If the floodlight also illuminates the DA associated with a `FALSE` assignment, we move it to the blue configuration, i.e., such that it illuminates the entire mouth and spike corresponding to setting the literal to `FALSE`.

(b) If the floodlight also illuminates the DA associated with a `TRUE` assignment, we move it to the red configuration.

(c) If the floodlight does not illuminate any DAs, we move it to the red configuration.

4. If a BM has more than one floodlight in either the red or blue configurations, we leave only one and move the extra floodlights to the configuration of the same color in the variable pattern of its variable.

5. If a BM has floodlights in both the red and blue configurations, switch all BMs of its variable to the red configuration and move the, at least one, extra floodlights to the red configuration in its variable pattern. If there is already a floodlight there, move the extra floodlights to the blue configuration instead. This ensures that all dents and spikes associated with this variable are illuminated. In addition, any clause dents that were illuminated in the input solution by floodlights in any gadget of this variable are still illuminated.

6. For floodlights inside a clause pattern but outside any BM or PFG, we have a number of cases. As shown in Figure 9, we need to consider floodlight configurations within the intersection of cones belonging to different dents. Observe the following: (a) The literal dents are set up such that no two can be illuminated by a single floodlight. (b) The design of the BM and the steps thus far outlined in the transformation process ensure that all BMs will have a floodlight in at least one of the red or blue configurations. (c) The PFG illuminates the entire clause pattern except for the dents and DA-2. .

   - Cases 1 and 4: Even if the floodlight can illuminate the DAs of both the literal dent and clause dent, the BM above associated with the dent in question must also be able to illuminate at least one of the two DAs. As such, it suffices to move the floodlight to the unoccupied configuration in this BM, if any.
   - Cases 2 and 3: Similar to the previous case, even if the floodlight can illuminate the DAs of both the literal dent and the spike, the BM must be able to illuminate at least one. Likewise, the floodlight is moved to the unoccupied configuration of the BM, if any.
   - Cases 5 and 6: Since the floodlight may be able to illuminate the DAs of both a literal dent and a spike belonging to a different literal, we will need to add a floodlight and move one to illuminate the spike from its corresponding configuration in the BM and move the other to illuminate the literal dent from the red configuration of its BM.
   - Cases 7, 8 and 9: Such floodlights may illuminate the DA of the clause dent or the spikes of some literal. By moving these floodlights to the corresponding configuration in the BM associated with the DA, we can still illuminate it.

7. For each variable pattern, move the floodlight that sees the DA of the variable pattern to the red configuration, if it also lies in a spike pattern containing the red point. Otherwise, move it to the blue configuration.

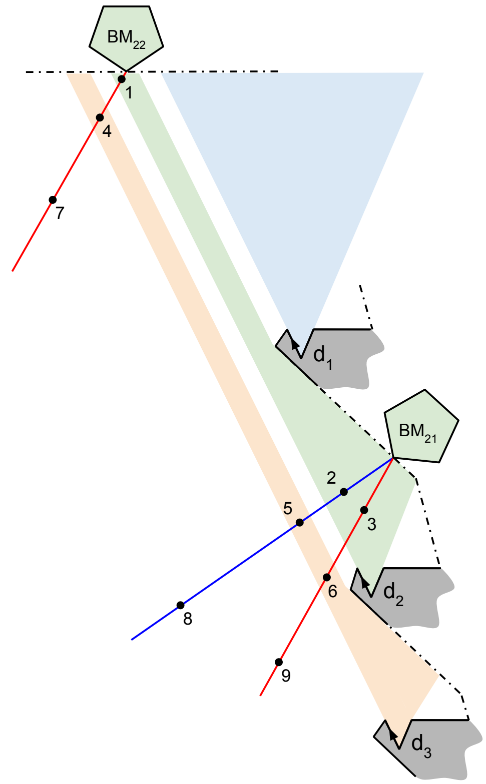


Figure 9: Additional cases for floodlight placement.

8. Move all floodlights that cover a single spike to the red or blue point of the spike pattern of that spike.

9. If a floodlight illuminates DAs of two spike pattens that connect literals to two different legs of variable patterns, add a floodlight and move one floodlight each to the two red or blue points of the variable patterns of these two spikes.

   This is the only case where we *add* an α-floodlight and increase the cost of the solution. Note that because of Lemma 1 [11], no floodlight can see the DAs of three spike patterns that belong to three different legs.

10. Any floodlights that can be removed without leaving any DA uncovered are moved, and fixed, to any red or blue point of any variable pattern, if there is no floodlight there already.

We iterate this process until the locations of all floodlights are fixed. One can verify that the transformed solution of S'' is still a feasible solution of I' as any element which was covered in S' remains covered in S''. To obtain the solution S of the `FOM-3SAT` instance I using S'', we set the truth values of the variables as follows. For variable x_k, if the corresponding variable pattern only has floodlights at the blue point, we set it to `False`. If it has only floodlights at the red point, we set it to `True`. If it has floodlights at both points, we assign x_k in such a way that makes the majority of its, at most 5, literals `True`.

Appendix E: Omitted theorems and proofs

We fill in the omitted steps following the analysis in [11].

5.1 $Satisfiable(I) \implies MinFloodlightCost(I')$

The proofs below also provide the *if* part needed for the NP-hardness results we obtained in Section 4. Note that for the *only if* part, we assign `True` to all variables having the α-floodlight at the red points of their variable patterns and `False` otherwise. In particular, these two proofs work for `FIP` and `PFIP` as well and show both the two problems and their restricted variants are NP-hard.

Lemma 11 *If an instance of* `FOM-3SAT`*, with n variables and $m \le \frac{5}{3}n$ clauses, is satisfiable, then there exists a feasible solution of the corresponding instance of* `RFIP` *with $19m + n + 1$ α-floodlights.*

Proof. Given a satisfying assignment of the n variables in the `FOM-3SAT` instance, we add α-floodlights to a solution of `RFIP` as follows. We start by placing 1 at the ear PFG, 1 for the PFG in all m clauses, and 2 in each of the 2 BMs for all $3m$ literal couples. Next, for each variable x_k, we do the following:

1. If x_k is true, place 1 α-floodlight at the red point in its variable pattern, 1 α-floodlight in the red point of each of its positive literals and 1 α-floodlight in blue point of its negative literals.
2. If x_k is true, place 1 α-floodlight at the blue point in its variable pattern, 1 α-floodlight in the blue point of each of its positive literals and 1 α-floodlight in red point of its negative literals.

For each positive literal, we place 1 α-floodlight in the red point of its coupled BM. For each false literal, we place 1 α-floodlight in the blue point of its coupled BM. This solution is feasible and costs $1 + m + (2 \times 2)3m + n + (1 + 1)3m = 19m + n + 1$. □

Lemma 12 *If an instance of* `FOM-3SAT`*, with n variables and $m \le \frac{5}{3}n$ clauses, is satisfiable, then there exists a feasible solution of the corresponding instance of* `RPFIP` *with $44m + 3n + 2$ α-floodlights.*

Proof. Given a satisfying assignment of the n variables in the `FOM-3SAT` instance, we add α-floodlights to a solution of `RPFIP` as follows. We start by placing 2 at the ear PFG, 2 for the outward wedges of the 2 PFGs in all n variable patterns, 2 for the PFG in all m clauses, and 6 in each of the 2 BMs for all $3m$ literal couples. We proceed as in the proof of Lemma 11. This solution is feasible and costs $2+2n+2m+(6\times2)3m+n+(1+1)3m = 44m+3n+2$. □

5.2 ϵ'-$APPROX(I') \implies \epsilon$-$APPROX(I)$

Given a feasible ϵ-approximate solution S' to I' of the α-floodlight illumination problem, we apply the transformation process described in Appendix D. The transformed solution S'' is still feasible, i.e., illuminates the entire polygon. However, due to the possibility of having α-floodlights at the intersection of two spike patterns, we resolved to adding α-floodlights and ended up having variable patterns with α-floodlights at both the red and blue points. Such variables were then assigned in a manner that satisfies the majority of their clauses, but we will not be able to guarantee satisfying all clauses.

Lemma 13 *If there exists an $\epsilon > 0$ and a feasible solution of the* `RPFIP` *instance I' with at most $44m + 3n + 2 + \epsilon m$ α-floodlights, then there exists an assignment of the variables of the corresponding* `FOM-3SAT` *instance I that satisfies at least $m(1 - 4\epsilon)$ clauses.*

Proof. Any feasible solution S' can be transformed into a *canonical* solution S'' that only illuminates the polygon using the gadgets the way we designed them. In such a canonical solution, we know the minimum number of α-floodlights required by the gadgets themselves. Clearly, the algorithm for the illumination problem could not illuminate the entire polygon using that minimum number, possibly because it was created using an unsatisfiable boolean formula. In both cases, we know the algorithm incurred *at most* an additional ϵm cost to ensure the entire polygon is covered. In the worst case, all these additional ϵm α-floodlights were placed in the intersections of two spike patterns. This means that when the transformation process terminates, *at most* $2\epsilon m$ variable patterns will have received an additional α-floodlight that results, in the worst case, in all the $2\epsilon m$ variable patterns having two α-floodlights at both their red and blue points. This leaves at least $n - 2\epsilon m$ variable patterns with only one α-floodlight. For all variables in the second group, they can be assigned a truth value unambiguously. For the variables in the first group, however, we assign truth values to satisfy the majority of their clauses. In the worst case, each such variable will satisfy only 3 out of its 5 clauses. In the worst case, all the 2 clauses left out by *each* of the variables in the second group will not be satisfied by any other literal. This means we may not be able to satisfy at most $2 \times 2\epsilon m = 4\epsilon m$ clauses. The number of satisfied clauses can then be lower bounded by $m - 4\epsilon m = m(1 - 4\epsilon)$. □

5.3 Don't make a promise that is *hard* to keep

Using Lemma 12 and the contraposition of Lemma 13, we obtain the following.

Theorem 14 *Let I be an instance of the promise problem of* `FOM-3SAT`*, with n variables in I, $m \le \frac{5}{3}n$ clauses. Let $OPT(I)$ denote the maximum number of clauses that can be satisfied using any assignment of the n variables. Furthermore, let I' be the corresponding instance of* `RPFIP` *and let $OPT(I')$ denote the minimum number of α-floodlights needed to illuminate the polygon in I'. Then the following hold:*

- *If $OPT(I) = m$, then $OPT(I') \le 44m + 3n + 2$.*
- *If $OPT(I) \le m(1 - 4\epsilon)$, then $OPT(I') \ge 44m + 3n + 2 + \epsilon m$.*

Theorem 14 shows that the reduction is indeed gap-preserving and that the promise problem of `RPFIP` with parameters $44m + 3n + 2$ and $44m + 3n + 2 + \epsilon m$ is NP-hard.