# Parameterized Hardness of Art Gallery Problems*

**Édouard Bonnet and Tillmann Miltzow**

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

## 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 $\mathcal{P}$. 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.

**Digital Object Identifier** 10.4230/LIPIcs...

## 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 $S$ 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. In what follows, $n$ refers to the number of vertices of $\mathcal{P}$ and $k$ to the size of an optimal set 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, three books [12, 28, 30] and two extensive surveys appeared [5, 29]. O'Rourke's book from 1987 has over a 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 shapes of guards, have been defined and analyzed. One of the first 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 [10].

The paper of Eidenbenz et al. showed NP-hardness and APX-hardness for most relevant variants [9]. See also [2, 19, 22] for more recent reductions. Due to those negative results, most papers concentrated on finding approximation algorithms and variants that are polynomially tractable [13, 20–22, 25]. However, considering the recent lack of progress in this direction, the study of other approaches becomes interesting. One such approach is to find heuristics to solve large instances of the art gallery problem [5]. The fundamental drawback of this approach is the lack of performance *guarantees.*

In the last twenty-five years, another fruitful approach gained popularity: parameterized complexity. The underlying idea is to study algorithmic problems with dependence on a natural parameter. If the dependence on the parameter is practical and the parameter is small for real-life instances, we attain algorithms that give optimal solutions with reasonable

* supported by the ERC grant PARAMTIGHT: "Parameterized complexity and the search for tight complexity results", no. 280152.

running times and performance *guarantees*. For a gentle introduction to parameterized complexity, we recommend Niedermeier's book [26]. For a thorough reading highlighting complexity classes, we suggest the book by Downey and Fellows [7]. For a recent book on the topic with an emphasize on algorithms, we advise to read the book by Cygan et al. [4]. An approach based on logic is given by Flum and Grohe [11]. Despite the recent successes of parameterized complexity, only very few results on the art gallery problem are known.

The first such result is the trivial algorithm for the vertex guard variant to check if a solution of size $k$ exists in a polygon with $n$ vertices. The algorithm runs in $O(n^{k+2})$ time, by checking all possible subsets of size $k$ of the vertices. The second *not so well-known* result is the fact that one can find in time $n^{O(k)}$ a set of $k$ guards for the point guard variant, if it exists [8], using tools from real algebraic geometry [1]. This was first observed by Sharir [8, Acknowledgment]. 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 want to mention another very natural variant of the art gallery problem, where we do not require to guard the whole polygon, but only $k$ distinct points. This variant is fixed parameter tractable, which was pointed out to us by Bart M. P. Jansen. The algorithm is to compute the visibility region of each picture and compute the arrangement formed by these visibility regions. This naturally induces a set-cover instance with a ground set of size $k$. The instance can be solved in $O(2^k)$ time after elimination of dublicate sets. Nilsson and Zylinski have a similar result for the watchman route problem [27].

The Exponential Time Hypothesis (ETH) asserts that there is no $2^{o(N)}$ time algorithm for Sat on $N$ variables. The ETH is used to attain more precise conditional lower bounds than the mere NP-hardness. A simple reduction from Set Cover by Eidenbenz et al. shows that there is no $n^{o(k)}$ algorithm for these problems, when we consider polygons with holes [9, Sec.4], unless the ETH fails. However, polygons with holes are very different from simple polygons. For instance, they have unbounded VC-dimension while simple polygons have bounded VC-dimension [14, 15, 18, 31].

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

▶ **Theorem 1** (Parameterized hardness point guard). *Assuming the ETH,* Point Guard Art Gallery *is not solvable in time $f(k)\, n^{o(k/\log k)}$, for any computable function $f$, even on simple polygons, where $n$ is the number of vertices of the polygon and $k$ is the number of guards allowed.*

▶ **Theorem 2** (Parameterized hardness vertex guard). *Assuming the ETH,* Vertex Guard Art Gallery *is not solvable in time $f(k)\, n^{o(k/\log k)}$, for any computable function $f$, even on simple polygons, where $n$ is the number of vertices of the polygon and $k$ is the number of guards allowed.*

These results imply that the previous noted algorithms are essentially tight, and suggest that there are no better parameterized algorithms. Our reductions are from Subgraph Isomorphism and therefore an $f(k)\, n^{o(k)}$-algorithm for the art gallery problem would also imply improved algorithms for Subgraph Isomorphism and for CSP parameterized by treewidth, which would be considered a major breakthrough [23]. Let us also mention that our results imply that both variants are $W[1]$-hard parameterized by the number of guards.

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

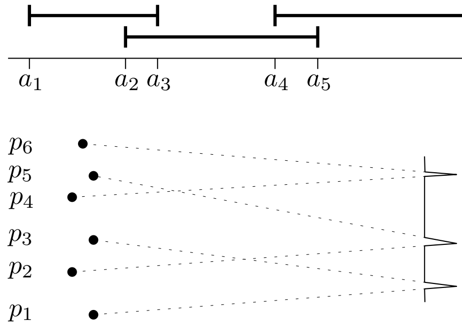


**Figure 1** Reduction from Hitting Set on interval graphs to a restricted version of the art gallery problem.

Let $I_1, \ldots, I_n$ be $n$ intervals with endpoints $a_1, \ldots, a_{2n}$. Then, we construct $2n$ points $p_1, \ldots, p_{2n}$ representing $a_1, \ldots, a_{2n}$. Further, we construct one pocket "far enough to the right" for each interval as described above. This way, we reduce Hitting Set on interval graphs to a restricted version of the art gallery problem. This observation is *not* so useful in itself since hitting set on interval graphs can be solved in polynomial time.

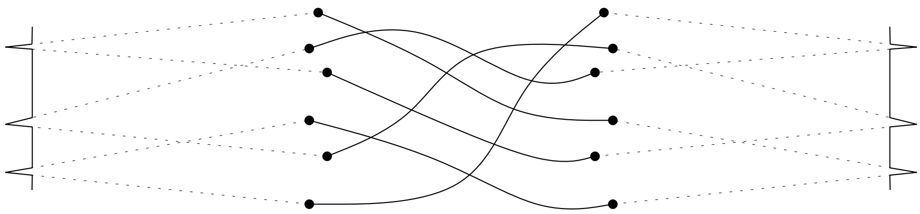

**Figure 2** Two instances of Hitting Set "magically" linked.

The situation changes rapidly if we consider Hitting Set on 2-track interval graphs, as described in Section 2. Unfortunately, we are not able to just "magically" link some specific pairs of points in the polygon of the art gallery instance. Therefore, we construct linker gadgets, which basically work as follows. We are given two set of points $P$ and $Q$ and a bijection $\sigma$ between $P$ and $Q$. The linker gadget is built in a way that it can be covered by two points $(p, q)$ of $P \times Q$, if and only if $q = \sigma(p)$. The Structured 2-Track Hitting Set problem will be specifically designed so that the linker gadget is the main remaining ingredient to show hardness.

**Organization of the paper.** In Section 2, we introduce some notations, discuss the encoding of the polygon, give some useful ETH-based lower bounds, and prove a technical lemma. Due to the space limitation, we cannot include all the proofs in the main body. In Section A of the appendix, we prove the lower bound for Structured 2-Track Hitting Set (Theorem 5). Lemma 9 shows the same hardness result for Set Cover restricted to sets formed as the union of two intervals, and contains the key arguments. In Section C of the appendix, we give an alternative and direct proof of Theorem 5. Should the reader be interested on how Theorem 5 can be proven, he or she can read Section C for a short and simple exposition of the ideas and/or read Section A for more technical and detailed proofs, together with pointers towards similar pre-existing results. From this point onwards, we can reduce from the particularly convenient Structured 2-Track Hitting Set. In Section 3, we show the lower bound for the Point Guard Art Gallery problem (Theorem 1). We design a linker gadget, show its correctness, and show how several linker gadgets can be combined consistently. In Section B of the appendix, we tackle the Vertex Guard Art Gallery problem (Theorem 2). We have to design a very different linker gadget, that has to be combined with other gadgets and ideas.

## 2 Preliminaries

For any two integers $x \leqslant y$, we set $[x, y] := \{x, x+1, \dots, y-1, y\}$, and for any positive integer $x$, $[x] := [1, x]$. Given two points $a, b$ in the plane, we define $\text{seg}(a, b)$ as the line segment with endpoints $a, b$. Given $n$ points $v_1, \dots, v_n \in \mathbb{R}^2$, we define a polygonal closed curve $c$ by $\text{seg}(v_1, v_2), \dots, \text{seg}(v_{n-1}, v_n), \text{seg}(v_n, v_1)$. If $c$ is not self intersecting, it partitions the plane into a closed bounded area and an unbounded area. The closed bounded area is a *simple polygon* on the vertices $v_1, \dots, v_n$. Note that we do not consider the boundary as the polygon but rather all the points bounded by the curve $c$ as described above. Given two points $a, b$ in a simple polygon $\mathcal{P}$, we say 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 “see through” vertices of the polygon. We say that $S$ is a set of *point guards* of $\mathcal{P}$, if every point $p \in \mathcal{P}$ 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 subset of the vertices of $\mathcal{P}$. The Point Guard Art Gallery problem and the Vertex Guard Art Gallery problem are formally defined as follows.

Point Guard Art Gallery
**Input:** The vertices of a simple polygon $\mathcal{P}$ in the plane and a natural number $k$.
**Question:** Does there exist a set of $k$ point guards for $\mathcal{P}$?

Vertex Guard Art Gallery
**Input:** A simple polygon $\mathcal{P}$ on $n$ vertices in the plane and a natural number $k$.
**Question:** Does there exist a set of $k$ vertex guards for $\mathcal{P}$?

For any two distinct points $v$ and $w$ in the plane we denote by $\text{ray}(v, w)$ the ray starting at $v$ and passing through $w$, and by $\ell(v, w)$ the supporting line passing through $v$ and $w$. 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 $y \in \mathcal{P}$ seen by $x$. 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 *sub-polygon* $\mathcal{P}'$ of a simple polygon $\mathcal{P}$ is defined 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 neighbors in $\mathcal{P}$) such that $v_1 v_l$ does not cross any edge of $\mathcal{P}$. In particular, $\mathcal{P}'$ is a simple polygon.

We assume that the vertices of the polygon are either given by integers or by rational numbers. We also assume that the output is given either by integers or by rational numbers. The instances we generate as a result of Theorem 1 and Theorem 2 have rational coordinates. We can represent them by specifying the nominator and denominator. The number of bits is bounded by $O(\log n)$ in both cases. We can transform the coordinates to integers by multiplying every coordinate with the least common multiple of all denominators. However, this leads to integers using $O(n \log n)$ bits.

**ETH-based lower bounds.** The *Exponential Time Hypothesis* (ETH) is a conjecture by Impagliazzo et al. [16] asserting that there is no $2^{o(n)}$-time algorithm for 3-SAT on instances with $n$ variables. The $k$-Multicolored-Clique problem has as input a graph $G = (V, E)$, where the set of vertices is partitioned into $V_1, \ldots, V_k$. It asks if there exists a set of $k$ vertices $v_1 \in V_1, \ldots, v_k \in V_k$ such that these vertices form a clique of size $k$. We will use the following lower bound proved by Chen et al. [3].

▶ **Theorem 3** ([3]). *There is no $f(k)\,n^{o(k)}$ algorithm for $k$-Multicolored-Clique, for any computable function $f$, unless the ETH fails.*

Marx showed that Subgraph Isomorphism cannot be solved in time $n^{o(k/\log k)}$ where $k$ is the number of *edges* of the pattern graph, under the ETH [23]. Usually, this result enables to improve a lower bound obtained by a reduction from Multicolored $k$-Clique with a quadratic blow-up on the parameter, from exponent $o(\sqrt{k})$ to exponent $o(k/\log k)$, by doing more or less the same reduction but from Multicolored Subgraph Isomorphism. The Multicolored Subgraph Isomorphism problem can be defined in the following way. One is given a graph with $n$ vertices partitioned into $l$ color classes $V_1, \ldots, V_l$ 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 color class so that the selected vertices induce $k$ edges. Observe that $l$ corresponds to the number of vertices of the pattern graph. The technique of color coding and the result of Marx imply that:

▶ **Theorem 4** ([23]). Multicolored Subgraph Isomorphism *cannot be solved in time $f(k)\,n^{o(k/\log k)}$ where $k$ is the number of edges of the solution and $f$ any computable function, unless the ETH fails.*

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

In the 2-Track Hitting Set problem, the input consists of an integer $k$, two totally ordered 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. 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, if possible, a set $S$ of $k$ 2-elements such that the first projection of $S$ is a hitting set of $\mathcal{S}_A$, and the second projection of $S$ is a hitting set of $\mathcal{S}_B$.

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 $\phi$. Given two integers $k$ and $t$, $A$ is partitioned into $(C_1, C_2, \ldots, C_k)$ where $C_j = \{a^j_1, a^j_2, \ldots, a^j_t\}$ for each $j \in [k]$. $A$ is ordered: $a^1_1, a^1_2, \ldots, a^1_t, a^2_1, a^2_2, \ldots, a^2_t, \ldots, a^k_1, a^k_2, \ldots, a^k_t$. We define $C'_j := \phi(C_j)$ and $b^j_i := \phi(a^j_i)$ for all $i \in [t]$ and $j \in [k]$. We now impose that $\phi$ is such that, for each $j \in [k]$, the set $C'_j$ is a $B$-interval. 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: $b^j_{\sigma_j(1)}, b^j_{\sigma_j(2)}, \ldots, b^j_{\sigma_j(t)}$. In what follows, it will be convenient to see an instance of Structured 2-Track Hitting Set as a tuple $\mathcal{I} = (k \in \mathbb{N}, t \in \mathbb{N}, \sigma \in \mathfrak{S}_k, \sigma_1 \in \mathfrak{S}_t, \ldots, \sigma_k \in \mathfrak{S}_t, \mathcal{S}_A, \mathcal{S}_B)$, where we recall that $\mathcal{S}_A$ is a set of $A$-intervals and $\mathcal{S}_B$ is a set of $B$-intervals. We denote by $[a^j_i, a^{j'}_{i'}]$ (resp. $[b^j_i, b^{j'}_{i'}]$) all the elements $a \in A$ (resp. $b \in B$) such that $a^j_i \leq_A a \leq_A a^{j'}_{i'}$ (resp. $b^j_i \leq_B b \leq_B b^{j'}_{i'}$).

Taking inspiration from previous results, we show hardness of Structured 2-Track Hitting Set by a reduction from Multicolored Subgraph Isomorphism (see Sections A and C of the appendix).

▶ **Theorem 5.** Structured 2-Track Hitting Set *is $W[1]$-hard, and not solvable in time $f(k)\,|\mathcal{I}|^{o(k/\log k)}$ for any computable function $f$, unless the ETH fails.*

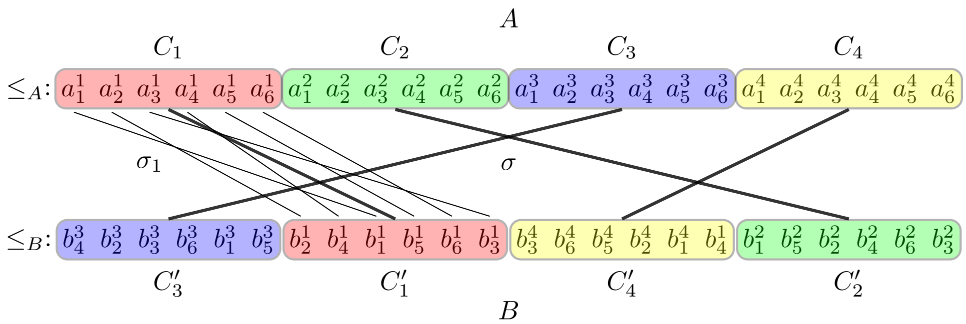


**Figure 3** An illustration of the $k+1$ permutations $\sigma \in \mathfrak{S}_k$, $\sigma_1 \in \mathfrak{S}_t$, ..., $\sigma_k \in \mathfrak{S}_t$ of an instance of Structured 2-Track Hitting Set, with $k = 4$ and $t = 6$.

**Bounding the coordinates.** We say a point $p = (p_x, p_y) \in \mathbb{Z}^2$ has coordinates bounded by $L$ if $|p_x|, |p_y| \leq L$. Given two vectors $v, w$, we denote their scalar product as $v \cdot w$. This technical lemma will prove useful to ensure that the polygon built in Section 3 can be described with integer coordinates.

▶ **Lemma 6.** *Let $p^1, q^1, p^2, q^2$ be four points with integer coordinates bounded by $L$. Then the intersection 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. The nominator and denominator of the coordinates are bounded by $O(L^2)$.*

**Proof.** The fact that $d$ lies on $\ell_i$ can be expressed as $v^i \cdot d = b^i$, with some appropriate vector $v^i$ and number $b^i$, for $i = 1, 2$. To be precise $v^i = (-p^i_x + q^i_x, p^i_y - q^i_y)$ and $b^i = v_i \cdot p^i$, for $i = 1, 2$. We define the matrix $A = (v^1, v^2)$ and the vector $b = (b^1, b^2)$. Then both conditions can be expressed 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 determinant of the matrix $M$. By Cramer's rule, it holds that $d_x = \frac{\det(A_1)}{\det(A)}$ and $d_y = \frac{\det(A_2)}{\det(A)}$. ◀

## 3 Parameterized hardness of the point guard variant

As exposed in the introduction, we give a reduction from the Structured 2-Track Hitting Set problem. The main challenge is to design a *linker* gadget that groups together specific pairs of points in the polygon. The following introductory lemma inspires the *linker* gadgets for both Point Guard Art Gallery and Vertex Guard Art Gallery.

▶ **Lemma 7.** *The only minimum hitting sets of the set-system $\mathcal{S} = \{S_i = \{1, 2, \ldots, i, \overline{i+1}, \overline{i+2}, \ldots, \overline{n}\} \mid i \in [n]\} \cup \{\overline{S_i} = \{\overline{1}, \overline{2}, \ldots, \overline{i}, i+1, i+2, \ldots, n\} \mid i \in [n]\}$ are $\{i, \overline{i}\}$, for each $i \in [n]$.*

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

▶ **Theorem 1** (Parameterized hardness point guard). *Assuming the ETH,* Point Guard Art Gallery *is not solvable in time $f(k)\,n^{o(k/\log k)}$, for any computable function $f$, even* *on simple polygons, where $n$ is the number of vertices of the polygon and $k$ is the number of guards allowed.*

**Proof.** Given an instance $\mathcal{I} = (k \in \mathbb{N}, t \in \mathbb{N}, \sigma \in \mathfrak{S}_k, \sigma_1 \in \mathfrak{S}_t, \ldots, \sigma_k \in \mathfrak{S}_t, \mathcal{S}_A, \mathcal{S}_B)$ of Structured 2-Track Hitting Set, 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.

**Outline.** We recall that $A$'s order is: $a^1_1, \ldots, a^1_t, \ldots, a^k_1, \ldots, a^k_t$ and $B$'s order is determined by $\sigma$ and the $\sigma_j$'s (see Figure 3). Let us focus on one color class $j \in [k]$ together with a permutation $\sigma_j : A \to B$. The global strategy of the reduction is to *allocate*, $2t$ special points for this polygon. The points $a^j_1, \ldots, a^j_t$ on track $A$ are represented by $\alpha^j_1, \ldots, \alpha^j_t$ points in $\mathcal{P}$. and the points $\sigma_j(a^j_1), \ldots, \sigma_j(a^j_t)$ on track $B$ are represented by $\beta^j_1, \ldots, \beta^j_t$ in the polygon. Placing a guard in $\alpha^j_i$ and $\beta^j_i$ shall correspond to picking the 2-element $(a^j_i, \sigma_j(b^j_i))$. The points $\alpha^j_i$'s and $\beta^j_i$'s ordered by increasing $y$-coordinates will match the order of the $a^j_i$'s along the order $\leq_A$ and then of the $b^j_i$'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 4).

The first critical issue will be to *link* point $\alpha^j_i$ to point $\beta^j_i$. 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 $\alpha^j_i$ and $\beta^j_{i'}$ with $i \neq i'$. The so-called *point linker* gadget will realize the intervals as described in Lemma 7.

The second critical issue is to enforce these positions. For this purpose, we will need to introduce a *copy* $\overline{\alpha}^j_i$ of each $\alpha^j_i$. 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 $\alpha^j_i$, $\overline{\alpha}^j_i$, and $\beta^j_i$ for some $i \in [t]$ (see Figure 7). 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.

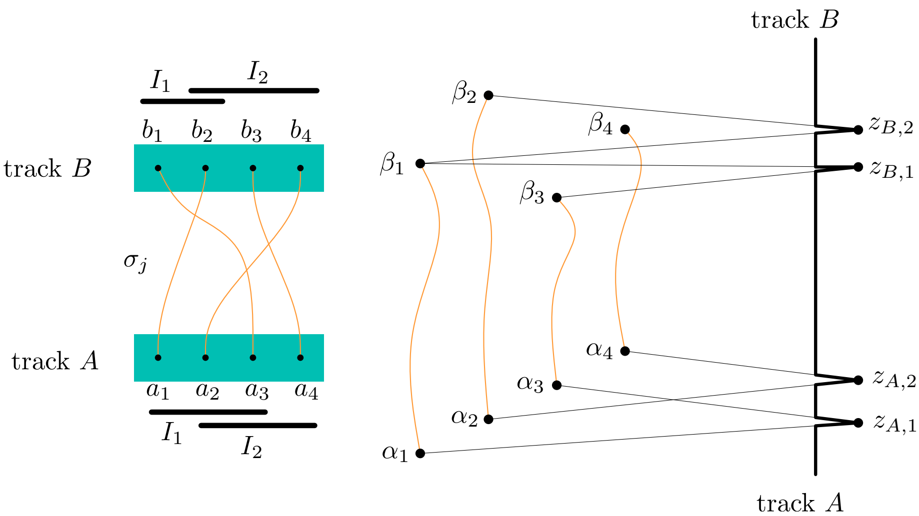


**Figure 4** The elements in one color class are represented by points $\alpha_1, \ldots, \alpha_t$ and $\beta_1, \ldots, \beta_t$. Here, we suppressed the super index $j$.

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 with an explicit specification of the coordinates. The description will be dependent on some parameters $x, y, L, D, F$ that we will specify later. The value $x$ represents the offset between elements with respect to the $x$-coordinate and likewise the value $y$ represents the offset between elements with respect to the $y$-coordinate. $D$ represents the vertical distance between different color classes and $L$ represents the horizontal distance between all the $\alpha's$ and the $\beta's$, see also Figure 8. The value $F$ will become relevant later and describes the distance of the points to the pockets to the far right. The crucial point of the construction is that the order of the $\alpha$'s corresponds exactly to the order of the $a$'s along track $A$ and the same relation holds between the $\beta$'s and $b$'s.

We recall that we want the points $\alpha_i^j$'s and $\beta_i^j$'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 $\alpha_1^1$), the $y$-coordinates of the $\alpha_i^j$'s are regularly spaced out by an offset $y$; that is, the $y$-coordinate of $\alpha_i^j$ 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 $\beta_i^j$ is $y_1 + (kt-1)y + L + (\text{ord}(b_i^j) - 1)y$ (for any $j \in [k]$ and $i \in [t]$) where $\text{ord}(b_i^j)$ is the rank of $b_i^j$ along the order $\leq_B$.

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, say[1], $tx$ such that the uppermost longer side of the rectangular pocket lies on the line $\ell(\alpha_1^j, \alpha_t^j)$ (see Figure 6). 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 $\beta_i^j$ to $x_j + (i-1)x$, so that $\beta_i^j$ and $\alpha_i^j$ share the same $x$-coordinate. One can check that it is consistent with the previous paragraph. We also observe that, by the choice of the $y$-coordinate for the $\beta_i^j$'s, we have both encoded the permutations $\sigma_j$'s and permutation $\sigma$ (see Figure 8 or Figure 6). This finishes the description of the coordinates.

Now, we will give a description how, we can encode intervals by on track $A$ and $B$ by small pockets and, we describe, where to place them. From hereon, for a vertex $v$ and two

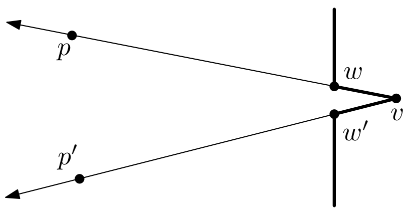


**Figure 5** A triangular pocket.

points $p$ and $p'$, we informally call *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), see Figure 5. 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 $p$

[1] the 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 $\ell(\alpha_1^j, \alpha_t^j)$

and $p'$. It is easy to see that each point that sees $v$ also sees the entire triangular pocket $P(v)$.

For each $A$-interval $I_q = [a_i^j, a_{i'}^{j'}] \in \mathcal{S}_A$ we construct one 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 $\text{ray}(z_{A,q}, \alpha_{i'}^{j'})$. The placement of this triangular pocket is very far to the right. The $x$-coordinate of $z_{A,q}$ equals $x_k + (t-1)x + F$, for some large value $F$ to be specified later. The $y$-coordinate shall be between $y_1$ and $y_k + (kt-1)y$. We place those $|\mathcal{S}_A|$ pockets along the $y$-axis, and space them out by some small distance $s$. To guarantee that we have enough room to place all those pockets, $s$ will be chosen sufficiently small ($s \ll y$).

We will show later, for appropriate values $y \ll x \ll D \ll F$, the only $\alpha_{i''}^{j''}$ seeing vertex $z_{A,q}$ should be the points such that $a_i^j \leq_A a_{i''}^{j''} \leq_A a_{i'}^{j'}$ (see Figure 8 and Figure 4).

Similarly, we represent each interval $I_q \in \mathcal{S}_B$ by a triangular pocket rooted at $z_{B,q}$. These pockets are placed at the $x$-coordinate $x_k + (t-1)x + F$ and spaced out by distance $s$ along the $y$-axis between $y$-coordinates $y_1 + (kt-1)y + L$ and $y_1 + 2(kt-1)y + L$. The $B$-interval $I_q = [b_i^j, b_{i'}^{j'}]$ is represented by the triangular pocket $\mathcal{P}(z_{B,q})$ rooted at vertex $z_{B,q}$ supported by $\text{ray}(z_{B,q}, \sigma_j(a_i^j))$ and $\text{ray}(z_{B,q}, \sigma_j(a_{i'}^{j'}))$. Note that $\sigma_j(a_i^j)$ is the point on track $B$ that corresponds to $\beta_i^j$. 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 8.

Now, we describe how we *link* each point $\alpha_i^j$ to its associate $\beta_i^j$. For each $j \in [k]$, let us mentally draw $\text{ray}(\alpha_t^j, \beta_1^j)$ and consider points slightly to the left of this ray at a distance, say, $L'$ from point $\alpha_t^j$. 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 the order $\alpha_1^j$, $\alpha_2^j$ up to $\alpha_t^j$, and then, $\beta_1^j$, $\beta_2^j$ up to $\beta_t^j$. This observation relies on the fact that $y \ll x \ll L$. So, from the distance, the points $\beta_1^j, \ldots, \beta_t^j$ 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 $\text{ray}(c_i^j, \alpha_{i+1}^j)$ and $\text{ray}(c_i^j, \beta_i^j)$. We place also a triangular pocket $\mathcal{P}(c_t^j)$ rooted at $c_t^j$ supported by $\text{ray}(c_i^j, \beta_1^j)$ and $\text{ray}(c_i^j, \beta_t^j)$. We place vertices $c_i^j$ and $c_{i+1}^j$ at the same $y$-coordinate and spaced out by distance $x$ along the $x$-axis (see Figure 6). Similarly, let us informally refer to the region slightly to the right of $\text{ray}(\alpha_1^j, \beta_t^j)$ at a distance $L'$ from point $\alpha_1^j$, as $\mathcal{R}_{\text{right}}^j$. Any point $\mathcal{R}_{\text{right}}^j$ sees, from right to left, in this order $\beta_1^j$, $\beta_2^j$ up to $\beta_t^j$, and then, $\alpha_1^j$, $\alpha_2^j$ up to $\alpha_t^j$. 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 $\text{ray}(d_i^j, \beta_{i+1}^j)$ and $\text{ray}(c_i^j, \alpha_i^j)$. We place also a triangular pocket $\mathcal{P}(d_t^j)$ rooted at $d_t^j$ supported by $\text{ray}(d_i^j, \alpha_1^j)$ and $\text{ray}(c_i^j, \alpha_t^j)$. Again, those $t$ pockets are placed at the same $y$-coordinate and spaced out horizontally by $x$ (see Figure 6). We denote by $\mathcal{P}_{j,\alpha,\beta}$ the set of pockets $\{\mathcal{P}(c_1^j), \ldots, \mathcal{P}(c_t^j), \mathcal{P}(d_1^j), \ldots, \mathcal{P}(d_t^j)\}$ and informally call it the *weak point linker* (or simply, *weak linker*) of $\alpha_1^j, \ldots, \alpha_t^j$ and $\beta_1^j, \ldots, \beta_t^j$. 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 $\mathcal{P}_{j,\alpha,\beta} = \{\mathcal{P}(c_1^j), \ldots, \mathcal{P}(c_t^j), \mathcal{P}(d_1^j), \ldots, \mathcal{P}(d_t^j)\}$ and one first decides to put a guard on point $\alpha_i^j$ (for some $i \in [t]$), then one is not forced to put the other guard on point $\beta_i^j$ but only on an area whose uppermost point is $\beta_i^j$ (see the shaded areas below the $b_i^j$'s in Figure 6). Now, if the points $\beta_1^j, \ldots, \beta_t^j$ would all lie on a common line $\ell$, we could shrink the shaded area of each $\beta_i^j$ (Figure 6) down to the single point $\beta_i^j$ by adding a thin rectangular pocket on $\ell$ (similarly to what we have for $\alpha_1^j, \ldots, \alpha_t^j$). Naturally, we need that $\beta_1^j, \ldots, \beta_t^j$ are *not* on a common line to be able to encode the permutation $\sigma_j$. The remedy we pursue is the following. For each $j \in [k]$, we allocate $t$ points $\overline{\alpha}_1^j, \overline{\alpha}_2^j, \ldots, \overline{\alpha}_t^j$ on a horizontal line, spaced out by distance $x$, say, $\approx \frac{D}{2}$ to the right and $\approx L$ above of $\beta_t^j$. We place a thin

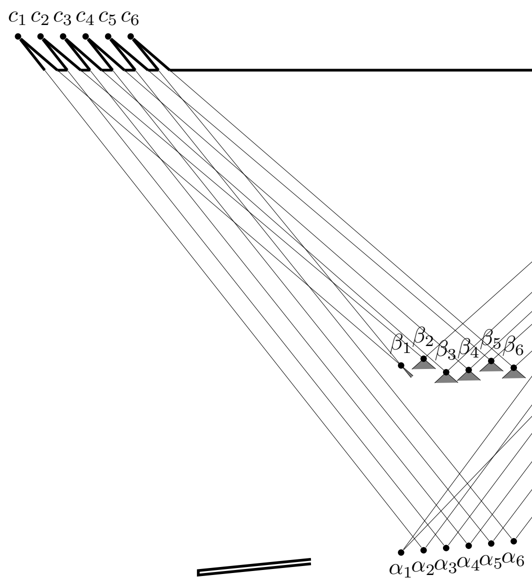


**Figure 6** Weak point linker gadget.

horizontal rectangular pocket $\mathcal{P}_{j,\overline{r}}$ of the same dimension as $\mathcal{P}_{j,r}$ such that the lowermost longer side of $\mathcal{P}_{j,\overline{r}}$ is on the line $\ell(\overline{\alpha}_1^j, \overline{\alpha}_t^j)$. We add the $2t$ pockets corresponding to a weak linker $\mathcal{P}_{j,\alpha,\overline{\alpha}}$ between $\alpha_1^j, \dots, \alpha_t^j$ and $\overline{\alpha}_1^j, \dots, \overline{\alpha}_t^j$ as well as the $2t$ pockets of a weak linker $\mathcal{P}_{j,\overline{\alpha},\beta}$ between $\overline{\alpha}_1^j, \dots, \overline{\alpha}_t^j$ and $\beta_1^j, \dots, \beta_t^j$ as pictured in Figure 7. We denote by $\mathcal{P}_j$ the union $\mathcal{P}_{j,r} \cup \mathcal{P}_{j,\overline{r}} \cup \mathcal{P}_{j,\alpha,\beta} \cup \mathcal{P}_{j,\alpha,\overline{\alpha}} \cup \mathcal{P}_{j,\overline{\alpha},\beta}$ of all the pockets involved in the encoding of color class $j$. Now, say, one wants to guard all the pockets of $\mathcal{P}_j$ with only three points, and chooses to put a guard on $\alpha_i^j$ (for some $i \in [t]$). Because of the pockets of $\mathcal{P}_{j,\alpha,\overline{\alpha}} \cup P_{j,\overline{r}}$, one is forced to place a second guard precisely on $\overline{\alpha}_i^j$. Now, because of the weak linker $\mathcal{P}_{j,\alpha,\beta}$ the third guard should be on a region whose uppermost point is $\beta_i^j$, while, because of $\mathcal{P}_{j,\overline{\alpha},\beta}$ the third guard should be on a region whose lowermost point is $\beta_i^j$. The conclusion is that the third guard should be put precisely on $\beta_i^j$. This *triangle* of weak linkers is called the *linker* of color class $j$. The $k$ linkers are placed accordingly to Figure 8. This ends the construction.

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

We can safely set $s$ to one, as it is the smallest length, we specified. We will put $|\mathcal{S}_a|$ pockets on track $A$ and $|\mathcal{S}_b|$ pockets on track $B$. It is sufficient to have an opening space of one between them. Thus, the space on the right side of $\mathcal{P}$, for all pockets of track $A$ is bounded by $2|\mathcal{S}_a|$. Thus setting $y$ to $|\mathcal{S}_a| + |\mathcal{S}_b|$ secures us that we have plenty of space to place all the pockets. We specify $F = (|\mathcal{S}_a| + |\mathcal{S}_b|)Dk = yDk$. We have to show that this is large enough to guarantee that the pockets on track $A$ distinguish the picked points only by the $y$-coordinate. Let $p$ and $q$ be two points among the $\alpha_i^j$. Their vertical distance is upper bounded by $Dk$ and their horizontal distance is lower 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 $\ell$ will be at least $F\frac{y}{Dk}$ above the pockets of track $A$. Note $F\frac{y}{Dk} = yDk\frac{y}{Dk} > y^2 > |\mathcal{S}_a|^2 > 2|\mathcal{S}_a|$. The same argument shows

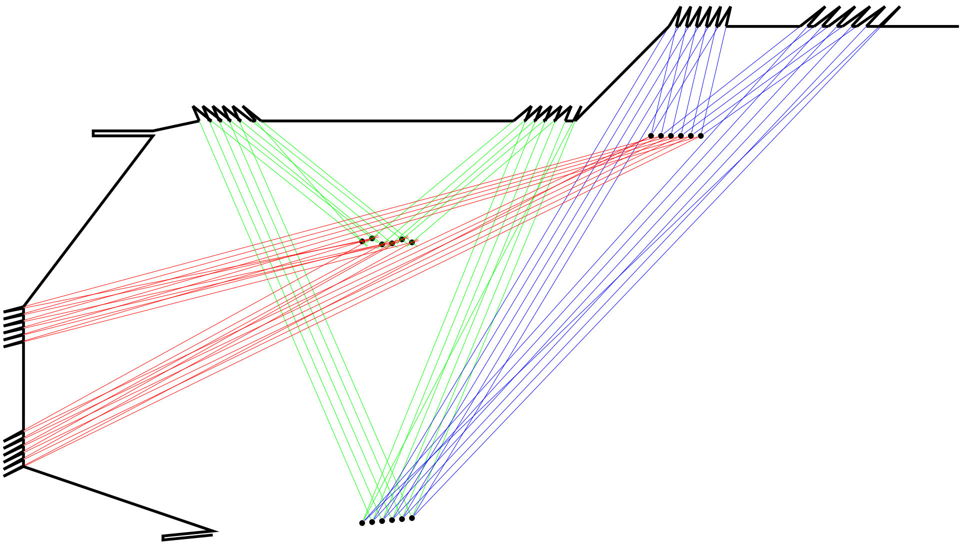

**Figure 7** Point linker gadget: a triangle of (three) weak point linkers.

that $F$ is sufficiently large for track $B$.

The remaining lengths $x, L, L'$, and $D$ can be specified in a similar fashion. For the construction of the pockets, let $s \in \mathcal{S}_a$ be an $A$-interval with endpoints $a$ and $b$, represented by some points $p$ and $q$ and assume the opening vertices $v$ and $w$ of the triangular pocket are already specified. Then 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 6, $x$ has rational coordinates and the integers to represent them can be expressed by the coordinates of $p, q, v$, and $w$. This way, all the pockets can be explicitly constructed using rational coordinates as claimed above.

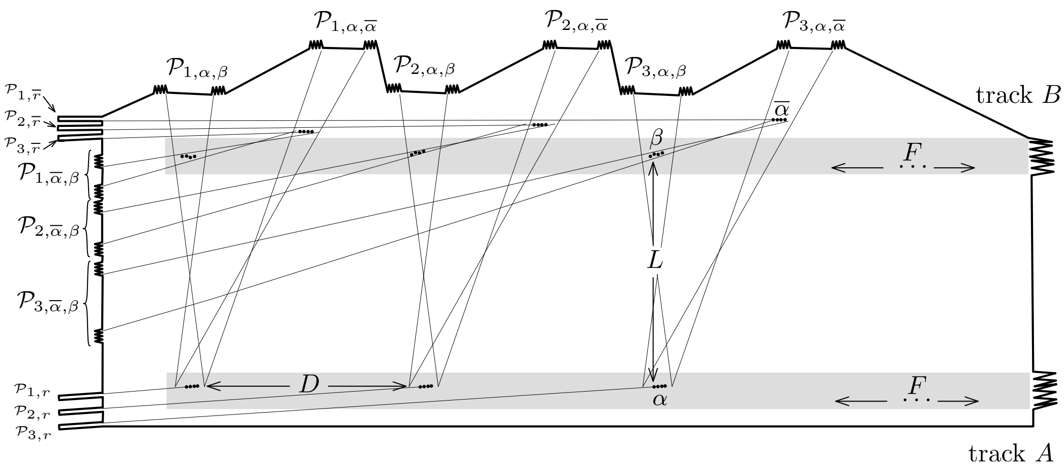


**Figure 8** The overall picture of the reduction with $k = 3$.

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

▶ **Lemma 8.** $\forall j \in [k]$, $\forall i \in [t]$, *the three associate points* $\alpha_i^j$, $\overline{\alpha}_i^j$, $\beta_i^j$ *guard entirely* $\mathcal{P}_j$.

**Proof.** The rectangular pockets $\mathcal{P}_{j,r}$ and $\mathcal{P}_{j,\overline{r}}$ are entirely seen by respectively $\alpha_i^j$ and $\overline{\alpha}_i^j$. The pockets $\mathcal{P}(c_1^j), \mathcal{P}(c_2^j), \dots \mathcal{P}(c_{i-1}^j)$ and $\mathcal{P}(d_i^j), \mathcal{P}(d_{i+1}^j), \dots \mathcal{P}(d_t^j)$ are all entirely seen by $\alpha_i^j$, while the pockets $\mathcal{P}(c_i^j), \mathcal{P}(c_{i+1}^j), \dots \mathcal{P}(c_t^j)$ and $\mathcal{P}(d_1^j), \mathcal{P}(d_2^j), \dots \mathcal{P}(d_{i-1}^j)$ are all entirely seen by $\beta_i^j$. This means that $\alpha_i^j$ and $\beta_i^j$ jointly see all the pockets of $\mathcal{P}_{j,\alpha,\beta}$. Similarly, $\alpha_i^j$ and $\overline{\alpha}_i^j$ jointly see all the pockets of $\mathcal{P}_{j,\alpha,\overline{\alpha}}$, and $\overline{\alpha}_i^j$ and $\beta_i^j$ jointly see all the pockets of $\mathcal{P}_{j,\overline{\alpha},\beta}$. Therefore, $\alpha_i^j$, $\overline{\alpha}_i^j$, $\beta_i^j$ jointly see all the pockets of $\mathcal{P}_j$. ◀

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 $G = \{\alpha_{s_1}^1, \overline{\alpha}_{s_1}^1, \beta_{s_1}^1, \dots, \alpha_{s_k}^k, \overline{\alpha}_{s_k}^k, \beta_{s_k}^k\}$ guard the whole polygon $\mathcal{P}$. By Lemma 8, $\forall j \in [k]$, $\mathcal{P}_j$ is guarded. For each $A$-interval (resp. $B$-interval) in $\mathcal{S}_A$ (resp. $\mathcal{S}_B$) there is at 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 guarded by $\alpha_{s_j}^j$ (resp. $\beta_{s_j}^j$). The rest of the polygon $\mathcal{P}$ (which is not part of pockets) is guarded by, for instance, $\{\overline{\alpha}_{s_1}^1, \dots, \overline{\alpha}_{s_k}^k\}$. So, $G$ is indeed a solution and it contains $3k$ points.

Assume now that there is no solution to the instance $\mathcal{I}$ of Structured 2-Track Hitting Set. We show that there is no set of $3k$ points guarding $\mathcal{P}$. We observe that no point of $\mathcal{P}$ sees inside two triangular pockets one being in $\mathcal{P}_{j,\alpha,\gamma}$ and the other in $\mathcal{P}_{j',\alpha,\gamma'}$ with $j \neq j'$ and $\gamma, \gamma' \in \{\beta, \overline{\alpha}\}$. Further, $V(r(\mathcal{P}_{j,\alpha,\beta} \cup \mathcal{P}_{j,\alpha,\overline{\alpha}})) \cap V(r(\mathcal{P}_{j',\alpha,\beta} \cup \mathcal{P}_{j',\alpha,\overline{\alpha}})) = \emptyset$ when $j \neq j'$, where $r$ maps a set of triangular pockets to the set of their root. Also, for each $j \in [k]$, seeing entirely $\mathcal{P}_{j,\alpha,\beta}$ and $\mathcal{P}_{j,\alpha,\overline{\alpha}}$ requires at least 3 points. This means that for each $j \in [k]$, one should place three guards in $V(r(\mathcal{P}_{j,\alpha,\beta} \cup \mathcal{P}_{j,\alpha,\overline{\alpha}}))$. Furthermore, one can observe among those three points one should guard a triangular pocket $\mathcal{P}_{j',r}$ and another should guard $\mathcal{P}_{j'',\overline{r}}$. Let us try to guard entirely $\mathcal{P}_1$ and two rectangular pockets $\mathcal{P}_{j',r}$ and $\mathcal{P}_{j'',\overline{r}}$, with only three guards. Let call $\ell_1$ (resp. $\ell_1'$) the line corresponding to the extension of the uppermost (resp. lowermost) longer side of $\mathcal{P}_{1,r}$ (resp. $\mathcal{P}_{1,\overline{r}}$). The only points of $\mathcal{P}$ that can see a rectangular pocket $\mathcal{P}_{j',r}$ and at least $t$ pockets of $\mathcal{P}_{1,\alpha,\overline{\alpha}}$ are on $\ell_1$: more specifically, they are the points $\alpha_1^1, \dots, \alpha_t^1$. The only points that can see a rectangular pocket $\mathcal{P}_{j'',\overline{r}}$ and at least $t$ pockets of $\mathcal{P}_{1,\alpha,\overline{\alpha}}$ are on $\ell_1'$: they are the points $\overline{\alpha}_1^1, \dots, \overline{\alpha}_t^1$. As $\mathcal{P}_{1,\alpha,\overline{\alpha}}$ has $2t$ pockets, one has to take a point $\alpha_i^1$ and $\overline{\alpha}_{i'}^1$. By the same argument argument as in Lemma 7, $i$ should be equal to $i'$ (otherwise, $i < i'$ and the left pocket pointing towards $\overline{\alpha}_{i'-1}^1$ and $\alpha_{i'}^1$ is not seen, or $i > i'$ and the right pocket pointing towards $\alpha_{i+1}^1$ and $\overline{\alpha}_i^1$ is not seen). We now denote 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 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 shaded area of Figure 6). That is, the third guard should be on a region in which $\beta_{s_1}^1$ is the uppermost point. The same argument with the pockets of $\mathcal{P}_{1,\overline{\alpha},\beta}$ implies that the third guard should also be on a region in which $\beta_{s_1}^1$ is the lowermost point. Thus, the position of the third guard has to be point $\beta_{s_1}^1$. Therefore, one should put guards on points $\alpha_{s_1}^1$, $\overline{\alpha}_{s_1}^1$, and $\beta_{s_1}^1$, for some $\alpha_1 \in [t]$.

As none of those three points see any pocket $\mathcal{P}_{j,\overline{\alpha},\beta}$ with $j > 1$ (we already mentioned that no pocket of $\mathcal{P}_{j,\alpha,\beta}$ and $\mathcal{P}_{j,\alpha,\overline{\alpha}}$ with $j > 1$ can be seen by those points), we can repeat the argument for the second color class; and so forth up to color class $k$. Thus, a potential solution with $3k$ guards should be of the form $\{\alpha_{s_1}^1, \overline{\alpha}_{s_1}^1, \beta_{s_1}^1, \dots, \alpha_{s_k}^k, \overline{\alpha}_{s_k}^k, \beta_{s_k}^k\}$. As there is no solution to $\mathcal{I}$, there should be a set in $\mathcal{S}_A \cup \mathcal{S}_B$ that is not hit by $\{(a_{s_1}^1, b_{s_1}^1), \dots, (a_{s_k}^k, b_{s_k}^k)\}$. By construction, the pocket associated to this set is not entirely seen. ◀

## References


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

2 B. Brodén, M. Hammar, and B. J. Nilsson. Guarding lines and 2-link polygons is APX-hard. 2001.

3 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.

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

5 P. J. de Rezende, C. C. de Souza, S. Friedrichs, M. Hemmer, A. Kröller, and D. C. Tozoni. Engineering art galleries. *CoRR*, abs/1410.8720, 2014.

6 M. Dom, M. R. Fellows, F. A. Rosamond, and S. Sikdar. The parameterized complexity of stabbing rectangles. *Algorithmica*, 62(1-2):564–594, 2012.

7 R. G. Downey and M. R. Fellows. *Parameterized complexity*. Springer Science & Business Media, 2012.

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

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

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

11 J. Flum and M. Grohe. Parameterized complexity theory, volume xiv of texts in theoretical computer science. an EATCS series, 2006.

12 S. K. Ghosh. *Visibility algorithms in the plane*. Cambridge University Press, 2007.

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

14 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.

15 A. Gilbers and R. Klein. A new upper bound for the VC-dimension of visibility regions. *Computational Geometry*, 47(1):61–74, 2014.

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

17 M. Jiang. On the parameterized complexity of some optimization problems related to multiple-interval graphs. *Theor. Comput. Sci.*, 411(49):4253–4262, 2010.

18 G. Kalai and J. Matoušek. Guarding galleries where every point sees a large area. *Israel Journal of Mathematics*, 101(1):125–139, 1997.

19 M. J. Katz and G. S. Roisman. On guarding the vertices of rectilinear domains. *Computational Geometry*, 39(3):219–228, 2008.

20 J. King. Fast vertex guarding for polygons with and without holes. *Comput. Geom.*, 46(3):219–231, 2013.

21 D. G. Kirkpatrick. An $O(lglgOPT)$-approximation algorithm for multi-guarding galleries. *Discrete & Computational Geometry*, 53(2):327–343, 2015.

22 E. Krohn and B. J. Nilsson. Approximate guarding of monotone and rectilinear polygons. *Algorithmica*, 66(3):564–594, 2013.

23 D. Marx. Can you beat treewidth? *Theory of Computing*, 6(1):85–112, 2010.

24 D. Marx and M. Pilipczuk. Optimal parameterized algorithms for planar facility location problems using Voronoi diagrams. In *ESA 2015*, pages 865–877, 2015.

25 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.

26 R. Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, 2006.
27 B. J. Nilsson and P. Żyliński. How to keep an eye on a few small things. In *EuroCG 2014*. 2014.
28 J. O'rourke. *Art gallery theorems and algorithms*, volume 57. Oxford University Press Oxford, 1987.
29 T. C. Shermer. Recent results in art galleries [geometry]. *Proceedings of the IEEE*, 80(9):1384–1399, 1992.
30 J. Urrutia et al. Art gallery and illumination problems. *Handbook of computational geometry*, 1(1):973–1027, 2000.
31 P. Valtr. Guarding galleries where no point sees a small area. *Israel Journal of Mathematics*, 104(1):1–16, 1998.

## A Parameterized hardness of Structured 2-Track Hitting Set

The purpose of this section is to show Theorem 5. As we will see at the end of the section, there already exist quite a few parameterized hardness results for set cover/hitting set problems restricted to instances with some geometric flavor. The crux of the proof of Theorem 5 lies in Lemma 9. We introduce a few notation and vocabulary to state and prove this lemma.

Given a finite totally ordered set $Y = \{y_1, \ldots, y_{|Y|}\}$ (that is, for any $i, j \in [|Y|]$, $y_i \leq y_j$ iff $i \leqslant j$), 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 of $Y$. A set-system $(X, \mathcal{S})$ is said *two-block* if $X$ can be partitioned into two totally ordered sets $A = \{a_1, \ldots, a_{|A|}\}$ and $B = \{b_1, \ldots, b_{|B|}\}$ such that each set $S \in \mathcal{S}$ is the union of an $A$-interval with a $B$-interval.

▶ **Lemma 9.** $k$-Set Cover *restricted to two-block instances with $N$ elements and $M$ sets is $W[1]$-hard and not solvable in time $f(k)\,(N+M)^{o(k/\log k)}$ for any computable function $f$, unless the ETH fails.*

**Proof.** We present the reduction from the more common problem Multicolored $k$-Clique. Though, in the end, we will see that we can do the same reduction from Multicolored Subgraph Isomorphism to obtain the claimed lower bound. Let $G = (V = V_1 \cup \ldots \cup V_k, E)$ be an instance of Multicolored $k$-Clique such that $\forall i \in [k]$, $V_i = \{v^i_1, \ldots, v^i_t\}$, $m = |E|$, and $n = |V| = tk$. We can indeed assume that each $V_i$ has the same cardinality $t$ by potentially adding dummy isolated vertices to the instance. For each pair $i < j \in [k]$, $E_{ij}$ denotes the set of edges $E(V_i, V_j)$ between $V_i$ and $V_j$. For each $E_{ij}$ we give an arbitrary order to the edges: $e^{ij}_1, \ldots, e^{ij}_{|E_{ij}|}$. We build an equivalent instance $(X, \mathcal{S})$ of $k$-Set Cover 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 call $A$ and $B$ the two sets of the partition of $X$ that realizes that $(X, \mathcal{S})$ is two-block.

For each of the color class $V_i$, we add $tk + 2$ elements to $A$ with the following order: $x_b(i), x(i,1,1), \ldots,\ x(i,1,t), x(i,2,1), \ldots, x(i,2,t), \ldots, x(i,i-1,1), \ldots,\ x(i,i-1,t), x(i,i+1,1), \ldots,\ x(i,i+1,t), \ldots, x(i,k+1,1), \ldots, x(i,k+1,t), x_e(i)$, and call $X(i)$ the set containing those elements. We also denote by $X(i,j)$ the set $\{x(i,j,1), x(i,j,2), \ldots, x(i,j,t)\}$ (hence, $X(i) = \bigcup_{j \neq i} X(i,j) \cup \{x_b(i), x_e(i)\}$). For each $E_{ij}$, we add $3|E_{ij}| + 2$ elements to $B$ with the order: $y_b(i,j), y(i,j,1), \ldots,\ y(i,j,3|E_{ij}|), y_e(i,j)$, and denote by $Y(i,j)$ the set containing them. For each pair $i < j \in [k]$ and for each edge $e^{ij}_c = v^i_a v^j_b$ in $E_{ij}$ (with $a, b \in [t]$ and $c \in [|E_{ij}|]$), we add to $\mathcal{S}$ the two sets $S(e^{ij}_c, v^i_a) = \{x(i,j,a), x(i,j,a+1), \ldots, x(i,j,t),\ x(i,j+1,1), \ldots, x(i,j+1,a-1)\} \cup \{y(i,j,c),\ \ldots, y(i,j,c+|E_{ij}|-1)\}$ and $S(e^{ij}_c, v^j_b) = \{x(j,i,b), x(j,i,b+1), \ldots, x(j,i,t),\ x(j,i+1,1), \ldots x(j,i+1,b-1)\} \cup \{y(i,j,c+|E_{ij}|), \ldots, y(i,j,c+2|E_{ij}|-1)\}$. Observe that in case $j = i+1$, then all the elements of the form $x(j,i+1,\cdot)$ in set $S(e^{ij}_c, v^j_b)$ 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 $S(e^{ij}_c, v^i_a)$ (resp. in set $S(e^{ij}_c, v^j_b)$). For each pair $i < j \in [k]$, we also add to $A$ the $|E_{ij}|+2$ elements of a set $Z(i,j)$ ordered: $z_b(i,j), z(i,j,1), \ldots, z(i,j,|E_{ij}|), z_e(i,j)$, and for each edge $e^{ij}_c$ in $E_{ij}$ (with $c \in [|E_{ij}|]$), we add to $\mathcal{S}$ the two sets $S(e^{ij}_c, \vdash) = \{z_b(i,j), z(i,j,1), \ldots, z(i,j,|E_{ij}|-c\} \cup \{y_b(i,j), y(i,j,1) \ldots y(i,j,c-1)\}$ and $S(e^{ij}_c, \dashv) = \{z(i,j,|E_{ij}|-c+1), \ldots, z(i,j,|E_{ij}|, z_e(i,j)\} \cup \{y(i,j,c+2|E_{ij}|) \ldots y(i,j,3|E_{ij}|), y_e(i,j)\}$. Finally, for each $i \in [k]$, we add to $B$ the $t+2$ elements of a set $W(i)$ ordered: $w_b(i), w(i,1), \ldots,$ $w(i,t), w_e(i)$, and for all $a \in [t]$, we add the sets $S(i,a,\vdash) = \{x_b(i), x(i,1,1), \ldots, x(i,1,a-1)\}$ $\cup \{w_b(i), w(i,1), \ldots, w(i,t-a+1)\}$ and $S(i,a,\dashv) = \{x(i,k+1,a), \ldots, x(i,k+1,t), x_e(i)\} \cup$ $\{w(i,t-a+2), \ldots, w(i,t), w_e(i)\}$.

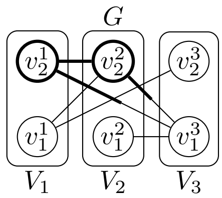


**Figure 9** A simple instance of MULTICOLORED $k$-CLIQUE. The elements in bold: vertices $v_2^1$ and $v_2^2$, edge $v_2^1v_2^2$, and half of the edges $v_2^1v_1^3$ and $v_2^2v_1^3$ correspond to the selection of sets depicted in Figure 10.

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 order: $X(1), X(2), \ldots, X(k), Z(1,2), Z(1,3), \ldots, Z(1,k), Z(2,3), \ldots, Z(2,k), \ldots, Z(k-2,k-1), Z(k-2,k), Z(k-1,k)$ for $A$ and $Y(1,2), Y(1,3), \ldots, Y(1,k), Y(2,3), \ldots, Y(2,k), \ldots, Y(k-2,k-1), Y(k-2,k), Y(k-1,k), W(1), W(2), \ldots, W(k)$ for $B$. We ask for a set cover with $2k^2$ sets. This ends the construction (see Figure 10 for an illustration of the construction for the instance graph of Figure 9).

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

Assume there is a multicolored clique $\mathcal{C} = \{v_{a_1}^1, \ldots, v_{a_k}^k\}$ in $G$. We show that $\mathcal{T} = \{S(v_{a_i}^i v_{a_j}^j, v_{a_i}^i) \mid i < j \in [k]\} \cup \{S(v_{a_i}^i v_{a_j}^j, v_{a_j}^j) \mid i < j \in [k]\} \cup \{S(i, a_i, \vdash) \mid i \in [k]\} \cup \{S(i, a_i, \dashv) \mid i \in [k]\} \cup \{S(v_{a_i}^i v_{a_j}^j, \vdash) \mid i < j \in [k]\} \cup \{S(v_{a_i}^i v_{a_j}^j, \dashv) \mid i < j \in [k]\}$ is a set cover of $(\mathcal{S}, X)$ of size $2k^2$. As $\mathcal{C}$ is a clique, $\mathcal{T}$ is well defined and it contains $2\binom{k}{2} + 2k + 2\binom{k}{2} = 2k^2$ sets. For each $i \in [k]$, the elements $x(i,1,a_i), \ldots, x(i,1,t), \ldots, x(i,k+1,1), \ldots, x(i,k+1,a_i-1)$ are covered by the sets $S(v_{a_1}^1 v_{a_i}^i, v_{a_i}^i), S(v_{a_2}^2 v_{a_i}^i, v_{a_i}^i), \ldots, S(v_{a_i}^i v_{a_k}^k, v_{a_i}^i)$. Indeed, $S(v_{a_j}^j v_{a_i}^i, v_{a_i}^i)$ (or $S(v_{a_i}^i v_{a_j}^j, v_{a_i}^i)$ if $j > i$) covers all the elements $x(i,j,a_i), \ldots, x(i,j,t), x(i,j+1,1), \ldots, x(i,j+1,a_i-1)$ (again, in case $i+1=j$, replace *j+1* by *i+1*). For each $i \in [k]$, the elements $x_b(i), x(i,1,1), \ldots, x(i,1,a_i-1), x(i,k+1,a_i), \ldots, x(i,k+1,t), x_e(i)$ and of $W(i)$ are covered by $S(i,a_i,\vdash)$ and $S(i,a_i,\dashv)$. For all $i < j \in [k]$, say $v_{a_i}^i v_{a_j}^j$ is the $c$-th edge $e_c^{ij}$ in the arbitrary order of $E_{ij}$. Then, the elements $y(i,j,c), y(i,j,c+1), \ldots, y(i,j,c+2|E_{ij}|-1)$ are covered by $S(v_{a_i}^i v_{a_j}^j, v_{a_i}^i)$ and $S(v_{a_i}^i v_{a_j}^j, v_{a_j}^j)$. Finally, the elements $y_b(i,j), y(i,j,1), \ldots, y(i,j,c-1), y(i,j,c+2|E_{ij}|), \ldots, y(i,j,3|E_{ij}|), y_e(i,j)$ and of $Z(i,j)$ are covered by $S(v_{a_i}^i v_{a_j}^j, \vdash)$ and $S(v_{a_i}^i v_{a_j}^j, \dashv)$.

Assume now that the set-system $(X, \mathcal{S})$ admits a set cover $\mathcal{T}$ of size $2k^2$. As mentioned above, this solution $\mathcal{T}$ should contain exactly one set in each partite set (of the partition of $\mathcal{S}$). For each $i \in [k]$, to cover all the elements of $W(i)$, one should take $S(i,a_i,\vdash)$ and

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

| | $x_b(1)$ | $x(1,2,1)$ | $x(1,2,2)$ | $x(1,3,1)$ | $x(1,3,2)$ | $x(1,4,1)$ | $x(1,4,2)$ | $x_e(1)$ | $x_b(2)$ | $x(2,1,1)$ | $x(2,1,2)$ | $x(2,3,1)$ | $x(2,3,2)$ | $x(2,4,1)$ | $x(2,4,2)$ | $x_e(2)$ | … | $z_b(1,2)$ | $z(1,2,1)$ | $z(1,2,2)$ | $z_e(1,2)$ | … | $y_b(1,2)$ | $y(1,2,1)$ | $y(1,2,2)$ | $y(1,2,3)$ | $y(1,2,4)$ | $y(1,2,5)$ | $y(1,2,6)$ | $y_e(1,2)$ | … | $w_b(1)$ | $w(1,1)$ | $w(1,2)$ | $w_e(1)$ | $w_b(2)$ | $w(2,1)$ | $w(2,2)$ | $w_e(2)$ | … |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $S(1,1,\vdash)$ | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 1 | 1 | 1 | | | | | | |
| $\mathbf{S(1,2,\vdash)}$ | **1** | **1** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | **1** | **1** | | | | | | | |
| $S(v_1^1v_2^2,v_1^1)$ | | 1 | 1 | | | | | | | | | | | | | | | | | | | | | 1 | 1 | | | | | | | | | | | | | | | |
| $\mathbf{S(v_2^1v_2^2,v_2^1)}$ | | | **1** | **1** | | | | | | | | | | | | | | | | | | | | | **1** | **1** | | | | | | | | | | | | | | |
| $S(v_1^1v_2^3,v_1^1)$ | | | | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| $\mathbf{S(v_2^1v_1^3,v_2^1)}$ | | | | | **1** | **1** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| $S(1,1,\dashv)$ | | | | | | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | 1 | | | | | |
| $\mathbf{S(1,2,\dashv)}$ | | | | | | | **1** | **1** | | | | | | | | | | | | | | | | | | | | | | | | | | **1** | **1** | | | | | |
| $S(2,1,\vdash)$ | | | | | | | | | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | 1 | 1 | 1 | | |
| $\mathbf{S(2,2,\vdash)}$ | | | | | | | | | **1** | **1** | | | | | | | | | | | | | | | | | | | | | | | | | | **1** | **1** | | | |
| $S(v_2^2v_1^1,v_2^2)$ | | | | | | | | | | | 1 | 1 | | | | | | | | | | | | | | 1 | 1 | | | | | | | | | | | | | |
| $\mathbf{S(v_2^2v_2^1,v_2^2)}$ | | | | | | | | | | | **1** | **1** | | | | | | | | | | | | | | | **1** | **1** | | | | | | | | | | | | |
| $S(v_1^2v_1^3,v_1^2)$ | | | | | | | | | | | | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| $\mathbf{S(v_2^2v_1^3,v_2^2)}$ | | | | | | | | | | | | | **1** | **1** | | | | | | | | | | | | | | | | | | | | | | | | | | |
| $S(2,1,\dashv)$ | | | | | | | | | | | | | | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | 1 | |
| $\mathbf{S(2,2,\dashv)}$ | | | | | | | | | | | | | | | **1** | **1** | | | | | | | | | | | | | | | | | | | | | | **1** | **1** | |
| $\mathbf{S(v_2^1v_2^2,\vdash)}$ | | | | | | | | | | | | | | | | | | **1** | | | | | **1** | **1** | | | | | | | | | | | | | | | | |
| $S(v_1^1v_2^2,\vdash)$ | | | | | | | | | | | | | | | | | | 1 | 1 | | | | 1 | | | | | | | | | | | | | | | | | |
| $\mathbf{S(v_2^1v_2^2,\dashv)}$ | | | | | | | | | | | | | | | | | | | **1** | **1** | **1** | | | | | | | | **1** | **1** | | | | | | | | | | |
| $S(v_1^1v_2^2,\dashv)$ | | | | | | | | | | | | | | | | | | | | 1 | 1 | | | | | | | 1 | 1 | 1 | | | | | | | | | | |

**Figure 10** The sets of $\mathcal{S}_b(1)$, $\mathcal{S}_b(2)$, $\mathcal{S}_e(1)$, $\mathcal{S}_e(2)$, $\mathcal{S}(1,2,\vdash)$, $\mathcal{S}(1,2,\dashv)$, $\mathcal{S}(1,2)$, $\mathcal{S}(2,1)$ for the graph of Figure 9. The sets of $\mathcal{S}(1,3)$ and $\mathcal{S}(2,3)$ are also represented but only their part in $A$.

$S(i,a'_i,\dashv)$ with $a_i \leqslant a'_i$. Now, each set of $\mathcal{S}(i,j)$ has their $A$-intervals containing exactly $t$ elements. This means that the only way of covering the $tk+2$ elements of $X(i)$ is to take $S(i,a_i,\vdash)$ and $S(i,a'_i,\dashv)$ with $a_i \geqslant a'_i$ (therefore $a_i = a'_i$), and to take all the $k-1$ sets of $\mathcal{S}(i,j)$ (for $j \in [k] \setminus \{i\}$) of the form $S(v^i_{a_i}v^j_{s_j}, v^i_{a_i})$, for some $s_j \in [t]$. So far, we showed that a potential solution of $k$-Set Cover should stick to the same vertex $v^i_{a_i}$ in each *color class*. We now show that if one selects $S(v^i_{a_i}v^j_{s_j}, v^i_{a_i})$, one should be consistent with this choice and also selects $S(v^i_{a_i}v^j_{s_j}, v^j_{s_j})$. In particular, it implies that, for each $i \in [k]$, $s_i$ should be equal to $a_i$. For each $i \neq j \in [k]$, to cover all the elements of $Z(i,j)$, one should take $S(e^{ij}_{c_{ij}},\vdash)$ and $S(e^{ij}_{c'_{ij}},\dashv)$ with $c_{ij} \geqslant c'_{ij}$. Now, each set of $\mathcal{S}(i,j)$ and each set of $\mathcal{S}(j,i)$ has their $B$-intervals containing exactly $|E_{ij}|$ elements. This means that the only way of covering the $3|E_{ij}|+2$ elements of $Y(i,j)$ is to take $S(e^{ij}_{c_{ij}},\vdash)$ and $S(e^{ij}_{c'_{ij}},\dashv)$ with $c_{ij} \leqslant c'_{ij}$ (therefore, $c_{ij} = c'_{ij}$), and to take the sets $S(v^i_{a_i}v^j_{a_j}, v^i_{a_i})$ and $S(v^i_{a_i}v^j_{a_j}, v^j_{a_j})$. Therefore, if there is a solution to the $k$-Set Cover instance, then there is a multicolored clique $\{v^1_{a_1},\dots,v^k_{a_k}\}$ in $G$.

In this reduction, there is a quadratic blow-up of the parameter. Under the ETH, it would only forbid, by Theorem 3, an algorithm solving $k$-Set Cover on two-block instances in time $(N+M)^{o(\sqrt{k})}$. We can do the previous reduction from Multicolored Subgraph Isomorphism and suppress $X(i,j)$, $X(j,i)$, $Z(i,j)$, and $Y(i,j)$, and the sets defined over these elements, whenever $E_{ij}$ is empty. One can check that the produced set cover instance is still two-block and that the way of proving correctness does not change. Therefore, by Theorem 4, for any computable function $f$, $k$-Set Cover restricted to two-block instances cannot be solved in time $f(k)\,(N+M)^{o(k/\log k)}$ unless the ETH fails. ◀

For the size $|\mathcal{I}|$ of an instance $\mathcal{I}$ of Structured 2-Track Hitting Set, one can take $kt + |\mathcal{S}_A| + |\mathcal{S}_B|$.

▶ **Theorem 5.** Structured 2-Track Hitting Set *is* $W[1]$*-hard, and not solvable in time* $f(k)\,|\mathcal{I}|^{o(k/\log k)}$ *for any computable function* $f$*, unless the ETH fails.*

**Proof.** This result is a consequence of Lemma 9. Let $(A \uplus B, \mathcal{S})$ be a two-block instance of $k$-Set Cover. We recall that each set $S$ of $\mathcal{S}$ is the union of an $A$-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 each element $u$ of the $k$-Set Cover instance into a set $T_u$ of the Structured 2-Track Hitting Set instance. We put element $x_{S,A}$ (resp. $x_{S,B}$) into set $T_u$ whenever $u \in S \cap A = I_A$ (resp. $u \in S \cap B = I_B$). 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 order of $A'$ and $B'$ so that the instance is *structured*. Keep in mind that elements in the Structured 2-Track Hitting Set instance corresponds to sets in the $k$-Set Cover instance. We order the elements of $A'$ accordingly to the following ordering of the sets of the $k$-Set Cover instance: $\mathcal{S}_b(1)$, $\mathcal{S}(1,2)$, $\ldots$, $\mathcal{S}(1,k)$, $\mathcal{S}_e(1)$, $\mathcal{S}_b(2)$, $\mathcal{S}(2,1)$, $\ldots$, $\mathcal{S}(2,k)$, $\mathcal{S}_e(2)$, $\ldots$, $\mathcal{S}_b(k)$, $\mathcal{S}(k,1)$, $\ldots$, $\mathcal{S}(k,k-1)$, $\mathcal{S}_e(k)$, $\mathcal{S}(1,2,\vdash)$, $\mathcal{S}(1,2,\dashv)$, $\mathcal{S}(1,3,\vdash)$, $\mathcal{S}(1,3,\dashv)$, $\ldots$, $\mathcal{S}(k-1,k,\vdash)$, $\mathcal{S}(k-1,k,\dashv)$. We order the elements of $B'$ accordingly to the following ordering of the sets of the $k$-Set Cover instance: $\mathcal{S}(1,2,\vdash)$, $\mathcal{S}(1,2)$, $\mathcal{S}(2,1)$, $\mathcal{S}(1,2,\dashv)$, $\mathcal{S}(1,3,\vdash)$, $\mathcal{S}(1,3)$, $\mathcal{S}(3,1)$, $\mathcal{S}(1,3,\dashv)$, $\ldots$, $\mathcal{S}(k-1,k,\vdash)$, $\mathcal{S}(k-1,k)$, $\mathcal{S}(k,k-1)$, $\mathcal{S}(k-1,k,\dashv)$, $\mathcal{S}_b(1)$, $\mathcal{S}_e(1)$, $\ldots$, $\mathcal{S}_b(k)$, $\mathcal{S}_e(k)$. Within all those sets of sets, we order by increasing left endpoint (and then, in case of a tie, by increasing right endpoint). One can now check that with those two orders $\leq_{A'}$ and $\leq_{B'}$, all the sets $T_u$'s are $A'$-interval or $B'$-interval. Also, one can check that the 2-Track Hitting Set instance is *structured* by taking as color classes the partite 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 class to hit all the sets $T_u$ corresponds to taking one set in each partite set of $\mathcal{S}$ to dominate all the elements of the $k$-Set Cover instance. ◀

**Domination problems on 2-track (unit) interval graphs.** 2-track (unit) interval graphs are the intersection graphs of (unit) 2-track intervals, where a (unit) 2-track interval is the union of two (unit) intervals in two disjoint copies of the real line, called the first track and the second track. Two 2-track intervals intersect if they intersect in either the first or the second track. In this context, we can refer to a 2-track interval as an *object* and we say that an object *dominates* another object if they intersect. We recall that the *intersection graph* has the objects as vertices and admits an edge between two vertices iff they represent intersecting objects. Here, we also say that an object (an interval) *dominates* an interval (an object) if they intersect. We observe that many dominating problems with some geometric flavor can be restated with the terminology of 2-track (unit) interval graphs.

In particular, a result very close to Theorem 5 was obtained recently:

▶ **Theorem 10** ([24]). *Given the representation of a 2-track unit interval graph, the problem of selecting $k$ objects to dominate all the* intervals *is* $W[1]$*-hard, and not solvable in time* $n^{o(k/\log k)}$*, unless the ETH fails.*

We still had to give an *alternative* proof of this result because we will need the additional property that the instance can be further assumed to have the structure depicted in Figure 3. This will be crucial for showing the hardness result for Vertex Guard Art Gallery.

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

▶ **Theorem 11** ([17]). *Given the representation of a 2-track unit interval graph, the problem of selecting $k$ objects to dominate all the objects is* $W[1]$*-hard.*

▶ **Theorem 12** ([6]). *Given the representation of a 2-track unit interval graph, the problem of selecting $k$* intervals *to dominate all the objects is* $W[1]$*-hard.*

The result of Dom et al. is formalized differently in their paper [6], where the problem is defined as stabbing axis-parallel rectangles with axis-parallel lines.

## B Parameterized hardness of the vertex guard variant

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

▶ **Theorem 2** (Parameterized hardness vertex guard). *Assuming the ETH,* Vertex Guard Art Gallery *is not solvable in time $f(k)\, n^{o(k/\log k)}$, for any computable function $f$, even on simple polygons, where $n$ is the number of vertices of the polygon and $k$ is the number of guards allowed.*

**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 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 $3k$ vertices.

**Linker gadget.** For each $j \in [k]$, permutation $\sigma_j$ will be encoded by a sub-polygon $\mathcal{P}_j$ that we call *vertex linker*, or simply *linker* (see Figure 11). We regularly set $t$ consecutive vertices $\alpha^j_1, \alpha^j_2, \dots, \alpha^j_t$ in this order, along the $x$-axis. Opposite to this *segment*, we place $t$ vertices $\beta^j_{\sigma_j(1)}, \beta^j_{\sigma_j(2)}, \dots, \beta^j_{\sigma_j(t)}$ in this order, along the $x$-axis, too. The $\beta^j_{\sigma_j(1)}, \dots, \beta^j_{\sigma_j(t)}$, contrary to $\alpha^j_1, \dots, \alpha^j_t$, are *not* consecutive; we will later add some reflex vertices between them. At mid-distance between $\alpha^j_1$ and $\beta^j_{\sigma_j(1)}$, to the left, we put a reflex vertex $r^j_\downarrow$. *Behind* this reflex vertex, we place a vertical *wall* $d^j e^j$ ($r^j_\downarrow$, $d^j$, and $e^j$ are three consecutive vertices of $\mathcal{P}$), so that $\text{ray}(\alpha^j_1, r^j_\downarrow)$ and $\text{ray}(\alpha^j_t, r^j_\downarrow)$ both intersect $\text{seg}(d^j, e^j)$. That implies that for each $i \in [t]$, $\text{ray}(\alpha^j_i, r^j_\downarrow)$ intersects $\text{seg}(d^j, e^j)$. We denote by $p^j_i$ this intersection. The greater $i$, the closer $p^j_i$ is to $d^j$. Similarly, at mid-distance between $\alpha^j_t$ and $\beta^j_{\sigma_j(t)}$, to the right, we put a reflex vertex $r^j_\uparrow$ and place a vertical wall $x^j y^j$ ($r^j_\uparrow$, $x^j$, and $y^j$ are consecutive), so that $\text{ray}(\alpha^j_1, r^j_\uparrow)$ and $\text{ray}(\alpha^j_t, r^j_\uparrow)$ both intersect $\text{seg}(x^j, y^j)$. For each $i \in [t]$, we denote by $q^j_i$ the intersection between $\text{ray}(\alpha^j_i, r^j_\uparrow)$ and $\text{seg}(x^j, y^j)$. The smaller $i$, the closer $q^j_i$ is to $x^j$.

For each $i \in [t]$, we put around $\beta^j_i$ two reflex vertices, one in $\text{ray}(\beta^j_i, p^j_i)$ and one in $\text{ray}(\beta^j_i, q^j_i)$. In Figure 11, we merged some reflex vertices but the essential part is that $V(\beta^j_i) \cap \text{seg}(d^j, e^j) = \text{seg}(d^j, p^j_i)$ and $V(\beta^j_i) \cap \text{seg}(x^j, y^j) = \text{seg}(x^j, q^j_i)$. Finally, we add a triangular pocket rooted at $g^j$ and supported by $\text{ray}(g^j, \alpha^j_1)$ and $\text{ray}(g^j, \alpha^j_t)$, as well as a triangular pocket rooted at $b^j$ and supported by $\text{ray}(g^j, \beta^j_{\sigma_j(1)})$ and $\text{ray}(g^j, \beta^j_{\sigma_j(t)})$. This ends the description of the vertex linker (see Figure 11).

The following lemma formalizes how exactly the vertices $\alpha^j_i$ and $\beta^j_i$ are linked: say, one chooses to put a guard on a vertex $\alpha^j_i$, then the only way to see entirely $\mathcal{P}_j$ by putting a second guard on a vertex of $\{\beta^j_1, \dots, \beta^j_t\}$ is to place it on the vertex $\beta^j_i$.

▶ **Lemma 13.** *For any $j \in [k]$, the sub-polygon $\mathcal{P}_j$ is seen entirely by $\{\alpha^j_v, \beta^j_w\}$ iff $v = w$.*

**Proof.** The regions of $\mathcal{P}_j$ not seen by $\alpha^j_v$ (i.e., $\mathcal{P}_j \setminus V(\alpha^j_v)$) consist of the triangles $d^j r^j_\downarrow p^j_v$, $x^j r^j_\uparrow q^j_v$ and partially the triangle $a^j b^j c^j$. The triangle $a^j b^j c^j$ is anyway entirely seen by the

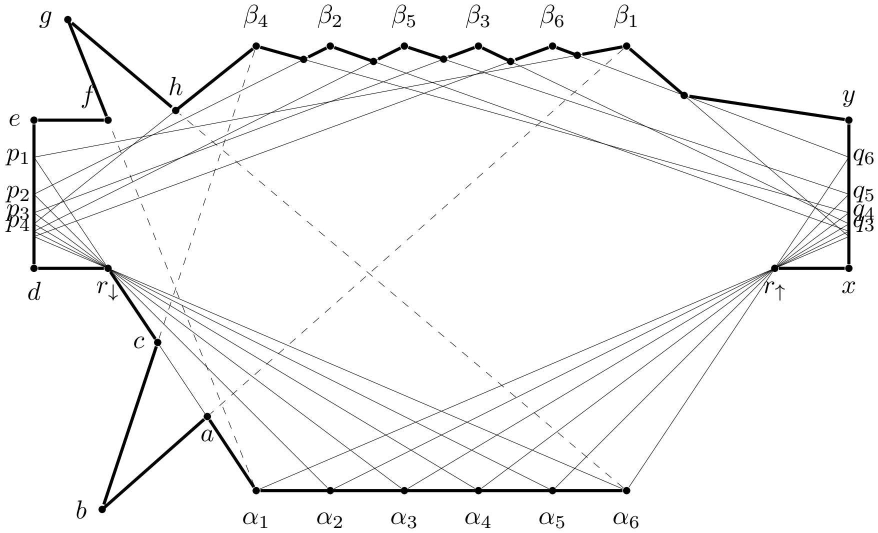


**Figure 11** Vertex linker gadget. We omitted the superscript $j$ in all the labels. Here, $\sigma_j(1) = 4,\ \sigma_j(2) = 2,\ \sigma_j(3) = 5,\ \sigma_j(4) = 3,\ \sigma_j(5) = 6,\ \sigma_j(6) = 1$.

vertex $\beta^j_i$, for any $i \in [t]$. It remains to prove that $d^j r^j_\downarrow p^j_v \cup x^j r^j_\uparrow q^j_v \subseteq V(\beta^j_w)$ iff $v = w$.

It holds that $d^j r^j_\downarrow p^j_v \cup x^j r^j_\uparrow q^j_v \subseteq V(\beta^j_v)$ since, by construction, the two reflex vertices neighboring $\beta^j_v$ are such that $\beta^j_v$ sees $\mathrm{seg}(d^j, p^j_\alpha)$ (hence, the whole triangle $d^j r^j_\downarrow p^j_v$) and $\mathrm{seg}(x^j, q^j_\alpha)$ (hence, the whole triangle $x^j r^j_\uparrow q^j_v$). Now, let us assume that $v \neq w$. If $v < w$, the interior of the segment $\mathrm{seg}(p_v, p_w)$ is not seen by $\{\alpha^j_v, \beta^j_w\}$, and if $v > w$, the interior of the segment $\mathrm{seg}(q_v, q_w)$ is not seen by $\{\alpha^j_v, \beta^j_w\}$. ◀

The issue we now have is that one could decide to place a guard on a vertex $\alpha^j_i$ and a second guard on a reflex vertex between $\beta^j_{\sigma_j(w)}$ and $\beta^j_{\sigma_j(w+1)}$ (for some $w \in [t-1]$). This is indeed another 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 gadget* (see Figure 12) satisfying the property that all its (triangular) pockets can be guarded by adding only one guard on a vertex of $\mathcal{F}_j$ iff there is already a guard on a vertex $\beta^j_i$ of $\mathcal{P}_j$. Therefore, the filter gadget will prevent one from placing a guard on a reflex vertex of $\mathcal{P}_j$. The functioning of the gadget is again based on Lemma 7.

**Filter gadget**. Let $d^j_1, \ldots, d^j_t$ be $t$ consecutive vertices of a regular, say, $20t$-gon, so that the angle made by $\mathrm{ray}(d^j_1, d^j_2)$ and the $x$-axis is a bit below 45°, while the angle made by $\mathrm{ray}(d^j_{t-1}, d^j_t)$ and the $x$-axis is a bit above 45°. The vertices $d^j_1, \ldots, d^j_t$ can therefore be seen as the discretization of an arc $\mathcal{C}$. We now mentally draw two lines $\ell_h$ and $\ell_v$; $\ell_h$ is a horizontal line a bit below $d^j_1$, while $\ell_v$ is a vertical line a bit to the right of $d^j_t$. We put, for each $i \in [t]$, a vertex $x^j_i$ at the intersection of $\ell_h$ and the tangent to $\mathcal{C}$ passing through $d^j_i$. Then, for each $i \in [t-1]$, we set a triangular pocket $\mathcal{P}(x^j_i)$ rooted at $x^j_i$ and supported by $\mathrm{ray}(x^j_i, d^j_1)$ and $\mathrm{ray}(x^j_i, \beta^j_{\sigma_j(i+1)})$. For convenience, each point $\beta^j_{\sigma_j(i)}$ is denoted by $c^j_i$ on Figure 12. We also set a triangular pocket $\mathcal{P}(x^j_t)$ rooted at $x^j_t$ and supported by $\mathrm{ray}(x^j_t, d^j_1)$ and $\mathrm{ray}(x^j_t, d^j_t)$. Similarly, we place, for each $i \in [t-1]$, a vertex $y^j_i$ at the intersection of $\ell_v$ and the tangent to $\mathcal{C}$ passing through $d^j_{i+1}$. Finally, we set a triangular pocket $\mathcal{P}(y^j_i)$ rooted at $y^j_i$ and supported by $\mathrm{ray}(y^j_i, \beta^j_{\sigma_j(i)})$ and $\mathrm{ray}(y^j_i, d^j_t)$, for each $i \in [t-1]$ (see Figure 12). We denote by $\mathcal{P}(\mathcal{F}_j)$ the $2t-1$ triangular pockets of $\mathcal{F}_j$.

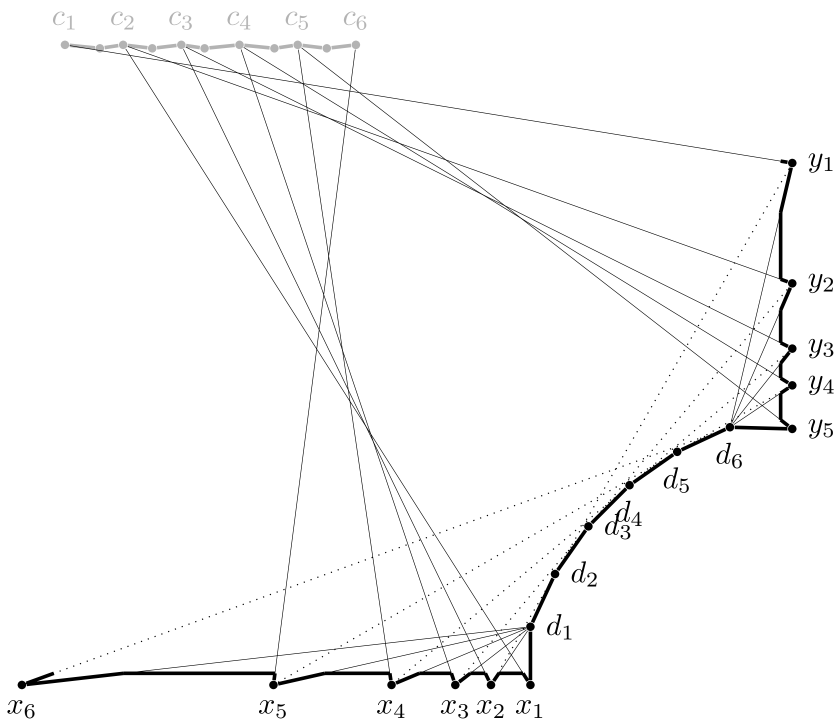


**Figure 12** The filter gadget $\mathcal{F}_j$. Again, we omit the superscript $j$ on the labels. Vertices $c_1, c_2, \ldots, c_t$ are not part of $\mathcal{F}_j$ and are in fact the vertices $\beta^j_{\sigma_j(1)}, \beta^j_{\sigma_j(2)}, \ldots, \beta^j_{\sigma_j(t)}$ and the vertices in between the $c_i$'s are the reflex vertices that we have to *filter out*.

▶ **Lemma 14.** *For each $j \in [k]$, the only ways to see entirely $\mathcal{P}(\mathcal{F}_j)$ and the triangle $a^j b^j c^j$ with only two guards on vertices of $\mathcal{P}_j \cup \mathcal{F}_j$ is to place them on vertices $c_i^j$ and $d_i^j$ (for any $i \in [t]$).*

**Proof.** Proving this lemma will, in particular, entail that it is not possible to see entirely $\mathcal{P}(\mathcal{F}_j)$ with only two vertices if one of them is a reflex vertex between $c_i^j$ and $c_{i+1}^j$. Let us call such a vertex an intermediate reflex vertex (in color class $j$). Because of the pocket $a^j b^j c^j$, one should put a guard on a $c_i^j$ (for some $i \in [t]$) or on an intermediate reflex vertex in class $j$. As 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 vertices cannot work as a consequence of what follows.

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

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

**Putting the pieces together.** The permutation $\sigma$ is encoded the following way. We position the vertex linkers $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_k$ such that $\mathcal{P}_{i+1}$ is below and slightly to the left of $\mathcal{P}_i$. Far below 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 connected to the leftmost vertex of $\mathcal{F}_{\sigma(i+1)}$, for all $i \in [t-1]$. We add a constant number of vertices in the vicinity of each $\mathcal{P}_j$, so that the only filter gadget that vertices $\beta_1^j, \ldots, \beta_t^j$ can see is $\mathcal{F}_{\sigma(j)}$ (see Figure 13). Similarly to the point guard version, we place vertically and far from the $\alpha_i^j$'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$(z_{A,q}, \alpha_{i'}^{j'})$, for each $A$-interval $I_q = [a_i^j, a_{i'}^{j'}] \in \mathcal{S}_A$ (Track $A$). Finally, we place vertically and far from the $d_i^j$'s, one triangular pocket $\mathcal{P}(z_{B,q})$ rooted at vertex $z_{B,q}$ and supported by ray$(z_{B,q}, d_i^j)$ and ray$(z_{B,q}, d_{i'}^{j'})$, for each $B$-interval $I_q = [b_{\sigma_j(i)}^j, b_{\sigma_{j'}(i')}^{j'}] \in \mathcal{S}_B$ (Track $B$). This ends the construction (see Figure 13).

**Soundness.** We now prove the correctness of the reduction. Assume that $\mathcal{I}$ is a YES-instance and let $\{(a_{s_1}^1, b_{s_1}^1), \ldots, (a_{s_k}^k, b_{s_k}^k)\}$ be a solution. We claim that the set of vertices $G = \{\alpha_{s_1}^1, \beta_{s_1}^1, d_{\sigma_1^{-1}(s_1)}^1, \ldots, \alpha_{s_k}^k, \beta_{s_k}^k, d_{\sigma_k^{-1}(s_k)}^k\}$ guards the whole polygon $\mathcal{P}$. Let $z^j := d_{\sigma_j^{-1}(s_j)}^j$ for notational convenience. By Lemma 13, for each $j \in [k]$, the sub-polygon $\mathcal{P}_j$ is entirely seen, since there are guards on $\alpha_{s_j}^j$ and $\beta_{s_j}^j$. By Lemma 14, for each $j \in [k]$, all the pockets of $\mathcal{F}_j$ are entirely seen, since there are guards 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 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 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 $z^{\sigma(k)}$.

Assume now that there is no solution to the instance $\mathcal{I}$ of Structured 2-Track Hitting Set, and, for the sake of contradiction, that there is a set $G$ of $3k$ vertices guarding $\mathcal{P}$. For each $j \in [k]$, vertices $b^j$, $g^j$, and $x_t^j$ are seen by three disjoint set of vertices. The first two sets are contained in the vertices of sub-polygon $\mathcal{P}_j$ and the third one is contained in the vertices of $\mathcal{F}_j$. Therefore, to see entirely $\mathcal{P}_j \cup \mathcal{P}(\mathcal{F}_j)$, three vertices are necessary. Summing that over the $k$ color classes, this corresponds already to $3k$ vertices which is the size of the supposed set $G$. Thus, there should *exactly* 3 guards placed among the vertices of $\mathcal{P}_j \cup \mathcal{F}_j$.

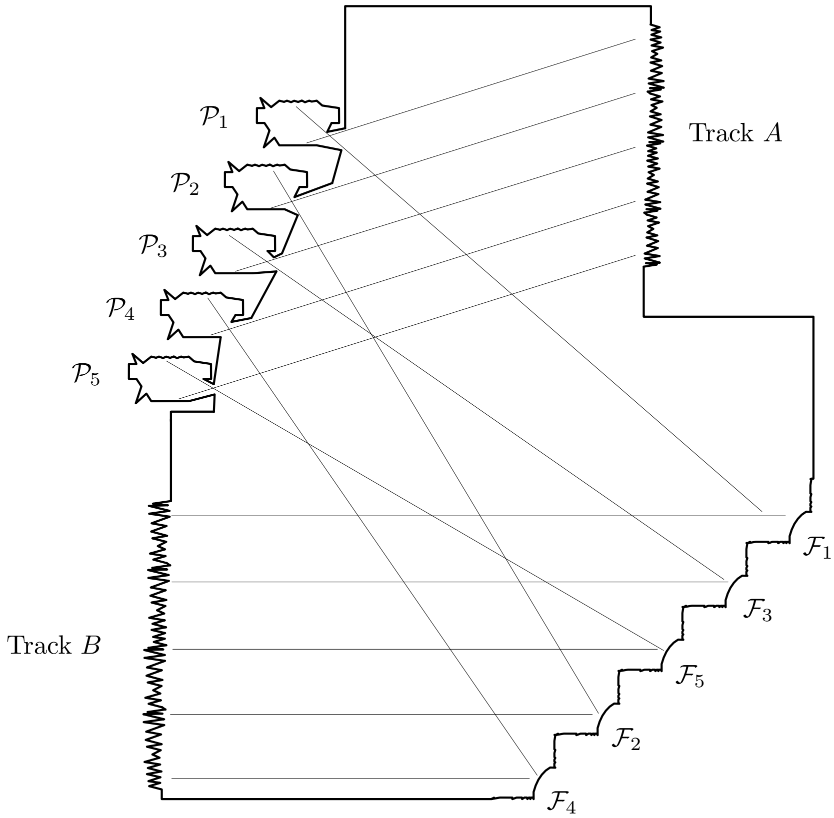


**Figure 13** Overall picture of the reduction with $k = 5$.

Therefore, by Lemma 14, there should be 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 $G$. Then, by Lemma 13, a guard should be placed at vertex $\alpha^j_{\sigma_j(s_j)}$. Indeed, the only vertices seeing $g^j$ are $f^j, g^j, h^j$ and $a^j_1, \ldots, a^j_t$; but, if the third guard is placed at vertex $f^j$, $g^j$, or $h^j$, then vertices $\beta^j_w$ (with $w \neq \sigma_j(i)$) are not seen. So far, we showed that $G$ should be of the form $\{\alpha^1_{\sigma_1(s_1)}, \beta^1_{\sigma_1(s_1)}, d^1_{s_1}, \ldots, \alpha^j_{\sigma_j(s_j)}, \beta^j_{\sigma_j(s_j)}, d^j_{s_j}, \ldots, \alpha^k_{\sigma_k(s_k)}, \beta^k_{\sigma_k(s_k)}, d^k_{s_k}, \}$. Though, as there is no solution to $\mathcal{I}$, there should be a set in $\mathcal{S}_A \cup \mathcal{S}_B$ that is not hit by $\{(a^1_{\sigma_1(s_1)}, b^1_{\sigma_1(s_1)}), \ldots, (a^k_{\sigma_k(s_k)}, b^k_{\sigma_k(s_k)})\}$. By construction, the pocket associated to this set is not entirely seen; a contradiction.

Let us bound the number of vertices of $\mathcal{P}$. Each sub-polygon $\mathcal{P}_j$ or $\mathcal{F}_j$ contains $O(t)$ vertices. *Track A* contains $3|\mathcal{S}_A|$ vertices and *Track B* contains $3|\mathcal{S}_B|$ vertices. Linking everything together requires $O(k)$ additional vertices. So, in total, there are $O(kt + |\mathcal{S}_A| + |\mathcal{S}_B|)$ vertices. Thus, this reduction together with Theorem 5 implies that VERTEX GUARD ART GALLERY is W[1]-hard and cannot be solved in time $f(k)\, n^{o(k/\log k)}$ for any computable function $f$, where $n$ is the number of vertices of the polygon and $k$ the number of guards, unless the ETH fails. ◀

## C Direct Proof of Structured 2-Track Hitting Set Parameterized Hardness

▶ **Theorem 5.** STRUCTURED 2-TRACK HITTING SET *is* $W[1]$*-hard, and not solvable in time* $f(k)\, |\mathcal{I}|^{o(k/\log k)}$ *for any computable function* $f$*, unless the ETH fails.*

We reduce from Multicolored Subgraph Isomorphism. Recall that, in the Multicolored Subgraph Isomorphism problem, one is given a host graph $G$ with $n$ vertices and a pattern graph $P$ with $l$ vertices and $k$ edges. Our reduction is linear in $k$. The vertices of $G$ are partitioned into $l$ color classes $V_1, \ldots, V_l$ 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 color class so that the selected vertices induce the pattern graph $P$. Note that each color class $V_i$ corresponds to a specific vertex $v_i$ in the pattern graph. The set $E_{ij}$ in the host graph is empty if and only if there is no edge between $v_i$ and $v_j$ in $P$.

The hardness reduction uses repeatedly certain gadgets in order to enforce consistent choices. We describe the functionality of these gadgets and show later how we combine them. Given $d$ color classes $C_1, \ldots, C_d$ each of size $t$, which are adjacent on track $A$. We say a choice over color class $C_1, \ldots, C_d$ on track $A$ is consistent if in each color class the $l$-th element on track $A$ was chosen for some $l \in [t]$. Note that we do not care (for the moment) about the elements on track $B$ of the color classes $C_1, \ldots, C_d$.

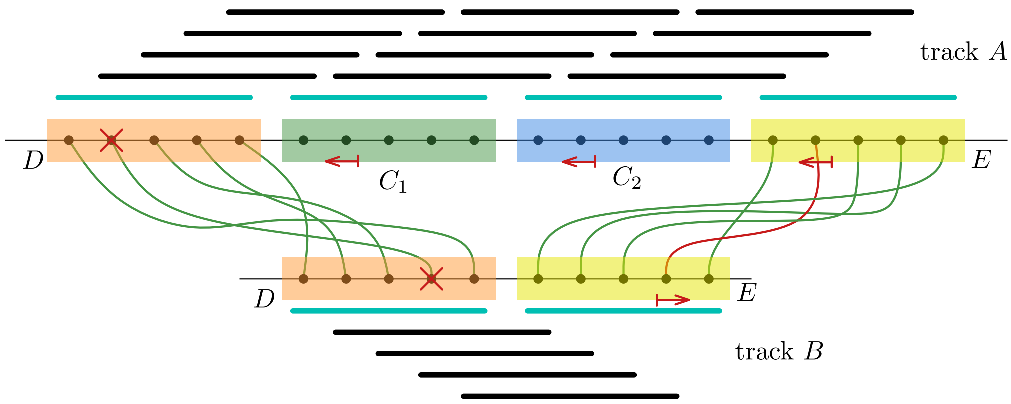


**Figure 14** A gadget to make choices consistent.

▶ **Lemma 15.** *Given $d$ color-classes $C_1, \ldots, C_d$ each of size $t$, which are adjacent on track $A$, we can build a gadget that enforces a consistent choice on track $A$.*

**Proof.** We use the same notations as above. We add two more color classes $D$ and $E$ surrounding $C_1, \ldots, C_d$ on track $A$ and being adjacent anywhere on track $B$ disjoint from all other color classes, as in Figure 14. To be precise, we give an explicit description, of the gadget in Figure 14. $D$ and $E$ have both $t$ 2-elements. $D$ is to the left of $C_1, \ldots, C_d$ on track $A$ and $E$ is to the right of $C_1, \ldots, C_d$ on track $A$. On track $B$, $D$ is to the left of $E$ and they are adjacent. The permutation between the elements of track $A$ and $B$ of $D$ and $E$ is the inversion (which, for each $i \in [t]$, maps the $i$-th element to the $t-i+1$-th element). We add all intervals of length $t$ that completely contained in the color classes $D, C_1, \ldots, C_d, E$ on track $A$, and we add all intervals of length $t$ that are completely contained in the color classes $D$ and $E$ on track $B$.

We claim that the only way to hit all those intervals with $d+2$ 2-elements is to make a consistent choice over all the color classes on track $A$. Let us sssume that we choose on track $A$ the $l_0$-th element of color class $D$, the $l_1$-th element of color class $C_1$, the $l_2$-th element of color class $C_2$ and so on, up to the $l_{d+1}$-th element of color class $E$. for all $i \in [0, d]$, $l_i \geqslant l_{i+1}$, since otherwise the interval of length $t$ starting at the $l_i+1$-th element of color class $C_i$ (with $C_0 = D$) and ending at the $l_{i+1}$-th element on color class $C_{i+1}$ (with $C_{d+1} = E$) would not be hit. In particular, it must hold that $l_0 \geqslant l_{d+1}$.

Now we switch to track $B$. Choosing the $l_0$-th element of $D$ on track $A$ implies that the $t - l_0 + 1$-th element of $D$ on track $B$ is chosen. Similary, choosing the $l_{d+1}$-th element of $E$ on track $A$ implies that the $t - l_{d+1} + 1$-th element of $E$ on track $B$ is chosen. To hit all the intervals on track $B$ we have to satisfy the condition $t - l_0 + 1 \geqslant t - l_{d+1} + 1$. Together with $l_0 \geqslant l_{d+1}$ we can conclude that $l_0 = l_{d+1}$, which implies $l_0 = l_i$, for all $i \in [d+1]$. ◀

We apply this lemma by representing the vertex set by consecutive elements on track $A$. We make several copies of these elements and place them consecutively on track $A$. Then, we can link each copy to a distinct element. With the help of Lemma 15, we ensure that all choices are consistent and correspond to the same vertex. The same procedure can be applied to the edge sets $E_{ij} = E(V_i, V_j)$.

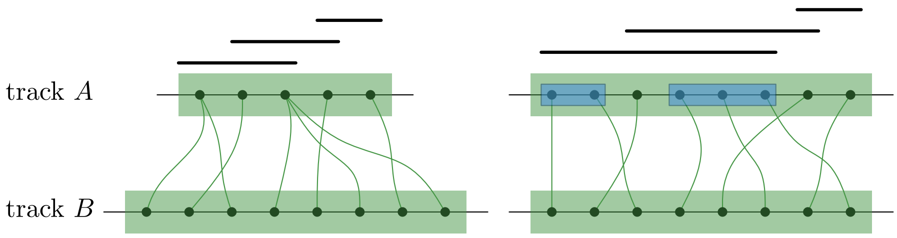


**Figure 15** Splitting an element into several elements; here, two elements got duplicated.

Let us consider a color class with a different number of elements on track $A$ and $B$. And, let us assume that we have given a surjective mapping $\sigma$ instead of a bijection. Then we can transform $\sigma$ easily into a bijection $\overline{\sigma}$ by duplicating each element by the size of its preimage under $\sigma$. Further, we say every new interval contains all duplicates of $x$, if and only if it contained $x$ before. It follows an easy observation.

▶ Observation 1. Let $(x, \sigma(x))$ be a pair of elements and $(x, \overline{\sigma}(x))$ be the corresponding pair after the transformation. Then, an interval is hit by $(x, \sigma(x))$ if and only if it is hit by $(x, \overline{\sigma}(x))$ after the transformation.

To summarize, we can simulate surjective mappings by this duplicating trick.

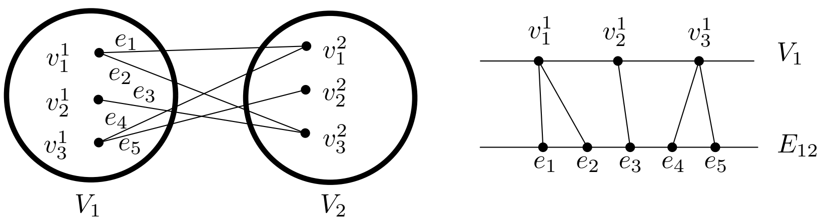


**Figure 16** The linking between a vertex and an edge set is displayed.

We use Observation 1 as follows. We define a surjective mapping $\sigma_i^{ij}$ from $E_{ij}$ to $V_i$, by $\sigma_i^{ij}(e) = v$ if and only if $e$ is incident to $v$ (a vertex $v$ cannot be in a solution if it is not incident to at least one edge in $E_{ij}$).

Assume that we have given $|E_{ij}|$ elements on track $B$ representing the edges of $E_{ij}$ and $|V_i|$ elements on track $A$ representing the vertices of $V_i$. Then, we simulate the surjective function $\sigma_i^{ij}$ by the duplicating trick as explained above. This enforces that it is only possible to pick vertex-edge $(v, e)$ pairs, where the vertex $v$ is incident to edge $e$.

We are know ready to describe the full reduction from Multicolored Subgraph Isomorphism. We denote by $d_i$ the degree of $v_i$ in the pattern graph $P$. For each color class

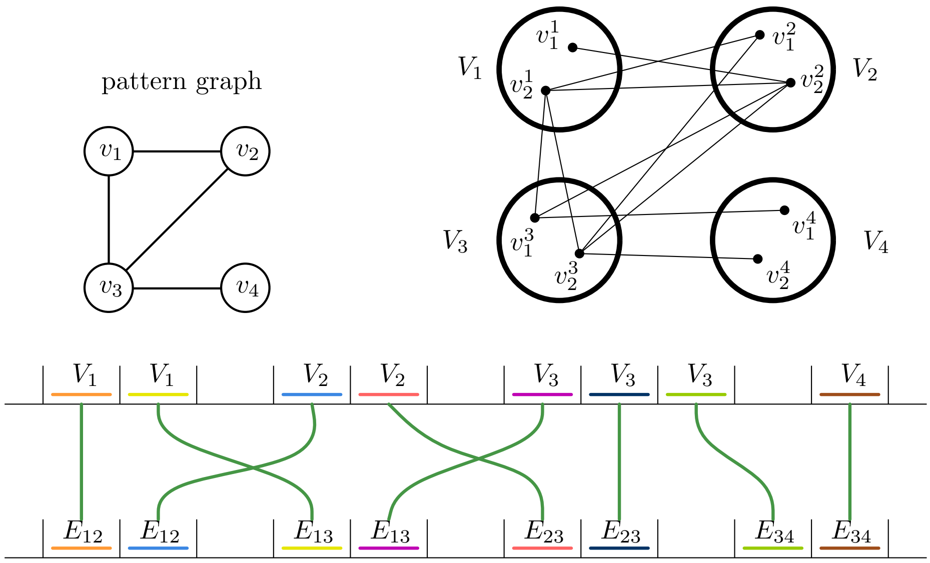


**Figure 17** At the top is an instance of subgraph isomorphism with a pattern and a host graph. Below is a schematic drawing of an equivalent instance of STRUCTURED 2-TRACK HITTING SET.

$V_i$ of size $n_i$, we place $n_i d_i$ consecutive elements on track $A$. They represent $d_i$ copies of $V_i$. For each edge set $E_{ij}$ of size $m_{ij}$, we place $2m_{ij}$ consecutive elements on track $B$. They represent two copies of the edge set $E_{ij}$. One copy of $E_{ij}$ will be linked to $V_i$ and the other to $V_j$.

With the help of Lemma 15, we make sure that each copy of elements representing vertices and edges is consistent. Now, we pair up one copy of a vertex set $V_i$ with one copy of the edge set $E_{ij}$ in an arbitrary fashion using $\sigma_i^{ij}$, see Figure 17. We have just created enough copies of each $V_i$. By Observation 1, the $\sigma_i^{ij}$ can be represented by duplicating elements.

The new parameter is linear in the number of edges of the pattern graph. In fact, the number of color classes of the STRUCTURED 2-TRACK HITTING SET instance is bounded by $6k + 2l \leqslant 8k + 2$. Recall that $k$ denotes the number of edges in the pattern graph and $l$ its number of vertices. Since we assume the pattern graph to be connected we know $l \leqslant k + 1$. To see correctness, assume that we are given a YES-instance of MULTICOLORED SUBGRAPH ISOMORPHISM. Thus, there are $l$ vertices, one in each color classes of $G$, which induce $k$ edges. We can pick the elements corresponding to these $k$ edges and $l$ vertices.

For the reverse direction, assume we can pick one 2-element in each color class of the STRUCTURED 2-TRACK HITTING SET problem and hit all the sets. Then, we can read off the corresponding vertices, because of the consistency gadget. Let $w_i$ be this choice in $V_i$, for each $i \in [k]$. We have to show that, for all $i \neq j \in [k]$, $w_i$ and $w_j$ are adjacent in the case that $E_{ij}$ is non-empty. In the linking between $V_i$ and $E_{ij}$, a 2-element corresponds to a vertex-edge pair $(w_i, e)$ with $w_i \in V_i$, $e \in E_{ij}$ and $e$ incident to $w_i$. The same holds for the linking between $V_j$ and $E_{ij}$: any 2-element corresponds to a vertex edge pair $(w_j, e')$ with $w_j \in V_j$, $e' \in E_{ij}$ and $e'$ incident to $w$. Due to the consistency gadget, it must hold $e = e'$. This implies $w_i \in V_i$ and $w_j \in V_j$ must be touched by the same edge $e \in E_{ij}$ and thus the two vertices are adjacent in the original graph $G$.