On the Parameterized Complexity of Red-Blue Points Separation

Édouard Bonnet[1], Panos Giannopoulos[1], and Michael Lampis[2]

1 **Middlesex University, Department of Computer Science, London, UK**
edouard.bonnet@dauphine.fr, p.giannopoulos@mdx.ac.uk
2 **Université Paris-Dauphine, PSL Research University, CNRS, LAMSADE, Paris, France** michail.lampis@dauphine.fr

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 smallest set).

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases red-blue points separation, geometric problem, W[1]-hardness, FPT algorithm, ETH-based lower bound

Digital Object Identifier 10.4230/LIPIcs...

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 the 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.

Apart from being interesting in its own right, Red-Blue Separation is also directly motivated by the problem of univariate discretization of continuous variables in the context of machine learning [4, 9]. For example, its two-dimensional version models problem instances with decision tables of two real-valued attributes and a binary decision function. The lines to be found represent cut points determining a partition of the values into intervals and one opts for a minimum-size set of cuts that is consistent with the given decision table. For the case where $k = 1$ and $k = 2$, Red-Blue Separation is solvable in $O(n)$ and $O(n \log n)$ time respectively [7]. When k is part of the input, it is known to be NP-hard [13] and APX-hard [2] even for the axis-parallel variant. The latter also admits a 2-approximation [2].

Results. 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 2, which has been recently used for showing hardness for another classical geometric optimization problem [1]. Then, in Section 3, 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 smallest set).

Related work. The following 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 [5], APX-hard [2] and, in the axis-parallel case, to admit a 2-approximation [2]. Very recently, the problem has been also shown to admit an $\mathrm{OPT}\log\mathrm{OPT}$-approximation [6]. 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 $\Theta(k^2)$. For results on several other related separation problems see [3, 7].

2 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. We reduce from Structured 2-Track Hitting Set [1], see below; more about this problem and its lower bound can be found in the appendix.

For a positive integer x, let $[x]$ be the set of integers between 1 and x, and $[x, y]$ the set of integers between x and y. If X is a totally ordered (finite) set, we call X*-interval* 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 color classes over the 2-elements and a restriction on the one-to-one mapping ϕ. 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 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: $b^j_{\sigma_j(1)}, b^j_{\sigma_j(2)}, \ldots, b^j_{\sigma_j(t)}$. 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. $b^j_i \leq_B b \leq_B b^{j'}_{i'}$).

If one deconstructs 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 (see Figure 3), thanks

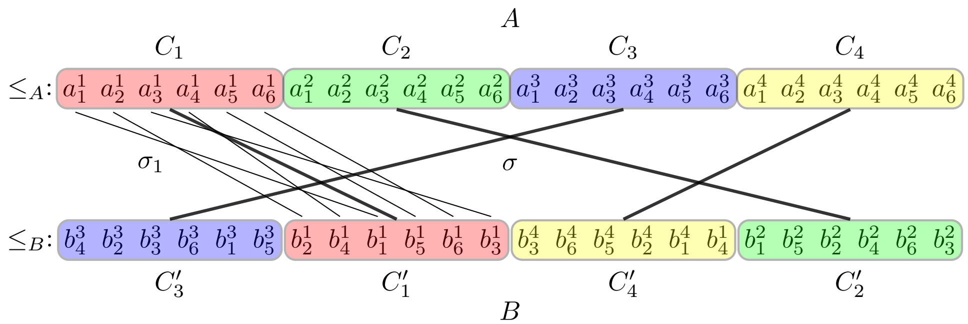


Figure 1 An illustration of the $k+1$ permutations $\sigma \in \mathfrak{S}_k$, $\sigma_1 \in \mathfrak{S}_t$, $\ldots$, $\sigma_k \in \mathfrak{S}_t$ of an instance of Structured 2-Track Hitting Set, with $k = 4$ and $t = 6$.

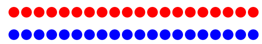
to their geometric nature, can be realized by two red points which have to be separated from a diagonal of blue points, while permutation σ (see Figure 4), being on k elements, can be designed straightforwardly without blowing-up the size of the solution. For these gadgets, we would like to force the chosen lines to be axis-parallel. We obtain that by surrounding the gadgets by *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 $\sigma_j(i)$ (see Figure 5). Our gadget actually only links the element of index i to elements of indices greater than $\sigma_j(i)$. We thus combine two of these gadgets to obtain the other inequality (see Figure 6).

▶ **Theorem 1.** 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.*

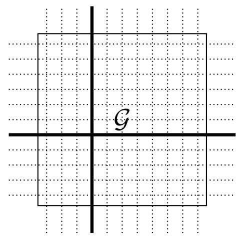
Proof. We reduce from S2-THS which is $W[1]$-hard and has the above lower bound under ETH [1]. Let $\mathcal{I} = (k \in \mathbb{N}, t \in \mathbb{N}, \sigma \in \mathfrak{S}_k, \sigma_1 \in \mathfrak{S}_t, \ldots, \sigma_k \in \mathfrak{S}_t, \mathcal{S}_A, \mathcal{S}_B)$ be an instance of S2-THS. We will build an instance $\mathcal{J} = (\mathcal{R}, \mathcal{B})$ 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* 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 $a < b$). We denote those points by $\mathcal{C}(a \to b, y)$ (resp. $\mathcal{C}(x, a \to b)$).

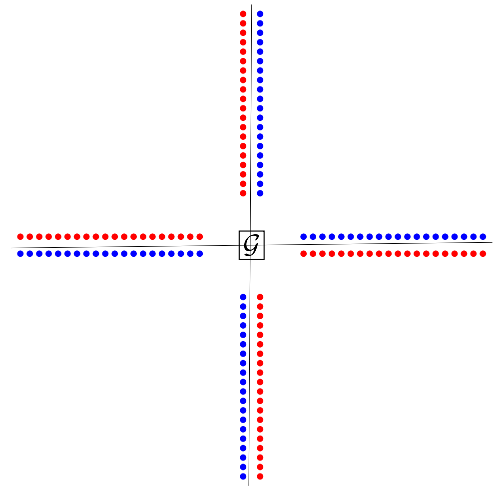
Long alley gadgets. In the gadgets encoding the intervals (see next paragraph), we will need to restrict the selected separating lines to be almost horizontal or almost vertical. To enforce that, we use the *long alley* gadgets. A *horizontal long alley* gadget is made of ℓ horizontally consecutive red points $\mathcal{C}(a \to a + \ell - 1, y)$ and ℓ horizontally consecutive blue points $\mathcal{C}(a \to a + \ell - 1, y')$ with $a, a + \ell - 1, y \neq y' \in [z]$ (see Figure 2a). A *vertical long alley* is defined analogously. Long alleys are called so because if $\ell \gg |y - y'|$, then separating the red points from the blue points of a horizontal (resp. vertical) long alley with a budget of only one line, will require this 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 subgrid of dimension g-by-g (with $g \ll z$). We place four long alley gadgets to the left, top, right, and bottom of $\mathcal{G}$: horizontal ones to



(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.

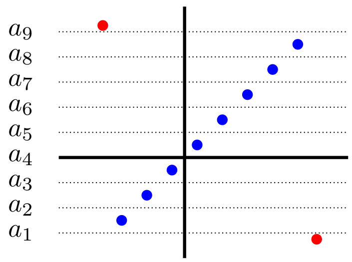
the left and right, vertical ones to the top and bottom (as depicted in Figure 2c). The left horizontal (resp. bottom vertical) long alley starts at the x-coordinate (resp. y coordinate) of 1, whereas the right horizontal (resp. top vertical) long alley ends at the x-coordinate (resp. y coordinate) of z; see Figure 6, 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.

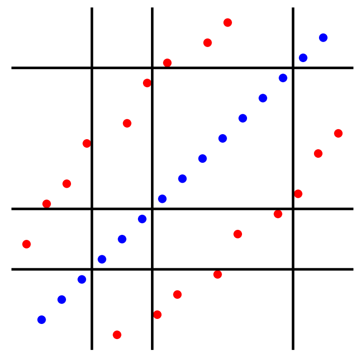
Interval gadgets and encoding track A. Elements of A are represented by a diagonal of $kt-1$ blue points. More precisely, we add $(x_0^A, y_0^A), (x_0^A+4, y_0^A+4), (x_0^A+8, y_0^A+8), \ldots, (x_0^A+4kt-8, y_0^A+4kt-8)$ to $\mathcal{B}$ for some offset $x_0^A, y_0^A \in [z]$ that we will specify later. We think those points as going from the first (x_0^A, y_0^A) to the last $(x_0^A+4kt-8, y_0^A+4kt-8)$. 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 $\leq_A$. The almost horizontal (resp. vertical) line just above (resp. just to the right of) the last blue point corresponds to selecting the kt-th, i.e., last, element of A.

For each interval $[a_i^j, a_{i'}^{j'}]$ in $\mathcal{S}_A$ (for some $i, i' \in [k]$, $j, j' \in [t]$), 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 $(x_0^A+4s-7, y_0^A+4s'-5)$ and one at $(x_0^A+4s'-5, y_0^A+4s-7)$ (see Figure 3a for one interval gadget and Figure 3b for track A). Let $R([a_i^j, a_{i'}^{j'}])$ be this pair of red points. Informally, one red point has its projection along the x-axis just to the left of the s-th blue point and its projection along the y-axis just above the s'-th blue point; the other one has its projection along the x-axis just to the right of the s'-th blue point and its projection along 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.

We surround this encoding of track A, which we denote by $\mathcal{G}(A)$, with $4k$ long alleys, whose width is $4t-4$, from x-coordinates $x_0^A+4(k-1)t-2$ to $x_0^A+4kt-6$ for horizontal alleys (from y-coordinates $y_0^A+4(k-1)t-2$ to $y_0^A+4kt-6$ for vertical alleys). We alternate



(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.

red-blue[1] alleys and blue-red alleys for two contiguous alleys so that there is no need to separate one from the other. 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 consistent with the choices of Figure 2c. This, together with the description of long alleys in the previous paragraph, fully defines the $4k$ long alleys (see Figure 6).

The general intention is that in order to separate those two red points 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 $x = x_0^A + 4\hat{s} - 6, y = y_0^A + 4\hat{s} - 6$ for $\hat{s} \in [s, s']$. 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 $y = y_0^A + 4s - 6$ and $\mathrm{VL}(s)$ the vertical line of equation $x = x_0^A + 4s - 6$. 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 its right as depicted in Figure 6. 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

[1] i.e., for horizontal (resp. vertical) alleys, the red points are above (resp. to the left of) the blue points.
[2] we use this term informally to avoid confusion with what we have been calling *grids* so far.

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 $\mathrm{Cell}(j, \sigma(j))$ a diagonal of $t-1$ blue points and two red points corresponding to the full interval $[a^j_1, a^j_t]$ (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 results in a vertical propagation of the initial choices accompanied by the desired reordering of the *color classes*. The vertical line matching the choice of $\mathrm{HL}(s)$ 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.

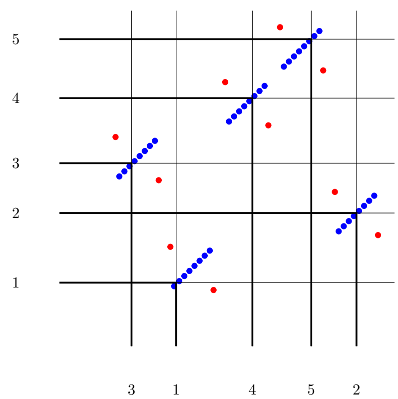


Figure 4 Encoding permutation $\sigma = 31452$. The choices within the five color classes are transferred from almost horizontal lines to almost vertical ones. This way, we obtain the desired reordering of the color classes.

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 denote by $\hat{v}$ the distance along the y-axis between $\mathcal{G}(\sigma)$ and $\mathcal{G}(B)$. Eventually $\hat{v}$ will be chosen 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 6. 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 points p_i^j corresponding to the intersection of a 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 (see Figure 5). 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 5 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 symmetry $\mathcal{G}_{\approx h}(\sigma_j)$ of gadget $\mathcal{G}_{\approx v}(\sigma_j)$ 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.

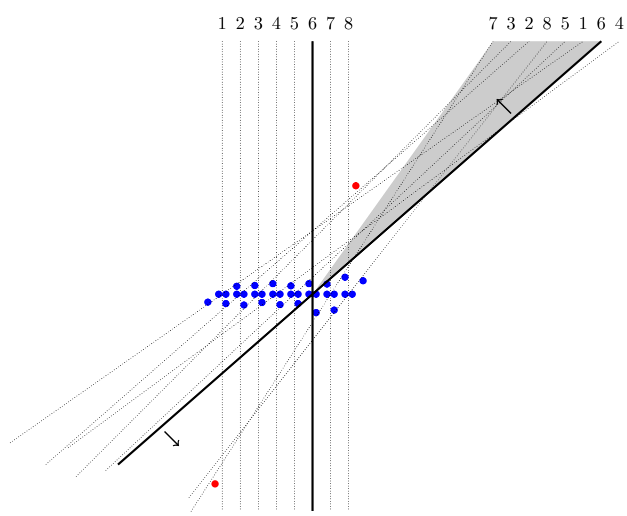


Figure 5 Half-encoding of the permutation $\sigma_j = 73285164$ of the j-th color class. Observe that the choice of the, say, sixth almost horizontal candidate line only forces to take the slanted line depicted in bold or a line having the same intersection with the almost horizontal line but a larger slope. For the sake of legibility, the angles between the vertical lines and the slanted lines are exaggerated.

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

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^j_i) 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 Γ. We set the length ℓ of the long alleys to $100(k^2 + 1)$. Point (x^A_0, y^A_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 6. 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 $4t - 4$, 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 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. The correctness of the reduction is deferred to the appendix. ◀

3 FPT Algorithm Parameterized by Size of Smaller Set

We now present an FPT algorithm for Axis-Parallel Red-Blue Separation parameterized by $\min\{|\mathcal{A}|, |\mathcal{B}|\}$. In the following, w.l.o.g., we assume that $\mathcal{B}$ is the smaller set.

▶ **Theorem 2.** *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 partitioning the plane into $|\mathcal{B}| + 1$ vertical strips, each consisting of the area between two horizontally consecutive blue points, and $|\mathcal{B}| + 1$ horizontal strips, each consisting of the area between two vertically consecutive blue points (see Figure 7a). Since all strips contain only red points, the optimal solution uses at most two lines contained in a single strip (Lemma 4). We therefore guess (by

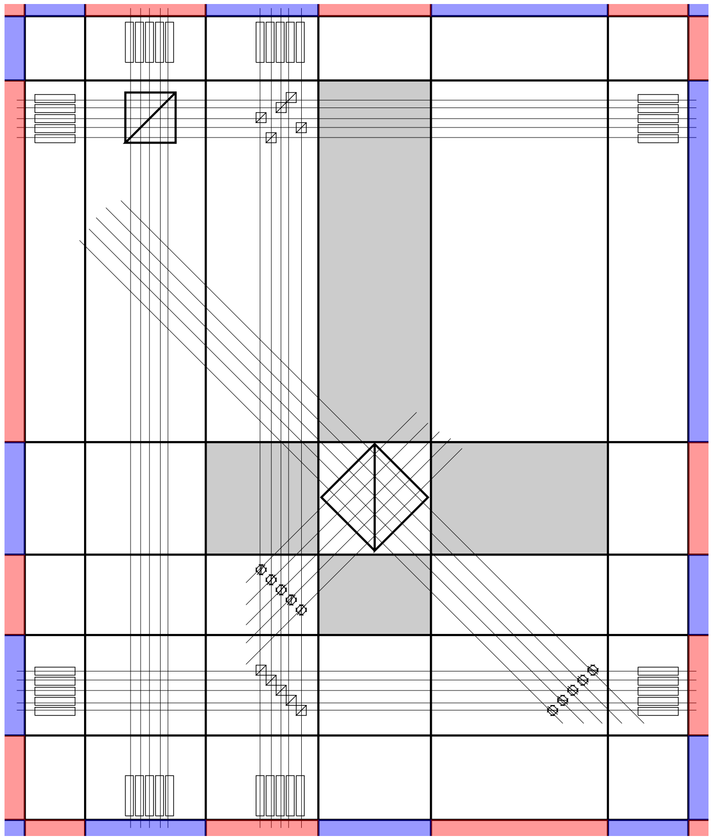


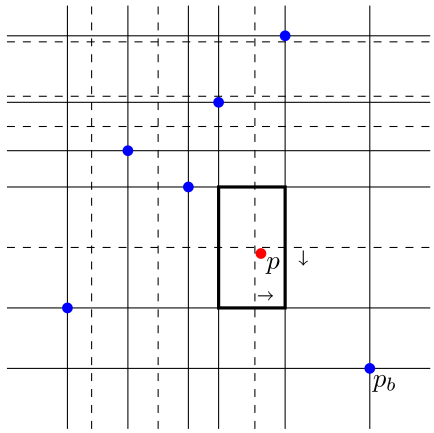
Figure 6 The overall picture. The thin rectangles are long alleys, the bold large squares with a diagonal are the encoding of track A, in the top left corner, and track B, slanted by 45 degrees (for the sake of fitting the whole construction on one page; in reality the encoding of B is only inclined 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.

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 the optimal solution uses two lines in a strip, these can be placed as far away from each other as possible (Lemma 5).

To complete the solution we must decide where to place the lines contained in strips that contain a single line of the 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, because the red point, if it is not already separated by the blue point, must belong in vertical and horizontal strips that contain at most one line from the optimal solution. Hence, there are at most two ways in which the placement of the lines can separate this pair. The last step of our algorithm is to produce a 2-SAT instance which is satisfiable if and only if there exists a placement of exactly one line in each remaining strip that produces a feasible solution.

We now proceed to a more formal description of our algorithm. We begin with some definitions. To ease notation, we set $b := |\mathcal{B}|$.

▶ **Definition 3.** For $i \in [0, b]$ we define the i-th horizontal strip of $\mathcal{R}$, denoted $\mathcal{R}_h(i)$ as



(a) The cell decomposition (solid lines), a guess of how S intersects it (dashed lines), and an interesting $(4, 2)$-cell (in bold) 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 7 Illustration of the algorithm and the two kinds of clauses of the 2-SAT instance.

follows: $p \in \mathcal{R}_h(i)$ if and only if there exist exactly i points of $\mathcal{B}$ with y-coordinate smaller than that of p. For $j \in [0, b]$ we define the j-th vertical strip of $\mathcal{R}$, denoted $\mathcal{R}_v(j)$ as follows: $p \in \mathcal{R}_v(j)$ if and only if there exist exactly j points of $\mathcal{B}$ with x-coordinate smaller than that of p.

The horizontal and vertical strips defined above essentially partition the plane into monochromatic regions. Thus, an optimal solution of Axis-Parallel Red-Blue Separation will place at most two lines in each strip. In the lemmas below, a vertical line $x = x_0$ has a point p to the left of it if p has an x coordinate smaller than x_0. Similarly, a horizontal line $y = y_0$ has p below it if p has y-coordinate smaller than y_0.

▶ **Lemma 4.** *Let S be an optimal solution to an instance of* Axis-Parallel Red-Blue Separation*. Then for any $i \in [0, b]$ there exists at most two vertical lines in S which have exactly i points of $\mathcal{B}$ to the left of them. Furthermore, there exist at most two horizontal lines in S with exactly i points of $\mathcal{B}$ below them.*

▶ **Lemma 5.** *For any instance of* Axis-Parallel Red-Blue Separation *there exists an optimal solution S with the following property: for any $i \in [0, b]$ if there exist two lines $\ell_1, \ell_2 \in S$ with exactly i points of $\mathcal{B}$ below them, then all the points of $\mathcal{R}_h(i)$ are between ℓ_1, ℓ_2. Furthermore, for all $i \in [0, b]$ if there exist two lines $\ell_1, \ell_2 \in S$ with exactly j points of $\mathcal{B}$ to the left of them, then all the points of $\mathcal{R}_v(j)$ are between ℓ_1, ℓ_2.*

Proof of Theorem 2. We describe an FPT algorithm which guesses how many lines the 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 points sorted by x or y coordinate (producing these lists takes time $O(n \log n)$).

Fix some optimal solution S that the algorithm will try to find. We first guess, for each $i \in [0, b]$, how many horizontal lines of S have exactly i blue points below them. By Lemma

4 there are 3^{b+1} possibilities, since S contains 0, 1, or 2 lines with i blue points below them. Similarly, for each $j \in [0, b]$ we guess how many vertical lines of S have exactly j blue points to their left. We thus consider a total of $(3^{b+1})^2 = 9^{b+2}$ possibilities.

In what follows, we assume that for each $i, j \in [0, b]$ we have fixed how many lines of S have i blue points below them and j blue points to their left. We describe an algorithm deciding in polynomial time if these specifications are feasible. Since these specification fully determine the number of lines of a solution, our algorithm simply goes through all specifications and selects among all feasible ones the one with minimum cost.

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 $i \in [0, b]$ such that we have decided S contains exactly one horizontal line with exactly i blue points below it, and for each $p \in \mathcal{R}_h(i)$ we define a variable y_p^i. Its informal meaning is "the line of S that has i blue points below it is below point p". Similarly, for each $j \in [0, b]$ such that S contains exactly one vertical line with exactly j blue points to the left of it, and for each $p \in \mathcal{R}_v(j)$ we define a variable x_p^j. Its informal meaning is "the line of S that has j blue points to its left is to the left of p". We have constructed $O(n)$ variables (at most two for each point of $\mathcal{R}$).

We now construct 2-CNF clauses imposing the informal meaning described. For each $i \in [0, b]$ such that S contains exactly one horizontal line with exactly i blue points below it, and for each $p, p' \in \mathcal{R}_h(i)$ such that p is below p', and there is no point whose y coordinate is between those of p, p', we add the clause $(y_p^i \to y_{p'}^i)$. We note that this can be done in time $O(n)$ by traversing the list that contains all points sorted by y coordinate, since p, p' are consecutive in this list. Similarly, for each $j \in [0, b]$ such that S contains exactly one vertical line with exactly j blue points to its left, and for each $p, p' \in \mathcal{R}_v(j)$ such that p is to the left of p' and no point has x coordinate between those of p, p', we add the clause $(x_p^j \to x_{p'}^j)$. Observe that from any (vertical or horizontal) line in a solution we can construct an assignment following the informal meaning described above satisfying 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. Before we proceed, let us give some helpful definitions. For $i, j \in [0, b]$ we will call the set $\mathcal{R}_h(i) \cap \mathcal{R}_v(j)$ the (i, j)-cell of the instance.

Consider a point $p_b \in \mathcal{B}$ that has exactly $j - 1$ blue points to its left and $i - 1$ blue points below it (i.e. that i-th blue point in terms of y-coordinate and the j-th blue point in terms of x-coordinate). We say that the (i', j')-cell is *interesting* for point p_b if the following hold: (i) S contains at most 1 horizontal line with exactly i' blue points below it and at most 1 vertical line with exactly j' blue points to its left; (ii) for all i'' such that $\min\{i, i'\} < i'' < \max\{i, i'\}$, S contains no horizontal lines with exactly i'' blue points below them, and for all j'' such that $\min\{j, j'\} < j'' < \max\{j, j'\}$, S contains no vertical lines with exactly j'' blue points to their left. The motivation for this definition is that if one of the two conditions is violated (either the cell contains two horizontal or two vertical lines, or there exists a strip strictly between p_b and the cell that contains a line) then from Lemma 5 and the high-level specification of the solution, p_b will be separated from all red points of the cell. Otherwise the cell is interesting, i.e., it is not obvious that its red points are separated from p_b, and we therefore have to add some clauses to express this constraint.

For each $p_b \in \mathcal{B}$ as in the previous paragraph and each (i', j')-cell that is interesting for p_b we construct a clause for every (red) point of the cell. Initially, the clause is empty. If the specifications say that there is exactly one line in S with exactly i' blue points below it then we add to the clause a literal as follows: if $i > i'$ (that is, if p_b is above all the points of the (i', j')-cell) we add the literal $\neg y_p^{i'}$ (meaning that the horizontal line is above p, and hence separates p from p_b); if $i \leqslant i'$ we add the literal $y_p^{i'}$. Furthermore, if the specifications say that there is exactly one line in S with exactly j' blue points to its left we add to the clause a literal as follows: if $j > j'$ (that is, if p_b is to the right of all the points of the (i', j')-cell) we add the literal $\neg x_p^{j'}$; if $j \leqslant j'$ we add the literal $x_p^{j'}$. 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 running through the (i', j')-cell, but this is desirable since in this case no feasible solution matches the specifications. Note that we have constructed $O(|\mathcal{B}||\mathcal{R}|)$ clauses in this way (at most one for each pair of a blue with a red point). Hence, the 2-SAT formula we have constructed 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.

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 red point p in an (i', j')-cell that is interesting for p_b the solution must be placing at least one of the lines going through the (i', j')-cell 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. For the converse direction, given an assignment to the 2-SAT instance we construct an Axis-Parallel Red-Blue Separation solution following the informal meaning of the variables and observe that every blue point p_b is immediately separated from every red point that belongs in a cell that is not interesting for p_b. Furthermore, p_b is separated from red points in cells that are interesting for it because of the additional 2-SAT clauses we added in the second part of the construction. ◀

4 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.

Acknowledgements. The authors would like to thank Sergio Cabello and Christian Knauer for fruitful discussions.

References

1 E. Bonnet and T. Miltzow. Parameterized hardness of art gallery problems. In *Proc. of 24th ESA 2016*, pages 19:1–19:17, 2016.
2 G. Calinescu, A. Dumitrescu, H. Karloff, and P. Wan. Separating points by axis-parallel lines. *Int. J. Comput. Geometry Appl.*, 15(6):575–590, 2005.
3 O. Devillers, F. Hurtado, M. Mora, and C. Seara. Separating several point sets in the plane. In *Proc. of the 13th Canad. Conf. Comput. Geom.*, pages 81–84, 2001.
4 U. Fayyad and K. 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.
5 R. Freimer, J. Mitchell, and C. Piatko. On the complexity of shattering using arrangements. TR 91-1197, Dept. Comput. Sci., Cornell Univ., NY, 1991.

6 S. Har-Peled and M. Jones. On separating points by lines. arXiv:1706.02004v1, 2017.

7 F. Hurtado, M. Mora, P. Ramos, and C. Seara. Separability by two lines and by nearly straight polygonal chains. *Disc. Appl. Mathematics*, 144(1-2):110–122, 2004.

8 R. Impagliazzo and R. Paturi. Complexity of k-SAT. In *Proc. of 14th Annual IEEE Conf. on Computational Complexity*, pages 237–240. IEEE, 1999.

9 J. Kujala and T. Elomaa. Improved algorithms for univariate discretization of continuous features. In *Proc. of the 11th PKDD*, volume 4702 of *LNCS*, pages 188–199, 2007.

10 D. Marx. Parameterized complexity of independence and domination on geometric graphs. In *Proc. of 2nd IWPEC*, volume 4169 of *LNCS*, pages 154–165, 2006.

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

12 D. Marx and M. Pilipczuk. Optimal parameterized algorithms for planar facility location problems using voronoi diagrams. In *Proc. of 23rd ESA*, volume 9294 of *LNCS*, pages 865–877, 2015.

13 N. Megiddo. On the complexity of polyhedral separability. *Discr. & Comput. Geom*, 3:325–337, 1988.

5 Appendix

5.1 Some more details on Structured 2-Track Hitting Set

ETH-based lower bounds. The *Exponential Time Hypothesis* (ETH) is a conjecture by Impagliazzo et al. [8] 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. 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 6** ([11]). 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 7** ([1]). 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 is proved for 2-Track Hitting Set in a paper by Marx and Pilipczuk [12]. The authors use that 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 7 to show the same lower bound for Point Guard Art Gallery and Vertex Guard Art Gallery, where one wishes to guard a simple polygon with k points, and k vertices, respectively. In this paper, we again utilize S2-THS as the starting point of a reduction to Red-Blue Separation. Thus, it seems as though (Structured) 2-Track Hitting Set can provide a healthy starting point for a wide variety of geometric problems and yield almost tight lower bounds, like Grid Tiling [10] has been doing in the last decade for geometric problems optimally solvable in $n^{\Theta(\sqrt{k})}$.

5.2 Why the lower bound might be elusive

Let us say a few words on the difficulty of showing such a result for Red-Blue Separation, compared to its NP-hardness. 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 us from encoding an adjacency matrix on n vertices with bichromatic gadgets, while one does not seem to achieve much with a monochromatic encoding.

A perhaps more concrete issue with encoding an adjacency matrix is the following. Say, 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.

5.3 Correctness of the reduction

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 6). Let $(a^1_{u_1}, b^1_{u_1}), (a^2_{u_2}, b^2_{u_2}), \ldots, (a^k_{u_k}, b^k_{u_k})$ 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 $j \in [k]$. $F \cup \bigcup_{j \in [k]} \{\mathrm{HL}(s_j), \mathrm{VL}(s_j), \mathrm{VL}'(s_j), \mathrm{HL}'(s_j), \mathrm{SL}(s_j), \mathrm{SL}'(s_j)\}$ 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 6. 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 $\mathcal{G}(A)$, $\mathcal{G}(B)$, $\mathcal{G}(\sigma)$, $\mathcal{G}(\mathrm{id})$, the $\mathcal{G}_{\approx v}(\sigma_j)$'s, and the $\mathcal{G}_{\approx h}(\sigma_j)$'s. The points in $\mathcal{G}(\sigma)$ and $\mathcal{G}(\mathrm{id})$ are separated as in Figure 3a, since the choice of $\mathrm{VL}'(s_j)$ matches the choices of $\mathrm{HL}(s_j)$ and $\mathrm{HL}'(s_j)$. As it can be observed by looking at Cell$(4, 3)$ and Cell$(5, 4)$ of Figure 4, there is not interaction between the red and blue points of diagonally adjacent faces of the k-by-k crossing (in $\mathcal{G}(\sigma)$ and $\mathcal{G}(\mathrm{id})$).

Since $a^1_{u_1}, a^2_{u_2}, \ldots, a^k_{u_k}$ (resp. $b^1_{u_1}, b^2_{u_2}, \ldots, b^k_{u_k}$) 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 an $j \in [k]$ such that $a^j_{u_j}$ hits I (resp. $b^j_{u_j}$ hits I), and the two red points encoding I are in the two quadrants defined by $\mathrm{HL}(s_j)$ and $\mathrm{VL}(s_j)$ (resp. defined by $\mathrm{SL}(s_j)$ and $\mathrm{SL}'(s_j)$) 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 $\mathrm{VL}'(s_j)$ and $\mathrm{SL}(s_j)$ (resp. $\mathrm{HL}'(s_j)$ and $\mathrm{SL}'(s_j)$) 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 8.** *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 8 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 6, 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 $a^j_{u_j}$ be the element of A corresponding to $\mathcal{L}_j$ (with the correspondence described in Figure 3a). 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, thanks to the following lemma, that $a^1_{u_1}, a^2_{u_2}, \ldots, a^k_{u_k}$ is a hitting set of $(A, \mathcal{S}_A)$.

▶ **Lemma 9.** *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 $R([a^j_1, a^j_t])$, 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 9, 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 $a^j_{u_j}$'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 $a^1_{u_1}, a^2_{u_2}, \ldots, a^k_{u_k}$ will force to take the lines corresponding to the elements $b^1_{u_1}, b^2_{u_2}, \ldots, b^k_{u_k}$. Still by Lemma 9, $\mathcal{G}(\sigma)$ transmits the choices of the $\mathcal{L}_j$'s *downwards* with the desired permutation σ of the color classes, while $\mathcal{G}(\text{id})$ transmits unchanged the choices of the vertical lines separating $\mathcal{G}(\sigma)$ *to the left*.

Similarly to the simple argument of Lemma 9, 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, $\text{SL}(s)$ when $\text{VL}'(s)$ has been selected, or $\text{SL}'(s)$ when $\text{HL}'(s)$ 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 5). 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 $b^j_{u_j}$, and selecting a relatively vertical line somewhere below the line corresponding to $b^j_{u_j}$ (see Figure 8).

Though, by Lemma 9, 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 $b^j_{u_j}$. This concludes to prove that choosing the lines corresponding to $a^1_{u_1}, a^2_{u_2}, \ldots, a^k_{u_k}$ to separate $\mathcal{G}(A)$ forces to select the lines corresponding to $b^1_{u_1}, b^2_{u_2}, \ldots, b^k_{u_k}$ to separate $\mathcal{G}(B)$. Finally, as we already observed for track A, the $b^j_{u_j}$'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 $a^1_{u_1}, a^2_{u_2}, \ldots, a^k_{u_k}$ is a hitting set of $(A, \mathcal{S}_A)$ and $b^1_{u_1}, b^2_{u_2}, \ldots, b^k_{u_k}$ is a hitting set of $(B, \mathcal{S}_B)$, $\mathcal{I}$ is a YES-instance.

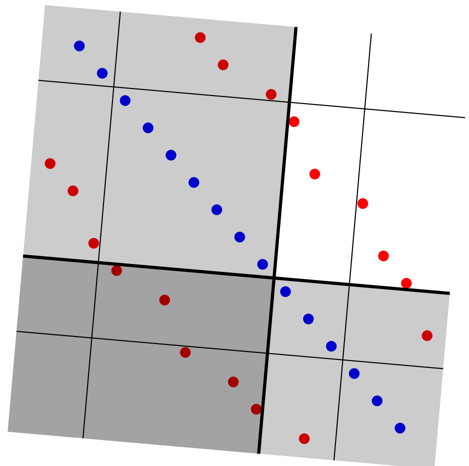


Figure 8 In bold, the horizontal and vertical lines in $\mathcal{G}(B)$ corresponding to selecting some element b_i^j of B. The grey regions materialize the potential positions for the slanted line in $\mathcal{G}_{\approx v}(\sigma_j)$ and the slanted line in $\mathcal{G}_{\approx h}(\sigma_j)$ once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

5.4 Proof of Lemma 4

Suppose for contradiction that there are three lines in S which have exactly i points to the left of them: they are the lines $x = x_1$, $x = x_2$, and $x = x_3$, with $x_1 < x_2 < x_3$. Then all points with x-coordinates between x_1, x_3 are red. Hence, removing the line $x = x_2$ from the solution produces a better, but still feasible, solution. The argument is identical for horizontal lines.

5.5 Proof of Lemma 5

We state the proof for horizontal lines, since the case of vertical lines is identical. Suppose that S contains the lines $y = y_1$, $y = y_2$ which both have exactly i points of $\mathcal{B}$ below them, and $y_1 < y_2$. Let y_1' be such that the line $y = y_1'$ contains exactly i blue points below it and as many red points above it a possible. Let y_2' be such that the line $y = y_2'$ contains exactly i blue points below it and as many red points below it as possible. We replace in the solution the lines $y = y_1$ and $y = y_2$ with the lines $y = y_1'$ and $y = y_2'$. Informally, we have moved the line $y = y_1$ as far down as possible and the line $y = y_2$ as far up as possible without changing the number of blue points each has below it. We claim that the new solution is still feasible because any cell contained between $y = y_1'$ and $y = y_2'$ contains only red points, while all other cells are either unchanged or have become smaller.