



HAL
open science

Parameterized Hardness of Art Gallery Problems

Edouard Bonnet, Miltzow Tillmann

► **To cite this version:**

Edouard Bonnet, Miltzow Tillmann. Parameterized Hardness of Art Gallery Problems. ACM Transactions on Algorithms, 2020, 16, pp.1 - 23. 10.1145/3398684 . hal-03015328

HAL Id: hal-03015328

<https://hal.science/hal-03015328>

Submitted on 23 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Parameterized Hardness of Art Gallery Problems

ÉDOUARD BONNET*, Univ Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668
TILLMANN MILTZOW†, Utrecht University

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} . A point in the set S is referred to as a guard. For both variants, we rule out any $f(k)n^{o(k/\log k)}$ algorithm, where $k := |S|$ is the number of guards, for any computable function f , unless the Exponential Time Hypothesis fails. These lower bounds almost match the $n^{O(k)}$ algorithms that exist for both problems.

CCS Concepts: • **Randomness, geometry and discrete structures** → **Computational geometry**; • **Design and analysis of algorithms** → **Parameterized complexity and exact algorithms**.

Additional Key Words and Phrases: Computational Geometry, Art Gallery, Parameterized Complexity, Intractability, ETH lower bound

ACM Reference Format:

Édouard Bonnet and Tillmann Miltzow. 2020. Parameterized Hardness of Art Gallery Problems. *ACM Trans. Algor.* 1, 1, Article 1 (January 2020), 23 pages. <https://doi.org/10.1145/3398684>

1 INTRODUCTION

Two points x, y in a simple polygon \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} . In both cases, such a set S is a *guarding set* and its elements are called *guards*. In the decision versions, given a simple polygon and an integer, one has to decide if there is a guarding set for the polygon of cardinality at most the integer. In what follows, n refers to the number of vertices of \mathcal{P} and k to the allowed number of guards.

The art gallery problem is arguably one of the most well-known problems in discrete and computational geometry. Since its introduction by Viktor Klee in 1976, numerous research papers were published on the subject. O'Rourke's early book from 1987 [41] has over two thousand citations, and each year, top conferences publish new results on the topic. Many variants of the art gallery problem, based on different definitions of visibility, restricted classes of polygons, different

*supported by the LABEX MILYON (ANR-10- LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR).

†supported by the ERC Consolidator Grant 615640-ForEFront. The author acknowledges generous support from the Netherlands Organisation for Scientific Research (NWO) under project no. 016.Veni.192.250.

Authors' addresses: Édouard Bonnet, edouard.bonnet@ens-lyon.fr, Univ Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668; Tillmann Miltzow, t.miltzow@gmail.com, Utrecht University.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2019 Association for Computing Machinery.

1549-6325/2020/1-ART1 \$15.00

<https://doi.org/10.1145/3398684>

29 shapes of guards, have been defined and analyzed. One of the first results is the elegant proof of Fisk
30 that $\lfloor n/3 \rfloor$ guards are always sufficient and sometimes necessary for a polygon with n vertices [23].

31 The art gallery problem was shown NP-hard by Aggarwal in his PhD thesis [3] and by Lee and
32 Lin [36]. Eidenbenz et al. [21] even showed APX-hardness for the most standard variants. See
33 also [13, 31, 35] for other hardness constructions. Very recently, Abrahamsen et al. [2] showed
34 that POINT GUARD ART GALLERY is $\exists\mathbb{R}$ -complete. In particular, this problem is unlikely to be in
35 NP. This is maybe intuitive, if we consider simple instances of the art gallery problem, which
36 need irrational numbers for an optimal guard placement [1]. In contrast, Dobbins, Holmsen and
37 Miltzow [17] showed how to find a solution with rational coordinates using the concept of smoothed
38 analysis. Due to those negative results, most papers focus on finding approximation algorithms
39 and on variants or restrictions that are polynomially tractable [25, 32, 34, 35, 39]. For the POINT
40 GUARD ART GALLERY problem on simple polygons, there is an $O(\log \text{OPT})$ -approximation under
41 some assumptions (integer coordinates and some special *general position* of the vertices) [12]. The
42 approximation relies on the construction of ε -nets and ideas from Efrat and Har-Peled [20]. For
43 polygons with h holes, there is a polynomial approximation algorithm with ratio $O(\log \text{OPT} \cdot \log h)$
44 which guards all but a δ -fraction of the polygon [22]. Recently, a constant-factor approximation was
45 announced for VERTEX GUARD ART GALLERY [9]. However, a mistake was later found [7]. Another
46 approach is to find heuristics to solve large instances of the art gallery problem [16]. Naturally, the
47 fundamental drawback of this approach is the lack of performance guarantees.

48 In the last twenty-five years, another fruitful approach gained popularity: parameterized complex-
49 ity. The underlying idea is to study algorithmic problems with dependence on a natural parameter.
50 If the dependence on the parameter is practical and the parameter is small for real-life instances,
51 we attain algorithms that give optimal solutions with reasonable running times. For a gentle in-
52 troduction to parameterized complexity, we recommend Niedermeier's book [40]. For a thorough
53 reading highlighting complexity classes, we suggest the book by Downey and Fellows [19]. For a
54 recent book on the topic with an emphasis on algorithms, we advise to read the book by Cygan et
55 al. [15]. An approach based on logic is given by Flum and Grohe [24]. Despite the recent successes
56 of parameterized complexity, only very few results on the art gallery problem are known prior to
57 this paper.

58 The first such result is the trivial algorithm for the vertex guard variant to check if a solution of
59 size k exists in a polygon with n vertices. The algorithm runs in $O(n^{k+2})$ time, by checking all possible
60 subsets of size k of the vertices. The second *not so well-known* result is the fact that one can find in
61 time $n^{O(k)}$ a set of k guards for the point guard variant, if it exists [20], using tools from real algebraic
62 geometry [8]. This was first observed by Sharir [20, Acknowledgment]. Despite the fact that the first
63 algorithm is extremely basic and the second algorithm, even with remarkably sophisticated tools,
64 uses almost no problem specific insights, no better exact parameterized algorithms are known.

65 The Exponential Time Hypothesis (ETH) asserts that there is no $2^{o(N)}$ time algorithm for SAT on
66 N variables. The ETH is used to attain more precise conditional lower bounds than the mere NP-
67 hardness. A simple reduction from SET COVER by Eidenbenz et al. shows that there is no $f(k)n^{o(k)}$
68 algorithm for these problems, when we consider polygons with holes [21, Sec.4], unless the ETH
69 fails. However, polygons with holes are very different from simple polygons. For instance, they
70 have unbounded VC-dimension while simple polygons have bounded VC-dimension [26, 27, 30, 42].

71 We present the first lower bounds for the parameterized art gallery problems restricted to *simple*
72 polygons. Here, the parameter is the optimal number k of guards to cover the polygon.

73 **THEOREM 1.1 (PARAMETERIZED HARDNESS POINT GUARD).** *POINT GUARD ART GALLERY is not*
74 *solvable in time $f(k)n^{o(k/\log k)}$, even on simple polygons, where n is the number of vertices of the*
75 *polygon and k is the number of guards allowed, for any computable function f , unless the ETH fails.*

76 THEOREM 1.2 (PARAMETERIZED HARDNESS VERTEX GUARD). VERTEX GUARD ART GALLERY is not
 77 solvable in time $f(k)n^{o(k/\log k)}$, even on simple polygons, where n is the number of vertices of the
 78 polygon and k is the number of guards allowed, for any computable function f , unless the ETH fails.

79 These results imply that the previous noted algorithms are essentially tight, and suggest that
 80 there are no significantly better parameterized algorithms. Our reductions are from SUBGRAPH
 81 ISOMORPHISM and therefore an $f(k)n^{o(k)}$ -algorithm for the art gallery problem would also imply
 82 improved algorithms for SUBGRAPH ISOMORPHISM and for CSP parameterized by treewidth, which
 83 would be considered a major breakthrough [37]. Let us also mention that our results imply that
 84 both variants are $W[1]$ -hard parameterized by the number of guards.

85 After the conference version of this paper appeared, the parameterized complexity of the art
 86 gallery and related problems was investigated further. The parameterized complexity of the terrain
 87 guarding problem was studied [6]. The terrain guarding problem is a particular case of the art gallery
 88 problem, where instead of a polygon, one should guard an x -monotone curve. This restriction is
 89 still NP-hard [33], even on rectilinear (that is, every edge is horizontal or vertical) terrains [10]. The
 90 authors of [6] present an $n^{O(\sqrt{k})}$ -time algorithm (hence $2^{O(n^{1/2} \log n)}$) for guarding general n -vertex
 91 terrains with k guards, and an FPT $k^{O(k)}n^{O(1)}$ -time algorithm for guarding the vertices of rectilinear
 92 terrains. Note that there is no $2^{o(n^{1/3})}$ algorithm for terrain guarding, unless the ETH fails [10].

93 The art gallery problem parameterized by the number of reflex vertices is considered by Agrawal
 94 et al. [5]. The authors present an FPT algorithm for VERTEX GUARD ART GALLERY under this
 95 parameterization. See also [4] for FPT algorithms on the (strong) conflict-free coloring of terrains.

96 2 PROOF IDEAS

97 In order to achieve these results, we slightly extend some known hardness results of geometric
 98 set cover/hitting set problems and combine them with problem-specific insights of the art gallery
 99 problem. One of the first problem-specific insights is the ability to encode HITTING SET on interval
 100 graphs. The reader can refer to Figure 1 for the following description. Assume that we have some
 101 fixed points p_1, \dots, p_n with increasing y -coordinates in the plane. We can build a pocket “far enough
 to the right” that can be seen only from $\{p_i, \dots, p_j\}$ for any $1 \leq i < j \leq n$.

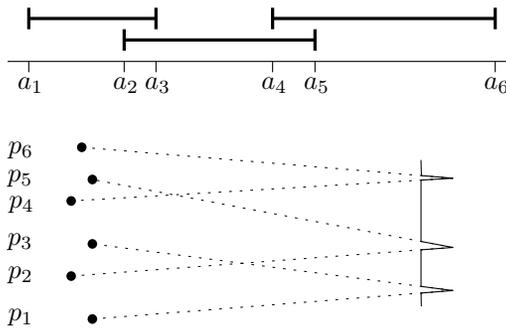


Fig. 1. Reduction from HITTING SET on interval graphs to a restricted version of the art gallery problem.

102 Let I_1, \dots, I_n be n intervals with endpoints a_1, \dots, a_{2n} . Then, we construct $2n$ points p_1, \dots, p_{2n}
 103 representing a_1, \dots, a_{2n} . Further, we construct one pocket “far enough to the right” for each interval
 104 as described above. This way, we reduce HITTING SET on interval graphs to a restricted version of
 105 the art gallery problem. This observation is *not* so useful in itself since HITTING SET on interval
 106 graphs can be solved in polynomial time.
 107

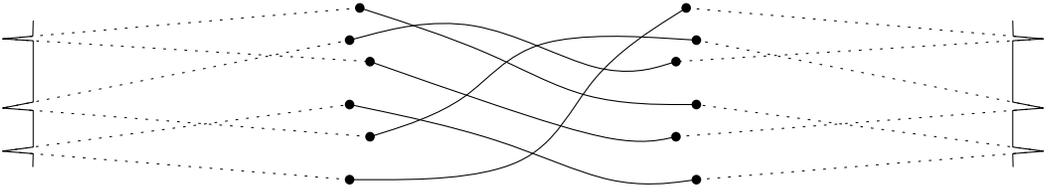


Fig. 2. Two instances of Hitting Set “magically” linked.

108 The situation changes rapidly if we consider HITTING SET on 2-track interval graphs, as described
 109 in the preliminaries. Unfortunately, we are not able to just “magically” link (see Figure 2) some
 110 specific pairs of points in the polygon of the art gallery instance. Instead, we construct linking
 111 gadgets, which work “morally” as follows. We are given two set of points P and Q and a bijection
 112 σ between P and Q . The linking gadget is built in a way that it can be covered by two points
 113 (p, q) of $P \times Q$, if and only if $q = \sigma(p)$. The STRUCTURED 2-TRACK HITTING SET problem will be
 114 specifically designed so that the linking gadget is the main remaining ingredient to show hardness.
 115 This intermediate problem is a convenient starting point for parameterized reductions to other
 116 geometric problems. For instance, the parameterized hardness of RED-BLUE POINTS SEPARATION,
 117 where given a set of blue points and a set of red points in the plane, one has to find at most k lines
 118 so that no cell of the arrangement is bichromatic, was obtained by a reduction from STRUCTURED
 119 2-TRACK HITTING SET [11].

120 **Organization.** The rest of the paper is organized as follows. In Section 3, we introduce some
 121 notations, discuss the encoding of the polygon, give some useful ETH-based lower bounds, and
 122 prove a technical lemma. In Section 4, we prove the lower bound for STRUCTURED 2-TRACK HITTING
 123 SET (Theorem 4.2). Lemma 4.1 contains the key arguments. From this point onward, we can reduce
 124 from STRUCTURED 2-TRACK HITTING SET. In Section 5, we show the lower bound for the POINT
 125 GUARD ART GALLERY problem (Theorem 1.1). We design a linking gadget, show its correctness,
 126 and show how several linking gadgets can be combined consistently. In Section 6, we tackle the
 127 VERTEX GUARD ART GALLERY problem (Theorem 1.2). We have to design a very different linking
 128 gadget, that has to be combined with other gadgets and ideas.

129 3 PRELIMINARIES

130 For any two integers $x \leq y$, we set $[x, y] := \{x, x + 1, \dots, y - 1, y\}$, and for any positive integer
 131 x , $[x] := [1, x]$. Given two points a, b in the plane, we define $\text{seg}(a, b)$ as the line segment with
 132 endpoints a, b . Given n points $v_1, \dots, v_n \in \mathbb{R}^2$, we define a polygonal closed curve c by $\text{seg}(v_1, v_2),$
 133 $\dots, \text{seg}(v_{n-1}, v_n), \text{seg}(v_n, v_1)$. If c is not self intersecting, it partitions the plane into a closed
 134 bounded area and an unbounded area. The closed bounded area is a *simple polygon* on the vertices
 135 v_1, \dots, v_n . Note that we do not consider the boundary as the polygon but rather all the points
 136 bounded by the curve c as described above. Given two points a, b in a simple polygon \mathcal{P} , we say
 137 that a *sees* b or a is *visible* from b if $\text{seg}(a, b)$ is contained in \mathcal{P} . By this definition, it is possible to
 138 “see through” vertices of the polygon. We say that S is a set of *point guards* of \mathcal{P} , if every point
 139 $p \in \mathcal{P}$ is visible from a point of S . We say that S is a set of *vertex guards* of \mathcal{P} , if additionally S is a
 140 subset of the vertices of \mathcal{P} . The POINT GUARD ART GALLERY problem and the VERTEX GUARD ART
 141 GALLERY problem are formally defined as follows.

142 POINT GUARD ART GALLERY

143 **Input:** The vertices of a simple polygon \mathcal{P} in the plane and a natural number k .

144 **Question:** Does there exist a set of k point guards for \mathcal{P} ?

145 **VERTEX GUARD ART GALLERY**

146 **Input:** A simple polygon \mathcal{P} on n vertices in the plane and a natural number k .

147 **Question:** Does there exist a set of k vertex guards for \mathcal{P} ?

148 For any two distinct points v and w in the plane we denote by $\text{ray}(v, w)$ the ray starting at
 149 v and passing through w , and by $\ell(v, w)$ the supporting line passing through v and w . For any
 150 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
 151 the set of all the points $y \in \mathcal{P}$ seen by x . We say that two vertices v and w of a polygon \mathcal{P} are
 152 *neighbors* or *consecutive* if vw is an edge of \mathcal{P} . A *sub-polygon* \mathcal{P}' of a simple polygon \mathcal{P} is defined
 153 by any l distinct consecutive vertices v_1, v_2, \dots, v_l of \mathcal{P} (that is, for every $i \in [l - 1]$, v_i and v_{i+1} are
 154 neighbors in \mathcal{P}) such that v_1v_l does not cross any edge of \mathcal{P} . In particular, \mathcal{P}' is a simple polygon.

155 **Encoding.** We assume that the vertices of the polygon are either given by integers or by rational
 156 numbers. We also assume that the output is given either by integers or by rational numbers. The
 157 instances we generate as a result of Theorem 1.1 and Theorem 1.2 have rational coordinates. We
 158 can represent each coordinate by specifying the nominator and denominator. The number of bits is
 159 bounded by $O(\log n)$ in both cases. We can transform the coordinates to integers by multiplying
 160 every coordinate with the least common multiple of all denominators. However, this leads to
 161 integers using $O(n \log n)$ bits.

162 **ETH-based lower bounds.** The *Exponential Time Hypothesis* (ETH) is a conjecture by Impagli-
 163 azzo et al. [28] asserting that there is no $2^{o(n)}$ -time algorithm for 3-SAT on instances with n variables.
 164 The k -MULTICOLORED-CLIQUE problem has as input a graph $G = (V, E)$, where the set of vertices is
 165 partitioned into V_1, \dots, V_k . It asks if there exists a set of k vertices $v_1 \in V_1, \dots, v_k \in V_k$ such that
 166 these vertices form a clique of size k . We will use the following lower bound proved by Chen et
 167 al. [14].

168 **THEOREM 3.1** ([14]). *There is no $f(k)n^{o(k)}$ algorithm for k -MULTICOLORED-CLIQUE, for any com-*
 169 *putable function f , unless the ETH fails.*

170 Marx showed that SUBGRAPH ISOMORPHISM cannot be solved in time $f(k)n^{o(k/\log k)}$ where k is the
 171 number of edges of the pattern graph, under the ETH [37]. Usually, this result enables to improve
 172 a lower bound obtained by a reduction from MULTICOLORED k -CLIQUE with a quadratic blow-up
 173 on the parameter, from exponent $o(\sqrt{k})$ to exponent $o(k/\log k)$, by doing more or less the same
 174 reduction but from MULTICOLORED SUBGRAPH ISOMORPHISM. In the MULTICOLORED SUBGRAPH
 175 ISOMORPHISM problem, one is given a graph with n vertices partitioned into l color classes V_1, \dots, V_l
 176 such that only k of the $\binom{l}{2}$ sets $E_{ij} = E(V_i, V_j)$ are non empty. The goal is to pick one vertex in each
 177 color class so that the selected vertices induce k edges. The technique of color coding and the result
 178 of Marx shows that:

179 **THEOREM 3.2** ([37]). *MULTICOLORED SUBGRAPH ISOMORPHISM cannot be solved in time $f(k)n^{o(k/\log k)}$*
 180 *where k is the number of edges of the solution, for any computable function f , unless the ETH fails.*

181 Naturally, this result still holds when restricted to connected input graphs. In that case, $k \geq l - 1$.

182 **Bounding the coordinates.** We say a point $p = (p_x, p_y) \in \mathbb{Z}^2$ has coordinates bounded by L
 183 if $|p_x|, |p_y| \leq L$. Given two vectors v, w , we denote their scalar product as $v \cdot w$. This technical
 184 lemma will prove useful to ensure that the polygon built in Section 5 can be described with integer
 185 coordinates.

186 **LEMMA 3.3.** *Let p^1, q^1, p^2, q^2 be four points with integer coordinates bounded by L . Then the inter-*
 187 *section point $d = (d_x, d_y)$ of the supporting lines $\ell_1 = \ell(p^1, q^1)$ and $\ell_2 = \ell(p^2, q^2)$ is a rational point.*
 188 *The nominator and denominator of d_x and d_y are bounded by $O(L^2)$.*

189 PROOF. The fact that d lies on ℓ_i can be expressed as $v_i \cdot d = b_i$, with some appropriate vector v^i
 190 and number b^i , for $i = 1, 2$. To be precise $v^i = (-p_x^i + q_x^i, p_y^i - q_y^i)$ and $b^i = v_i \cdot p^i$, for $i = 1, 2$. We
 191 define the matrix $A = (v^1, v^2)$ and the vector $b = (b^1, b^2)$. Then both conditions can be expressed
 192 as $A \cdot d = b$. We denote by A_i the matrix i with the i -th column replaced by b . And by $\det(M)$ the
 193 determinant of the matrix M . By Cramer's rule, it holds that $d_x = \frac{\det(A_1)}{\det(A)}$ and $d_y = \frac{\det(A_2)}{\det(A)}$. \square

194 4 PARAMETERIZED HARDNESS OF STRUCTURED 2-TRACK HITTING SET

195 The purpose of this section is to show Theorem 4.2. As we will see at the end of the section, there
 196 already exist quite a few parameterized hardness results for set cover/hitting set problems restricted
 197 to instances with some geometric flavor. The crux of the proof of Theorem 4.2 lies in Lemma 4.1.
 198 We introduce a few notation and vocabulary to state and prove this lemma.

199 Given a finite totally ordered set $Y = \{y_1, \dots, y_{|Y|}\}$ (that is, for any $i, j \in [|Y|]$, $y_i \leq y_j$ iff $i \leq j$),
 200 a subset $S \subseteq Y$ is a Y -interval if $S = \{y \mid y_i \leq y \leq y_j\}$ for some i and j . We denote by \leq_Y the order
 201 of Y . A set-system (X, \mathcal{S}) is said to be *two-block* if X can be partitioned into two totally ordered
 202 sets $A = \{a_1, \dots, a_{|A|}\}$ and $B = \{b_1, \dots, b_{|B|}\}$ such that each set $S \in \mathcal{S}$ is the union of an A -interval
 203 with a B -interval.

204 Given a set \mathcal{S} of subsets of X , k -SET COVER asks to find k sets of \mathcal{S} whose union is X . We first
 205 show an ETH lower bound and $W[1]$ -hardness for k -SET COVER restricted to two-block instances.
 206 We reduce from MULTICOLORED k -CLIQUE for simplicity sake (then from MULTICOLORED SUBGRAPH
 207 ISOMORPHISM to improve the ETH lower bound). On a high-level, we encode adjacencies in the
 208 MULTICOLORED k -CLIQUE instance by pairs of disjoint sets particularly effective to cover X . On the
 209 contrary, pairs of non-adjacent vertices will be mapped to pairs of sets overlapping and missing an
 210 important part of X . This trick will be a recurring theme throughout the paper.

211 LEMMA 4.1. *k -SET COVER restricted to two-block instances with N elements and M sets is $W[1]$ -hard
 212 and not solvable in time $f(k)(N + M)^{o(k/\log k)}$ for any computable function f , unless the ETH fails.*

213 PROOF. We reduce from MULTICOLORED k -CLIQUE which remains $W[1]$ -hard when each color
 214 class has the same number t of vertices. Let $G = (V_1 \cup \dots \cup V_k, E)$ be an instance of MULTICOLORED
 215 k -CLIQUE with $V = \bigcup_{i \in [k]} V_i$, $\forall i \in [k]$, $V_i = \{v_1^i, \dots, v_t^i\}$, $m = |E|$, and $n = |V| = tk$. For each
 216 pair $i < j \in [k]^1$, E_{ij} denotes the set of edges $E(V_i, V_j)$ between V_i and V_j . For each E_{ij} we give an
 217 arbitrary order to the edges: $e_1^{ij}, \dots, e_{|E_{ij}|}^{ij}$. We build an equivalent instance (X, \mathcal{S}) of k -SET COVER
 218 with $4\binom{k}{2} + 4m + tk(k + 1) + 4k$ elements and $4m + 2kt$ sets, and such that (X, \mathcal{S}) is two-block. We
 219 call A and B the two sets of the partition of X that realizes that (X, \mathcal{S}) is two-block.

220 For each of the color class V_i , we add $tk + 2$ elements to A with the following order:

$$\begin{aligned}
 & x_b(i), \\
 & x(i, 1, 1), \dots, x(i, 1, t), \\
 & x(i, 2, 1), \dots, x(i, 2, t), \\
 & \dots \\
 & x(i, i - 1, 1), \dots, x(i, i - 1, t), \\
 & x(i, i + 1, 1), \dots, x(i, i + 1, t), \\
 & \dots \\
 & x(i, k + 1, 1), \dots, x(i, k + 1, t), \\
 & x_e(i),
 \end{aligned}$$

¹By $i < j \in [k]$, we mean that $i \in [k]$, $j \in [k]$, and $i < j$.

and call $X(i)$ the set containing those elements. We also set

$$X(i, j) := \{x(i, j, 1), x(i, j, 2), \dots, x(i, j, t)\}$$

(hence, $X(i) = \bigcup_{j \neq i} X(i, j) \cup \{x_b(i), x_e(i)\}$). For each E_{ij} , we add to B the $3|E_{ij}| + 2$ of a set $Y(i, j)$ ordered:

$$y_b(i, j), y(i, j, 1), \dots, y(i, j, 3|E_{ij}|), y_e(i, j).$$

For each pair $i < j \in [k]$ and for each edge $e_c^{ij} = v_a^i v_b^j$ in E_{ij} (with $a, b \in [t]$ and $c \in [|E_{ij}|]$), we add to \mathcal{S} the two sets

$$\begin{aligned} S(e_c^{ij}, v_a^i) &:= \{x(i, j, a), x(i, j, a+1), \dots, x(i, j, t), x(i, j+1, 1), \dots, x(i, j+1, a-1)\} \\ &\cup \{y(i, j, c), \dots, y(i, j, c+|E_{ij}|-1)\} \text{ and} \\ S(e_c^{ij}, v_b^j) &:= \{x(j, i, b), x(j, i, b+1), \dots, x(j, i, t), x(j, i+1, 1), \dots, x(j, i+1, b-1)\} \\ &\cup \{y(i, j, c+|E_{ij}|), \dots, y(i, j, c+2|E_{ij}|-1)\}. \end{aligned}$$

Observe that in case $j = i + 1$, then all the elements of the form $x(j, i+1, \cdot)$ in set $S(e_c^{ij}, v_b^j)$ are in fact of the form $x(j, i+2, \cdot)$. We may also notice that in case $a = 1$ (resp. $b = 1$), then there is no element of the form $x(i, j+1, \cdot)$ (resp. $x(j, i+1, \cdot)$) in set $S(e_c^{ij}, v_a^i)$ (resp. in set $S(e_c^{ij}, v_b^j)$). For each pair $i < j \in [k]$, we also add to A the $|E_{ij}| + 2$ elements of a set $Z(i, j)$ ordered:

$$z_b(i, j), z(i, j, 1), \dots, z(i, j, |E_{ij}|), z_e(i, j),$$

and for each edge e_c^{ij} in E_{ij} (with $c \in [|E_{ij}|]$), we add to \mathcal{S} the two sets

$$\begin{aligned} S(e_c^{ij}, \vdash) &= \{z_b(i, j), z(i, j, 1), \dots, z(i, j, |E_{ij}| - c)\} \cup \{y_b(i, j), y(i, j, 1) \dots y(i, j, c-1)\} \text{ and} \\ S(e_c^{ij}, \dashv) &= \{z(i, j, |E_{ij}| - c + 1), \dots, z(i, j, |E_{ij}|), z_e(i, j)\} \cup \{y(i, j, c + 2|E_{ij}|) \dots y(i, j, 3|E_{ij}|), y_e(i, j)\}. \end{aligned}$$

Finally, for each $i \in [k]$, we add to B the $t + 2$ elements of a set $W(i)$ ordered:

$$w_b(i), w(i, 1), \dots, w(i, t), w_e(i),$$

and for all $a \in [t]$, we add the sets

$$\begin{aligned} S(i, a, \vdash) &:= \{x_b(i), x(i, 1, 1), \dots, x(i, 1, a-1)\} \cup \{w_b(i), w(i, 1), \dots, w(i, t-a+1)\} \text{ and} \\ S(i, a, \dashv) &:= \{x(i, k+1, a), \dots, x(i, k+1, t), x_e(i)\} \cup \{w(i, t-a+2), \dots, w(i, t), w_e(i)\}. \end{aligned}$$

No matter the order in which we put the $X(i)$'s and $Z(i, j)$'s in A (respectively the $Y(i, j)$'s and $W(i)$'s in B), the sets we defined are all unions of an A -interval with a B -interval, provided we keep the elements within each $X(i)$, $Z(i, j)$, $Y(i, j)$, and $W(i)$ consecutive (and naturally, in the order we specified). Though, to clarify the construction, we fix the following orders for A and for B :

$$X(1), \dots, X(k), Z(1, 2), \dots, Z(1, k), Z(2, 3), \dots, Z(2, k), \dots, Z(k-2, k-1), Z(k-2, k), Z(k-1, k)$$

$$Y(1, 2), \dots, Y(1, k), Y(2, 3), \dots, Y(2, k), \dots, Y(k-2, k-1), Y(k-2, k), Y(k-1, k), W(1), \dots, W(k).$$

229 We ask for a set cover with $2k^2$ sets. This ends the construction (see Figure 4 for an illustration of
230 the construction for the instance graph of Figure 3).

231 For each $i \in [k]$, let us denote by $\mathcal{S}_b(i)$ (resp. $\mathcal{S}_e(i)$), all the sets in \mathcal{S} that contains element $x_b(i)$
232 (resp. $x_e(i)$). For each pair $i \neq j \in [k]$, we denote by $\mathcal{S}(i, j)$ all the sets in \mathcal{S} that contains element
233 $x(i, j, t)$. Finally, for each pair $i < j \in [k]$, we denote by $\mathcal{S}(i, j, \vdash)$ (resp $\mathcal{S}(i, j, \dashv)$) all the sets in \mathcal{S}
234 that contains element $y_b(i, j)$ (resp. $y_e(i, j)$). One can observe that the $\mathcal{S}_b(i)$'s, $\mathcal{S}_e(i)$'s, $\mathcal{S}(i, j)$'s, $\mathcal{S}(i, j, \vdash)$'s,
235 and $\mathcal{S}(i, j, \dashv)$'s partition \mathcal{S} into $k + k + k(k-1) + 2\binom{k}{2} = 2k^2$ partite sets². Thus, as each of the $2k^2$
236 partite sets \mathcal{S}' has a private element which is only contained in sets of \mathcal{S}' , a solution has to contain
237 one set in each partite set.

²We do not call them *color classes* to avoid the confusion with the color classes of the instance of MULTICOLORED k -CLIQUE.

249 the elements $y_b(i, j), y(i, j, 1), \dots, y(i, j, c-1), y(i, j, c+2|E_{ij}|), \dots, y(i, j, 3|E_{ij}|), y_e(i, j)$ and of $Z(i, j)$
 250 are covered by $S(v_{a_i}^i v_{a_j}^j, \vdash)$ and $S(v_{a_i}^i v_{a_j}^j, \dashv)$.

251 Assume now that the set-system (X, \mathcal{S}) admits a set cover \mathcal{T} of size $2k^2$. As mentioned above,
 252 this solution \mathcal{T} should contain exactly one set in each partite set (of the partition of \mathcal{S}). For each
 253 $i \in [k]$, to cover all the elements of $W(i)$, one should take $S(i, a_i, \vdash)$ and $S(i, a_i', \dashv)$ with $a_i \leq a_i'$. Now,
 254 each set of $\mathcal{S}(i, j)$ has their A -intervals containing exactly t elements. This means that the only way
 255 of covering the $tk + 2$ elements of $X(i)$ is to take $S(i, a_i, \vdash)$ and $S(i, a_i', \dashv)$ with $a_i \geq a_i'$ (therefore
 256 $a_i = a_i'$), and to take all the $k - 1$ sets of $\mathcal{S}(i, j)$ (for $j \in [k] \setminus \{i\}$) of the form $S(v_{a_i}^i v_{s_j}^j, v_{a_i}^i)$, for some
 257 $s_j \in [t]$. So far, we showed that a potential solution of k -SET COVER should stick to the same vertex
 258 $v_{a_i}^i$ in each *color class*. We now show that if one selects $S(v_{a_i}^i v_{s_j}^j, v_{a_i}^i)$, one should be consistent with
 259 this choice and also selects $S(v_{a_i}^i v_{s_j}^j, v_{s_j}^j)$. In particular, it implies that, for each $i \in [k]$, s_i should
 260 be equal to a_i . For each $i \neq j \in [k]$, to cover all the elements of $Z(i, j)$, one should take $S(e_{c_{ij}}^{ij}, \vdash)$
 261 and $S(e_{c_{ij}}^{ij}, \dashv)$ with $c_{ij} \geq c_{ij}'$. Now, each set of $\mathcal{S}(i, j)$ and each set of $\mathcal{S}(j, i)$ has their B -intervals
 262 containing exactly $|E_{ij}|$ elements. This means that the only way of covering the $3|E_{ij}| + 2$ elements
 263 of $Y(i, j)$ is to take $S(e_{c_{ij}}^{ij}, \vdash)$ and $S(e_{c_{ij}}^{ij}, \dashv)$ with $c_{ij} \leq c_{ij}'$ (therefore, $c_{ij} = c_{ij}'$), and to take the sets
 264 $S(v_{a_i}^i v_{a_j}^j, v_{a_i}^i)$ and $S(v_{a_i}^i v_{a_j}^j, v_{a_j}^j)$. Therefore, if there is a solution to the k -SET COVER instance, then
 265 there is a multicolored clique $\{v_{a_1}^1, \dots, v_{a_k}^k\}$ in G .

266 In this reduction, there is a quadratic blow-up of the parameter. Under the ETH, it would
 267 only forbid, by Theorem 3.1, an algorithm solving k -SET COVER on two-block instances in time
 268 $f(k)(N + M)^{o(\sqrt{k})}$. We can do the previous reduction from MULTICOLORED SUBGRAPH ISOMORPHISM
 269 and suppress $X(i, j), X(j, i), Z(i, j)$, and $Y(i, j)$, and the sets defined over these elements, whenever
 270 E_{ij} is empty. One can check that the produced set cover instance is still two-block and that the
 271 way of proving correctness does not change. Therefore, by Theorem 3.2, k -SET COVER restricted to
 272 two-block instances cannot be solved in time $f(k)(N + M)^{o(k/\log k)}$ for any computable function f ,
 273 unless the ETH fails. \square

274 In the 2-TRACK HITTING SET problem, the input consists of an integer k , two totally ordered
 275 ground sets A and B of the same cardinality, and two sets \mathcal{S}_A of A -intervals, and \mathcal{S}_B of B -intervals.
 276 In addition, the elements of A and B are in one-to-one correspondence $\phi : A \rightarrow B$ and each pair
 277 $(a, \phi(a))$ is called a 2-element. The goal is to find, if possible, a set S of k 2-elements such that the
 278 first projection of S is a hitting set of \mathcal{S}_A , and the second projection of S is a hitting set of \mathcal{S}_B .

279 STRUCTURED 2-TRACK HITTING SET is the same problem with color classes over the 2-elements,
 280 and a restriction on the one-to-one mapping ϕ . Given two integers k and t , A is partitioned into
 281 (C_1, C_2, \dots, C_k) where $C_j = \{a_1^j, a_2^j, \dots, a_t^j\}$ for each $j \in [k]$. A is ordered: $a_1^1, a_2^1, \dots, a_t^1, a_1^2, a_2^2, \dots, a_t^2,$
 282 $\dots, a_1^k, a_2^k, \dots, a_t^k$. 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
 283 impose that ϕ is such that, for each $j \in [k]$, the set C_j' is a B -interval. That is, B is ordered:
 284 $C'_{\sigma(1)}, C'_{\sigma(2)}, \dots, C'_{\sigma(k)}$ for some permutation on $[k]$, $\sigma \in \mathfrak{S}_k$. For each $j \in [k]$, the order of the
 285 elements within C_j' can be described by a permutation $\sigma_j \in \mathfrak{S}_t$ such that the ordering of C_j' is:
 286 $b_{\sigma_j(1)}^j, b_{\sigma_j(2)}^j, \dots, b_{\sigma_j(t)}^j$. In what follows, it will be convenient to see an instance of STRUCTURED
 287 2-TRACK HITTING SET as a tuple $\mathcal{I} = (k \in \mathbb{N}, t \in \mathbb{N}, \sigma \in \mathfrak{S}_k, \sigma_1 \in \mathfrak{S}_t, \dots, \sigma_k \in \mathfrak{S}_t, \mathcal{S}_A, \mathcal{S}_B)$, where
 288 we recall that \mathcal{S}_A is a set of A -intervals and \mathcal{S}_B is a set of B -intervals. The size $|\mathcal{I}|$ of \mathcal{I} is defined
 289 as $kt + |\mathcal{S}_A| + |\mathcal{S}_B|$. We denote by $[a_i^j, a_{i'}^j]$ (resp. $[b_i^j, b_{i'}^j]$) all the elements $a \in A$ (resp. $b \in B$) such
 290 that $a_i^j \leq_A a \leq_A a_{i'}^j$ (resp. $b_i^j \leq_B b \leq_B b_{i'}^j$).

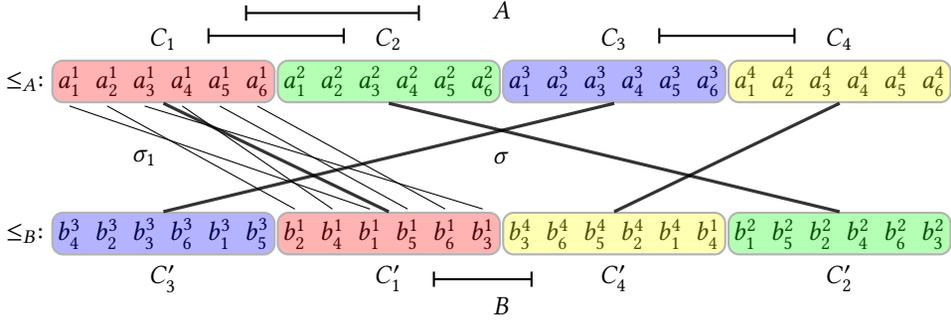


Fig. 5. An illustration of a STRUCTURED 2-TRACK HITTING SET instance, with $k = 4$ and $t = 6$. The permutation $\sigma \in \mathfrak{S}_k$ is represented with thick edges. Among $\sigma_1 \in \mathfrak{S}_t, \dots, \sigma_k \in \mathfrak{S}_t$, we only represented σ_1 , for the sake of legibility. We also only represented four intervals of the instance, three A -intervals, $[a_5^1, a_2^1] = \{a_5^1, a_6^1, a_1^2, a_2^2\}$, $[a_6^1, a_4^1]$, $[a_3^3, a_2^4]$, and one B -interval $[b_6^1, b_3^4] = \{b_6^1, b_3^1, b_3^4\}$.

291 Again a solution is a set of k 2-elements $\{(a_{i(1)}^1, b_{i(1)}^1), \dots, (a_{i(k)}^k, b_{i(k)}^k)\}$, each from a distinct color
 292 class, such that $a_{i(1)}^1, \dots, a_{i(k)}^k$ is a hitting set of \mathcal{S}_A , and $b_{i(1)}^1, \dots, b_{i(k)}^k$ is a hitting set of \mathcal{S}_B .

293 We show the ETH lower bound and $W[1]$ -hardness for STRUCTURED 2-TRACK HITTING SET. The
 294 reduction is from k -SET COVER on two-block instances. We transform the unions of two intervals
 295 into 2-elements, and the elements of the k -SET COVER instance into A -intervals or B -intervals of
 296 the STRUCTURED 2-TRACK HITTING SET instance.

297 **THEOREM 4.2.** *STRUCTURED 2-TRACK HITTING SET is $W[1]$ -hard. Furthermore it is not solvable in*
 298 *time $f(k)|\mathcal{I}|^{o(k/\log k)}$ for any computable function f , unless the ETH fails.*

299 **PROOF.** This result is a consequence of Lemma 4.1. Let $(A \uplus B, \mathcal{S})$ be a hard two-block instance of
 300 k -SET COVER, obtained from the previous reduction. We recall that each set S of \mathcal{S} is the union of an
 301 A -interval with a B -interval: $S = S_A \uplus S_B$. We transform each set S into a 2-element $(x_{S,A}, x_{S,B})$, and
 302 each element u of the k -SET COVER instance into a set T_u of the STRUCTURED 2-TRACK HITTING SET
 303 instance. We put element $x_{S,A}$ (resp. $x_{S,B}$) into set T_u whenever $u \in S \cap A = I_A$ (resp. $u \in S \cap B = I_B$).
 304 We call A' (resp. B') the set of all the elements of the form $x_{S,A}$ (resp. $x_{S,B}$). We shall now specify an
 305 order of A' and B' so that the instance is *structured*. Keep in mind that elements in the STRUCTURED
 306 2-TRACK HITTING SET instance corresponds to sets in the k -SET COVER instance. We order the
 307 elements of A' accordingly to the following ordering of the sets of the k -SET COVER instance: $\mathcal{S}_b(1)$,
 308 $\mathcal{S}(1, 2), \dots, \mathcal{S}(1, k), \mathcal{S}_e(1), \mathcal{S}_b(2), \mathcal{S}(2, 1), \dots, \mathcal{S}(2, k), \mathcal{S}_e(2), \dots, \mathcal{S}_b(k), \mathcal{S}(k, 1), \dots, \mathcal{S}(k, k-1), \mathcal{S}_e(k)$,
 309 $\mathcal{S}(1, 2, \vdash), \mathcal{S}(1, 2, \dashv), \mathcal{S}(1, 3, \vdash), \mathcal{S}(1, 3, \dashv), \dots, \mathcal{S}(k-1, k, \vdash), \mathcal{S}(k-1, k, \dashv)$. We order the elements of
 310 B' accordingly to the following ordering of the sets of the k -SET COVER instance: $\mathcal{S}(1, 2, \vdash), \mathcal{S}(1, 2)$,
 311 $\mathcal{S}(2, 1), \mathcal{S}(1, 2, \dashv), \mathcal{S}(1, 3, \vdash), \mathcal{S}(1, 3), \mathcal{S}(3, 1), \mathcal{S}(1, 3, \dashv), \dots, \mathcal{S}(k-1, k, \vdash), \mathcal{S}(k-1, k), \mathcal{S}(k, k-1)$,
 312 $\mathcal{S}(k-1, k, \dashv), \mathcal{S}_b(1), \mathcal{S}_e(1), \dots, \mathcal{S}_b(k), \mathcal{S}_e(k)$. Within all those sets of sets, we order by increasing
 313 left endpoint (and then, in case of a tie, by increasing right endpoint). One can now check that
 314 with those two orders $\leq_{A'}$ and $\leq_{B'}$, all the sets T_u 's are A' -interval or B' -interval. Also, one can
 315 check that the 2-TRACK HITTING SET instance is *structured* by taking as color classes the partite
 316 sets $\mathcal{S}_b(i)$'s, $\mathcal{S}_e(i)$'s, $\mathcal{S}(i, j)$'s, $\mathcal{S}(i, j, \vdash)$'s, and $\mathcal{S}(i, j, \dashv)$'s. Now, taking one 2-element in each color
 317 class to hit all the sets T_u corresponds to taking one set in each partite set of \mathcal{S} to dominate all the
 318 elements of the k -SET COVER instance. \square

319 2-track (unit) interval graphs are the intersection graphs of (unit) 2-track intervals, where a
 320 (unit) 2-track interval is the union of a (unit) interval in each of two parallel lines, called the first

321 track and the second track. A (unit) 2-track interval may be referred to as an *object*. Two 2-track
 322 intervals intersect if they intersect in either the first or the second track. We observe here that
 323 many dominating problems with some geometric flavor can be restated with the terminology of
 324 2-track (unit) interval graphs.

325 In particular, a result very close to Theorem 4.2 was obtained recently:

326 **THEOREM 4.3** ([38]). *Given the representation of a 2-track unit interval graph, the problem of*
 327 *selecting k objects to dominate all the intervals is $W[1]$ -hard, and not solvable in time $f(k)n^{o(k/\log k)}$*
 328 *for any computable function f , unless the ETH fails.*

329 We still had to give an *alternative* proof of this result because we will need the additional property
 330 that the instance can be further assumed to have the structure depicted in Figure 5. This will be
 331 crucial for showing the hardness result for VERTEX GUARD ART GALLERY.

332 Other results on dominating problems in 2-track unit interval graphs include:

333 **THEOREM 4.4** ([29]). *Given the representation of a 2-track unit interval graph, the problem of*
 334 *selecting k objects to dominate all the objects is $W[1]$ -hard.*

335 **THEOREM 4.5** ([18]). *Given the representation of a 2-track unit interval graph, the problem of*
 336 *selecting k intervals to dominate all the objects is $W[1]$ -hard.*

337 The result of Dom et al. is formalized differently in their paper [18], where the problem is defined
 338 as stabbing axis-parallel rectangles with axis-parallel lines.

339 5 PARAMETERIZED HARDNESS OF THE POINT GUARD VARIANT

340 As exposed in Section 2, we give a reduction from the STRUCTURED 2-TRACK HITTING SET problem.
 341 The main challenge is to design a *linker* gadget that groups together specific pairs of points in the
 342 polygon. The following introductory lemma inspires the *linker* gadgets for both POINT GUARD ART
 343 GALLERY and VERTEX GUARD ART GALLERY.

344 **LEMMA 5.1.** *The only minimum hitting sets of the set-system $\mathcal{S} = \{S_i = \{1, 2, \dots, i, \overline{i+1}, \overline{i+2},$
 345 $\dots, \overline{n}\} \mid i \in [n]\} \cup \{\bar{S}_i = \{\bar{1}, \bar{2}, \dots, \bar{i}, i+1, i+2, \dots, n\} \mid i \in [n]\}$ are $\{i, \bar{i}\}$, for each $i \in [n]$.*

346 **PROOF.** First, for each $i \in [n]$, one may easily observe that $\{i, \bar{i}\}$ is a hitting set of \mathcal{S} . Now, because
 347 of the sets S_n and \bar{S}_n one should pick one element i and one element \bar{j} for some $i, j \in [n]$. If $i < j$,
 348 then set \bar{S}_i is not hit, and if $i > j$, then S_j is not hit. Therefore, i should be equal to j . \square

349 Henceforth we keep this bar notation to denote pairs of homologous objects (points, vertices)
 350 that we wish to link together.

351 **THEOREM 1.1** (PARAMETERIZED HARDNESS POINT GUARD). *POINT GUARD ART GALLERY is not*
 352 *solvable in time $f(k)n^{o(k/\log k)}$, even on simple polygons, where n is the number of vertices of the*
 353 *polygon and k is the number of guards allowed, for any computable function f , unless the ETH fails.*

354 **PROOF.** Given an instance $\mathcal{I} = (k \in \mathbb{N}, t \in \mathbb{N}, \sigma \in \mathfrak{S}_k, \sigma_1 \in \mathfrak{S}_t, \dots, \sigma_k \in \mathfrak{S}_t, \mathcal{S}_A, \mathcal{S}_B)$ of
 355 STRUCTURED 2-TRACK HITTING SET, we build a simple polygon \mathcal{P} with $O(kt + |\mathcal{S}_A| + |\mathcal{S}_B|)$ vertices,
 356 such that \mathcal{I} is a YES-instance iff \mathcal{P} can be guarded by $3k$ points.

357 **Outline.** We recall that A 's order is: $a_1^1, \dots, a_t^1, \dots, a_1^k, \dots, a_t^k$ and B 's order is determined by σ
 358 and the σ_j 's (see Figure 5). The global strategy of the reduction is to *allocate*, for each color class
 359 $j \in [k]$, $2t$ special points in the polygon $\alpha_1^j, \dots, \alpha_t^j$ and $\beta_1^j, \dots, \beta_t^j$. Placing a guard in α_i^j (resp. β_i^j)
 360 shall correspond to picking a 2-element whose first (resp. second) component is a_i^j (resp. b_i^j). The
 361 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 the

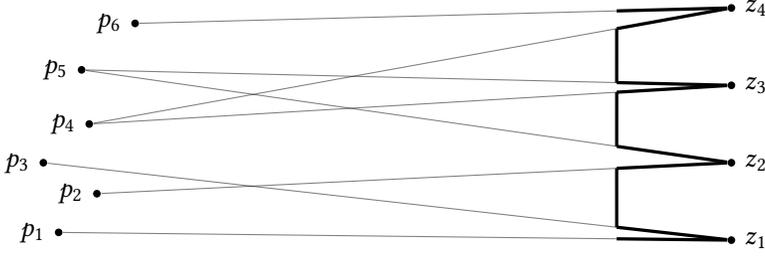


Fig. 6. Interval gadgets encoding $\{p_1, p_2, p_3\}$, $\{p_2, p_3, p_4, p_5\}$, $\{p_4, p_5\}$, and $\{p_4, p_5, p_6\}$.

order \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 6).

The critical issue will be to *link* point α_i^j to point $\beta_{i'}^j$. 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 $i \neq i'$. The so-called *point linker* gadget will be grounded in Lemma 5.1. Due to a technicality, we will need to introduce a *copy* $\bar{\alpha}_i^j$ 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 will be to place them in α_i^j , $\bar{\alpha}_i^j$, and β_i^j for some $i \in [t]$ (see Figure 8). 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 get into the details of the reduction. We will introduce several characteristic lengths and compare them; when $l_1 \ll l_2$ means that l_1 should be thought as really small compared to l_2 , and $l_1 \approx l_2$ means that l_1 and l_2 are roughly of the same order. The motivation is to guide the intuition of the reader without bothering her/him too much about the details. At the end of the construction, we will specify more concretely how those lengths are chosen.

Construction. We start by formalizing the positions of the α_i^j 's and β_i^j 's. We recall that we want the points α_i^j 's and β_i^j 's ordered by increasing y -coordinates, to match the order of the a_i^j 's and b_i^j 's along \leq_A and \leq_B , with first all the elements of A and then all the elements of B . Starting from some y -coordinate y_1 (which is the one given to point α_1^1), the y -coordinates of the α_i^j 's are regularly spaced out by an offset y ; that is, the y -coordinate of α_i^j is $y_1 + (i + (j-1)t)y$. Between the y -coordinate of the last element in A (i.e., a_t^k whose y -coordinate is $y_1 + (kt - 1)y$) and the first element in B , there is a large offset L , such that the y -coordinate of β_i^j is $y_1 + (kt - 1)y + L + (\text{ind}(b_i^j) - 1)y$ (for any $j \in [k]$ and $i \in [t]$) where $\text{ind}(b_i^j)$ is the *index* of b_i^j along the order \leq_B , that is the number of $b \in B$ such that $b \leq_B b_i^j$.

For each color class $j \in [k]$, let $x_j := x_1 + (j-1)D$ for some x -coordinate x_1 and value D , and $y_j := y_1 + (j-1)ty$. The allocated points $\alpha_1^j, \alpha_2^j, \alpha_3^j, \dots, \alpha_t^j$ are on a line at coordinates: $(x_j, y_j), (x_j + x, y_j + y), (x_j + 2x, y_j + 2y), \dots, (x_j + (t-1)x, y_j + (t-1)y)$, for some value x . We place, to the left of those points, a rectangular pocket $\mathcal{P}_{j,r}$ of width, say, y and length, say³, tx such that the uppermost longer side of the rectangular pocket lies on the line $\ell(\alpha_1^j, \alpha_t^j)$ (see Figure 7). The y -coordinates of $\beta_1^j, \beta_2^j, \beta_3^j, \dots, \beta_t^j$ have already been defined. We set, for each $i \in [t]$, the x -coordinate of β_i^j to $x_j + (i-1)x$, so that β_i^j and α_i^j share the same x -coordinate. One can check that it is consistent with

³the exact width and length of this pocket are not relevant; the reader may just think of $\mathcal{P}_{j,r}$ as a thin pocket which forces to place a guard on a thin strip whose uppermost boundary is $\ell(\alpha_1^j, \alpha_t^j)$

the previous paragraph. We also observe that, by the choice of the y -coordinate for the β_i^j 's, we have both encoded the permutations σ_j 's and permutation σ (see Figure 9 or Figure 7).

Our construction almost exclusively rely on so-called *triangular pockets*. Henceforth, for a vertex v and two points p and p' , we call a *triangular pocket rooted at vertex v and supported by ray(v, p) and ray(v, p')* a sub-polygon w, v, w' (a triangle) such that ray(v, w) passes through p , ray(v, w') passes through p' , while w and w' are close to v (sufficiently close not to interfere with the rest of the construction). We say that v is the *root* of the triangular pocket, that we often denote by $\mathcal{P}(v)$. We also say that the pocket $\mathcal{P}(v)$ *points* towards p and p' .

We now encode the A -intervals and B -intervals with triangular pockets. At the x -coordinate $x_k + (t-1)x + F$, for some large value F , we put between y -coordinates y_1 and $y_k + (kt-1)y$, for each A -interval $I_q = [a_i^j, a_{i'}^j] \in \mathcal{S}_A$ we put 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($z_{A,q}, \alpha_{i'}^j$). Intuitively, if $y \ll x \ll D \ll F$, the only $\alpha_{i''}^j$ seeing vertex $z_{A,q}$ should be all the points such that $a_i^j \leq_A \alpha_{i''}^j \leq_A a_{i'}^j$ (see Figure 9 and Figure 6). We place those $|\mathcal{S}_A|$ pockets along the y -axis, and space them out by distance s . To guarantee that we have enough room to place all those pockets, $s \ll y$ shall later hold. Similarly, we place at the same x -coordinate $x_k + (t-1)x + F$ each of the $|\mathcal{S}_B|$ triangular pockets $\mathcal{P}(z_{B,q})$ rooted at vertex $z_{B,q}$ and supported by ray($z_{B,q}, \beta_i^j$) and ray($z_{B,q}, \beta_{i'}^j$) for B -interval $[b_i^j, b_{i'}^j] \in \mathcal{S}_B$; and we space out those pockets by distance s along the y -axis between x -coordinates $y_1 + (kt-1)y + L$ and $y_1 + 2(kt-1)y + L$. We do not specify an order to the $z_{A,q}$'s (resp. the $z_{B,q}$'s) along the y -axis since we do not need that to prove the reduction correct. The different values (s, x, y, D, L , and F) introduced so far compare in the following way: $s \ll y \ll x \ll D \ll F$, and $x \ll L \ll F$ (see Figure 9).

We now describe the *linker gadget*, or how to force consistent pairs of guards α_i^j and its associate β_i^j . The idea is that pairs of guards α_i^j, β_i^j will be very effective since the two points see disjoint sets of pockets, whereas pairs $\alpha_i^j, \beta_{i'}^j$ (with $i \neq i'$) will overlap on some pockets, and miss some other pockets completely.

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 at a distance, say, L' from point α_t^j . 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, β_1^j, β_2^j up to β_t^j . This observation relies on the fact that $y \ll x \ll L$. So, from the distance, the points $\beta_1^j, \dots, \beta_t^j$ look almost *flat*. It makes the following construction possible. 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_t^j, β_1^j) and ray(c_t^j, β_t^j). We place the vertices c_i^j ($i \in [t]$) at the same y -coordinate and we space them out by distance x along the x -axis (see Figure 7). Similarly, let us informally refer to the region slightly to the right of ray(α_1^j, β_t^j) at a distance L' from point α_1^j , as $\mathcal{R}_{\text{right}}^j$. Any point $\mathcal{R}_{\text{right}}^j$ sees, from right to left, in this order β_1^j, β_2^j up to β_t^j , and then, α_1^j, α_2^j up to α_t^j . Therefore, one can place in $\mathcal{R}_{\text{left}}^j$, for each $i \in [t-1]$, a triangular pocket $\mathcal{P}(d_i^j)$ rooted at d_i^j supported by ray(d_i^j, β_{i+1}^j) and ray(d_i^j, α_i^j). We place also a triangular pocket $\mathcal{P}(d_t^j)$ rooted at d_t^j supported by ray(d_t^j, α_1^j) and ray(d_t^j, α_t^j). Again, those t pockets can be put at the same y -coordinate and spaced out horizontally by x (see Figure 7). 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 informally call it the *weak point linker* (or simply, *weak linker*) of $\alpha_1^j, \dots, \alpha_t^j$ and $\beta_1^j, \dots, \beta_t^j$. We may call the pockets of $\mathcal{R}_{\text{left}}^j$ (resp. $\mathcal{R}_{\text{right}}^j$) *left* pockets (resp. *right* pockets).

As we will show later, if one wants to guard with only two points all the pockets of $\mathcal{P}_{j,\alpha,\beta} = \{\mathcal{P}(c_1^j), \dots, \mathcal{P}(c_t^j), \mathcal{P}(d_1^j), \dots, \mathcal{P}(d_t^j)\}$ and one first decides to put a guard on point α_i^j (for some

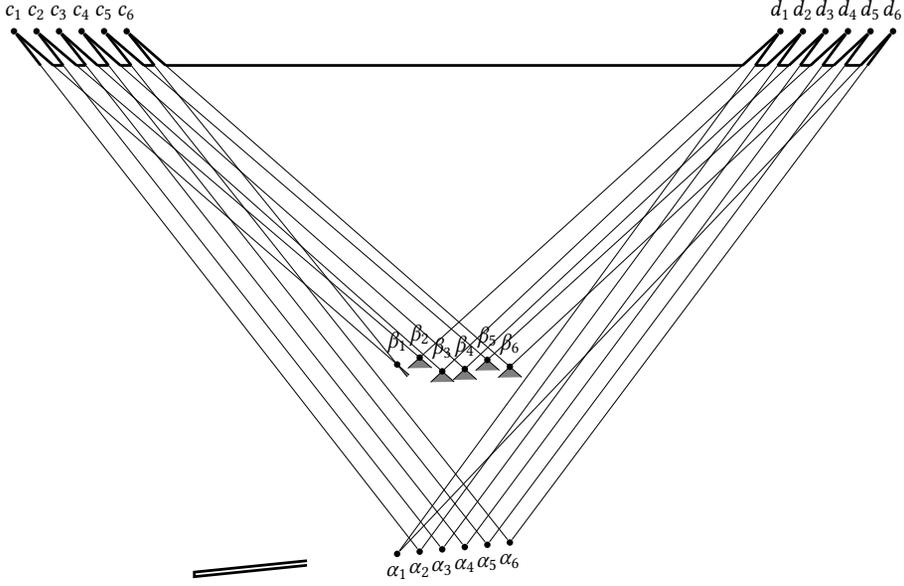


Fig. 7. Weak point linker gadget $\mathcal{P}_{j,\alpha,\beta}$ with $t = 6$. We omit the superscript j in all the labels.

437 $i \in [t]$), then one is not forced to put the other guard on point β_i^j but only on an area whose
 438 uppermost point is β_i^j (see the shaded areas below the β_i^j 's in Figure 7). Now, if $\beta_1^j, \dots, \beta_t^j$
 439 lie on a same line ℓ , we could shrink the shaded area of each β_i^j (Figure 7) down to the single point
 440 β_i^j by adding a thin rectangular pocket on ℓ (similarly to what we have for $\alpha_1^j, \dots, \alpha_t^j$). Naturally
 441 we need that $\beta_1^j, \dots, \beta_t^j$ are *not* on the same line, in order to encode σ_j .

442 The remedy we suggest is to make a triangle of weak linkers. For each $j \in [k]$, we allocate
 443 t points $\bar{\alpha}_1^j, \bar{\alpha}_2^j, \dots, \bar{\alpha}_t^j$ on a horizontal line, spaced out by distance x , say, $\approx \frac{D}{2}$ to the right and
 444 $\approx L$ to the up of β_1^j . We put a thin horizontal rectangular pocket $\mathcal{P}_{j,\bar{r}}$ of the same dimension as
 445 $\mathcal{P}_{j,r}$ such that the lowermost longer side of $\mathcal{P}_{j,\bar{r}}$ is on the line $\ell(\bar{\alpha}_1^j, \bar{\alpha}_t^j)$. We add the $2t$ pockets
 446 corresponding to a weak linker $\mathcal{P}_{j,\alpha,\bar{\alpha}}$ between $\alpha_1^j, \dots, \alpha_t^j$ and $\bar{\alpha}_1^j, \dots, \bar{\alpha}_t^j$ as well as the $2t$ pockets
 447 of a weak linker $\mathcal{P}_{j,\bar{\alpha},\beta}$ between $\bar{\alpha}_1^j, \dots, \bar{\alpha}_t^j$ and $\beta_1^j, \dots, \beta_t^j$ as pictured in Figure 8. We denote by \mathcal{P}_j
 448 the union $\mathcal{P}_{j,r} \cup \mathcal{P}_{j,\bar{r}} \cup \mathcal{P}_{j,\alpha,\beta} \cup \mathcal{P}_{j,\alpha,\bar{\alpha}} \cup \mathcal{P}_{j,\bar{\alpha},\beta}$ of all the pockets involved in the encoding of color
 449 class j . Now, say, one wants to guard all the pockets of \mathcal{P}_j with only three points, and chooses to
 450 put a guard on α_i^j (for some $i \in [t]$). Because of the pockets of $\mathcal{P}_{j,\alpha,\bar{\alpha}} \cup \mathcal{P}_{j,\bar{r}}$, one is forced to place a
 451 second guard precisely on $\bar{\alpha}_i^j$. Now, because of the weak linker $\mathcal{P}_{j,\alpha,\beta}$ the third guard should be
 452 on a region whose uppermost point is β_i^j , while, because of $\mathcal{P}_{j,\bar{\alpha},\beta}$ the third guard should be on a
 453 region whose lowermost point is β_i^j . The conclusion is that the third guard should be put precisely
 454 on β_i^j . This *triangle* of weak linkers is called the *linker* of color class j . The k linkers are placed
 455 accordingly to Figure 9. This ends the construction.

456 **Specification of the distances.** We can specify the coordinates of positions of all the vertices
 457 by fractions of integers. These integers are polynomially bounded in n . If we want to get integer
 458 coordinates, we can transform the rational coordinates to integer coordinates by multiplying all of
 459 them with the least common multiple of all the denominators, which is not polynomially bounded
 460 anymore. The length of the integers in binary is still polynomially bounded.

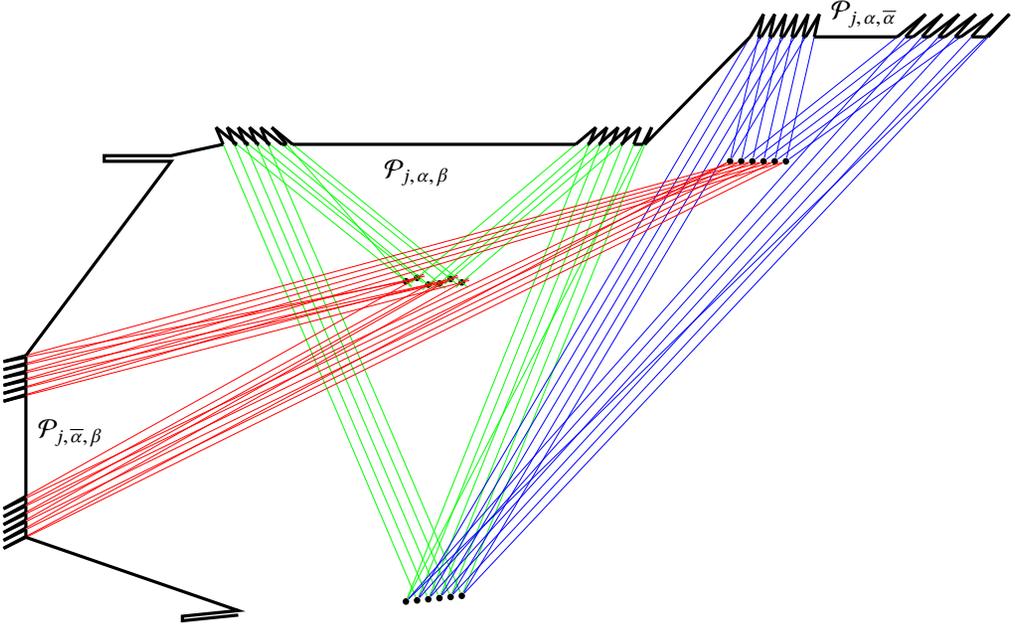


Fig. 8. Point linker gadget \mathcal{P}_j : a triangle of (three) weak point linkers $\mathcal{P}_{j,\alpha,\beta}$, $\mathcal{P}_{j,\alpha,\bar{\alpha}}$, $\mathcal{P}_{j,\bar{\alpha},\beta}$, and two rectangular pockets forcing one guard on the lines $\ell(\alpha_1^j, \alpha_2^j) = \ell(\alpha_1^j, \alpha_t^j)$ and $\ell(\bar{\alpha}_1^j, \bar{\alpha}_2^j) = \ell(\bar{\alpha}_1^j, \bar{\alpha}_t^j)$.

461 We can safely set s to one, as it is the smallest length, we specified. We will put $|\mathcal{S}_A|$ pockets
 462 on track 1 and $|\mathcal{S}_B|$ pockets on track 2. It is sufficient to have an opening space of one between
 463 them. Thus, the space on the right side of \mathcal{P} , for all pockets of track 1 is bounded by $2 \cdot |\mathcal{S}_A|$. Thus
 464 setting y to $|\mathcal{S}_A| + |\mathcal{S}_B|$ secures us that we have plenty of space to place all the pockets. We specify
 465 $F = (|\mathcal{S}_A| + |\mathcal{S}_B|)Dk = y \cdot D \cdot k$. We have to show that this is large enough to guarantee that the
 466 pockets on track 1 distinguish the picked points only by the y -coordinate. Let p and q be two points
 467 among the α_i^j . Their vertical distance is upper bounded by Dk and their horizontal distance is lower
 468 bounded by y . Thus the slope of $\ell = \ell(p, q)$ is at least $\frac{y}{Dk}$. At the right side of \mathcal{P} the line ℓ will be at
 469 least $F \frac{y}{Dk}$ above the pockets of track 1. Note $F \frac{y}{Dk} = yDk \cdot \frac{y}{Dk} > y^2 > |\mathcal{S}_A|^2 > 2 \cdot |\mathcal{S}_A|$. The same
 470 argument shows that F is sufficiently large for track 2.

471 The remaining lengths x, L, L' , and D can be specified in a similar fashion. For the construction
 472 of the pockets, let $s \in \mathcal{S}_A$ be an A -interval with endpoints a and b , represented by some points p
 473 and q and assume the opening vertices v and w of the triangular pocket are already specified. Then
 474 the two lines $\ell(p, v)$ and $\ell(q, w)$ will meet at some point x to the right of v and w . By Lemma 3.3, x
 475 has rational coordinates and the integers to represent them can be expressed by the coordinates of
 476 p, q, v , and w . This way, all the pockets can be explicitly constructed using rational coordinates as
 477 claimed above.

478 **Correctness.** We now show that the reduction is correct. The following lemma is the main
 479 argument for the easier implication: *if \mathcal{I} is a YES-instance, then the gallery that we build can be*
 480 *guarded with $3k$ points.*

481 LEMMA 5.2. $\forall j \in [k], \forall i \in [t]$, the three associate points $\alpha_i^j, \bar{\alpha}_i^j, \beta_i^j$ guard \mathcal{P}_j entirely.

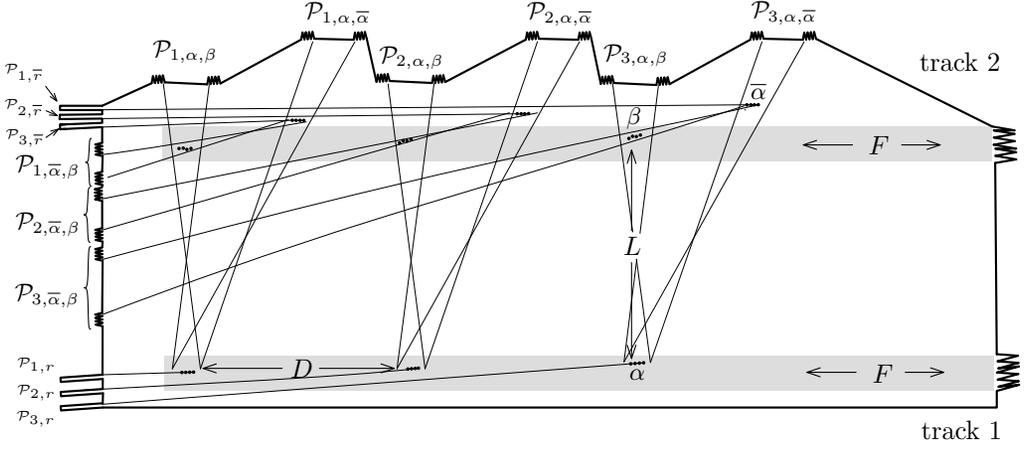


Fig. 9. The overall picture of the reduction with $k = 3$. The combination of $\mathcal{P}_{j,\alpha,\beta}$, $\mathcal{P}_{j,\alpha,\bar{\alpha}}$, $\mathcal{P}_{j,\bar{\alpha},\beta}$, $\mathcal{P}_{j,r}$, and $\mathcal{P}_{j,\bar{r}}$ forces to place pairs of guards at $\alpha_{i(j)}^j, \beta_{i(j)}^j$, analogously to the STRUCTURED 2-TRACK HITTING SET semantics. The y -coordinates of these points encode the total orders over A and B . The A -intervals are encoded by triangular pockets in track 1, while the B -intervals are encoded in track 2.

482 PROOF. The rectangular pockets $\mathcal{P}_{j,r}$ and $\mathcal{P}_{j,\bar{r}}$ are entirely seen by α_i^j and $\bar{\alpha}_i^j$, respectively. The
 483 pockets $\mathcal{P}(c_1^j), \mathcal{P}(c_2^j), \dots, \mathcal{P}(c_{i-1}^j)$ and $\mathcal{P}(d_i^j), \mathcal{P}(d_{i+1}^j), \dots, \mathcal{P}(d_i^j)$ are all entirely seen by α_i^j , while
 484 the pockets $\mathcal{P}(c_i^j), \mathcal{P}(c_{i+1}^j), \dots, \mathcal{P}(c_i^j)$ and $\mathcal{P}(d_1^j), \mathcal{P}(d_2^j), \dots, \mathcal{P}(d_{i-1}^j)$ are all entirely seen by β_i^j . This
 485 means that α_i^j and β_i^j jointly see all the pockets of $\mathcal{P}_{j,\alpha,\beta}$. Similarly, α_i^j and $\bar{\alpha}_i^j$ jointly see all the
 486 pockets of $\mathcal{P}_{j,\alpha,\bar{\alpha}}$, and $\bar{\alpha}_i^j$ and β_i^j jointly see all the pockets of $\mathcal{P}_{j,\bar{\alpha},\beta}$. Therefore, $\alpha_i^j, \bar{\alpha}_i^j, \beta_i^j$ jointly
 487 see all the pockets of \mathcal{P}_j . \square

488 Assume that \mathcal{I} is a YES-instance and let $\{(a_{s_1}^1, b_{s_1}^1), \dots, (a_{s_k}^k, b_{s_k}^k)\}$ be a solution. We claim that
 489 $G = \{\alpha_{s_1}^1, \bar{\alpha}_{s_1}^1, \beta_{s_1}^1, \dots, \alpha_{s_k}^k, \bar{\alpha}_{s_k}^k, \beta_{s_k}^k\}$ guard the whole polygon \mathcal{P} . By Lemma 5.2, $\forall j \in [k]$, \mathcal{P}_j is
 490 guarded. For each A -interval (resp. B -interval) in \mathcal{S}_A (resp. \mathcal{S}_B) there is at least one 2-element
 491 $(a_{s_j}^j, b_{s_j}^j)$ such that $a_{s_j}^j \in \mathcal{S}_A$ (resp. $b_{s_j}^j \in \mathcal{S}_B$). Thus, the corresponding pocket is guarded by $\alpha_{s_j}^j$
 492 (resp. $\beta_{s_j}^j$). The rest of the polygon \mathcal{P} (which is not part of pockets) is guarded by, for instance,
 493 $\{\bar{\alpha}_{s_1}^1, \dots, \bar{\alpha}_{s_k}^k\}$. So, G is indeed a solution and it contains $3k$ points.

494 We now assume that there is a set G of $3k$ points guarding \mathcal{P} . We will then show that \mathcal{I} is a
 495 YES-instance. We observe that no point of \mathcal{P} sees inside two triangular pockets one being in $\mathcal{P}_{j,\alpha,\gamma}$
 496 and the other in $\mathcal{P}_{j',\alpha,\gamma'}$ with $j \neq j'$ and $\gamma, \gamma' \in \{\beta, \bar{\alpha}\}$. Further, $V(r(\mathcal{P}_{j,\alpha,\beta} \cup \mathcal{P}_{j,\alpha,\bar{\alpha}})) \cap V(r(\mathcal{P}_{j',\alpha,\beta} \cup$
 497 $\mathcal{P}_{j',\alpha,\bar{\alpha}})) = \emptyset$ when $j \neq j'$, where r maps a set of triangular pockets to the set of their root. Also, for
 498 each $j \in [k]$, seeing $\mathcal{P}_{j,\alpha,\beta}$ and $\mathcal{P}_{j,\alpha,\bar{\alpha}}$ entirely requires at least 3 points. This means that for each
 499 $j \in [k]$, one should place three guards in $V(r(\mathcal{P}_{j,\alpha,\beta} \cup \mathcal{P}_{j,\alpha,\bar{\alpha}}))$. Furthermore, one can observe that,
 500 among those three points, one should guard a triangular pocket $\mathcal{P}_{j',r}$ and another should guard
 501 $\mathcal{P}_{j'',\bar{r}}$. Thus a set S_1 , consisting of three guards of G , sees \mathcal{P}_1 and two rectangular pockets $\mathcal{P}_{j',r}$ and
 502 $\mathcal{P}_{j'',\bar{r}}$.

503 Let us call ℓ_1 (resp. ℓ'_1) the line corresponding to the extension of the uppermost (resp. lowermost)
 504 longer side of $\mathcal{P}_{1,r}$ (resp. $\mathcal{P}_{1,\bar{r}}$). The only points of \mathcal{P} that can see a rectangular pocket $\mathcal{P}_{j',r}$ and at
 505 least t pockets of $\mathcal{P}_{1,\alpha,\bar{\alpha}}$ are on ℓ_1 : more specifically, they are the points $\alpha_1^1, \dots, \alpha_t^1$. The only points
 506 that can see a rectangular pocket $\mathcal{P}_{j'',\bar{r}}$ and at least t pockets of $\mathcal{P}_{1,\alpha,\bar{\alpha}}$ are on ℓ'_1 : they are the points

507 $\bar{\alpha}_1^1, \dots, \bar{\alpha}_{i'}^1$. As $\mathcal{P}_{1, \alpha, \bar{\alpha}}$ has $2t$ pockets, S_1 should contain two points α_i^1 and $\bar{\alpha}_{i'}^1$. By the argument of
 508 Lemma 5.1, i should be equal to i' (otherwise, $i < i'$ and the left pocket pointing towards $\bar{\alpha}_{i'-1}^1$ and
 509 $\alpha_{i'}^1$ is not seen, or $i > i'$ and the right pocket pointing towards α_{i+1}^1 and $\bar{\alpha}_i^1$ is not seen). We denote
 510 by s_1 this shared value. Now, to see the left pocket $\mathcal{P}(c_{s_1}^1)$ and the right pocket $\mathcal{P}(d_{s_1-1}^1)$ (that should
 511 still be seen), the third guard should be to the left of $\ell(c_{s_1}^1, \beta_{s_1}^1)$ and to the right of $\ell(d_{s_1-1}^1, \beta_{s_1}^1)$ (see
 512 shaded area of Figure 7). That is, the third guard of S_1 should be on a region in which $\beta_{s_1}^1$ is the
 513 uppermost point. The same argument with the pockets of $\mathcal{P}_{1, \bar{\alpha}, \beta}$ implies that the third guard should
 514 also be on a region in which $\beta_{s_1}^1$ is the lowermost point. Thus, the third guard of S_1 has to be the
 515 point $\beta_{s_1}^1$. Therefore $S_1 = \{\alpha_{s_1}^1, \bar{\alpha}_{s_1}^1, \beta_{s_1}^1\}$, for some $s_1 \in [t]$.

516 As none of those three points see any pocket $\mathcal{P}_{j, \bar{\alpha}, \beta}$ with $j > 1$ (we already mentioned that
 517 no pocket of $\mathcal{P}_{j, \alpha, \beta}$ and $\mathcal{P}_{j, \alpha, \bar{\alpha}}$ with $j > 1$ can be seen by those points), we can repeat the
 518 argument for the second color class; and so forth up to color class k . Thus, G is of the form
 519 $\{\alpha_{s_1}^1, \bar{\alpha}_{s_1}^1, \beta_{s_1}^1, \dots, \alpha_{s_k}^k, \bar{\alpha}_{s_k}^k, \beta_{s_k}^k\}$. As G also guards all the pockets of tracks 1 and 2, the set of k
 520 2-elements $\{(a_{s_1}^1, b_{s_1}^1), \dots, (a_{s_k}^k, b_{s_k}^k)\}$ hits all the A -intervals of \mathcal{S}_A , and the B -intervals of \mathcal{S}_B . \square

521 6 PARAMETERIZED HARDNESS OF THE VERTEX GUARD VARIANT

522 We now turn to the vertex guard variant and show the same hardness result. Again, we reduce from
 523 STRUCTURED 2-TRACK HITTING SET and our main task is to design a *linker gadget*. Though, *linking*
 524 pairs of vertices turns out to be very different from *linking* pairs of points. Therefore, we have to
 525 come up with fresh ideas to carry out the reduction. In a nutshell, the principal ingredient is to
 526 *link* pairs of convex vertices by introducing reflex vertices at strategic places. As placing guards on
 527 those reflex vertices is not supposed to happen in the STRUCTURED 2-TRACK HITTING SET instance,
 528 we design a so-called *filter gadget* to prevent any solution from doing so.

529 **THEOREM 1.2 (PARAMETERIZED HARDNESS VERTEX GUARD).** *VERTEX GUARD ART GALLERY is not*
 530 *solvable in time $f(k)n^{o(k/\log k)}$, even on simple polygons, where n is the number of vertices of the*
 531 *polygon and k is the number of guards allowed, for any computable function f , unless the ETH fails.*

532 **PROOF.** From an instance $\mathcal{I} = (k \in \mathbb{N}, t \in \mathbb{N}, \sigma \in \mathfrak{S}_k, \sigma_1 \in \mathfrak{S}_t, \dots, \sigma_k \in \mathfrak{S}_t, \mathcal{S}_A, \mathcal{S}_B)$, we build
 533 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
 534 guarded by $3k$ vertices.

535 **Linker gadget.** This gadget encodes the 2-elements. We build a sub-polygon that can be seen
 536 entirely by pairs of convex vertices if and only if they correspond to the same 2-element.

537 For each $j \in [k]$, permutation σ_j will be encoded by a sub-polygon \mathcal{P}_j that we call *vertex linker*,
 538 or simply *linker* (see Figure 10). We regularly set t consecutive vertices $\alpha_1^j, \alpha_2^j, \dots, \alpha_t^j$ in this order,
 539 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,
 540 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 will later
 541 add some reflex vertices between them. At mid-distance between α_1^j and $\beta_{\sigma_j(1)}^j$, to the left, we put
 542 a reflex vertex r_\downarrow^j . To the left of this reflex vertex, we place a vertical *wall* $d^j e^j$ (r_\downarrow^j, d^j , and e^j are
 543 three consecutive vertices of \mathcal{P}), so that $\text{ray}(\alpha_1^j, r_\downarrow^j)$ and $\text{ray}(\alpha_t^j, r_\downarrow^j)$ both intersect $\text{seg}(d^j, e^j)$. That
 544 implies that for each $i \in [t]$, $\text{ray}(\alpha_i^j, r_\downarrow^j)$ intersects $\text{seg}(d^j, e^j)$. We denote by p_i^j this intersection. The
 545 greater i , the closer p_i^j is to d^j . Similarly, at mid-distance between α_t^j and $\beta_{\sigma_j(t)}^j$, to the right, we put
 546 a reflex vertex r_\uparrow^j and place a vertical wall $x^j y^j$ (r_\uparrow^j, x^j , and y^j are consecutive), so that $\text{ray}(\alpha_1^j, r_\uparrow^j)$
 547 and $\text{ray}(\alpha_t^j, r_\uparrow^j)$ both intersect $\text{seg}(x^j, y^j)$. For each $i \in [t]$, we denote by q_i^j the intersection between
 548 $\text{ray}(\alpha_i^j, r_\uparrow^j)$ and $\text{seg}(x^j, y^j)$. The smaller i , the closer q_i^j is to x^j .

549 For each $i \in [t]$, we put around β_i^j two reflex vertices, one in $\text{ray}(\beta_i^j, p_i^j)$ and one in $\text{ray}(\beta_i^j, q_i^j)$.
 550 Later we may refer to these reflex vertices as *intermediate reflex vertices*. In Figure 10, we merged
 551 some reflex vertices but the essential part is that $V(\beta_i^j) \cap \text{seg}(d^j, e^j) = \text{seg}(d^j, p_i^j)$ and $V(\beta_i^j) \cap$
 552 $\text{seg}(x^j, y^j) = \text{seg}(x^j, q_i^j)$. Finally, we add a triangular pocket rooted at g^j and supported by $\text{ray}(g^j, \alpha_1^j)$
 553 and $\text{ray}(g^j, \alpha_t^j)$, as well as a triangular pocket rooted at b^j and supported by $\text{ray}(g^j, \beta_{\sigma_j(1)}^j)$ and
 554 $\text{ray}(g^j, \beta_{\sigma_j(t)}^j)$. This ends the description of the vertex linker (see Figure 10).

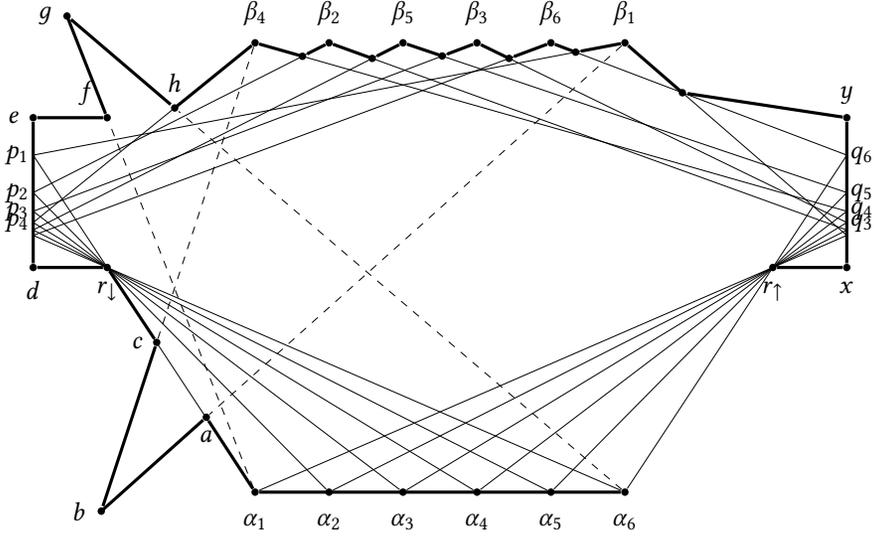


Fig. 10. Vertex linker gadget \mathcal{P}_j . We omitted the superscript j in all the labels. Here, $\sigma_j(1) = 4$, $\sigma_j(2) = 2$, $\sigma_j(3) = 5$, $\sigma_j(4) = 3$, $\sigma_j(5) = 6$, $\sigma_j(6) = 1$.

555 The following lemma formalizes how exactly the vertices α_i^j and β_i^j are linked: say, one chooses
 556 to put a guard on a vertex α_i^j , then the only way to see \mathcal{P}_j entirely, by putting a second guard on a
 557 vertex of $\{\beta_1^j, \dots, \beta_t^j\}$ is to place it on the vertex β_i^j .

558 LEMMA 6.1. For any $j \in [k]$, the sub-polygon \mathcal{P}_j is seen entirely by $\{\alpha_v^j, \beta_w^j\}$ iff $v = w$.

559 PROOF. The regions of \mathcal{P}_j not seen by α_v^j (i.e., $\mathcal{P}_j \setminus V(\alpha_v^j)$) consist of the triangles $d^j r_1^j p_v^j$, $x^j r_2^j q_v^j$
 560 and partially the triangle $a^j b^j c^j$. The triangle $a^j b^j c^j$ is anyway entirely seen by the vertex β_i^j , for
 561 any $i \in [t]$. It remains to prove that $d^j r_1^j p_v^j \cup x^j r_2^j q_v^j \subseteq V(\beta_w^j)$ iff $v = w$.

562 It holds that $d^j r_1^j p_v^j \cup x^j r_2^j q_v^j \subseteq V(\beta_v^j)$ since, by construction, the two reflex vertices neighboring
 563 β_v^j are such that β_v^j sees $\text{seg}(d^j, p_v^j)$ (hence, the whole triangle $d^j r_1^j p_v^j$) and $\text{seg}(x^j, q_v^j)$ (hence, the
 564 whole triangle $x^j r_2^j q_v^j$). Now, let us assume that $v \neq w$. If $v < w$, the interior of the segment
 565 $\text{seg}(p_v, p_w)$ is not seen by $\{\alpha_v^j, \beta_w^j\}$, and if $v > w$, the interior of the segment $\text{seg}(q_v, q_w)$ is not
 566 seen by $\{\alpha_v^j, \beta_w^j\}$. \square

567 The issue we now have is that one could decide to place a guard on a vertex α_i^j and a second
 568 guard on a reflex vertex between $\beta_{\sigma_j(w)}^j$ and $\beta_{\sigma_j(w+1)}^j$ (for some $w \in [t-1]$). This is indeed another

569 way to guard the whole \mathcal{P}_j . We will now describe a sub-polygon \mathcal{F}_j (for each $j \in [k]$) called *filter*
 570 *gadget* (see Figure 11) satisfying the property that all its (triangular) pockets can be guarded by
 571 adding only one guard on a vertex of \mathcal{F}_j iff there is already a guard on a vertex β_i^j of \mathcal{P}_j . Therefore,
 572 the filter gadget will prevent one from placing a guard on a reflex vertex of \mathcal{P}_j . The functioning of
 573 the gadget is again based on Lemma 5.1.

574 **Filter gadget.** Let d_1^j, \dots, d_t^j be t consecutive vertices of a regular, say, $20t$ -gon, so that the angle
 575 made by $\text{ray}(d_1^j, d_2^j)$ and the y -axis is a bit below 45° , while the angle made by $\text{ray}(d_{t-1}^j, d_t^j)$ and
 576 the y -axis is a bit above 45° . The vertices d_1^j, \dots, d_t^j therefore lie equidistantly on a circular arc C .
 577 We now mentally draw two lines ℓ_h and ℓ_v ; ℓ_h is a horizontal line a bit below d_1^j , while ℓ_v is a
 578 vertical line a bit to the right of d_t^j . We put, for each $i \in [t]$, a vertex x_i^j at the intersection of ℓ_h and
 579 the tangent to C passing through d_i^j . Then, for each $i \in [t-1]$, we set a triangular pocket $\mathcal{P}(x_i^j)$
 580 rooted at x_i^j and supported by $\text{ray}(x_i^j, d_i^j)$ and $\text{ray}(x_i^j, \beta_{\sigma_j(i)}^j)$. For convenience, each point $\beta_{\sigma_j(i)}^j$ is
 581 denoted by c_i^j on Figure 11. We also set a triangular pocket $\mathcal{P}(x_t^j)$ rooted at x_t^j and supported by
 582 $\text{ray}(x_t^j, d_1^j)$ and $\text{ray}(x_t^j, d_t^j)$. Similarly, we place, for each $i \in [t-1]$, a vertex y_i^j at the intersection of
 583 ℓ_v and the tangent to C passing through d_{i+1}^j . Finally, we set a triangular pocket $\mathcal{P}(y_i^j)$ rooted at y_i^j
 584 and supported by $\text{ray}(y_i^j, \beta_{\sigma_j(i)}^j)$ and $\text{ray}(y_i^j, d_{i+1}^j)$, for each $i \in [t-1]$ (see Figure 11). We denote by
 585 $\mathcal{P}(\mathcal{F}_j)$ the $2t-1$ triangular pockets of \mathcal{F}_j .

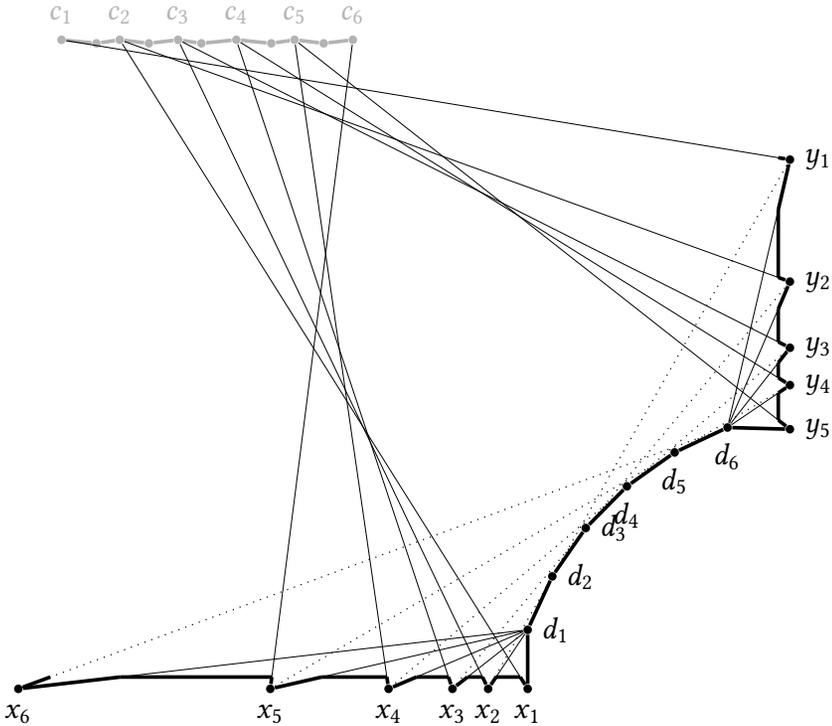


Fig. 11. The filter gadget \mathcal{F}_j . Again, we omit the superscript j on the labels. Vertices c_1, c_2, \dots, c_t are not part of \mathcal{F}_j and are in fact the vertices $\beta_{\sigma_j(1)}^j, \beta_{\sigma_j(2)}^j, \dots, \beta_{\sigma_j(t)}^j$ and the vertices in between the c_i 's are the reflex vertices that we have to *filter out*.

586 LEMMA 6.2. For each $j \in [k]$, the only ways to see $\mathcal{P}(\mathcal{F}_j)$ and the triangle $a^j b^j c^j$ entirely with only
587 two guards on vertices of $\mathcal{P}_j \cup \mathcal{P}(\mathcal{F}_j)$ is to place them on vertices c_i^j and d_i^j (for any $i \in [t]$).

588 PROOF. Proving this lemma will, in particular, entail that it is not possible to see $\mathcal{P}(\mathcal{F}_j)$ entirely
589 with only two vertices if one of them is a reflex vertex between c_i^j and c_{i+1}^j . We recall that such a
590 vertex is called an intermediate reflex vertex (in color class j). Because of the pocket $a^j b^j c^j$, one
591 should put a guard on a c_i^j (for some $i \in [t]$) or on an intermediate reflex vertex in class j . As
592 vertices a^j , b^j , and c^j do not see anything of $\mathcal{P}(\mathcal{F}_j)$, placing the first guard at one of those three
593 vertices cannot work as a consequence of what follows.

594 Say, the first guard is placed at $c_i^j (= \beta_{\sigma(i)}^j)$. The pockets $\mathcal{P}(x_1^j), \mathcal{P}(x_2^j), \dots, \mathcal{P}(x_{i-1}^j)$ and $\mathcal{P}(y_1^j),$
595 $\mathcal{P}(y_{i+1}^j), \dots, \mathcal{P}(y_{t-1}^j)$ are entirely seen, while the vertices $x_i^j, x_{i+1}^j, \dots, x_t^j$ and $y_1^j, y_2^j, \dots, y_{i-1}^j$ are
596 not. The only vertex that sees simultaneously all those vertices is d_i^j . The vertex d_i^j even sees the
597 whole pockets $\mathcal{P}(x_i^j), \mathcal{P}(x_{i+1}^j), \dots, \mathcal{P}(x_t^j)$ and $\mathcal{P}(y_1^j), \mathcal{P}(y_2^j), \dots, \mathcal{P}(y_{i-1}^j)$. Therefore, all the pockets
598 $\mathcal{P}(\mathcal{F}_j)$ are fully seen.

599 Now, say, the first guard is put on an intermediate reflex vertex r between c_i^j and c_{i+1}^j (for some
600 $i \in [t-1]$). Both vertices x_i^j and y_i^j , as well as x_{i+1}^j , are not seen by r and should therefore be seen by
601 the second guard. However, no vertex simultaneously sees those three vertices. \square

602 **Putting the pieces together.** The permutation σ is encoded the following way. We position
603 the vertex linkers $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_k$ such that \mathcal{P}_{i+1} is below and slightly to the left of \mathcal{P}_i . Far below
604 and to the right of the \mathcal{P}_i 's, we place the \mathcal{F}_i 's such that the uppermost vertex of $\mathcal{F}_{\sigma(i)}$ is close and
605 connected to the leftmost vertex of $\mathcal{F}_{\sigma(i+1)}$, for all $i \in [t-1]$. We add a constant number of vertices
606 in the vicinity of each \mathcal{P}_j , so that the only filter gadget that vertices $\beta_1^j, \dots, \beta_t^j$ can see is \mathcal{F}_j (see
607 Figure 12). Similarly to the point guard version, we place vertically and far from the α_i^j 's, one
608 triangular pocket $\mathcal{P}(z_{A,q})$ rooted at vertex $z_{A,q}$ and supported by $\text{ray}(z_{A,q}, \alpha_i^j)$ and $\text{ray}(z_{A,q}, \alpha_{i'}^j)$,
609 for each A -interval $I_q = [a_i^j, a_{i'}^j] \in \mathcal{S}_A$ (Track 1). Finally, we place vertically and far from the d_i^j 's,
610 one triangular pocket $\mathcal{P}(z_{B,q})$ rooted at vertex $z_{B,q}$ and supported by $\text{ray}(z_{B,q}, d_i^j)$ and $\text{ray}(z_{B,q}, d_{i'}^j)$,
611 for each B -interval $I_q = [b_{\sigma_j(i)}^j, b_{\sigma_j(i')}^j] \in \mathcal{S}_B$ (Track 2). We make sure that, all projected on the
612 x -axis, $\mathcal{F}_{\sigma(1)}$ is to the right of \mathcal{P}_1 and to the left of Track 1, so that, for every $i \in [t]$, the vertex $d_i^{\sigma(1)}$
613 sees the top edge of the gallery entirely. This ends the construction (see Figure 12).

614 **Correctness.** We now prove the correctness of the reduction. Assume that \mathcal{I} is a YES-instance
615 and let $\{(a_{s_1}^1, b_{s_1}^1), \dots, (a_{s_k}^k, b_{s_k}^k)\}$ be a solution. We claim that the set of vertices $G = \{\alpha_{s_1}^1, \beta_{s_1}^1, d_{\sigma_1^{-1}(s_1)}^1,$
616 $\dots, \alpha_{s_k}^k, \beta_{s_k}^k, d_{\sigma_k^{-1}(s_k)}^k\}$ guards the whole polygon \mathcal{P} . Let $z^j := d_{\sigma_j^{-1}(s_j)}^j$ for notational convenience. By
617 Lemma 6.1, for each $j \in [k]$, the sub-polygon \mathcal{P}_j is entirely seen, since there are guards on $\alpha_{s_j}^j$ and
618 $\beta_{s_j}^j$. By Lemma 6.2, for each $j \in [k]$, all the pockets of \mathcal{F}_j are entirely seen, since there are guards
619 on $\beta_{s_j}^j = c_{\sigma_j^{-1}(s_j)}^j$ and $d_{\sigma_j^{-1}(s_j)}^j = z^j$. For each A -interval (resp. B -interval) in \mathcal{S}_A (resp. \mathcal{S}_B) there is at
620 least one 2-element $(a_{s_j}^j, b_{s_j}^j)$ such that $a_{s_j}^j \in \mathcal{S}_A$ (resp. $b_{s_j}^j \in \mathcal{S}_B$). Thus, the corresponding pocket is
621 guarded by $\alpha_{s_j}^j$ (resp. $\beta_{s_j}^j$). The rest of the polygon is seen by, for instance, $z^{\sigma(1)}$ and $z^{\sigma(k)}$.

622 We now assume that there is a set G of $3k$ vertices guarding \mathcal{P} . We will show that \mathcal{I} is a YES-
623 instance. For each $j \in [k]$, vertices b^j, g^j , and x_t^j are seen by three pairwise-disjoint sets of vertices.
624 The first two sets are contained in the vertices of sub-polygon \mathcal{P}_j and the third one is contained
625 in the vertices of \mathcal{F}_j . Therefore, to see $\mathcal{P}_j \cup \mathcal{P}(\mathcal{F}_j)$ entirely, three vertices are necessary. Summing
626 that over the k color classes, this corresponds already to $3k$ vertices which is the size of G . Thus, G
627 contains a set S_j of exactly 3 guards among the vertices of $\mathcal{P}_j \cup \mathcal{P}(\mathcal{F}_j)$.

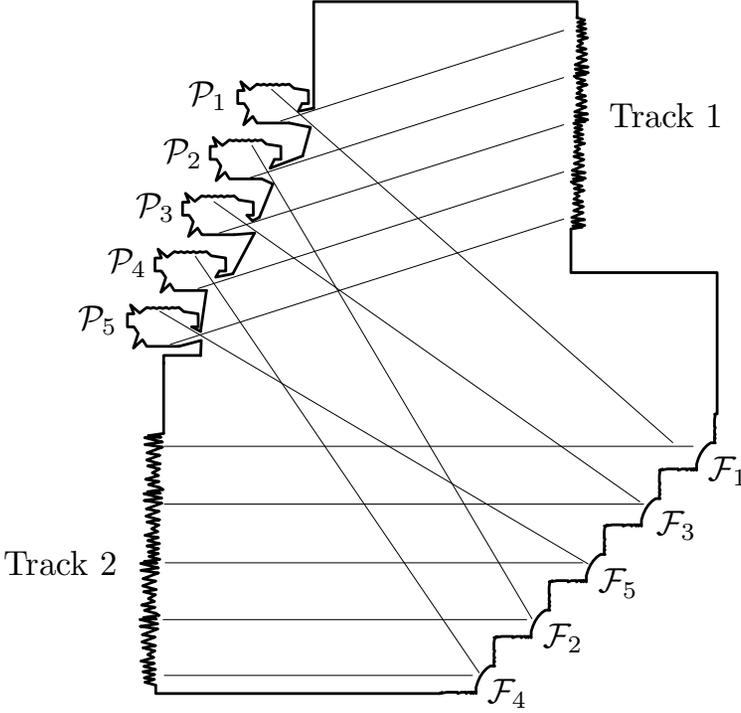


Fig. 12. Overall picture of the reduction with $k = 5$, and $\sigma = 42531$. The linker gadgets \mathcal{P}_j , together with \mathcal{F}_j , force guards at vertices $\alpha^j_{i(j)}, \beta^j_{i(j)}$. The filter gadgets \mathcal{F}_j transmit the choice of $\beta^j_{i(j)}$ and ensure that no other guard placement can be made in \mathcal{P}_j . The A -intervals of the STRUCTURED 2-TRACK HITTING SET instance are encoded by triangular pockets on Track 1, while the B -intervals are encoded on Track 2.

628 The guard of S_j responsible for seeing g^j does not see b^j nor any pockets of $P(\mathcal{F}_j)$. Hence there
 629 are only two guards of S_j performing the latter task. Therefore, by Lemma 6.2, there should be
 630 an $s_j \in [t]$ such that both $d^j_{s_j}$ and $c^j_{s_j} = \beta^j_{\sigma_j(s_j)}$ are in G . The only vertices seeing g^j are f^j, g^j, h^j
 631 and a^j_1, \dots, a^j_t . As $d^j_{s_j}$ and the $3k - 3$ guards of $G \setminus S_j$ do not see the edges $d^j e^j$ and $x^j y^j$ at all, by
 632 Lemma 6.1, among a^j_1, \dots, a^j_t the only possibility for the third guard of S_j is $\alpha^j_{\sigma_j(s_j)}$. We can assume
 633 that the third guard of S_j is indeed $\alpha^j_{\sigma_j(s_j)}$, since f^j, g^j, h^j do not see any pockets outside of \mathcal{P}_j
 634 (whereas $\alpha^j_{\sigma_j(s_j)}$, in principle, does in Track 1).

635 So far, we showed that G is of the form $\{\alpha^1_{\sigma_1(s_1)}, \beta^1_{\sigma_1(s_1)}, d^1_{s_1}, \dots, \alpha^j_{\sigma_j(s_j)}, \beta^j_{\sigma_j(s_j)}, d^j_{s_j}, \dots, \alpha^k_{\sigma_k(s_k)},$
 636 $\beta^k_{\sigma_k(s_k)}, d^k_{s_k}\}$. It means that $\alpha^1_{\sigma_1(s_1)}, \dots, \alpha^k_{\sigma_k(s_k)}$ see all the pockets of Track 1, while $d^1_{s_1}, \dots, d^k_{s_k}$ see
 637 all the pockets of Track 2. Therefore the set of k 2-elements $\{(a^1_{\sigma_1(s_1)}, b^1_{\sigma_1(s_1)}), \dots, (a^k_{\sigma_k(s_k)}, b^k_{\sigma_k(s_k)})\}$
 638 is a hitting set of both \mathcal{S}_A and \mathcal{S}_B , hence I is a YES-instance.

639 Let us bound the number of vertices of \mathcal{P} . Each sub-polygon \mathcal{P}_j or \mathcal{F}_j contains $O(t)$ vertices.
 640 Track 1 contains $3|\mathcal{S}_A|$ vertices and Track 2 contains $3|\mathcal{S}_B|$ vertices. Linking everything together
 641 requires $O(k)$ additional vertices. So, in total, there are $O(kt + |\mathcal{S}_A| + |\mathcal{S}_B|)$ vertices. Thus, this
 642 reduction together with Theorem 4.2 implies that VERTEX GUARD ART GALLERY is $W[1]$ -hard and
 643 cannot be solved in time $f(k)n^{o(k/\log k)}$, where n is the number of vertices of the polygon and k the
 644 number of guards, for any computable function f , unless the ETH fails. \square

REFERENCES

- 645
- 646 [1] M. Abrahamsen, A. Adamaszek, and T. Miltzow. Irrational guards are sometimes needed. In B. Aronov and M. J. Katz, editors, *33rd International Symposium on Computational Geometry, SoCG 2017, July 4-7, 2017, Brisbane, Australia*, volume 77 of *LIPICs*, pages 3:1–3:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.
- 647
- 648 [2] M. Abrahamsen, A. Adamaszek, and T. Miltzow. The art gallery problem is $\exists\mathbb{R}$ -complete. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 65–73, 2018.
- 649
- 650 [3] A. Aggarwal. The art gallery theorem: its variations, applications and algorithmic aspects. *PhD thesis*, 1984.
- 651
- 652 [4] A. Agrawal, P. Ashok, M. M. Reddy, S. Saurabh, and D. Yadav. FPT Algorithms for Conflict-free Coloring of Graphs and Chromatic Terrain Guarding. *arXiv preprint arXiv:1905.01822*, 2019.
- 653
- 654 [5] A. Agrawal, K. Knudsen, D. Lokshtanov, S. Saurabh, and M. Zehavi. The Parameterized Complexity of Guarding Almost Convex Polygons. *to appear in SoCG*, 2020.
- 655
- 656 [6] P. Ashok, F. V. Fomin, S. Kolay, S. Saurabh, and M. Zehavi. Exact algorithms for terrain guarding. *ACM Trans. Algorithms*, 14(2):25:1–25:20, Apr. 2018.
- 657
- 658 [7] S. Ashur, O. Filtser, and M. J. Katz. A Constant-Factor Approximation Algorithm for Vertex Guarding a WV-Polygon. *arxiv*, 2019.
- 659
- 660 [8] S. Basu, R. Pollack, and M.-F. Roy. Algorithms in real algebraic geometry. *AMC*, 10:12, 2011.
- 661
- 662 [9] P. Bhattacharya, S. K. Ghosh, and S. Pal. Constant approximation algorithms for guarding simple polygons using vertex guards. *arXiv preprint arXiv:1712.05492*, 2017.
- 663
- 664 [10] É. Bonnet and P. Giannopoulos. Orthogonal Terrain Guarding is NP-complete. *JoCG*, 10(2):21–44, 2019.
- 665
- 666 [11] É. Bonnet, P. Giannopoulos, and M. Lampis. On the parameterized complexity of red-blue points separation. In *12th International Symposium on Parameterized and Exact Computation, IPEC 2017, September 6-8, 2017, Vienna, Austria*, pages 8:1–8:13, 2017.
- 667
- 668 [12] É. Bonnet and T. Miltzow. An approximation algorithm for the art gallery problem. In *33rd International Symposium on Computational Geometry, SoCG 2017, July 4-7, 2017, Brisbane, Australia*, pages 20:1–20:15, 2017.
- 669
- 670 [13] B. Brodén, M. Hammar, and B. J. Nilsson. Guarding lines and 2-link polygons is apx-hard. Proc. 13th Canadian Conference on Computational Geometry, 2001.
- 671
- 672 [14] J. Chen, X. Huang, I. A. Kanj, and G. Xia. Strong computational lower bounds via parameterized complexity. *Journal of Computer and System Sciences*, 72(8):1346–1367, 2006.
- 673
- 674 [15] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.
- 675
- 676 [16] P. J. de Rezende, C. C. de Souza, S. Friedrichs, M. Hemmer, A. Kröller, and D. C. Tozoni. Engineering art galleries. In *Algorithm Engineering*, pages 379–417. Springer, 2016.
- 677
- 678 [17] M. G. Dobbins, A. Holmsen, and T. Miltzow. Smoothed analysis of the art gallery problem. *CoRR*, abs/1811.01177, 2018.
- 679
- 680 [18] M. Dom, M. R. Fellows, F. A. Rosamond, and S. Sikdar. The parameterized complexity of stabbing rectangles. *Algorithmica*, 62(1-2):564–594, 2012.
- 681
- 682 [19] R. G. Downey and M. R. Fellows. *Parameterized complexity*. Springer Science & Business Media, 2012.
- 683
- 684 [20] A. Efrat and S. Har-Peled. Guarding galleries and terrains. *Inf. Process. Lett.*, 100(6):238–245, 2006.
- 685
- 686 [21] S. Eidenbenz, C. Stamm, and P. Widmayer. Inapproximability results for guarding polygons and terrains. *Algorithmica*, 31(1):79–113, 2001.
- 687
- 688 [22] K. Elbassioni. Finding small hitting sets in infinite range spaces of bounded vc-dimension. In *33rd International Symposium on Computational Geometry, SoCG 2017, July 4-7, 2017, Brisbane, Australia*, pages 40:1–40:15, 2017.
- 689
- 690 [23] S. Fisk. A short proof of Chvátal’s watchman theorem. *J. Comb. Theory, Ser. B*, 24(3):374, 1978.
- 691
- 692 [24] J. Flum and M. Grohe. Parameterized complexity theory, volume XIV of texts in theoretical computer science. An EATCS Series, 2006.
- 693
- 694 [25] S. K. Ghosh. Approximation algorithms for art gallery problems in polygons. *Discrete Applied Mathematics*, 158(6):718–722, 2010.
- 695
- 696 [26] M. Gibson, E. Krohn, and Q. Wang. The VC-dimension of visibility on the boundary of a simple polygon. In *Algorithms and Computation*, pages 541–551. Springer, 2015.
- 697
- 698 [27] A. Gilbers and R. Klein. A new upper bound for the VC-dimension of visibility regions. *Computational Geometry*, 47(1):61–74, 2014.
- 699
- 700 [28] R. Impagliazzo and R. Paturi. Complexity of k-SAT. In *Computational Complexity, 1999. Proceedings. Fourteenth Annual IEEE Conference on*, pages 237–240. IEEE, 1999.
- 701
- [29] M. Jiang. On the parameterized complexity of some optimization problems related to multiple-interval graphs. *Theor. Comput. Sci.*, 411(49):4253–4262, 2010.
- [30] G. Kalai and J. Matoušek. Guarding galleries where every point sees a large area. *Israel Journal of Mathematics*, 101(1):125–139, 1997.

- 702 [31] M. J. Katz and G. S. Roisman. On guarding the vertices of rectilinear domains. *Computational Geometry*, 39(3):219–228,
703 2008.
- 704 [32] J. King. Fast vertex guarding for polygons with and without holes. *Comput. Geom.*, 46(3):219–231, 2013.
- 705 [33] J. King and E. Krohn. Terrain Guarding is NP-Hard. *SIAM J. Comput.*, 40(5):1316–1339, 2011.
- 706 [34] D. G. Kirkpatrick. An $O(lg lg OPT)$ -approximation algorithm for multi-guarding galleries. *Discrete & Computational*
707 *Geometry*, 53(2):327–343, 2015.
- 708 [35] E. Krohn and B. J. Nilsson. Approximate guarding of monotone and rectilinear polygons. *Algorithmica*, 66(3):564–594,
709 2013.
- 710 [36] D. Lee and A. Lin. Computational complexity of art gallery problems. *IEEE Transactions on Information Theory*,
711 32(2):276–282, 1986.
- 712 [37] D. Marx. Can you beat treewidth? *Theory of Computing*, 6(1):85–112, 2010.
- 713 [38] D. Marx and M. Pilipczuk. Optimal parameterized algorithms for planar facility location problems using voronoi
714 diagrams. In *ESA 2015*, pages 865–877, 2015.
- 715 [39] R. Motwani, A. Raghunathan, and H. Saran. Covering orthogonal polygons with star polygons: The perfect graph
716 approach. *J. Comput. Syst. Sci.*, 40(1):19–48, 1990.
- 717 [40] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
- 718 [41] J. O’Rourke. *Art gallery theorems and algorithms*, volume 57. Oxford University Press Oxford, 1987.
- 719 [42] P. Valtr. Guarding galleries where no point sees a small area. *Israel Journal of Mathematics*, 104(1):1–16, 1998.