Smoothed Analysis of the Art Gallery Problem

Michael Gene Dobbins[1], Andreas Holmsen[2], and Tillmann Miltzow[3]

1 **Department of Mathematical Sciences, Binghamton University**
2 **Department of Mathematical Sciences, KAIST**
3 **Department of Information and Computing Sciences, Utrecht University**

Abstract

In the Art Gallery Problem we are given a polygon P on n vertices and a number k. We want to find a guard set G of size k, such that each point in P is *seen* by a guard in G. Formally, a guard g sees a point $p \in P$ if the line segment pg is fully contained inside P.

We analyze the Art Gallery Problem under the lens of Smoothed Analysis. The significance of our results is that algebraic methods are not needed to solve the Art Gallery Problem in *typical* instances. This is the first time an $\exists\mathbb{R}$-complete problem was analyzed by Smoothed Analysis. Details can be found in the full-version [14].

A short video explaining the result is available at `youtu.be/Axs7k-qL2zY`.

1 Introduction

In the Art Gallery Problem we are given a polygon P and a number k. We want to find a guard set G of size k, such that each point in P is *seen* by a guard in G. Formally, a guard g sees a point $p \in P$ if the line segment pg is fully contained inside P. We usually denote the vertices of P by $v_1, \ldots, v_n$, and the number of vertices by n.

One of the most fundamental questions on the Art Gallery Problem is whether it is contained in the complexity-class NP. A first doubt of NP-membership was raised in 2017, when Abrahamsen, Adamaszek and Miltzow showed that there exist polygons with vertices given by integer coordinates, that can be guarded by three guards, in which case some guards must necessarily have irrational coordinates [1]. (It is an open problem whether irrational guards may be required for polygons which can be guarded by two guards.) Shortly after, the same authors could show that the Art Gallery Problem is complete for the complexity class $\exists\mathbb{R}$ [2].

The class $\exists\mathbb{R}$ is the class of all decision problems that are many-one reducible in polynomial time to deciding whether a given polynomial $Q \in \mathbb{Z}[x_1, \ldots, x_n]$ has a real root, i.e. a solution $x \in \mathbb{R}^n$ such that $Q(x) = 0$. From the field of real algebraic geometry [4], we know that

$$\text{NP} \subseteq \exists\mathbb{R} \subseteq \text{PSPACE}.$$

The complexity class $\exists\mathbb{R}$ provides a tool to give much more compelling arguments that a problem may not lie in NP than merely observing that the naive way of placing the problem into NP does not work. Indeed various problems have been shown to be $\exists\mathbb{R}$-complete [8, 9, 13, 16, 17, 20–23] and thus either non of them lie in NP or all of them do.

While those theoretical results on the Art Gallery Problem are quite negative, the history and practical experiences tell a more positive story. First of all, it took more than four decades before an example could be found that requires irrational guards [1]. Regarding the practical study of the Art Gallery Problem, we want to point out that several researchers have implemented heuristics, that were capable of finding optimal solutions for a large class of simulated instances [3, 5–7, 10–12, 15, 19]. Even up to 5000 vertices.

35th European Workshop on Computational Geometry, Utrecht, The Netherlands, March 18–20, 2019.
This is an extended abstract of a presentation given at EuroCG'19. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

$\exists\mathbb{R}$-completeness is our main motivation, to see if there is a *simple* algorithm that solves the Art Gallery Problem. As we don't expect that such an algorithm is correct in the worst case, we turn our attention to different ways to analyze algorithms.

Smoothed Analysis

Some algorithms perform much better than predicted by their worst case analysis. The most famous example seems to be the Simplex-Algorithm. It is an algorithm that solves linear programming efficiently in practice, although it is known that there are instances for seemingly all variants of the algorithm that take an exponential amount of time (see for instance [18]). There are several possible ways to explain this behavior. For example, it could be that all practical instances have some structural properties, which we have not yet discovered. We could imagine that a more clever analysis of the Simplex-Algorithm would yield that it runs in polynomial time, assuming the property is presented. To the best of our knowledge such a property has not yet been identified. Another approach would be to argue that worst case examples are just very "rare in practice". The problem with this approach is that it is difficult to formalize. Smoothed Analysis is a nice combination of the average case and the worst case analysis and generally referred to as Smoothed Analysis, as it *smoothly* interpolates between the two. It was developed by Spielman and Teng [24], who introduced the field in their celebrated seminal paper "Smoothed Analysis of algorithms: Why the simplex algorithm usually takes polynomial time". Both authors received the Gödel Prize in 2008, and the paper was one of the winners of the Fulkerson Prize in 2009. In 2010 Spielman received the Nevanlinna Prize for developing Smoothed Analysis.

The rough idea is to take the worst instance and perturb it slightly in a random way. The smoothed expected running time can be defined as follows: Let us fix some δ, which describes the maximum magnitude of perturbation. We denote by $(\Omega_\delta, \mu_\delta)$ a corresponding probability space where each $x \in \Omega_\delta$ defines for each instance I a new 'perturbed' instance I_x. We denote by $T(I_x)$, the time to solve the instance I_x. Now the smoothed expected running time of instance I equals

$$T_\delta(I) = \mathop{\mathbb{E}}_{x\in\Omega_\delta} T(I_x) = \int_{x\in\Omega_\delta} T(x)\mu_\delta(x).$$

If we denote by Γ_n the set of instances of size n, then the smoothed running time equals:

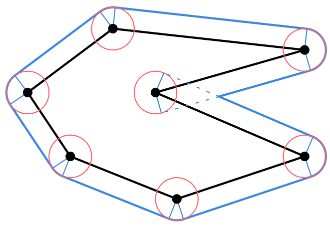
$$T_{\text{smooth}}(n) = \max_{I\in\Gamma_n} \mathop{\mathbb{E}}_{x\in\Omega_\delta} T(I_x).$$

Roughly speaking this can be interpreted as saying, that not only do the majority of instances have to behave nicely, but actually in every neighborhood the majority of instances behave nicely. The expected running time is measured in terms of n and δ. If the expected running time is small in terms of $1/\delta$ then this means that difficult instances are *fragile* with respect to perturbations. This serves as theoretical explanation why such instances may not appear in practice.

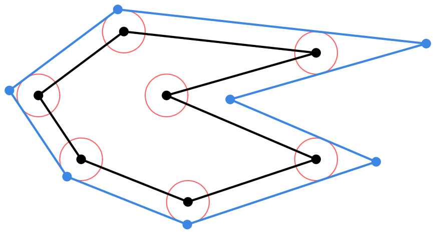
Although the concept of Smoothed Analysis is more complicated than simple worst case analysis, it is a new success story in theoretical computer science. It could be shown that various algorithms actually run in smoothed polynomial time, explaining very well their practical performance.

1.1 Defintions

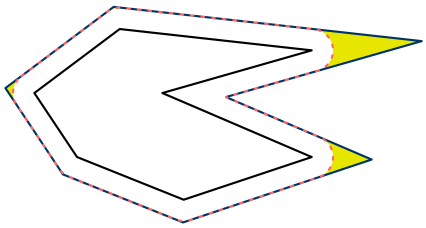
The different models of perturbation are illustrated in Figure 1. A rigorous definition can be found in the full-version of the paper [14].



(a) The Minkowski-sum of a polygon together with a disk. This is also called a Minkowski-Inflation.



(b) The polygon together with an Edge-Inflation. Roughly speaking, every edge of the polygon is "pushed" to the outside by the same amount.



(c) If we continue the edges, of a Minkowski-Inflation, we get an Edge-Inflation.

Figure 1 Overview, over various models, how a polygon can be perturbed. We use the uniform distribution in each case.

1.2 Results

Our main result states that typical instances do not require irrational guards and the expected number of bits per guard is logarithmic in L, δ and n. The result establishes that algebraic methods are *not* needed in typical instances.

▶ **Theorem 1.1** (Bit-complexity). *Let P be a polygon on n vertices, suppose $P \subset [0, L]^2$ for some positive integer L. If $\delta > 0$ is the magnitude of a Minkowski-Inflation or Edge-Inflation, then the expected number of bits per guard to describe an optimal solution equals $O\left(\log\left(\frac{nL}{\delta}\right)\right)$.*

As a simple corollary of the proof, we get that a fine grid of expected width $w = 2^{O(\log(nL/\delta))} = (nL/\delta)^{O(1)}$ will contain an optimal guarding set. This may appear at first sight as a *candidate set* of polynomial size, however recall that the vertices are given in binary and thus L may be exponential in the input size.

It can be argued (see the full-version [14]) that this also leads to expected NP algorithms in a specific sense. A very careful discussion of the different models of computation is needed [14] to make the above statement precise. Our results can also be extended to other types of perturbations [14].

Notation

We write $f(n) \leq_c g(n)$, to indicate $f(n) = O(g(n))$ or equivalently $f(n) \leq cg(n)$, for some large enough constant c. (Note that this is, in turn, equivalent to $f(n) \leq c_1 g(n) + c_2$. To see this note that $g(n) \geq 1$ and choose $c = c_1 + c_2$.)

2 Preliminaries

In this section we establish some general facts that will be needed throughout the paper.

The key idea of the paper are some monotonicity properties of Minkowski-Inflation and Edge-Inflation. Roughly speaking guarding can only get easier after inflations. (We denote by $OPT(P)$ a guarding set of P of optimal size. We denote by $OPT(P, C)$ a guarding set of P of optimal size, when restricted to the set C.)

▶ **Lemma 2.1** (Fixed Minkowski-Inflation). *Let P be a polygon, $t > 0$ and P_t its Minkowski-Inflation by magnitude t. Then $|OPT(P)| \geq |OPT(P_t, w\mathbb{Z}^2)|$, for any $w \leq \sqrt{2}t$.*

Proof. Given $OPT = OPT(P)$, we define a set $G \subseteq w\mathbb{Z}^2$ of guards of size $|G| = |OPT|$, by rounding every point in OPT to its closest grid point in $w\mathbb{Z}^2$. We will show that G guards P_t. See Figure 2 for an illustration.

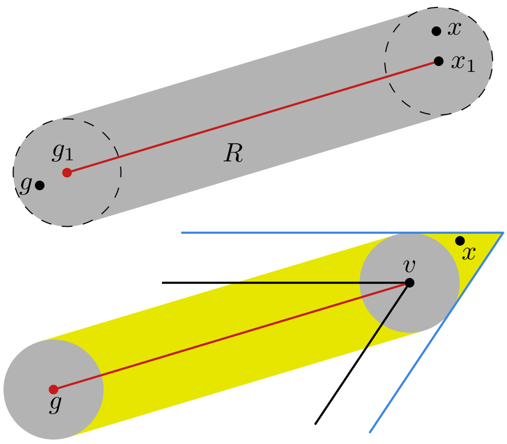


Figure 2 Top: The Region R is convex, and contains a guard $g \in G$ and the point x. Thus x is guarded by g. Bottom: The Region R' is easily seen to be convex.

Let us fix some arbitrary point $x \in P_t$. It is sufficient to show that G guards x. By definition of P_t, there exists an $x_1 \in P$ and an $x_2 \in \text{disk}(t)$ such that $x = x_1 + x_2$. Furthermore let g_1 be a guard of OPT that guards x_1. (disk(t) is a disk of radius t.) Consider the region $R = g_1x_1 \oplus \text{disk}(t)$, i.e., the Minkowski-sum of the segment g_1x_1 with a disk of radius t. As the segment g_1x_1 is contained in P, it holds that R is contained in P_t. Also as both the segment and the disk are convex, so is R. At last notice that R contains a point $g \in G$, as every disk of radius t contains a point of the grid $w\mathbb{Z}^2$ with $w = \sqrt{2}t$. As R is convex, $g \in G$ guards x. ◀

▶ **Lemma 2.2** (Fixed Edge Inflation). *Let P be a polygon with integer coordinates and $t > 0$ and P_t the Edge-Inflation of P by t. Then $|OPT(P)| \geq |OPT(P_t, w\mathbb{Z}^2)|$, for any $w \leq \sqrt{2}t$.*

Proof. We follow closely the proof of Lemma 2.1. See Figure 2 for an illustration.

Given $OPT = OPT(P)$, we define a set $G \subseteq w\mathbb{Z}^2$ of guards of size $|G| = |OPT|$, by rounding every point in OPT to its closest grid point in $w\mathbb{Z}^2$. We will show that G guards the shape P_t. Note that in an edge inflation by t, we get the same shape as by a Minkowski-Inflation by t, except that we have to add some small regions at the convex corners, as illustrated in Figure 1c. We already know that G guards the Minkowski t-inflation of P. So it remains to show that G guards those little extra regions, as discussed above.

Let us fix some arbitrary point $x \in P_t$ inside one of those extra regions. We will show that G guards x. Let v be the vertex according to the region that x sits in. Furthermore let g_1 be a guard of OPT that guards v. Consider the region $R = g_1 v \oplus \text{disk}(0, t)$. We define R' as the region R together with the region that x sits in. Obviously $x \in R'$ and also there exists a point of G in R. It holds by construction that R' is convex. This finishes the proof. ◀

3 Expected Number of Bits

This section is devoted to show the main theorem.

Proof. Let us assume that there are some numbers $0 = t_0 < t_1 < \ldots < t_\ell = \delta$ such that for all i and $s \in [t_{i-1}, t_i)$ holds that $|OPT(P_s)|$ is constant. As $|OPT(P_s)|$ is monotonically decreasing, for increasing s, it holds that $\ell \leq n$. We denote by $\delta_i = t_i - t_{i-1}$.

Note that if the perturbation happens to be $s \in [t_{i-1}, t_i]$ then a grid of width $w = \sqrt{2}(s - t_{i-1})$ contains an optimal solution to guard P_s, see Lemma 2.1 and 2.2. Note that we use the lemmas on the shape $P_{t_{i-1}}$ inflated by $s - t_{i-1}$. Then the number of bits per guard to describe the solution equals $O(\log(L/w))$ per guard. To see this note that we can use $b = \lceil 1/w \rceil$ as denominator of all coordinates and the numerators are upper bounded by $\lceil L/w \rceil$. Thus $O(\log(L/w))$ bits suffice. Let us denote the number of bits after a perturbation by s as $B(s)$. We denote by $\mathbb{E}(B_i)$ the expected number of bits for $s \in [t_{i-1}, t_i)$. The expected number of bits $\mathbb{E}(B_i)$ can be calculated as

$$\mathbb{E}(B_i) = \frac{1}{\delta_i} \int_{s \in [t_{i-1}, t_i)} B(s) \leq_c \frac{1}{\delta_i} \int_{s \in [t_{i-1}, t_i)} \log(L/(s - t_{i-1})) = \frac{1}{\delta_i} \int_{s \in [0, \delta_i]} \log(L/s).$$

Using some computer algebra system and concavity of $\log(1/x)$, we get

$$= \frac{1}{\delta_i} \delta_i \left(1 + \log(L/\delta_i)\right) \leq_c \log(L/\delta_i).$$

We are now ready to compute $\mathbb{E}(B)$.

$$\mathbb{E}(B) = \frac{1}{\delta} \sum_{i=1,\ldots,\ell} \delta_i \mathbb{E}(B_i) \leq_c \frac{1}{\delta} \sum_{i=1,\ldots,\ell} \delta_i \log(L/\delta_i).$$

As the function $x \log(1/x)$ is concave the maximum is attained, if $\delta_1 = \ldots = \delta_\ell = \delta/\ell$. Thus we get

$$\mathbb{E}(B) \leq_c \frac{1}{\delta} \sum_{i=1,\ldots,\ell} \delta/\ell \log(L\ell/\delta) = \log(L\ell/\delta) \leq_c \log(Ln/\delta).$$ ◀

Acknowledgments. We want to thank Stefan Langerman, Jean Cardinal, John Iacono and Mikkel Abrahamsen for helpful discussion on the presentation of the results. We want to thank Joseph O'Rourke for pointing us to an algorithm to check if a given set of guard positions is correct. Part of the research was conducted while visiting KAIST, and we thank KAIST and BK21 for their support and for providing an excellent working environment. Andreas Holmsen was supported by the Basic Science research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2016R1D1A1B03930998). Tillmann Miltzow acknowledges the generous support from the ERC Consolidator Grant 615640-ForEFront and the Veni EAGER.

References

1 Mikkel Abrahamsen, Anna Adamaszek, and Tillmann Miltzow. Irrational guards are sometimes needed. In *SoCG 2017*, pages 3:1–3:15, 2017. Arxiv 1701.05475.

2 Mikkel Abrahamsen, Anna Adamaszek, and Tillmann Miltzow. The art gallery problem is ∃ℝ-complete. In *STOC 2018*, pages 65–73, 2018. Arxiv 1704.06969. URL: `http://doi.acm.org/10.1145/3188745.3188868`, `doi:10.1145/3188745.3188868`.

3 Yoav Amit, Joseph S.B. Mitchell, and Eli Packer. Locating guards for visibility coverage of polygons. *International Journal of Computational Geometry & Applications*, 20(05):601–630, 2010.

4 Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in real algebraic geometry*. Springer-Verlag Berlin Heidelberg, 2006.

5 Dorit Borrmann, Pedro J. de Rezende, Cid C. de Souza, Sándor P. Fekete, Stephan Friedrichs, Alexander Kröller, Andreas Nüchter, Christiane Schmidt, and Davi C. Tozoni. Point guards and point clouds: Solving general art gallery problems. In *SoCG 2013*, pages 347–348, 2013. URL: `http://doi.acm.org/10.1145/2462356.2462361`, `doi:10.1145/2462356.2462361`.

6 Andrea Bottino and Aldo Laurentini. A nearly optimal sensor placement algorithm for boundary coverage. *Pattern Recognition*, 41(11):3343–3355, 2008.

7 Andrea Bottino and Aldo Laurentini. A nearly optimal algorithm for covering the interior of an art gallery. *Pattern recognition*, 44(5):1048–1056, 2011.

8 Jean Cardinal, Stefan Felsner, Tillmann Miltzow, Casey Tompkins, and Birgit Vogtenhuber. Intersection graphs of rays and grounded segments. *Journal of Graph Algorithms and Applications*, 22:273–295, 2018.

9 Jean Cardinal and Udo Hoffmann. Recognition and complexity of point visibility graphs. *Discrete & Computational Geometry*, 57(1):164–178, 2017.

10 Marcelo C. Couto, Pedro J. de Rezende, and Cid C. de Souza. An exact algorithm for minimizing vertex guards on art galleries. *International Transactions in Operational Research*, 18(4):425–448, 2011.

11 Marcelo C. Couto, Cid C. De Souza, and Pedro J. De Rezende. Experimental evaluation of an exact algorithm for the orthogonal art gallery problem. In *International Workshop on Experimental and Efficient Algorithms*, pages 101–113. Springer, 2008.

12 Pedro J. de Rezende, Cid C. de Souza, Stephan Friedrichs, Michael Hemmer, Alexander Kröller, and Davi C. Tozoni. Engineering art galleries. In Peter Sanders Lasse Kliemann, editor, *Algorithm Engineering – Selected Results and Surveys*, pages 379–417. Springer International Publishing, 2016. URL: `http://dx.doi.org/10.1007/978-3-319-49487-6_12`, `doi:10.1007/978-3-319-49487-6_12`.

13 Michael G. Dobbins, Linda Kleist, Tillmann Miltzow, and Paweł Rzążewski. ∀∃ℝ-completeness and area-universality. *WG 2018*, 2018. Arxiv 1712.05142.

14 Michael Gene Dobbins, Andreas Holmsen, and Tillmann Miltzow. Smoothed analysis of the art gallery problem. *CoRR*, abs/1811.01177, 2018. URL: `http://arxiv.org/abs/1811.01177`, `arXiv:1811.01177`.

15 S. Friedrichs. Integer solutions for the art gallery problem using linear programming. Masterthesis, 2012.

16 Jugal Garg, Ruta Mehta, Vijay V. Vazirani, and Sadra Yazdanbod. ETR-completeness for decision versions of multi-player (symmetric) Nash equilibria. In *ICALP 2015*, pages 554–566, 2015.

17 Ross J. Kang and Tobias Müller. Sphere and dot product representations of graphs. In *SoCG*, pages 308–314. ACM, 2011.

18 Victor Klee and George J. Minty. How good is the simplex algorithm. Technical report, Washington Univ. Seattle Dept. of Mathematics, 1970.

19 Alexander Kröller, Tobias Baumgartner, Sándor P Fekete, and Christiane Schmidt. Exact solutions and bounds for general art gallery problems. *Journal of Experimental Algorithmics (JEA)*, 17:2–3, 2012.

20 Anna Lubiw, Tillmann Miltzow, and Debajyoti Mondal. The complexity of drawing a graph in a polygonal region. *Arxiv*, 2018. Graph Drawing 2018.

21 Jürgen Richter-Gebert and Günter M. Ziegler. Realization spaces of 4-polytopes are universal. *Bulletin of the American Mathematical Society*, 32(4):403–412, 1995.

22 Marcus Schaefer. Realizability of graphs and linkages. In *Thirty Essays on Geometric Graph Theory*, pages 461–482. Springer, 2013.

23 Yaroslav Shitov. A universality theorem for nonnegative matrix factorizations. *Arxiv 1606.09068*, 2016.

24 Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM (JACM)*, 51(3):385–463, 2004.