Parameterized Hardness of Art Gallery Problems

Édouard Bonnet * Tillmann Miltzow †

Abstract

Given a simple polygon $\mathcal{P}$ on n vertices, two points x, y in $\mathcal{P}$ are said to be visible to each other if the line segment between x and y is contained in $\mathcal{P}$. The point guard art gallery problem asks for a minimum set S such that every point in $\mathcal{P}$ is visible from a point in S. The vertex guard art gallery problem asks for such a set S subset of the vertices of $\mathcal{P}$. The set S is referred to as guards. We show $W[1]$-hardness of both variants, when parameterized by the number k of guards. We even rule out any $n^{o(k/\log k)}$ algorithm under the exponential time hypothesis.

1 Introduction

Given a simple polygon $\mathcal{P}$ on n vertices, two points x, y in $\mathcal{P}$ are said to be visible to each other if the line segment between x and y is contained in $\mathcal{P}$. The point-guard art gallery problem asks for a minimum set S of points called *guards* such that every point in P is visible from a point in S. The vertex guard art gallery problem asks for such a set of guards S subset of the vertices of $\mathcal{P}$.

One of the first combinatorial results is the elegant proof of Fisk that $\lfloor n/3 \rfloor$ guards are always sufficient and sometimes necessary for a polygon with n vertices [8]. On the algorithmic side, very few variants are solvable in polynomial time [5, 11], but most results are on approximating the minimum number of guards [3, 4, 6, 9]. On the lower bound side the paper of Eidenbenz et al. showed for most relevant variants NP-hardness and inapproximability [7]. In particular, their reduction from Set-Cover implies that the art gallery is $W[2]$-hard on polygons with holes and that there is no $n^{o(k)}$ algorithm, to determine if k guards are sufficient for a given gallery with n vertices, under the exponential time hypothesis [7, Sec.4]. However, polygons with holes are very different to simple polygons as they have unbounded VC-dimension [12]. In particular none of these reductions rule out a fixed parameter tractable algorithm (i.e., whose running time is $O(f(k)n^c)$ where f is any computable function and c is a constant) for simple polygons (see [2] for an introduction to parameterized complexity.).

Obviously, the vertex guard variant can be solved in time $O(n^{k+2})$ by trying out all possible subsets of size k of the vertices and checking if one of those subsets sees the whole polygon. Not obvious at all is the algorithm running in time $n^{O(k)}$ for the point guard variant using standard tools from real algebraic geometry [1]. Despite the fact that the first algorithm is extremely basic and the second algorithm, even with remarkably sophisticated tools, uses almost no problem specific insights, no better exact parameterized algorithms are known.

We present the first conditional lower bounds for the parameterized art gallery problem for simple polygons:

Theorem 1 (Point guard hardness) Point Guard Art Gallery *parameterized by the number of guards k is $W[1]$-hard, and is not solvable in time $n^{o(k/\log k)}$, under the ETH.*

Theorem 2 (Vertex guard hardness) Vertex Guard Art Gallery *is $W[1]$-hard, and is not solvable in time $n^{o(k/\log k)}$, under the ETH.*

2 Preliminaries

For any two integers $x < y$, we set $[x, y] := \{x, x+1, \ldots, y-1, y\}$, and for any positive integer x, $[x] := [1, x]$. The *Exponential Time Hypothesis* (ETH) is a conjecture by Impagliazzo et al. [10] asserting that there is no $2^{o(n)}$-time algorithm for 3-SAT on instances with n variables.

Polygons and visibility. For any two distinct points v and w in the plane, we denote by $\text{seg}(v, w)$ the segment whose two endpoints are v and w, by $\text{ray}(v, w)$ the ray starting at v and passing through w, by $\ell(v, w)$ the supporting line passing through v and w.

A polygon is *simple* if it is not self-crossing and has no holes. For any point x in a polygon $\mathcal{P}$, $V_{\mathcal{P}}(x)$, or simply $V(x)$, denotes the *visibility region* of x within $\mathcal{P}$, that is the set of all the points $y \in \mathcal{P}$ such that segment $\text{seg}(x, y)$ is entirely contained in $\mathcal{P}$. We say that two vertices v and w of a polygon $\mathcal{P}$ are *neighbors* or *consecutive* if vw is an edge of $\mathcal{P}$. A *subpolygon* $\mathcal{P}'$ of a simple polygon $\mathcal{P}$ is defined by any l distinct consecutive vertices $v_1, v_2, \ldots, v_l$ of $\mathcal{P}$ (that is, for every

*Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), edouard.bonnet@lamsade.dauphine.fr

†Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), t.miltzow@gmail.com

$i \in [l-1]$, v_i and v_{i+1} are neighbors in $\mathcal{P}$) such that v_1v_l does not cross any edge of $\mathcal{P}$.

Given a vertex v and two points p and p', we call *triangular pocket rooted at vertex v and supported by ray(v,p) and ray(v,p')* a sub-polygon w, v, w' such that ray(v, w) passes through p, ray(v, w') passes through p'. We say that v is the *root* of the triangular pocket that we denote $\mathcal{P}(v)$. We also say that the pocket $\mathcal{P}(v)$ *points* towards p and p'.

Structured 2-Track Hitting Set. We introduce a new problem which will constitute a handy starting point to show Theorem 1 and 2. In the 2-Track Hitting Set problem, the input consists of an integer k, two sets A and B of the same cardinality totally ordered by $\leq_A$ and $\leq_B$, and two sets $\mathcal{S}_A$ of A-intervals (that is a set of consecutive elements of A according to $\leq_A$), and $\mathcal{S}_B$ of B-intervals. In addition, the elements of A and B are in one-to-one correspondence $\phi : A \to B$ and each pair $(a, \phi(a))$ is called a 2-*element*. The goal is to find a set S of k 2-elements such that the first projection of S is a hitting set of A, and the second projection of S is a hitting set of B. Structured 2-Track Hitting Set is the same problem with color classes over the 2-elements, and a restriction on the one-to-one mapping ϕ. A is partitioned into k classes $(C_1, C_2, \ldots, C_k)$ where $C_j = \{a_1^j, a_2^j, \ldots, a_t^j\}$ for each $j \in [k]$, where $|A| = tk$, and is ordered: $a_1^1, a_2^1, \ldots, a_t^1, a_1^2, a_2^2, \ldots, a_t^2, \ldots, a_1^k, a_2^k, \ldots, a_t^k$. We define $C'_j := \phi(C_j)$ and $b_i^j := \phi(a_i^j)$ for all $i \in [t]$ and $j \in [k]$. We now impose that ϕ is such that, for each $j \in [k]$, the t elements of C'_j are consecutive along $\leq_B$. That is, B is ordered: $C'_{\sigma(1)}, C'_{\sigma(2)}, \ldots, C'_{\sigma(k)}$ for some permutation on $[k]$, $\sigma \in \mathfrak{S}_k$. For each $j \in [k]$, the order of the elements within C'_j can be described by a permutation $\sigma_j \in \mathfrak{S}_t$ such that the ordering of C'_j is: $b^j_{\sigma_j(1)}, b^j_{\sigma_j(2)}, \ldots, b^j_{\sigma_j(t)}$. Due to space limitations, we omit the proof of the following theorem.

Theorem 3 Structured 2-Track Hitting Set *is $W[1]$-hard, and not solvable in time $|\mathcal{I}|^{o(k/\log k)}$, unless the ETH fails.*

3 Point Guard

Overview of the reduction. Given an instance $\mathcal{I} = (k \in \mathbb{N}, t \in \mathbb{N}, \sigma \in \mathfrak{S}_k, \sigma_1 \in \mathfrak{S}_t, \ldots, \sigma_k \in \mathfrak{S}_t, \mathcal{S}_A, \mathcal{S}_B)$, we build a simple polygon $\mathcal{P}$ with $O(kt + |\mathcal{S}_A| + |\mathcal{S}_B|)$ vertices, such that $\mathcal{I}$ is a YES-instance iff $\mathcal{P}$ can be guarded by $3k$ points.

The global strategy of the reduction is to *allocate*, for each color class $j \in [k]$, $2t$ special points in the polygon $\alpha_1^j, \ldots, \alpha_t^j$ and $\beta_1^j, \ldots, \beta_t^j$. Placing a guard in α_i^j (resp. β_i^j) shall correspond to picking a 2-element whose first (resp. second) component is a_i^j (resp. b_i^j).

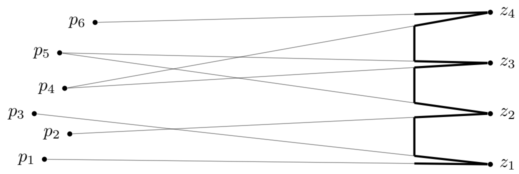

Figure 1: Interval gadgets encoding $\{p_1, p_2, p_3\}$, $\{p_2, p_3, p_4, p_5\}$, $\{p_4, p_5\}$, and $\{p_4, p_5, p_6\}$.

The points α_i^j's and β_i^j's ordered by increasing y-coordinates will match the order of the a_i^j's along $\leq_A$ and then of the b_i^j's along $\leq_B$. Then, far in the horizontal direction, we will place pockets to encode each A-interval of $\mathcal{S}_A$, and each B-interval of $\mathcal{S}_B$ (see Figure 1).

The critical issue will be to *link* point α_i^j to point β_i^j. Indeed, in the Structured 2-Track Hitting Set problem, one selects 2-elements (one per color class), so we should prevent one from placing two guards in α_i^j and $\beta_{i'}^j$ with $i \neq i'$. Due to a technicality, we will introduce a *copy* $\overline{\alpha}_i^j$ of each α_i^j. In each part of the gallery encoding a color class $j \in [k]$, the only way of guarding all the pockets with only three guards is to place them in α_i^j, $\overline{\alpha}_i^j$, and β_i^j for some $i \in [t]$. Hence, $3k$ guards will be necessary and sufficient to guard the whole $\mathcal{P}$ iff there is a solution to the instance of Structured 2-Track Hitting Set.

We now sketch the construction.

Allocated points and interval gadgets. The position of the α_i^j's and β_i^j can be seen on Figure 2 and Figure 4. It is such that the ordering of the α_i^j's (resp. β_i^j) by increasing y-coordinate matches the order $\leq_A$ on the a_i^j's (resp. $\leq_B$ on the b_i^j's). Also, α_i^j and β_i^j shares the same x-coordinate for each $j \in [k], i \in [t]$. There is a quite large gap D along the x-axis between a point α_t^j and α_t^{j+1}.

For each A-interval $I_q = [a_i^j, a_{i'}^{j'}] \in \mathcal{S}_A$, we put, at a very large distance F to the right of the α_i^j's, one triangular pocket $\mathcal{P}(z_{A,q})$ rooted at vertex $z_{A,q}$ and supported by ray($z_{A,q}, \alpha_i^j$) and ray($z_{A,q}, \alpha_{i'}^{j'}$). This way, the only $\alpha_{i''}^{j''}$ seeing vertex $z_{A,q}$ are all the points such that $a_i^j \leq_A a_{i''}^{j''} \leq_A a_{i'}^{j'}$ (see Figure 1 and Figure 4). We do the same for the B-intervals.

Weak linkers. We now describe how we *link* each point α_i^j to its associate β_i^j. See Figure 2 for a description of the following *weak linker* gadget.

For each $j \in [k]$, let us mentally draw ray(α_t^j, β_1^j) and consider points slightly to the left of this ray and quite far. Let us call $\mathcal{R}_{\text{left}}^j$ that informal region of points. Any point in $\mathcal{R}_{\text{left}}^j$ sees, from right to left, in this order α_1^j, α_2^j up to α_t^j, and then,

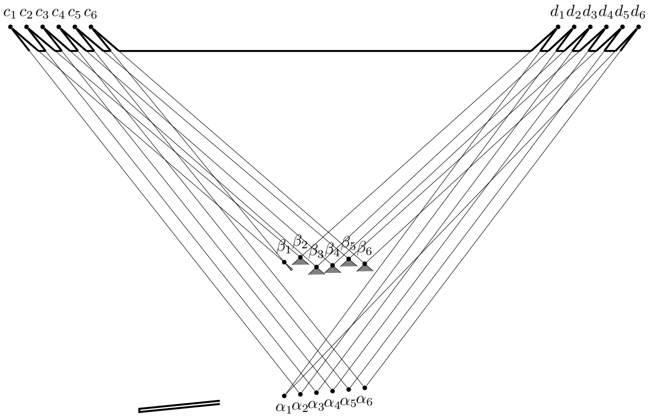

Figure 2: Weak point linker gadget.

β_1^j, β_2^j up to β_t^j. In $\mathcal{R}_{\text{left}}^j$, for each $i \in [t-1]$, we place a triangular pocket $\mathcal{P}(c_i^j)$ rooted at vertex c_i^j and supported by ray(c_i^j, α_{i+1}^j) and ray(c_i^j, β_i^j). We place also a triangular pocket $\mathcal{P}(c_t^j)$ rooted at c_t^j supported by ray(c_i^j, β_1^j) and ray(c_i^j, β_t^j). We place mirroring pockets $P(d_1^j), \dots, \mathcal{P}(d_t^j)$ in the region slightly to the right of ray(α_1^j, β_t^j) and quite far. Finally, we add a thin rectangular pocket $\mathcal{P}_{j,r}$ to the left of the points $\alpha_1^j, \dots, \alpha_t^j$ such that the uppermost longer side of the rectangular pocket lies on the line $\ell(\alpha_1^j, \alpha_t^j)$ (see Figure 2). We denote by $\mathcal{P}_{j,\alpha,\beta}$ the set of pockets $\{\mathcal{P}(c_1^j), \dots, \mathcal{P}(c_t^j), \mathcal{P}(d_1^j), \dots, \mathcal{P}(d_t^j)\}$ and call it *weak linker*.

If one wants to guard $\mathcal{P}_{j,\alpha,\beta}$ with only two points and place the first guard on α_i^j, one is not forced to place the second guard on β_i^j, as we would desire, but anywhere on an area whose uppermost point is β_i^j (see the shaded areas below the b_i^j's in Figure 2).

Linkers. For each $j \in [k]$, we allocate t points $\overline{\alpha}_1^j, \dots, \overline{\alpha}_t^j$ on a horizontal line above and to the right of β_t^j at a quite large distance. We add two weak linkers $\mathcal{P}_{j,\alpha,\overline{\alpha}}$ and $\mathcal{P}_{j,\overline{\alpha},\beta}$, one linking $\alpha_1^j, \dots, \alpha_t^j$ and $\overline{\alpha}_1^j, \dots, \overline{\alpha}_t^j$, the other linking $\overline{\alpha}_1^j, \dots, \overline{\alpha}_t^j$ and $\beta_1^j, \dots, \beta_t^j$ (see Figure 3). We also add a thin horizontal pocket whose lowermost side is in the same line as the points $\overline{\alpha}_1^j, \dots, \overline{\alpha}_t^j$. Pockets of $\mathcal{P}_{j,\alpha,\overline{\alpha}}$ and the two thin rectangular pockets force to put guards on α_i^j and $\overline{\alpha}_i^j$ (for a same $i \in t$), if we have only two guards to spare. Now, pockets of $\mathcal{P}_{j,\alpha,\beta}$ forces to place the third guard below β_i^j while pockets of $\mathcal{P}_{j,\overline{\alpha},\beta}$ forces to place the third guard above β_i^j (again if we have only three guards to spare). So, the only solution is to place the third guard exactly on β_i^j. The k linkers are placed accordingly to Figure 4.

Lemma 4 *$\forall j \in [k]$, $\forall i \in [t]$, the three associate points $\alpha_{i_1}^j$, $\overline{\alpha}_{i_2}^j$, $\beta_{i_3}^j$ guard entirely $\mathcal{P}_j$ iff $i_1 = i_2 = i_3$.*

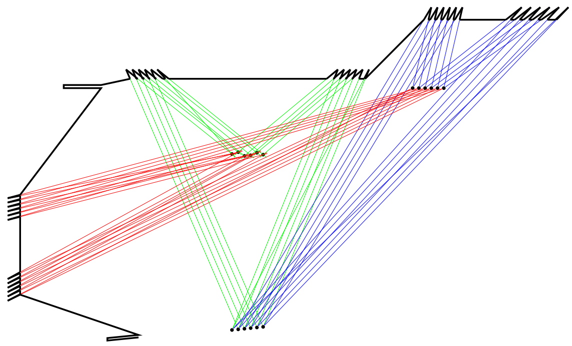

Figure 3: Point linker gadget: a triangle of (three) weak point linkers.

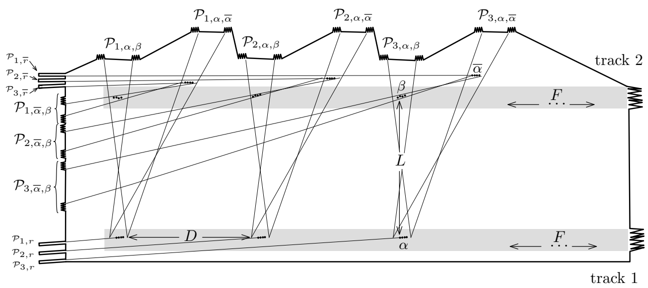

Figure 4: The overall picture of the reduction with $k = 3$.

4 Vertex Guard

The reduction is again from Structured 2-Track Hitting Set.

Vertex linkers. For each $j \in [k]$, permutation σ_j is encoded by a sub-polygon $\mathcal{P}_j$ that we call *vertex linker*, or simply *linker* (see Figure 5). We regularly set t consecutive vertices $\alpha_1^j, \alpha_2^j, \dots, \alpha_t^j$ in this order, along the x-axis. Opposite to this *segment*, we place t vertices $\beta_{\sigma_j(1)}^j, \beta_{\sigma_j(2)}^j, \dots, \beta_{\sigma_j(t)}^j$ in this order, along the x-axis, too. The $\beta_{\sigma_j(1)}^j, \dots, \beta_{\sigma_j(t)}^j$, contrary to $\alpha_1^j, \dots, \alpha_t^j$, are *not* consecutive. We put reflex vertices in between the vertices $\beta_{\sigma_j(1)}^j, \dots, \beta_{\sigma_j(t)}^j$ to ensure that the only way of seeing entirely the *walls* $d^j e^j$ and $x^j y^j$ by taking two vertices α_i^j and $\beta_{i'}^j$ is that $i = i'$.

Lemma 5 *For any $j \in [k]$, the sub-polygon $\mathcal{P}_j$ is seen entirely by $\{\alpha_v^j, \beta_w^j\}$ iff $v = w$.*

What we should now prevent is that one puts a guard in a reflex vertex of the linker.

Filter gadget. The only way to see all the pockets of the filter gadget $\mathcal{F}_j$ (see Figure 6) with two guards is to place them on c_i and d_i for the same i. In the overall construction, the c_i's are in fact vertices $\beta_{\sigma_j(1)}^j, \dots, \beta_{\sigma_j(t)}^j$. Thus, if one wants to guard

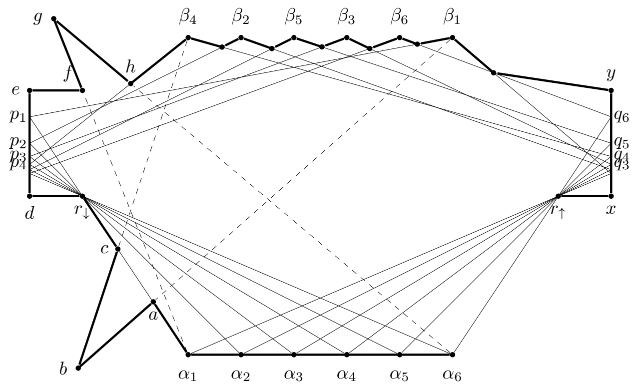

Figure 5: Vertex linker gadget. We omitted the superscript j in all the labels. Here, $\sigma_j(1) = 4,\ \sigma_j(2) = 2,\ \sigma_j(3) = 5,\ \sigma_j(4) = 3,\ \sigma_j(5) = 6,\ \sigma_j(6) = 1$.

all the pockets of $\mathcal{F}_j$ and $\mathcal{P}_j$ with only three guards, one should place them at vertices α_i^j, β_i^j, and $d_{\sigma_j(i)}^j$.

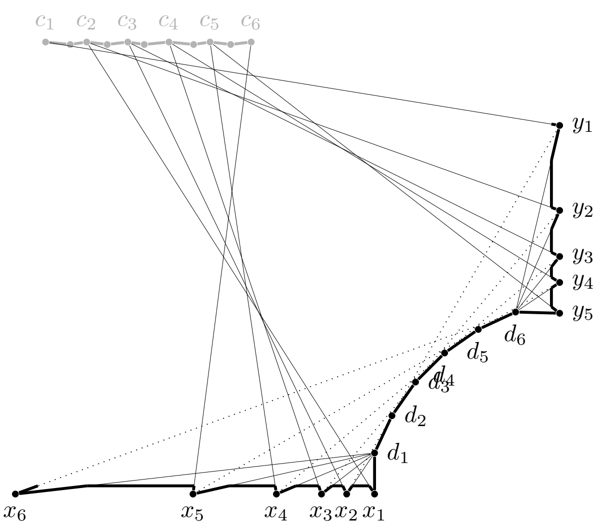

Figure 6: The filter gadget $\mathcal{F}_j$.

Overall construction. Permutation σ is encoded in the way depicted on Figure 7 by limiting the visibility of the vertices $\beta_{\sigma_j(1)}^j, \dots, \beta_{\sigma_j(t)}^j$ to only one filter gadget, namely $\mathcal{F}_j$. Finally, as for the point guard variant, for each A- and B-interval, we place a triangular pocket seeing the corresponding vertices (see Track 1 and 2 of Figure 7).

Acknowledgments

Both authors are supported by the ERC grant PARAMTIGHT: "Parameterized complexity and the search for tight complexity results", no. 280152. We thank Meirav Zehavi and Saeed Merhabi for useful discussions during a preliminary stage of the project.

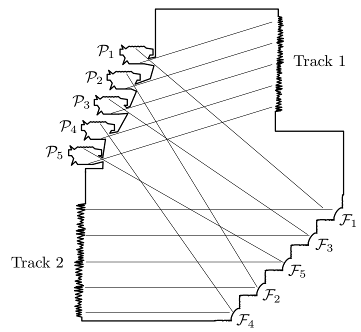

Figure 7: Overall picture of the reduction with $k = 5$.

References

[1] S. Basu, R. Pollack, and M.-F. Roy. Algorithms in real algebraic geometry. *AMC*, 10:12, 2011.

[2] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

[3] A. Deshpande. A pseudo-polynomial time $O(\log^2 n)$-approximation algorithm for art gallery problems. Master's thesis, Department of Mechanical Engineering, Department of Electrical Engineering and Computer Science, MIT, 2006.

[4] A. Deshpande, T. Kim, E. D. Demaine, and S. E. Sarma. A pseudopolynomial time $O(\log n)$-approximation algorithm for art gallery problems. In *WADS 2007*, pages 163–174, 2007.

[5] S. Durocher and S. Mehrabi. Guarding orthogonal art galleries using sliding cameras: algorithmic and hardness results. In *MFCS 2013*, pages 314–324. Springer, 2013.

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

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

[8] S. Fisk. A short proof of chvátal's watchman theorem. *J. Comb. Theory, Ser. B*, 24(3):374, 1978.

[9] S. K. Ghosh. Approximation algorithms for art gallery problems in polygons. *Discrete Applied Mathematics*, 158(6):718–722, 2010.

[10] R. Impagliazzo and R. Paturi. Complexity of k-sat. In *Computational Complexity, 1999. Proceedings. Fourteenth Annual IEEE Conference on*, pages 237–240. IEEE, 1999.

[11] R. Motwani, A. Raghunathan, and H. Saran. Covering orthogonal polygons with star polygons: The perfect graph approach. *J. Comput. Syst. Sci.*, 40(1):19–48, 1990.

[12] P. Valtr. Guarding galleries where no point sees a small area. *Israel Journal of Mathematics*, 104(1):1–16, 1998.