On the Parameterized Complexity of Red-Blue Points Separation[*]

Édouard Bonnet[†] *Panos Giannopoulos*[‡] *and Michael Lampis*[§]

ABSTRACT. We study the following geometric separation problem: Given a set $\mathcal{R}$ of red points and a set $\mathcal{B}$ of blue points in the plane, find a minimum-size set of lines that separate $\mathcal{R}$ from $\mathcal{B}$. We show that, in its full generality, parameterized by the number of lines k in the solution, the problem is unlikely to be solvable significantly faster than the brute-force $n^{O(k)}$-time algorithm, where n is the total number of points. Indeed, we show that an algorithm running in time $f(k)n^{o(k/\log k)}$, for any computable function f, would disprove ETH. Our reduction crucially relies on selecting lines from a set with a large number of different slopes (i.e., this number is not a function of k).

Conjecturing that the problem variant where the lines are required to be axis-parallel is FPT in the number of lines, we show the following preliminary result. Separating $\mathcal{R}$ from $\mathcal{B}$ with a minimum-size set of axis-parallel lines is FPT in the size of either set, and can be solved in time $O^*(9^{|\mathcal{B}|})$ (assuming that $\mathcal{B}$ is the smaller set).

1 Introduction

We study the parameterized complexity of the following RED-BLUE SEPARATION problem: Given a set $\mathcal{R}$ of red points and a set $\mathcal{B}$ of blue points in the plane and a positive integer k, find a set of at most k lines that together separate $\mathcal{R}$ from $\mathcal{B}$ (or report that such a set does not exist). Separation here means that each cell in the arrangement induced by the lines in the solution is either monochromatic, i.e., contains points of one color only, or empty. Equivalently, $\mathcal{R}$ is separated from $\mathcal{B}$ if every straight-line segment with one endpoint in $\mathcal{R}$ and the other one in $\mathcal{B}$ is intersected by at least one line in the solution. Note here that we opt for *strict* separation that is, no point in $\mathcal{R} \cup \mathcal{B}$ is on a separating line. Let $n := |\mathcal{R} \cup \mathcal{B}|$.

The variant where the separating lines sought must be axis-parallel will be simply referred to as AXIS-PARALLEL RED-BLUE SEPARATION.

Background and related work. Geometric separability problems have been studied extensively in computational geometry. Linear separability of two point sets in the plane, i.e., the case $k = 1$ for RED-BLUE SEPARATION is, perhaps, the simplest version and can

[†]*Univ Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668, France,* `edouard.bonnet@ens-lyon.fr`

[‡]*giCentre, Department of Computer Science, City University of London, EC1V 0HB, London, United Kingdom,* `panos.giannopoulos@city.ac.uk`

[§]*Université Paris-Dauphine, PSL Research University, CNRS, LAMSADE, Paris, France,* `michail.lampis@dauphine.fr`

be solved in $O(n)$ time by reduction to linear programming, as observed by Megiddo [26]; the same linear time bound holds for separating two point sets by a hyperplane in any fixed dimension. For the case where $k = 2$, RED-BLUE SEPARATION is solvable in $O(n \log n)$ time [16]. When k is part of the input, the problem was shown to be NP-hard first by Megiddo [27] and again much later (under the name minimum linear classification) with a similar reduction by Lu et al. [21]. It is also known to be APX-hard [5] even for the axis-parallel variant. By a simple reduction to set cover, RED-BLUE SEPARATION admits a $O(\log n)$-approximation [21], while for AXIS-PARALLEL RED-BLUE SEPARATION there is also a 2-approximation [5]. None of the hardness results above imply hardness in the parameterized setting and we are not aware of any previous results on the parameterized complexity of the problem.

In a more applied setting, AXIS-PARALLEL RED-BLUE SEPARATION is directly motivated by the problem of discretization of continuous variables, which is used in extraction of decision rules from decision tables in the area of data analysis and data mining, see, for example, [9, 33, 32, 19]. In its two-dimensional version, one is given a decision table with points (objects) of two real-valued coordinates (attributes) and a binary decision function. The function can be seen as assigning one of two colors to each point. Discretization boils down to partitioning the attribute values into intervals by finding a minimum-size set of axis-parallel lines (cuts on the axes) that is consistent with the decision table. Consistency is equivalent to asking for the corresponding lines to partition the plane into regions such that any two points in the same region have the same color. For heuristics that do not offer any approximation guarantees, see [33, 7, 19].

There is quite some work on other types of separators. For example, separability of two point sets in the plane by a circle can be decided in linear time [34] (by a reduction to linear separability in $\mathbb{R}^3$), while separability by a wedge or a strip can be decided in $O(n \log n)$ time [17] and this is optimal [1]. Additionally, van Kreveld et al. [37] showed how to compute all orientations for which a separating rectangle exists in $O(n \log n)$ time and Sheikhi et al. [35] gave a worst-case optimal $O(n^2)$-time algorithm that computes all orientations for which an L-shaped separator exists. Closer to the optimization problem we study, Edelsbrunner and Preparata [8] gave an $O(n \log n)$-time algorithm for computing a convex polygon with a minimum number of edges that separates two point sets (if it exists), while Fekete [10] showed the problem to be NP-hard for general simple polygons. For the latter, Mitchell [31] gave a polynomial-time $O(\log n)$-approximation algorithm.

The following closely related monochromatic points separation problem has also been studied: Given a set of points in the plane, find a smallest set of lines that separates every point from every other point in the set (i.e., each cell in the induced arrangement must contain at most one point). It has been shown to be NP-hard [11], APX-hard [5] and, in the axis-parallel case, to admit a 2-approximation [5]. Very recently, the problem has been also shown to admit an $\text{OPT} \log \text{OPT}$-approximation [14]. Note here that it is trivially FPT in the number of lines, as the number of cells in the arrangement of k lines is at most $\Theta(k^2)$.

Geometric separators have been studied also in a more combinatorial setting, where a separator for a set of geometric objects is a subset of small size that intersects few of the objects and whose removal disconnects the intersection graph of the original set into two

parts of roughly equal size. When such separators exist, they can be used in the design of divide and conquer algorithms for various fundamental geometric problems. There is a fair amount of work on geometric separators, a full account of which is outside the scope of this paper. Indicatively, we mention the early work on sphere separators in two and three dimensions by Miller and Thurston [30], their generalization in higher dimensions by Miller et al. [29], where an alternative proof of the planar separator theorem based on circle separators was also presented, and the work on hypercube separators by Smith and Wormald [36]. Algorithmically, their applications include nearest neighbor search [29], TSP and rectilinear Steiner tree in higher dimensions [36], mesh partitioning and compression [28, 2], geometric hitting set [6] and protein folding [12].

Finally, a different notion of separation comes in the form of ham-sandwich cuts. In the discrete, two-dimensional setting, given two point sets in the plane, a ham-sandwich cut is a line that simultaneously bisects both sets, that is, it has at most half of the points of each set on either side. The existence of such a cut is guaranteed by the well-known Ham-Sandwich theorem, see, for example, Matoušek [25], while it can be computed in optimal $O(n)$ time [20].

Results in this paper. We first show that RED-BLUE SEPARATION is W[1]-hard in the solution size k and that it cannot be solved in $f(k)n^{o(k/\log k)}$ time (for any computable function f) unless ETH fails. Our reduction is from STRUCTURED 2-TRACK HITTING SET, see Section 3, which has been recently used for showing hardness for another classical geometric optimization problem [4]. Then, in Section 4, we show that AXIS-PARALLEL RED-BLUE SEPARATION is FPT in the size of either of $\mathcal{R}$ and $\mathcal{B}$. Our algorithm is simple and is based on reducing the problem to $9^{|\mathcal{B}|+2}$ instances of 2-SAT (assuming, w.l.o.g., that $\mathcal{B}$ is the smaller set).

2   Preliminaries

For positive integers x, y, let $[x]$ be the set of integers between 1 and x, and $[x, y]$ the set of integers between x and y.

For a totally ordered (finite) set X, an X-*interval* is any subset of X of consecutive elements. 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. 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 decide if there is 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 $\mathcal{S}_B$. We will refer to the interval systems $(A, \mathcal{S}_A)$ and $(B, \mathcal{S}_B)$ as track A and track B.

STRUCTURED 2-TRACK HITTING SET (S2-THS for short) is the same problem with k color classes over the 2-elements and a restriction on the one-to-one mapping ϕ; see Figure 1 for an illustration. Given two integers k and t, A is partitioned into $(C_1, C_2, \ldots, C_k)$ where $C_j = \{a_1^j, a_2^j, \ldots, a_t^j\}$ for each $j \in [k]$. A is ordered: $a_1^1, a_2^1, \ldots, a_t^1, a_1^2, a_2^2, \ldots, a_t^2, \ldots, a_1^k, a_2^k,$ $\ldots, a_t^k$. We define $C'_j := \phi(C_j)$ and $b_i^j := \phi(a_i^j)$ for all $i \in [t]$ and $j \in [k]$. We now impose

that ϕ is such that, for each $j \in [k]$, the 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: $b^j_{\sigma_j(1)}, b^j_{\sigma_j(2)}, \ldots, b^j_{\sigma_j(t)}$. The goal is to find k 2-elements, one per color class, that together hit all the A- and the B-intervals.

In what follows, it will be convenient to see an instance of S2-THS 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 $\mathcal{S}_A$ 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. $b^j_i \leq_B b \leq_B b^{j'}_{i'}$).
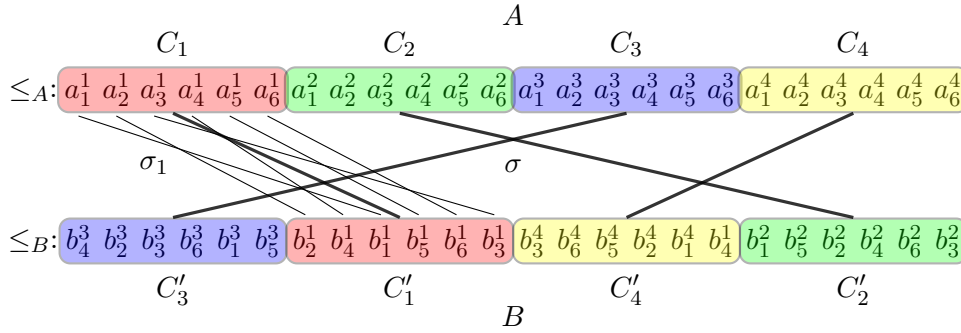

Figure 1: An illustration of the $k+1$ permutations $\sigma \in \mathfrak{S}_k$, $\sigma_1 \in \mathfrak{S}_t$, $\ldots$, $\sigma_k \in \mathfrak{S}_t$ of an instance of STRUCTURED 2-TRACK HITTING SET, with $k = 4$ and $t = 6$.

ETH-based lower bounds. The *Exponential Time Hypothesis* (ETH) is a conjecture by Impagliazzo et al. [18] asserting that there is no $2^{o(n)}$-time algorithm for 3-SAT on instances with n variables.

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 (where $E(V_i, V_j)$ is the set of edges with one endpoint in V_i and the other endpoint in V_j). 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 a result by Marx imply that:

Theorem 1 ([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.*

Bonnet and Miltzow showed the following conditional lower bound for STRUCTURED 2-TRACK HITTING SET by a reduction from MULTICOLORED SUBGRAPH ISOMORPHISM linearly preserving the parameter:

Theorem 2 ([4]). STRUCTURED 2-TRACK HITTING SET *is $W[1]$-hard and, unless the ETH fails, cannot be solved in time $f(k)n^{o(k/\log k)}$ for any computable function f.*

The same lower bound has been shown for the very close problem 2-TRACK HITTING SET by Marx and Pilipczuk [24]. They use this intermediate result to show that covering a given set of points in the plane with k axis-parallel rectangles taken from a prescribed set cannot be solved in time $f(k)n^{o(k/\log k)}$, even if the rectangles are almost squares. Bonnet and Miltzow used Theorem 2 to show the same lower bound for POINT GUARD ART GALLERY and VERTEX GUARD ART GALLERY, where one wants to guard a simple polygon with k points, and k vertices, respectively. In this paper, we again utilize S2-THS for a reduction to RED-BLUE SEPARATION. Thus, it seems as though (STRUCTURED) 2-TRACK HITTING SET can be a good starting point for a wide variety of geometric problems and yield almost tight lower bounds, like GRID TILING [22] has been doing in the last decade for geometric problems optimally solvable in $n^{\Theta(\sqrt{k})}$.

3   Parameterized hardness for arbitrary slopes

In this section, we show that RED-BLUE SEPARATION is unlikely to be FPT with respect to the number of lines k and establish that, unless the ETH fails, the $n^{O(k)}$-time brute-force algorithm is almost optimal.

Let us say a few words about the difficulty of showing such a result for RED-BLUE SEPARATION, compared to its mere NP-hardness. The $W[1]$-hardness of a problem is very often obtained by a reduction from a (W[1]-hard) *graph problem* such as MULTICOLORED CLIQUE. Sometimes, the reduction is from a constraint satisfaction problem (CSP), like variants of SAT. In either cases, this supposes to encode binary, ternary, or r-ary (with $r > 3$) relations (adjacency matrix, associated hypergraph of a CSP, etc.). The two following paragraphs argue that it is challenging to do so for RED-BLUE SEPARATION without blowing-up the size of the parameter.

A set of k lines creates at most $h(k) := \binom{k+1}{2} + 1$ cells. Therefore, any YES-instance can be covered by $h(k)$ pairwise-disjoint monochromatic convex sets. This prevents one from encoding an adjacency matrix on n vertices with bichromatic gadgets. Indeed, the number of monochromatic convex sets necessary to cover such an encoding could not be bounded by a function of k. On the other hand, one does not seem to achieve much with a monochromatic encoding.

A more concrete issue with encoding an adjacency matrix is the following. Suppose we try to reduce directly from MULTICOLORED SUBGRAPH ISOMORPHISM (or its special case MULTICOLORED CLIQUE), and we want a horizontal line $L(u)$ to represent the choice of a vertex u within one set, a vertical line $L(v)$ to represent the choice of a vertex v in another set, and the lines are compatible iff uv is an edge. Here is the pitfall: if uv and uw are edges, then $L(u)$ should form a feasible solution with $L(v)$ and with $L(w)$; but then, it can be observed that every vertical line in between $L(v)$ and $L(w)$ also completes $L(u)$ into a feasible solution (which is undesirable as soon as there are vertices between v and w which are not adjacent to u).

We overcome those issues by reducing from STRUCTURED 2-TRACK HITTING SET. Deconstructing S2-THS, one finds intervals, a permutation of the color classes σ, and k permutations σ_j's of the elements within the classes. Intervals, thanks to their geometric

nature, can be realized by two red points which have to be separated from a diagonal of blue points (see Figure 3), while permutation σ, being on k elements, can be designed straightforwardly without blowing-up the size of the solution (see Figure 4). For these gadgets, we would like to force the chosen lines to be axis-parallel. We obtain this by surrounding them with *long alleys* made of long red paths parallel and next to long blue paths (see Figure 2). The main challenge is to get the permutations σ_j's on t elements. To attain this, we match a selected line L_i (corresponding to an element of index $i \in [t]$) to a specific angle α_i, which *leads* to the intended position of the element of index $\sigma_j(l) = i$, for some $l \in [t]$ (see Figure 6). Note that the depicted gadget actually links the element of index i to elements equal to or smaller than the element indexed at $\sigma_j(l)$. By combining two of these gadgets we can easily obtain only the intended position (see Figure 7).

Theorem 3. RED-BLUE SEPARATION *is $W[1]$-hard w.r.t. the number of lines k, and unless ETH fails, cannot be solved in time $f(k)n^{o(k/\log k)}$ for any computable function f.*

Proof. We reduce from S2-THS, which is $W[1]$-hard and has the above lower bound under ETH [4]. Let $\mathcal{I} = (k \in \mathbb{N}, t \in \mathbb{N}, \sigma \in \mathfrak{S}_k, \sigma_1 \in \mathfrak{S}_t, \ldots, \sigma_k \in \mathfrak{S}_t, \mathcal{S}_A, \mathcal{S}_B)$ be an instance of S2-THS. We will build an instance $\mathcal{J} = (\mathcal{R}, \mathcal{B}, 6k + 14)$ of RED-BLUE SEPARATION such that $\mathcal{I}$ is a YES-instance for S2-THS if and only if $\mathcal{R}$ and $\mathcal{B}$ can be separated with $6k + 14$ lines.

The points in $\mathcal{R}$ and $\mathcal{B}$ will have rational coordinates. More precisely, most points will be pinned to a z-by-z grid where z is polynomial in the size of $\mathcal{I}$. The rest will have rational coordinates with nominator and denominator polynomial in z. Let Γ be the z-by-z grid corresponding to the set of points with coordinates in $[z] \times [z]$. We call *horizontally* (resp. *vertically*) *consecutive points from a to b in row y* (resp. *in column x*) a set of points of Γ with coordinates $(a, y), (a + 1, y), \ldots (b - 1, y), (b, y)$ for $a, b, y \in [z]$ and $a < b$ (resp. $(x, a), (x, a + 1), \ldots (x, b - 1), (x, b)$ for $a, b, x \in [z]$ and $a < b$).

Long alley gadgets. In the gadgets encoding the intervals (see next paragraph), we will need to restrict the selected separating lines to be almost horizontal or almost vertical. To enforce that, we use the *long alley* gadgets. A *horizontal long alley* gadget is made of ℓ horizontally consecutive red points from a to $a + \ell - 1$ in row y, and ℓ horizontally consecutive blue points from a to $a + \ell - 1$ in row y', with $a, a + \ell - 1, y \neq y' \in [z]$ (see Figure 2(a)). A *vertical long alley* is defined analogously. Long alleys are called so because $\ell \gg |y - y'|$ thus, separating the red points from the blue points of a horizontal (resp. vertical) long alley with a budget of only one line, requires the line to be almost horizontal (resp. vertical). The use of the long alleys will be the following. Let $\mathcal{G}$ be a gadget for which we wish the separating lines to be almost horizontal or vertical. Say, $\mathcal{G}$ occupies a tiny square subgrid of Γ. We place four long alley gadgets to the left, top, right, and bottom of $\mathcal{G}$: horizontal ones to the left and right, vertical ones to the top and bottom (as depicted in Figure 2(c)). The left horizontal (resp. bottom vertical) long alley starts at the x-coordinate (resp. y coordinate) of 1, whereas the right horizontal (resp. top vertical) long alley ends at the x-coordinate (resp. y coordinate) of z; see Figure 7, where the long alleys are depicted by thin rectangles.

Note that we will not surround each and every gadget of the construction by four long alleys. At some places, it will indeed be crucial that the lines can have arbitrary slopes.
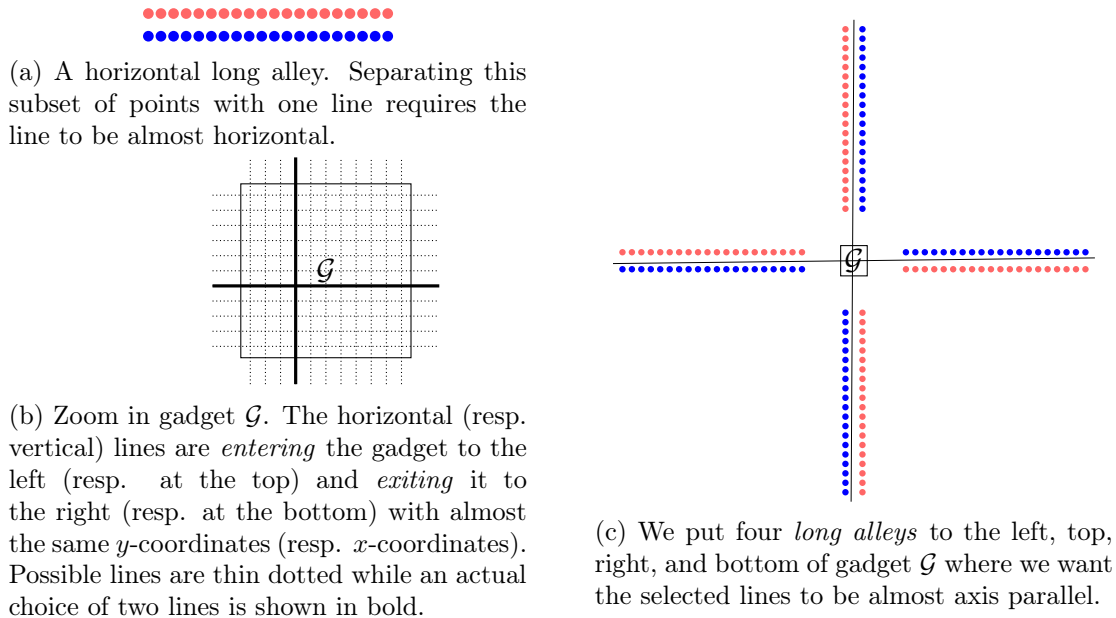
(a) A horizontal long alley. Separating this subset of points with one line requires the line to be almost horizontal.

(b) Zoom in gadget $\mathcal{G}$. The horizontal (resp. vertical) lines are *entering* the gadget to the left (resp. at the top) and *exiting* it to the right (resp. at the bottom) with almost the same y-coordinates (resp. x-coordinates). Possible lines are thin dotted while an actual choice of two lines is shown in bold.

(c) We put four *long alleys* to the left, top, right, and bottom of gadget $\mathcal{G}$ where we want the selected lines to be almost axis parallel.

Figure 2: The long alley gadget and its use in combination with another gadget.

Interval gadgets and encoding track A. The elements of A are represented by a diagonal of kt blue points. More precisely, we add the points $(x_0, y_0), (x_0 + 4, y_0 + 4), (x_0 + 8, y_0 + 8), \ldots, (x_0 + 4kt - 4, y_0 + 4kt - 4)$ to $\mathcal{B}$ for some offset $x_0, y_0 \in [z]$ that we will specify later. An almost horizontal (resp. vertical) line just below (resp. just to the left of) the s-th blue point of this diagonal translates as selecting the s-th element of A in the order fixed by $\leq_A$.

For each interval $[a_i^j, a_{i'}^{j'}]$ in $\mathcal{S}_A$ (for some $i, i' \in [t]$, $j, j' \in [k]$), that is, the interval between the $s = ((j-1)t + i)$-th and the $s' = ((j'-1)t + i')$-th elements of A, we add two red points: one at $(x_0 + 4s - 7, y_0 + 4s' - 5)$ and one at $(x_0 + 4s' - 5, y_0 + 4s - 7)$ (see Figure 3(a) for one interval gadget and Figure 3(b) for track A). Let $R([a_i^j, a_{i'}^{j'}])$ be this pair of red points. Informally, one red point has its projection on the x-axis just to the left of the s-th blue point and its projection on the y-axis just above the $s' - 1$-st blue point; the other one has its projection on the x-axis just to the right of the $s' - 1$-st blue point and its projection on the y-axis just below the s-th blue point. For technical reasons, we add, for every $j \in [k]$, the pair $R([a_1^j, a_t^j])$ encoding the interval formed by all the elements of the j-th color class of A. Adding these intervals to $\mathcal{S}_A$ does not constrain the problem more. Indeed, by definition of STRUCTURED 2-TRACK HITTING SET, one is required to take one pair of elements per color class.

We surround this encoding of track A, which we denote by $\mathcal{G}(A)$, with $4k$ long alleys, whose width is $4t$, from x-coordinates $x_0 + 4(j-1)t - 4$ to $x_0 + 4jt - 4$ for vertical alleys (from y-coordinates $y_0 + 4(j-1)t - 4$ to $y_0 + 4jt - 4$ for horizontal alleys). Note that half of the points of one alley coincides with half of the points of the next alley. We alternate red-blue[1] alleys and blue-red alleys for two contiguous alleys, so that we do not place at the

---

[1] i.e., for horizontal (resp. vertical) alleys, the red points are above (resp. to the left of) the blue points.

same position a red point and a blue point.

We start with a red-blue alley for the left horizontal and top vertical groups of alleys, and with a blue-red alley for the right horizontal and bottom vertical. This last detail is not in any way crucial but permits the construction to be defined uniquely and consistently with the choices of Figure 2(c). This, together with the description of long alleys in the previous paragraph, fully defines the $4k$ long alleys (see Figure 7). It can be observed that the blue points of the diagonal with index $t, 2t, 3t, \ldots, kt$ (that is, those sharing a coordinate with the long alleys) do not play any role. They can be removed or kept.

The general intention is that in order to separate two red points (encoding the same interval) from the blue diagonal with a budget of two almost axis-parallel lines, one should take two lines (one almost horizontal and one almost vertical) corresponding to the selection of the same element of A which hits the corresponding interval. In particular, taking two almost horizontal lines (resp. two almost vertical lines) is made impossible due to those vertical (resp. horizontal) long alleys. More precisely, the intended pairs of lines separating the red points $R([a_i^j, a_{i'}^{j'}])$ from the blue diagonal are of the form $x = x_0 + 4\hat{s} - 6, y = y_0 + 4\hat{s} - 6$ for $\hat{s} \in [(j-1)t + i, (j'-1)t + i']$. Furthermore, the $4k$ long alleys force a pair of (almost) horizontal and vertical lines corresponding to one element per color class to be taken.

For any $s \in [tk]$, $i \in [t]$, and $j \in [k]$, such that $s = (j-1)t + i$, let $\mathrm{HL}(s)$ be the horizontal line of equation $y = y_0 + 4s - 6$ and $\mathrm{VL}(s)$ the vertical line of equation $x = x_0 + 4s - 6$. They *correspond* to selecting a_i^j, the i-th element in the j-th color class of A. The goal of the remaining gadgets is to ensure that when the lines $\mathrm{HL}(s)$ and $\mathrm{VL}(s)$ (with $s = (j-1)t + i$) are chosen, additional lines corresponding to selecting element b_i^j of B have to be expressly selected. We define $\mathrm{HL} := \{\mathrm{HL}(s) \mid s \in [tk]\}$ and $\mathrm{VL} := \{\mathrm{VL}(s) \mid s \in [tk]\}$.

Encoding inter-class permutation σ. To encode the permutation σ of the k color classes of $\mathcal{I}$, we allocate a square subgrid of the same dimension as the space used for the encoding of track A, roughly $4tk$-by-$4tk$, and we place it to the right of A as depicted in Figure 7. This square subgrid is naturally and regularly split into k^2 smaller square subgrids of equal dimension (roughly $4t$-by-$4t$). This decomposition can be seen as the k color classes of $\mathcal{I}$, or equivalently, the k-by-k crossing[2] obtained by drawing horizontal lines between two contiguous horizontal long alleys and vertical lines between two contiguous vertical long alleys. We only put points in exactly one smaller square subgrid per column and per row. Let $\sigma := \sigma(1)\sigma(2)\ldots\sigma(k)$ and $\mathrm{Cell}(a, b)$ be the smaller square subgrid in the a-th row and b-th column of the k-by-k crossing. For each $j \in [k]$, we put in $\mathrm{Cell}(\sigma(j), j)$ a diagonal of $t-1$ blue points and two red points corresponding to the full interval $[a_1^j, a_t^j]$ (see Figure 4). We denote by $\mathcal{G}(\sigma)$ those sets of red and blue points in the encoding of σ. We surround $\mathcal{G}(\sigma)$ by $2k$ vertical long alleys similar to the $2k$ long alleys surrounding $\mathcal{G}(A)$. Notice that $\mathcal{G}(\sigma)$ and $\mathcal{G}(A)$ share the same $2k$ surrounding horizontal long alleys.

The way the gadget $\mathcal{G}(\sigma)$ works is quite intuitive. Given k choices of horizontal lines originating from a separation in $\mathcal{G}(A)$ and a budget of k extra lines for the separation within $\mathcal{G}(\sigma)$, the only option is to copy with the vertical line the choice of the horizontal line. It

---

[2]we use this term informally to avoid confusion with what we have been calling *grids* so far.
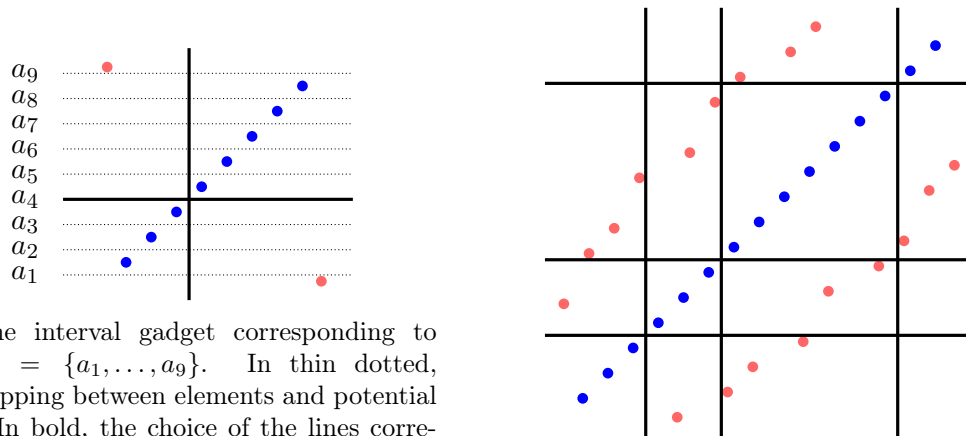
(a) The interval gadget corresponding to $[a_1, a_9] = \{a_1, \ldots, a_9\}$. In thin dotted, the mapping between elements and potential lines. In bold, the choice of the lines corresponding to picking a_4. If one wants to separate these points with two lines, one almost horizontal and one almost vertical, the choice of the former imposes the latter.

(b) The interval gadgets put together. A representation of one track. Separating these points with the fewest axis-parallel lines requires taking the horizontal and vertical lines associated to a minimum hitting set.

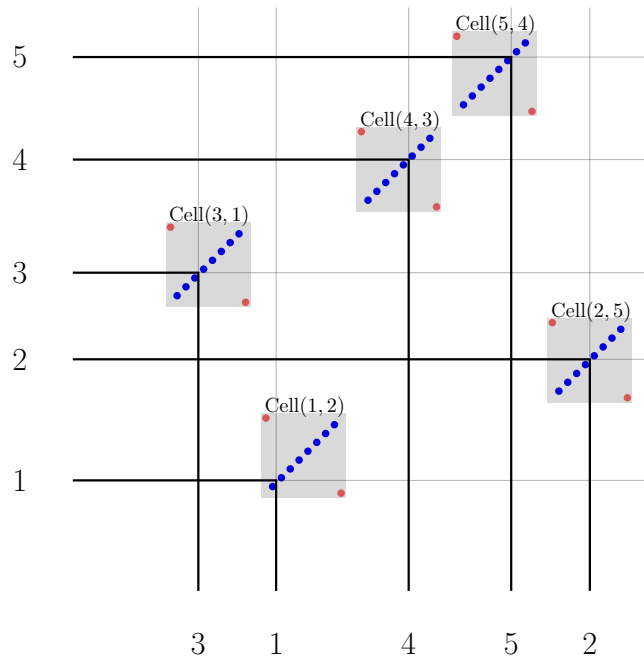Figure 3: To the left, one interval. To the right, several put together to form one track.

Figure 4: Encoding permutation $\sigma = 31452$. The choices within the five color classes are transferred from almost horizontal lines to almost vertical ones. This way, we obtain the desired reordering of the color classes.

results in a vertical propagation of the initial choices accompanied by the desired reordering of the *color classes* (see Figure 5). The vertical line matching the choice of HL(s) in the

corresponding cell of $\mathcal{G}(\sigma)$ is denoted by $\mathrm{VL}'(s)$. Let $\mathrm{VL}' := \{\mathrm{VL}'(s) \mid s \in [tk]\}$. Note that corresponding lines in VL and in VL' have a different order from left to right.
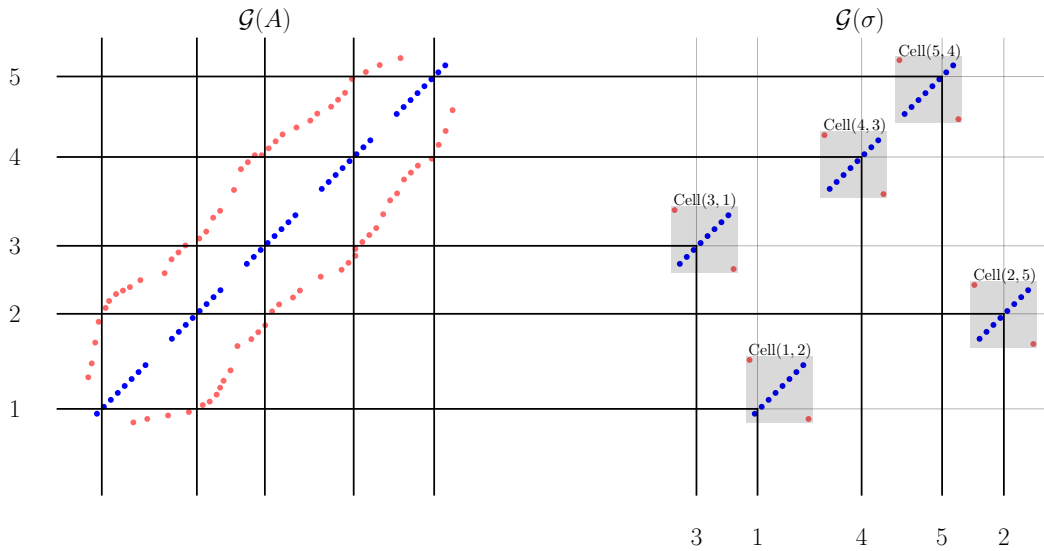

Figure 5: Interaction between the gadgets $\mathcal{G}(A)$ and $\mathcal{G}(\sigma)$ with the permutation $\sigma = 31452$.

Encoding of the intra-class permutations σ_j's and track B. If the encoding of permutation σ is conceptually simple, the number of intended lines separating red and blue points in $\mathcal{G}(\sigma)$ has to be linear in the number of permuted elements. Since we wish to encode a permutation σ_j (for every $j \in [k]$) on t elements, we cannot use the same mechanism as it would blow-up our parameter dramatically and would not result in an FPT reduction.

For the gadget $\mathcal{G}_{\approx v}(\sigma_j)$ partially encoding the permutation σ_j, we will crucially use the fact that separating lines can have arbitrary slopes. Slightly to the right (at distance at least ℓ) of the vertical line bounding the right end of $\mathcal{G}(\sigma)$ and far in the south direction, we place a gadget $\mathcal{G}(B)$ encoding track B similarly to the encoding of track A up to some symmetry that we will make precise later. We also incline the whole encoding of track B with a small, say 5, degree angle, in a way that its top-left corner is to the right of its bottom-left corner. We round up the real coordinates that this rotation incurs to rationals at distance less than, say, $(kt)^{-10}$. We denote by $\hat{v}$ the distance along the y-axis between $\mathcal{G}(\sigma)$ and $\mathcal{G}(B)$. Eventually $\hat{v}$ will be chosen very large, much larger than $\Theta(kt)$, which is the size of $\mathcal{G}(A), \mathcal{G}(B), \mathcal{G}(\sigma)$. Below $\mathcal{G}(\sigma)$ at a distance $2\hat{v}$ along the y-axis, we place gadgets $\mathcal{G}_{\approx v}(\sigma_j)$'s; from left to right, we place $\mathcal{G}_{\approx v}(\sigma_{\sigma(1)}), \mathcal{G}_{\approx v}(\sigma_{\sigma(2)}), \ldots, \mathcal{G}_{\approx v}(\sigma_{\sigma(k)})$ such that for every $i \in [k]$, $G_{\approx v}(\sigma_{\sigma(i)})$ falls below the i-th column of the k-by-k crossing of $\mathcal{G}(\sigma)$. Gadgets $\mathcal{G}_{\approx v}(\sigma_j)$'s are represented by small round shapes in Figure 7. Notwithstanding what is drawn on the overall picture, the $\mathcal{G}_{\approx v}(\sigma_j)$'s can be all placed at the same y-coordinates. Let $y_1 := y_0 - 2\hat{v}$ (the exact value of y_1 is not crucial). Also, we represent track B slanted by a 45 degree angle, instead of the actual 5 degree angle, to be able to fit everything on one page and convey the main ideas of the construction. In general, for the figure to be readable, the true proportions are not respected. The size of every gadget is much smaller than the distance between two different groups of gadgets, so that every line *entering* a gadget traverses it in

an axis-parallel fashion.

Gadget $\mathcal{G}_{\approx v}(\sigma_j)$ is built in the following way. For each $i \in [t]$ and $j \in [k]$, we draw a fictitious point p_i^j at the intersection of the close to vertical line corresponding to picking element b_i^j in gadget $\mathcal{G}(B)$ with the bottom end of $\mathcal{G}(B)$. Read from left to right, the p_i^j's have the same order as the b_i^j's in $(B, \leq_B)$. For every $s = (j - 1)t + i$ (with $j \in [k]$ and $i \in [t]$), let q_i^j be the point of y-coordinate y_1 on the line $\mathrm{VL}'(s)$. We define the line $\mathrm{SL}(s)$ as going through p_i^j and q_i^j, and set $\mathrm{SL} := \{\mathrm{SL}(s) \mid s \in [tk]\}$. We add two blue points just to the left and just to the right of q_i^j at distance $\epsilon := z^{-10}$. We also add two blue points on line $\mathrm{SL}(s)$, one to the left of q_i^j and one to the right of q_i^j. Finally, we place two red points for each $\mathcal{G}_{\approx v}(\sigma_j)$ at the bottom-left and top-right of the gadget, in a way that the bottom-left red point is below all the lines $\mathrm{SL}(s)$ of the gadget, and the top-right red point is above all those lines. Furthermore, the former (resp. latter) red point is to the left of the leftmost (resp. to the right of the rightmost) line $\mathrm{VL}'(s)$ of $\mathcal{G}_{\approx v}(\sigma_j)$ (see Figure 6). Note that in the figure, the lines in SL form a large angle with the y-axis, while in fact they are quite close to a 5 degree angle and behave like *relatively vertical*[3] lines within $\mathcal{G}(B)$ (since $\mathcal{G}(B)$ is also inclined by 5 degrees).

Assuming that line $\mathrm{VL}'(s = (j - 1)t + i)$ has been selected, it might be observed from Figure 6 that separating the red points from the blue points in $\mathcal{G}_{\approx v}(\sigma_j)$ with a budget of one additional line requires to take a line crossing $\mathrm{VL}'(s)$ at (or very close to) q_i^j and with a higher or equal slope to $\mathrm{SL}(s)$. It is not quite what we wanted. What we achieved so far is only to link the *choice of* a_i^j with the *choice of an element smaller or equal to* b_i^j. We will use a gadget $\mathcal{G}_{\approx v}(\sigma_j)$, which is symmetric to $\mathcal{G}_{\approx h}(\sigma_j)$, to get the other inequality so that choosing some lines corresponding to a_i^j actually forces to take some lines corresponding to b_i^j.

We add a gadget $\mathcal{G}(\mathrm{id})$ below the $\mathcal{G}_{\approx v}(\sigma_j)$'s. $\mathcal{G}(\mathrm{id})$ is obtained by mimicking $\mathcal{G}(\sigma)$ for the identity permutation. We surround $\mathcal{G}(\mathrm{id})$ by $2k$ new horizontal long alleys. The horizontal line matching the choice of $\mathrm{VL}'(s)$ in $\mathcal{G}(\mathrm{id})$ is denoted by $\mathrm{HL}'(s)$. At a distance $\hat{h} \approx \hat{v}/(cos(5°) \cdot sin(5°))$ to the right of $\mathcal{G}(\mathrm{id})$ we place gadgets $\mathcal{G}_{\approx h}(\sigma_j)$'s analogously to the $\mathcal{G}_{\approx v}(\sigma_j)$'s. The fictitious points p'^j_i (analogous of p_i^j) used for the construction of the lines $\mathrm{SL}'(s)$ (analogous of $\mathrm{SL}(s)$) are located at the right end of $\mathcal{G}(B)$ and ordered as B when read from top to bottom. The slight difference in the construction of $\mathcal{G}(B)$ from the B-intervals (compared to $\mathcal{G}(A)$ from the A-intervals) is that the diagonal of blue points go from the top-left corner to the bottom-right corner (instead of bottom-left to top-right). Similarly to our previous definitions, we define $\mathrm{HL}' := \{\mathrm{HL}'(s) \mid s \in [tk]\}$ and $\mathrm{SL}' := \{\mathrm{SL}'(s) \mid s \in [tk]\}$. Note that the choice of $\hat{h}$ makes the lines of SL' form a close to 5 degree angle with the x-axis and so *arrive relatively horizontal* within $\mathcal{G}(B)$.

Putting the pieces together. We already hinted at how the different gadgets are combined together. We choose the different typical values so that: $kt \ll \hat{v} < \hat{h} \ll z$. For instance, $\hat{v} := 100((kt)^2 + 1)$ and $z := 100(\hat{h}^5 + 1)$. An important and somewhat hidden consequence of z being much greater than $\hat{v}$ and $\hat{h}$ is that the bulk of the construction (say, all the gadgets which are not long alleys) occupies a tiny space in the top-left corner of

---

[3]By that, we mean that the lines are close to vertical for axes aligned with the encoding of track B.

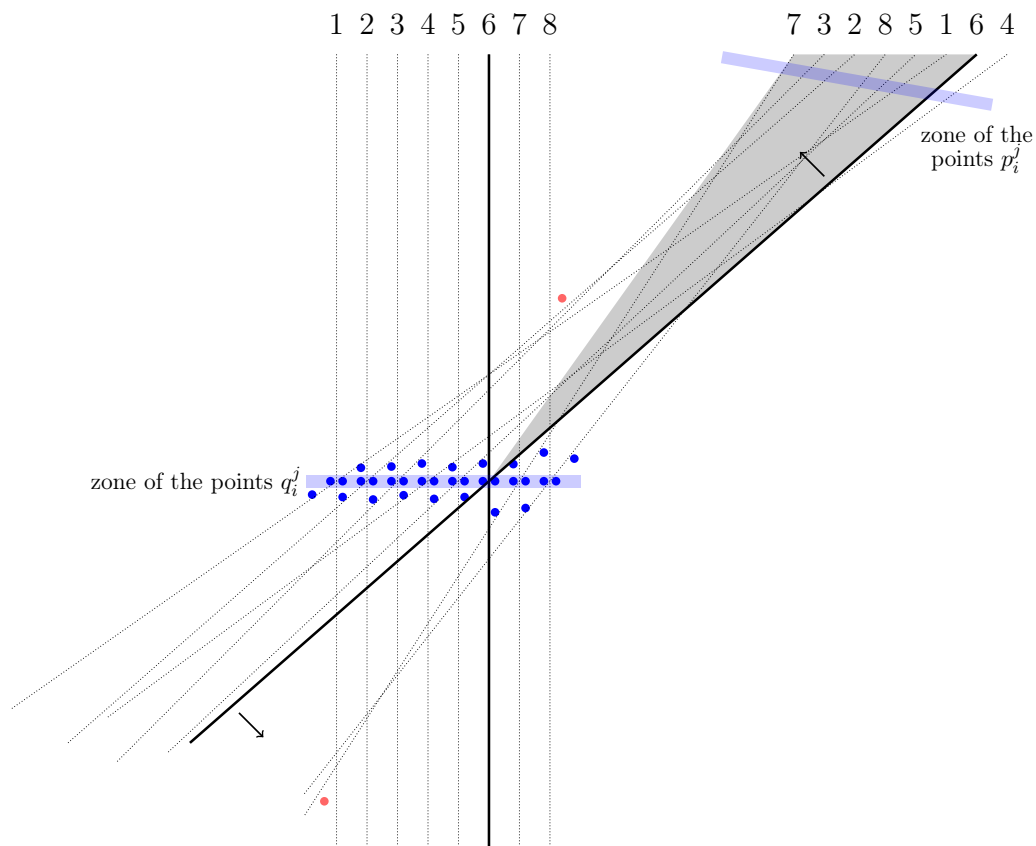Figure 6: Half-encoding of permutation $\sigma_j = 73285164$ of the j-th color class. Observe that the choice of the, say, sixth almost horizontal candidate line only forces to take the slanted line depicted in bold or a line having the same intersection with the almost horizontal line but a larger slope. For the sake of legibility, the angles between the vertical lines and the slanted lines are exaggerated.

Γ. We set the length ℓ of the long alleys to $100(k^2 + 1)$. Point (x_0, y_0) corresponds to the bottom-left corner of the square in bold with a diagonal close to the overall top-left corner.

Slightly outside grid Γ we place 14 pairs of long alleys (7 horizontal and 7 vertical) of width, say, $(kt)^{-10}$ to force the 14 lines in bold in Figure 7. Note that, on the figure, we do not explicitly represent those long alleys but only the lines they force. The purpose of those new long alleys is to separate groups of gadgets from each other. Going clockwise all around the grid Γ, we alternate red-blue and blue-red alleys so that two consecutive long alleys do not need a further separation. The even parity of those alleys make this alternation possible. Each one of the 64 faces that those 14 lines define is called a *super-cell*.

The four lines in bold surrounding $\mathcal{G}(B)$ are close (say, at distance $10t$) to the north, south, west, and east ends of that gadget. On the four super-cells adjacent to the super-cell containing $\mathcal{G}(B)$, shown in gray, we place $4k$ long alleys each of width $4t$, analogously to what was done for $\mathcal{G}(A)$, but slanted by a 5 degree angle (as the gadget $\mathcal{G}(B)$). As for track A, these alleys force, relatively to the orientation of $\mathcal{G}(B)$, one *close to horizontal* line and
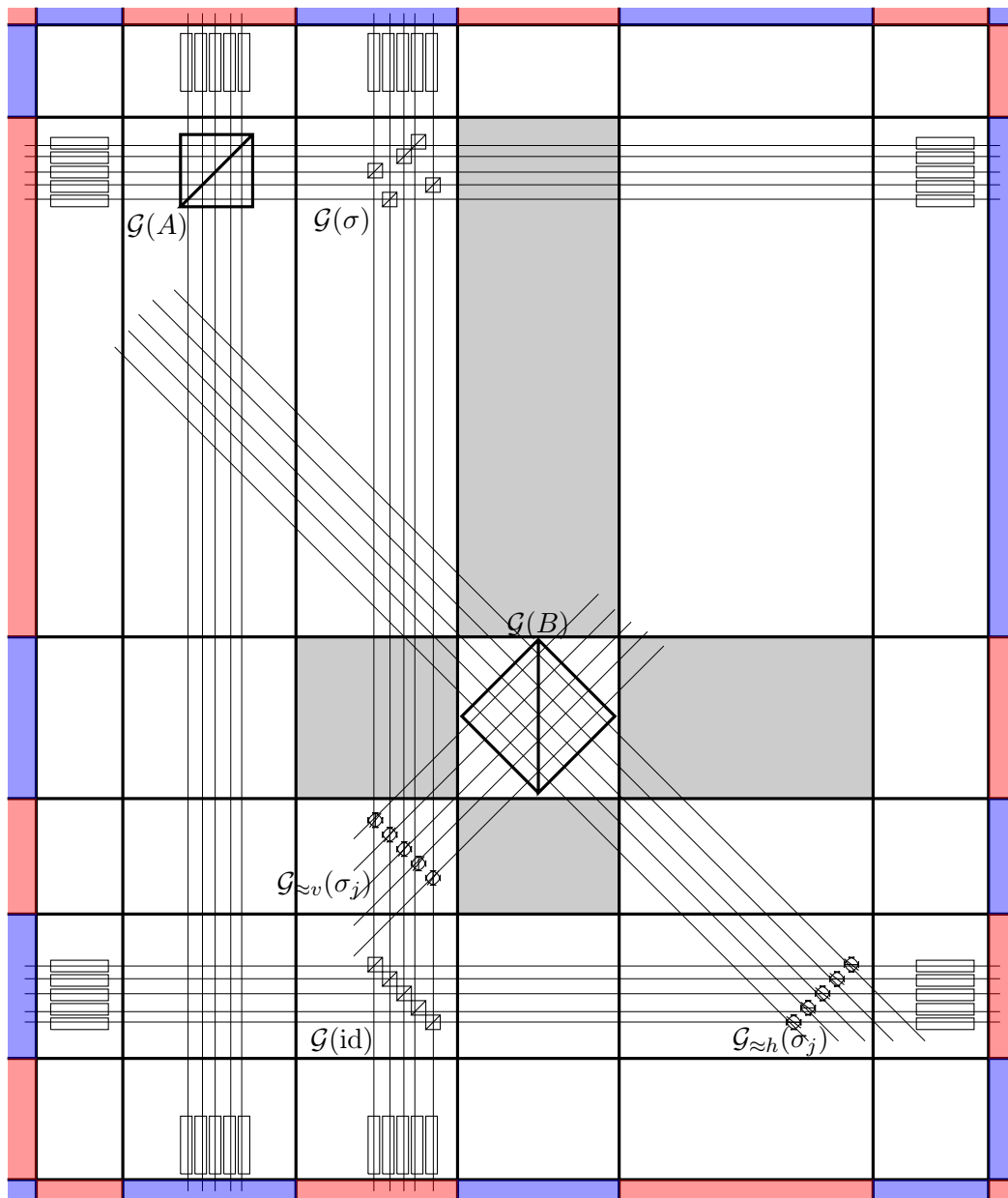
Figure 7: The overall picture. The thin rectangles are long alleys, the bold large squares with a diagonal are the encoding of track A, in the top left corner, and track B, slanted by 45 degrees (for the sake of fitting the whole construction on one page; in reality the encoding of B is only inclined by 5 degrees). The smaller squares with a diagonal are simple interval gadgets and the small round gadgets are half-encodings of the permutations σ_i's. The four super-cells filled with grey contain $4k$ long alleys slanted by 5 degrees. The (super-)cells filled with red and blue match their color, and are monochromatic once the 14 lines imposed by the outermost long alleys have been selected.

one *close to vertical line* per color class. The long alleys are placed just next to $\mathcal{G}(B)$ and are not crossed by any other candidate lines.

This finishes the construction. We ask for a separation of $\mathcal{R}$ and $\mathcal{B}$ with $6k + 14$ lines. We now show the correctness of the reduction.

If $\mathcal{I}$ is a YES-instance for S2-THS, then $6k + 14$ lines are sufficient. Let F be the set of 14 lines forced by the outermost long alleys (lines in bold in Figure 7). Let $(a_{u_1}^1, b_{u_1}^1), (a_{u_2}^2, b_{u_2}^2), \ldots, (a_{u_k}^k, b_{u_k}^k)$ be a solution of S2-THS $(u_1, u_2, \ldots, u_k \in [t])$. Let $s_j := (j-1)t + u_j$ for every $j \in [k]$. $F \cup \bigcup_{j \in [k]} \{\text{HL}(s_j), \text{VL}(s_j), \text{VL}'(s_j), \text{HL}'(s_j), \text{SL}(s_j), \text{SL}'(s_j)\}$ is a set of $6k + 14$ lines. We claim that it is a solution.

Due to F, we only need to check that the red and blue points of the *same* super-cell are separated. The constant number of outermost long alleys are well separated: see the alternating coloring of Figure 7. As the other long alleys also alternates red-blue and blue-red, the super-cells containing k long alleys are all well separated.

This leaves us 6 super-cells to check: namely those of $\mathcal{G}(A)$, $\mathcal{G}(B)$, $\mathcal{G}(\sigma)$, $\mathcal{G}(\text{id})$, the $\mathcal{G}_{\approx v}(\sigma_j)$'s, and the $\mathcal{G}_{\approx h}(\sigma_j)$'s. The points in $\mathcal{G}(\sigma)$ and $\mathcal{G}(\text{id})$ are separated as in Figure 3a, since the choice of $\text{VL}'(s_j)$ matches the choices of $\text{HL}(s_j)$ and $\text{HL}'(s_j)$. As it can be observed by looking at Cell$(4, 3)$ and Cell$(5, 4)$ of Figure 4, there is no interaction between the red and blue points of diagonally adjacent faces of the k-by-k crossing (in $\mathcal{G}(\sigma)$ and $\mathcal{G}(\text{id})$).

Since $a_{u_1}^1, a_{u_2}^2, \ldots, a_{u_k}^k$ (resp. $b_{u_1}^1, b_{u_2}^2, \ldots, b_{u_k}^k$) is a hitting set of $\mathcal{S}_A$ (resp. $\mathcal{S}_B$), the points in $\mathcal{G}(A)$ (resp. $\mathcal{G}(B)$) are separated as in Figure 3b. Indeed for each interval $I \in \mathcal{S}_A$ (resp $I \in \mathcal{S}_B$), there is a $j \in [k]$ such that $a_{u_j}^j$ hits I (resp. $b_{u_j}^j$ hits I), and the two red points encoding I are in the two quadrants defined by $\text{HL}(s_j)$ and $\text{VL}(s_j)$ (resp. defined by $\text{SL}(s_j)$ and $\text{SL}'(s_j)$) where there is no blue point.

Similarly the two red points of a gadget $\mathcal{G}_{\approx v}(\sigma_j)$ (resp. $\mathcal{G}_{\approx h}(\sigma_j)$) are separated from the blue points: they are in the two regions defined by $\text{VL}'(s_j)$ and $\text{SL}(s_j)$ (resp. $\text{HL}'(s_j)$ and $\text{SL}'(s_j)$) where there is no blue point. Two consecutive gadgets $\mathcal{G}_{\approx v}(\sigma_j)$ and $\mathcal{G}_{\approx v}(\sigma_{j+1})$ (resp. $\mathcal{G}_{\approx h}(\sigma_j)$ and $\mathcal{G}_{\approx h}(\sigma_{j+1})$) do not interact. In fact, all the blue points land in the quadrangular faces touching two consecutive gadgets.

If $6k + 14$ lines are sufficient, then $\mathcal{I}$ is a YES-instance for S2-THS. Let S be a feasible solution consisting of at most $6k + 14$ lines. The lines of S should separate *all* the straight-line segments whose one extremity is at a red point and the other is at a blue point. We call such a segment a *red/blue segment* or a *red/blue pair* (or simply *pair*).

First, we can assume that $F \subseteq S$, where F is the set of 14 lines forced by the 28 outermost long alleys. Indeed, in each of those long alleys there should be a line of S separating at least two red/blue segments, such that the two segments and the line have *not* a common intersection. For every line $\mathcal{L}$ satisfying this property, the line in F responsible from separating this long alley separates a superset of the red/blue pairs separated by $\mathcal{L}$; and therefore can be chosen.

We will now focus on a particular subset of red/blue pairs. Consider the set $\mathcal{X}$ of the red/blue segments within each of the $12k$ remaining long alleys between two points with the same x-coordinate (resp. y-coordinate) in a horizontal alley (resp. vertical alley), and

by generalizing in the natural way this notion for the close to horizontal (resp. vertical) alleys surrounding $\mathcal{G}(B)$. There are ℓ such red/blue pairs per long alley, hence $|\mathcal{X}| = 12k\ell$. We partition the $12k$ long alleys into eight *groups*: $\mathcal{A}_W, \mathcal{A}_E, \mathcal{A}_N, \mathcal{A}_S$, the axis-parallel long alleys to the west, east, north, and respectively, south of Γ, and $\mathcal{B}_W, \mathcal{B}_E, \mathcal{B}_N, \mathcal{B}_S$ the slightly slanted long alleys to the west, east, north, and respectively, south of $\mathcal{G}(B)$.

Lemma 4. *No line separates strictly more than 2ℓ red/blue pairs of $\mathcal{X}$. Furthermore, the only way for a line to separate 2ℓ red/blue pairs of $\mathcal{X}$ is to separate all the red/blue pairs of $\mathcal{X}$ of two long alleys belonging to a pair in $\{(\mathcal{A}_W, \mathcal{A}_E), (\mathcal{A}_N, \mathcal{A}_S), (\mathcal{B}_W, \mathcal{B}_E), (\mathcal{B}_N, \mathcal{B}_S)\}$ (and no other pair of $\mathcal{X}$).*

Proof. Within the same group of long alleys, a line separates at most ℓ red/blue pairs of $\mathcal{X}$. Indeed, say, the group of long alleys consists of horizontal alleys. Then a line cannot separate two red/blue pairs sharing the same x-coordinate. Furthermore, it can be observed that to separate within the same group exactly ℓ red/blue pairs of $\mathcal{X}$, the line has to separate the red/blue pairs of the *same* long alley.

We also observe that a line intersects a positive number of red/blue pairs of $\mathcal{X}$ in at most two groups among $\mathcal{A}_W, \mathcal{A}_E, \mathcal{A}_N$, and $\mathcal{A}_S$ (resp. $\mathcal{B}_W, \mathcal{B}_E, \mathcal{B}_N$, and $\mathcal{B}_S$) and at most three of the eight groups.

If a line intersects red/blue pairs of $\mathcal{X}$ in three groups, then those groups have to be (a) $\mathcal{B}_W, \mathcal{B}_E$, and $\mathcal{A}_W$, or (b) $\mathcal{B}_W, \mathcal{B}_E$, and $\mathcal{A}_E$, or (c) $\mathcal{B}_N, \mathcal{B}_S$, and $\mathcal{A}_N$, or (d) $\mathcal{B}_N, \mathcal{B}_S$, and $\mathcal{A}_S$. Here we use the fact that $\hat{h} \ll z$. Hence, all the other gadgets are much closer to the long alleys in $\mathcal{A}_W$ and $\mathcal{A}_N$ than to the long alleys in $\mathcal{A}_E$ and $\mathcal{A}_S$. Thus, a line separating red/blue pairs in, say, $\mathcal{A}_E$ and $\mathcal{B}_E$ looks horizontal between $\mathcal{B}_E$ and the west end of Γ, and therefore cannot separate red/blue pairs in $\mathcal{A}_W$.

The cases (a), (b), (c), and (d) being symmetric, we only treat case (a). A line corresponding to case (a), cannot separate 2ℓ red/blue pairs of $\mathcal{X}$. Here we use the fact that the distance between two groups of gadgets is much larger than the size of the gadgets. So a line $\mathcal{L}$ separating some red/blue pairs in $\mathcal{A}_W$ and $\mathcal{B}_W$ looks horizontal between $\mathcal{B}_W$ and $\mathcal{B}_E$. As the long alleys of $\mathcal{B}_W$ and $\mathcal{B}_E$ are slanted by a 5 degree angle, $\mathcal{L}$ cannot separate more than $100k < \ell$ red/blue pairs of $\mathcal{X}$ in $\mathcal{B}_W \cup \mathcal{B}_E$. Indeed, a close to horizontal line cannot separate more than a constant (smaller than 50) number of red/blue pairs of $\mathcal{X}$ per long alley of $\mathcal{B}_W \cup \mathcal{B}_E$.

At this point, one can eventually observe that the only ways to separate 2ℓ red/blue pairs of $\mathcal{X}$ with one line, is to separate ℓ pairs in $\mathcal{A}_W$ (resp. $\mathcal{B}_W$) and ℓ pairs in $\mathcal{A}_E$ (resp. $\mathcal{B}_E$), or ℓ pairs in $\mathcal{A}_N$ (resp. $\mathcal{B}_N$) and ℓ pairs in $\mathcal{A}_S$ (resp. $\mathcal{B}_S$). By a previous remark, the separated pairs within a group come from the *same* long alley. □

As the remaining budget is $6k$ lines, it follows from Lemma 4 that *all* the lines of $S \setminus F$ have to separate exactly 2ℓ pairwise-disjoint red/blue pairs of $\mathcal{X}$. Furthermore, in $S \setminus F$, there are $2k$ almost horizontal lines separating one long alley in $\mathcal{A}_W$ and the other in $\mathcal{A}_E$, $2k$ almost vertical lines separating one long alley in $\mathcal{A}_N$ and the other in $\mathcal{A}_S$, k lines separating one long alley in $\mathcal{B}_W$ and the other in $\mathcal{B}_E$, and k lines separating one long alley in $\mathcal{B}_N$ and the other in $\mathcal{B}_S$.

Let us draw a small parenthesis. Despite what is represented in Figure 7, the line of S separating the h-th topmost long alley of $\mathcal{A}_W$ (resp. the v-th leftmost long alley of $\mathcal{A}_N$) does not necessarily separate the h-th topmost long alley of $\mathcal{A}_E$ (resp. the v-th leftmost long alley of $\mathcal{A}_S$). Instead, this line separates *one* long alley of $\mathcal{A}_E$ (resp. $\mathcal{A}_S$); it does not matter which one. Therefore, the exact position of the long alleys of $\mathcal{A}_E \cup \mathcal{A}_S$ is not crucial. What is important is that there are $2k$ horizontal long alleys very far east, and $2k$ vertical long alleys very far south. We nevertheless chose to align those alleys with the ones in $\mathcal{A}_W \cup \mathcal{A}_N$, since we think it leads to a more intuitive construction for the reader. This closes the parenthesis.

Let us focus on the k lines of S separating the k topmost long alleys of $\mathcal{A}_W$. For each $j \in [k]$, we denote by $\mathcal{L}_j$ the one separating the j-th bottommost of those k long alleys. As we already observed those lines behave like horizontal lines in the smallest subgrid enclosing all the gadgets which are *not* in $\mathcal{A}_E \cup \mathcal{A}_S$ (nor the 14 outermost long alleys). For each $j \in [k]$, let $a_{u_j}^j$ be the element of A corresponding to $\mathcal{L}_j$ (with the correspondence described in Figure 3(a)). In particular, by the position of the k topmost long alleys of $\mathcal{A}_W$, it is indeed true that the k lines $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k$ translates to exactly one element per color class of track A. We show, using the following lemma, that $a_{u_1}^1, a_{u_2}^2, \ldots, a_{u_k}^k$ is a hitting set of $(A, \mathcal{S}_A)$.

Lemma 5. *The only ways to separate a simple interval gadget with one horizontal line and one vertical line is to make them meet at the diagonal defined by the blue points.*

Proof. If the lines meet *above* the diagonal, then the bottom red point is not separated from the blue point just to the right of the vertical line. If the lines meet *below* the diagonal, then the top red point is not separated from the blue point just to the left of the vertical line. □

Recall that we added for convenience the pairs of red points $R([a_1^j, a_t^j])$, for each $j \in [k]$. We consider the simple interval gadget that each pair induces, that is, the two red points and the diagonal of blue points contained in the smallest square subgrid enclosing them. Because of the long alleys in $\mathcal{A}_W$ and $\mathcal{A}_N$, we have a budget of exactly one horizontal line and one vertical line to separate each of those k simple intervals. By Lemma 5, the k vertical lines of S separating the k leftmost long alleys of $\mathcal{A}_N$ have to *agree* with the choices of the horizontal lines $\mathcal{L}_j$'s. More formally, the j-th bottommost horizontal line intersects the j-th leftmost vertical line at the diagonal defined by the blue points.

This implies that all the intervals of $\mathcal{S}_A$ are hit by the $a_{u_j}^j$'s. Indeed, if an interval I is not hit, the smallest square subgrid γ_I enclosing the corresponding pair of red points would not be intersected by S; and those red points would not be separated from any diagonal blue point in γ_I.

We now show that the choice of the lines corresponding to the elements $a_{u_1}^1, a_{u_2}^2, \ldots, a_{u_k}^k$ will force to take the lines corresponding to the elements $b_{u_1}^1, b_{u_2}^2, \ldots, b_{u_k}^k$. Still by Lemma 5, $\mathcal{G}(\sigma)$ transmits the choices of the $\mathcal{L}_j$'s *downwards* with the desired permutation σ of the color classes, while $\mathcal{G}(\mathrm{id})$ transmits unchanged the choices of the vertical lines separating $\mathcal{G}(\sigma)$ *to the left*.

Similarly to the simple argument of Lemma 5, once the axis-parallel line has been selected in a gadget $\mathcal{G}_{\approx v}(\sigma_j)$ or $\mathcal{G}_{\approx h}(\sigma_j)$, to separate the two red points from the four blue points on or close to the intended line (that is, $\mathrm{SL}(s)$ when $\mathrm{VL}'(s)$ has been selected, or $\mathrm{SL}'(s)$ when $\mathrm{HL}'(s)$ has been selected), one should choose the intended line itself or any line having the same intersection with the axis-parallel line and closer to this axis (see Figure 6). The way the gadgets $\mathcal{G}_{\approx v}(\sigma_j)$'s, $\mathcal{G}_{\approx h}(\sigma_j)$'s, and $\mathcal{G}(B)$ are placed, it results in, for each color class j of track B, selecting a relatively horizontal line somewhere to the left of the line corresponding to $b_{u_j}^j$, and selecting a relatively vertical line somewhere below the line corresponding to $b_{u_j}^j$ (see Figure 8).
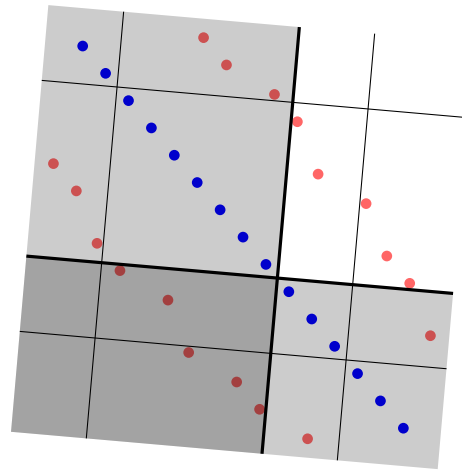

Figure 8: In bold, the horizontal and vertical lines in $\mathcal{G}(B)$ corresponding to selecting some element b_i^j of B. The grey regions materialize the potential positions for the slanted line in $\mathcal{G}_{\approx v}(\sigma_j)$ and the slanted line in $\mathcal{G}_{\approx h}(\sigma_j)$ once the lines corresponding to selecting a_i^j have been chosen.

Though, by Lemma 5, those two lines have to meet at the diagonal formed by the blue points. The only way to realize that is that both lines agree on the choice of $b_{u_j}^j$. This concludes to prove that choosing the lines corresponding to $a_{u_1}^1, a_{u_2}^2, \ldots, a_{u_k}^k$ to separate $\mathcal{G}(A)$ forces to select the lines corresponding to $b_{u_1}^1, b_{u_2}^2, \ldots, b_{u_k}^k$ to separate $\mathcal{G}(B)$. Finally, as we already observed for track A, the $b_{u_j}^j$'s have to be a hitting set of $(B, \mathcal{S}_B)$; otherwise, the non hit interval would induce some non separated red/blue pairs.

As $a_{u_1}^1, a_{u_2}^2, \ldots, a_{u_k}^k$ is a hitting set of $(A, \mathcal{S}_A)$ and $b_{u_1}^1, b_{u_2}^2, \ldots, b_{u_k}^k$ is a hitting set of $(B, \mathcal{S}_B)$, $\mathcal{I}$ is a YES-instance. $\square$

4   FPT algorithm parameterized by the size of the smaller set

In this section, we focus on AXIS-PARALLEL RED-BLUE SEPARATION. This version can be reduced to the problem of stabbing with axis-parallel lines the axis-parallel rectangles spanned by a point in $\mathcal{R}$ and a point in $\mathcal{B}$. In general, however, when parameterized by the number of lines in the solution, stabbing axis-parallel rectangles with axis-parallel lines is

W[1]-hard [13]. It is also known to be FPT when the rectangles are disjoint [15]. We have not been able to adapt the hardness proof to our special version of the problem nor to show it is FPT, so the parameterized complexity of AXIS-PARALLEL RED-BLUE SEPARATION w.r.t. the number of lines remains open.

Here, we present a simple FPT algorithm for AXIS-PARALLEL RED-BLUE SEPARA-TION parameterized by $\min\{|\mathcal{R}|, |\mathcal{B}|\}$. In the following, w.l.o.g., we assume that $\mathcal{B}$ is the smaller set. Arguably, this parameterization is somewhat less interesting from a practical point of view. Indeed, as the hardness reduction for the general problem in the previous section suggests, the difficult cases seem to be the ones where $\mathcal{R}$ and $\mathcal{B}$ are (more or less) balanced in size. Nevertheless, we believe that this positive result has its merits as it is non-trivial and provides a first step towards understanding the complexity of AXIS-PARALLEL RED-BLUE SEPARATION.

Theorem 6. *An optimal solution of* AXIS-PARALLEL RED-BLUE SEPARATION *can be computed in $O(n \log n + n|\mathcal{B}|9^{|\mathcal{B}|})$ time.*

We first give a high-level description of the algorithm. It begins by subdividing the plane into at most $|\mathcal{B}| + 1$ vertical strips, each consisting of the area "between" two horizontally successive blue points, and at most $|\mathcal{B}| + 1$ horizontal strips, each consisting of the area "between" two vertically successive blue points (see Figure 9(a)). Since each strip can contain only red points in its interior, an optimal solution uses at most two lines inside a single strip (Lemma 7(a)). We can therefore guess (by exhaustive enumeration) the number of lines used in each strip in an optimal solution. This gives a running time of roughly $9^{|\mathcal{B}|}$. A second observation is that if an optimal solution uses two lines in a strip, these can be placed as far away from each other as possible (Lemma 7(b)). To complete the solution we must decide where to place the lines in strips that contain only one line of an optimal solution. We consider every pair of blue and red points whose separation may depend on the exact placement of these lines. The key idea is that the separation of two such points can be expressed as a 2-CNF constraint. If the upcoming formal exposition seems a bit more complicated than this informal idea, it is because we have to deal with points sharing the same x- or y-coordinates.

We now proceed to a formal description of our algorithm, beginning with some definitions. For a point $p \in \mathbb{R}^2$, let $p(x)$ and $p(y)$ be its x-coordinate and y-coordinate, respectively. Also, let X, Y be the sets of x, y coordinates of the points in $\mathcal{B}$. In order to ease presentation later on, with a slight terminology abuse, we add $-\infty, +\infty$ to both X and Y. Let $X(i), Y(i)$ be the respective i-th elements of these sets in increasing order with $0 \leqslant i$, and let $k = |X| - 2$ and $l = |Y| - 2$; $k \leqslant |B|$ and $l \leqslant |B|$.

Definition 1. *The vertical strips are defined as $V_i = \{p \in \mathbb{R}^2 \mid X(i) \leqslant p(x) \leqslant X(i+1)\}$ for $i \in [0, k]$.*

Definition 2. *The horizontal strips are defined as $H_i = \{p \in \mathbb{R}^2 \mid Y(i) \leqslant p(y) \leqslant Y(i+1)\}$ for $i \in [0, l]$.*

The horizontal and vertical strips defined above essentially partition the plane into open monochromatic (red) or empty regions, while the boundaries of the strips may contain both red and blue points. As a result, we have the following properties of an optimal solution.
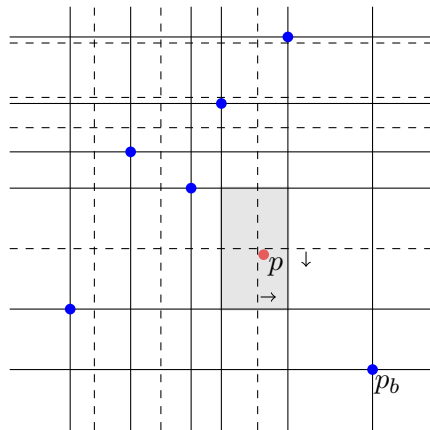
(a) The cell decomposition (solid lines), a guess of how S intersects it (dashed lines), and an interesting cell (in gray) for a point p_b (bottom-right corner). The red point p cannot be in the south-east quadrant of this cell which translates to the 2-clause $y_p^2 \vee \neg x_p^4$. Indeed, it should be that the horizontal line of S is below it or that the vertical line is to its right.

(b) Two consecutive red points in a horizontal strip $\mathcal{R}_h(i)$. If the corresponding line of S is below p, then it is also below p' which translates to $y_p^i \to y_{p'}^i$.

(c) Two consecutive red points in a vertical strip $\mathcal{R}_v(j)$. If the corresponding line of S is to the left of p, then it is also to the left of p' which translates to $x_p^i \to x_{p'}^i$.

Figure 9: Illustration of the algorithm and the two kinds of clauses of the 2-SAT instance.

Lemma 7. *(a) An optimal solution of* Axis-Parallel Red-Blue Separation *contains at most two lines in each horizontal or vertical strip. (b) In the case where a strip has two lines, these lines can be assumed to be placed in a way such that all red points in the interior of the strip lie between them.*

Proof. Recall that our notion of separation forbids lines from passing through input points. As the interior of every strip contains only red points, in any solution, every line that is between two other lines within a strip can be safely removed without affecting feasibility. Moreover, two lines within a strip can be translated in opposite directions towards the boundaries of the strip such that they enclose between them all red points that lie in the interior of the strip but no blue point. $\square$

We are now ready to give the proof of the main theorem of this section.

Proof of Theorem 6. We describe an FPT algorithm which guesses how many lines an optimal solution uses in each strip and then produces a 2-SAT instance of size $O(|\mathcal{B}|n)$ in order to check if its guess is feasible. We assume that we have access to two lists containing the input points sorted lexicographically by their (x, y) and (y, x) coordinates; producing these lists takes $O(n \log n)$ time.

Let S be some optimal solution. We first guess how many lines of S are in each horizontal and each vertical strip. Since, by Lemma 7, S contains at most two lines per strip, and there are $l + 1 \leqslant |\mathcal{B}| + 1$ horizontal strips and $k + 1 \leqslant |\mathcal{B}| + 1$ vertical strips, there are at most $3^{|\mathcal{B}|+1}$ possibilities to guess from for each direction thus, $O(9^{|\mathcal{B}|})$ in total.
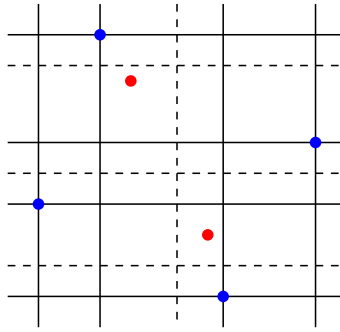
Figure 10: For the specification where the leftmost and rightmost vertical strips contain no lines while every other strip contains exactly one line, the line in the middle vertical strip must be placed between the two red points, otherwise the corresponding solution is not feasible.

In what follows, we assume that we have fixed how many lines of S are in each strip. We describe an algorithm deciding in polynomial time if such a specification gives a feasible solution. Since a specification fully determines the number of lines of a solution, the algorithm simply goes through all specifications and selects one with minimum cost among all feasible ones.

Note here that unlike in the case of strips that, according to a specification, contain exactly two lines (Lemma 7), for a strip that contains exactly one line, this line cannot always be placed in an 'extremal' (i.e., rightmost or leftmost) position within the strip; see Figure 10 for an example.

We now produce a 2-SAT instance which will be satisfiable if and only if a given specification is feasible. We first define the variables: for each horizontal strip H_i that contains exactly one line from S and for each red point $p \in H_i$, we define a variable y_p^i. Its informal meaning is "the line of S in H_i is below point p". Note that when p lies on the upper (lower) boundary of H_i, y_p^i is set to true (false) by default. Similarly, for each vertical strip V_j that contains exactly one line from S and for each red point $p \in V_j$, we define a variable x_p^j. Its informal meaning is "the line of S in V_j is to the left of p". It is set to true (false) by default when p lies on the right (left) boundary of V_j. We have constructed $O(n)$ variables (at most four for each point of $\mathcal{R}$).

Next, we construct 2-CNF clauses imposing the informal meaning described. For each strip H_i that contains exactly one horizontal line from S and each pair of red points $p, p' \in H_i$ that are consecutive in lexicographic (y, x) order, we add the clause $(y_p^i \to y_{p'}^i)$. Note that we can skip pairs that have a point lying on the upper or lower boundary of H_i as the corresponding variable has been already set to true or false respectively and the clause is satisfied; see the description in the previous paragraph. Similarly, for each strip V_j that contains exactly one vertical line from S and each pair of red points $p, p' \in V_j$ that are consecutive in lexicographic (x, y) order, we add the clause $(x_p^j \to x_{p'}^j)$; as before, pairs that have a point lying on the left or right boundary of V_j do not produce any clauses. Observe that given any solution, we can construct from its lines an assignment following the informal

meaning described above that satisfies all clauses added so far, while from any satisfying assignment we can find lines according to the informal meaning. We call the $O(n)$ clauses constructed so far the *coherence part* of our instance.

What remains is to add some further clauses to our instance to ensure not only that each satisfying assignment encodes a solution, but also that the solution is feasible, that is, it separates all pairs of red and blue points.

Consider a cell $C_{ij} = H_i \cap V_j$, where $i \in [0, l]$ and $j \in [0, k]$. A red point $p \in C_{ij}$ is called C_{ij}-*separable* for a point $p_b \in \mathcal{B}$, if p can be separated from p_b by a vertical or horizontal line running through the interior of C_{ij}. We will sometimes call p just separable when C_{ij} and p_b are obvious from the context. We say that C_{ij} is *interesting* for a point $p_b \in \mathcal{B}$ if the following conditions hold: (i) C_{ij} contains at least one red point that is C_{ij}-separable for p_b; (ii) at least one of H_i or V_j contains at most one horizontal or one vertical line from S respectively; (iii) if $X(j+1) < p_b(x)$ or $p_b(x) < X(j)$, then there is no vertical line from S in a strip between p_b and V_j; and (iv) if $Y(i+1) < p_b(y)$ or $p_b(y) < Y(i)$, then there is no horizontal line from S in a strip between p_b and H_i. Note that even if C_{ij} is interesting for p_b, it may contain a red point p that is *already separated* from p_b by a line going through C_{ij}: this happens exactly when H_i or V_j contains two horizontal or vertical lines from S respectively and p lies either in the interior of C_{ij} or on its boundary but not on the same side of H_i or V_j as p_b.

The motivation behind these definitions is that the cells that are interesting for p_b contain exactly the red points that need to be separated from p_b by lines going through the cells and whose positions cannot be predetermined. We therefore have to add some clauses to express these constraints.

For each $p_b \in \mathcal{B}$ and each cell C_{ij} that is interesting for p_b we construct a clause for every red point $p \in C_{ij}$ that is separable and not already separated from p_b. Initially, the clause is empty. If the specification says that there is exactly one line from S in H_i, we add to the clause a literal as follows: if $y(p_b) \geqslant Y(i+1)$, we add $\neg y_p^i$ (meaning that the horizontal line is above p, and hence separates p from p_b); if $y(p_b) \leqslant Y(i)$, we add y_p^i. Furthermore, if the specification says that there is exactly one line from S in V_j, we add to the clause a literal as follows: if $x(p_b) \geqslant X(i+1)$, we add the literal $\neg x_p^j$; if $x(p_b) \leqslant X(i)$, we add x_p^j. Observe that this process produces clauses of size at most two. It may produce an empty clause, rendering the 2-SAT unsatisfiable, in the case where there is no line of S in H_i or V_j, but this is desirable since in this case no feasible solution matches the specification. A point $p \in \mathcal{R}$ can belong to (at most) four cells C_{ij} but it can be C_{ij}-separable only for (at most) three of the cells for any point $p_b \in \mathcal{B}$. Thus, we have at most three clauses for each pair of a blue and a red point. In total, we have constructed $O(|\mathcal{B}||\mathcal{R}|)$ clauses in this way and the 2-SAT formula has $O(n)$ variables and $O(|\mathcal{B}|n)$ clauses. Since 2-SAT can be solved in linear time, we obtain the promised running time for our algorithm.

To complete the proof we rely on the informal correspondence between assignments to the 2-SAT instance and AXIS-PARALLEL RED-BLUE SEPARATION solutions. In particular, if there exists a solution that agrees with the guessed specification, this solution can easily be translated to an assignment that satisfies the coherence part of the 2-SAT formula. Furthermore, for any blue point p_b and any separable and not already separated red point

p in a cell C_{ij} that is interesting for p_b, the solution must be placing at least one line going through C_{ij} in a way that separates p_b from p (this follows from the fact that the cell is interesting). Hence, the corresponding 2-SAT clauses are also satisfied. Conversely, given an assignment to the 2-SAT instance, we construct an AXIS-PARALLEL RED-BLUE SEPARATION solution following the informal meaning of the variables. We first note that for every blue point p_b, every red point is C_{ij}-separable for p_b for at least one cell C_{ij}. Observe that for any cell C_{ij} that is not interesting for p_b and contains at least one separable point, we have that either all red points in the cell are separated from p_b by lines outside the cell or all separable red points in the cell are separated from p_b by the four lines running through the cell. Furthermore, if C_{ij} is interesting for p_b, then all separable (and not already separated) red points in the cell are separated from p_b because of the additional 2-SAT clauses we added in the second part of the construction.   □

5   Open problems

The most intriguing open problem is settling the complexity of AXIS-PARALLEL RED-BLUE SEPARATION w.r.t. the number of lines. We conjecture it to be FPT. Other problems include the complexity of RED-BLUE SEPARATION when the lines can have three different slopes and of AXIS-PARALLEL RED-BLUE SEPARATION in 3-dimensions.

References

[1] E. M. Arkin, F. Hurtado, J. S. B. Mitchell, C. Seara, and S. Skiena. Some lower bounds on geometric separability problems. *Int. J. Comput. Geometry Appl.*, 16(1):1–26, 2006. URL: https://doi.org/10.1142/S0218195906001902, doi:10.1142/S0218195906001902.

[2] D. K. Blandford, G. E. Blelloch, D. E. Cardoze, and C. Kadow. Compact representations of simplicial meshes in two and three dimensions. *Int. J. Comput. Geometry Appl.*, 15(1):3–24, 2005. URL: https://doi.org/10.1142/S0218195905001580, doi:10.1142/S0218195905001580.

[3] É. Bonnet, P. Giannopoulos, and M. Lampis. On the parameterized complexity of red-blue points separation. In D. Lokshtanov and N. Nishimura, editors, *12th International Symposium on Parameterized and Exact Computation, IPEC 2017, September 6-8, 2017, Vienna, Austria*, volume 89 of *LIPIcs*, pages 8:1–8:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. URL: https://doi.org/10.4230/LIPIcs.IPEC.2017.8, doi:10.4230/LIPIcs.IPEC.2017.8.

[4] Édouard Bonnet and Tillmann Miltzow. Parameterized hardness of art gallery problems. In Piotr Sankowski and Christos D. Zaroliagis, editors, *24th Annual European*

Symposium on Algorithms, ESA 2016, August 22-24, 2016, Aarhus, Denmark, volume 57 of *LIPIcs*, pages 19:1–19:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. URL: https://doi.org/10.4230/LIPIcs.ESA.2016.19, doi:10.4230/LIPIcs.ESA.2016.19.

[5] G. Calinescu, A. Dumitrescu, H.J. Karloff, and P. Wan. Separating points by axis-parallel lines. *Int. J. Comput. Geometry Appl.*, 15(6):575–590, 2005. URL: https://doi.org/10.1142/S0218195905001865, doi:10.1142/S0218195905001865.

[6] T. M. Chan. Polynomial-time approximation schemes for packing and piercing fat objects. *J. Algorithms*, 46(2):178–189, February 2003. URL: http://dx.doi.org/10.1016/S0196-6774(02)00294-8, doi:10.1016/S0196-6774(02)00294-8.

[7] B. S. Chlebus and S. H. Nguyen. On finding optimal discretizations for two attributes. In L. Polkowski and A. Skowron, editors, *Rough Sets and Current Trends in Computing, First International Conference, RSCTC'98, Warsaw, Poland, June 22-26, 1998, Proceedings*, volume 1424 of *Lecture Notes in Computer Science*, pages 537–544. Springer, 1998. URL: https://doi.org/10.1007/3-540-69115-4_74, doi:10.1007/3-540-69115-4_74.

[8] H. Edelsbrunner and F. P. Preparata. Minimum polygonal separation. *Inf. Comput.*, 77(3):218–232, 1988. URL: https://doi.org/10.1016/0890-5401(88)90049-1, doi:10.1016/0890-5401(88)90049-1.

[9] U.M. Fayyad and K.B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. of 13th Int. Joint Conf. on Artificial Intelligence*, pages 1022–1029, 1993. URL: https://www.ijcai.org/Proceedings/93-2/Papers/022.pdf.

[10] S Fekete. On the complexity of min-link red-blue separation. *Manuscript, Department of Applied Mathematics, SUNY Stony Brook, NY*, 1992.

[11] R. Freimer, J.S.B. Mitchell, and C.D. Piatko. On the complexity of shattering using arrangements. TR 91-1197, Dept. Comput. Sci., Cornell Univ., NY, 1991. URL: https://hdl.handle.net/1813/7037.

[12] B. Fu and W. Wang. Geometric separators and their applications to protein folding in the HP-model. *SIAM J. Comput.*, 37(4):1014–1029, 2007. URL: https://doi.org/10.1137/S0097539704440727, doi:10.1137/S0097539704440727.

[13] P. Giannopoulos, C. Knauer, G. Rote, and D. Werner. Fixed-parameter tractability and lower bounds for stabbing problems. *Comput. Geom.*, 46(7):839–860, 2013. URL: https://doi.org/10.1016/j.comgeo.2011.06.005, doi:10.1016/j.comgeo.2011.06.005.

[14] S. Har-Peled and M. Jones. On separating points by lines. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '18, pages 918–932, Philadelphia, PA, USA, 2018. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=3174304.3175329.

[15] P. Heggernes, D. Kratsch, D. Lokshtanov, V. Raman, and S. Saurabh. Fixed-parameter algorithms for cochromatic number and disjoint rectangle stabbing via iterative localization. *Inf. Comput.*, 231:109–116, 2013. URL: http://dx.doi.org/10.1016/j.ic.2013.08.007, doi:10.1016/j.ic.2013.08.007.

[16] F. Hurtado, M. Mora, P.A. Ramos, and C. Seara. Separability by two lines and by nearly straight polygonal chains. *Disc. Appl. Mathematics*, 144(1-2):110–122, 2004. URL: https://doi.org/10.1016/j.dam.2003.11.014, doi:10.1016/j.dam.2003.11.014.

[17] F. Hurtado, M. Noy, P. A. Ramos, and C. Seara. Separating objects in the plane by wedges and strips. *Discrete Applied Mathematics*, 109(1-2):109–138, 2001. URL: https://doi.org/10.1016/S0166-218X(00)00230-4, doi:10.1016/S0166-218X(00)00230-4.

[18] R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367 – 375, 2001. URL: http://www.sciencedirect.com/science/article/pii/S0022000000917276, doi:https://doi.org/10.1006/jcss.2000.1727.

[19] J. Kujala and T. Elomaa. Improved algorithms for univariate discretization of continuous features. In J. N. Kok, J. Koronacki, R. L. de Mantaras, S. Matwin, D. Mladenič, and A. Skowron, editors, *11th European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD 2007*, volume 4702 of *LNCS*, pages 188–199. Springer, 2007. URL: https://doi.org/10.1007/978-3-540-74976-9_20.

[20] C.-Y. Lo, J. Matoušek, and W. L. Steiger. Algorithms for ham-sandwich cuts. *Discrete & Computational Geometry*, 11:433–452, 1994. URL: https://doi.org/10.1007/BF02574017, doi:10.1007/BF02574017.

[21] B. Lu, H. Du, X. Jia, Y. Xu, and B. Zhu. On a minimum linear classification problem. *J. of Global Optimization*, 35(1):103–109, 2006. URL: https://doi.org/10.1023/A:1024779931668.

[22] D. Marx. Parameterized complexity of independence and domination on geometric graphs. In H. L. Bodlaender and M. A. Langston, editors, *Parameterized and Exact Computation, Second International Workshop, IWPEC 2006, Zürich, Switzerland, September 13-15, 2006, Proceedings*, volume 4169 of *LNCS*, pages 154–165. Springer, 2006. URL: https://doi.org/10.1007/11847250_14, doi:10.1007/11847250_14.

[23] D. Marx. Can you beat treewidth? *Theory of Computing*, 6(1):85–112, 2010. URL: http://dx.doi.org/10.4086/toc.2010.v006a005, doi:10.4086/toc.2010.v006a005.

[24] D. Marx and M. Pilipczuk. Optimal parameterized algorithms for planar facility location problems using Voronoi diagrams. In N. Bansal and I. Finocchi, editors, *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, volume 9294 of *LNCS*, pages 865–877. Springer, 2015. URL: https://doi.org/10.1007/978-3-662-48350-3_72, doi:10.1007/978-3-662-48350-3_72.

[25] J. Matoušek. *Lectures on Discrete Geometry*. Springer-Verlag, New York, 2002. `doi:10.1007/978-1-4613-0039-7`.

[26] N. Megiddo. Linear-time algorithms for linear programming in R^3 and related problems. *SIAM J. Comput.*, 12(4):759–776, 1983. URL: `https://doi.org/10.1137/0212052`, `doi:10.1137/0212052`.

[27] N. Megiddo. On the complexity of polyhedral separability. *Discr. & Comput. Geom*, 3:325–337, 1988. URL: `https://doi.org/10.1007/BF02187916`.

[28] G. L. Miller, S. Teng, W. Thurston, and S. A. Vavasis. Automatic mesh partitioning. In Alan George, John Gilbert, and Joseph Liu, editors, *Graphs Theory and Sparse Matrix Computation*, The IMA Volumes in Mathematics and its Application, pages 57–84. Springer-Verlag, 1993. Vol 56. URL: `https://doi.org/10.1007/978-1-4613-8369-7_3`.

[29] G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis. Separators for sphere-packings and nearest neighbor graphs. *J. ACM*, 44(1):1–29, January 1997. URL: `http://doi.acm.org/10.1145/256292.256294`, `doi:10.1145/256292.256294`.

[30] G. L. Miller and W. Thurston. Separators in two and three dimensions. In *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing*, STOC '90, pages 300–309, New York, NY, USA, 1990. ACM. URL: `http://doi.acm.org/10.1145/100216.100255`, `doi:10.1145/100216.100255`.

[31] J. S. B. Mitchell. Approximation algorithms for geometric separation problems. Technical report, State University of New York at Stony Brook, 1993. URL: `http://www.ams.sunysb.edu/~jsbm/papers/sep-2-10-94.pdf`.

[32] H. S. Nguyen. Approximate boolean reasoning: Foundations and applications in data mining. In James F. Peters and Andrzej Skowron, editors, *Transactions on Rough Sets V*, pages 334–506. Springer-Verlag, Berlin, Heidelberg, 2006. URL: `http://dl.acm.org/citation.cfm?id=2167843.2167862`.

[33] H. S. Nguyen and A. Skowron. Quantization of real value attributes - rough set and boolean reasoning approach. In *Proc. of the Second Joint Annual Conference on Information Sciences, Wrightsville Beach, North Carolina, Sept 28 - Oct 1*, pages 34–37, 1995.

[34] J. O'rourke, S. Rao Kosaraju, and N. Megiddo. Computing circular separability. *Discrete Comput. Geom.*, 1(1):105–113, December 1986. URL: `https://doi.org/10.1007/BF02187688`, `doi:10.1007/BF02187688`.

[35] F. Sheikhi, A. Mohades, M. de Berg, and M. Davoodi. Separating bichromatic point sets by L-shapes. *Comput. Geom.*, 48(9):673–687, 2015. URL: `https://doi.org/10.1016/j.comgeo.2015.06.008`, `doi:10.1016/j.comgeo.2015.06.008`.

[36] W. D. Smith and N. C. Wormald. Geometric separator theorems and applications. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*,

FOCS '98, pages 232–, Washington, DC, USA, 1998. IEEE Computer Society. URL: http://dl.acm.org/citation.cfm?id=795664.796397.

[37] M. van Kreveld, T. van Lankveld, and R. Veltkamp. Identifying well-covered minimal bounding rectangles in 2d point data. In *Abstracts of the 25th Europ. Workshop Comput. Geom.*, pages 277–280, 2009.