Red-Blue Point Separation for Points on a Circle

Neeldhara Misra* Harshil Mittal† Aditi Sethia‡

Abstract

Given a set R of red points and a set B of blue points in the plane, the Red-Blue point separation problem asks if there are at most k lines that separate R from B, that is, each cell induced by the lines of the solution is either empty or monochromatic (containing points of only one color). A common variant of the problem is when the lines are required to be axis-parallel. The problem is known to be NP-complete for both scenarios [1, 10], and W[1]-hard parameterized by k in the former setting [5] and FPT in the latter [8]. We demonstrate a polynomial time algorithm for the special case when the points lie on a circle. Further, we also demonstrate the W-hardness of a related problem in the axis-parallel setting, where the question is if there are p horizontal and q vertical lines that separate R from B. The hardness here is shown in the parameter p.

1 Introduction

Given a set R of red points and a set B of blue points in the plane, the Red-Blue Separation (RBS) problem asks if there are at most k lines that separate R from B, that is, each cell induced by the lines of the solution is either empty or monochromatic (containing points of only one color). Equivalently, R is separated from B if, for every straight-line segment ℓ with one endpoint in R and the other one in B, there is at least one line in the solution that intersects ℓ. A common variant of the problem is when the solution lines are required to be axis-parallel (APRBS). Questions about the discrete geometry on red and blue points in general, and their separability using geometric objects in particular, are of fundamental interest. This makes RBS a well-studied problem on its own right. It is also motivated by the problem of univariate discretization of continuous variables in the context of machine learning [4, 9].

RBS is known to be NP-complete [10], APX-hard [1], and W[1]-hard when parameterized by the solution size [5]. The approximation hardness holds for the APRBS problem also, while in contrast the parameterized intractability is known only for the general RBS problem. Specifically, it is known that an algorithm running in time $f(k)n^{o(k/\log k)}$, for any computable function f, would disprove ETH [5]. This reduction crucially relies on selecting lines from a set with a large number of different slopes — in particular, the number of distinct slopes of the lines used is not bounded by a function of k.

For the case where $k = 1$ and $k = 2$, the problem is solvable in $O(n)$ and $O(n \log n)$ time respectively [7]. The problem is known to be FPT parameterized by the number of blue points (or the number of red points). A 2-approximation algorithm is also known for APRBS [1] by casting the separation problem as a special case of the rectangle stabbing problem[1]. We note that the 2-approximation algorithm and APX-hardness applies even to the separation of monochromatic point sets (where the goal is to separate *all* points from each other) and this problem is also known to admit a approximation (OPT log OPT)-approximation [6].

Our Contributions We first address a question raised in the discussions from [1]: *Do special cases admit better approximation ratios or even exact solutions?* We make partial progress on this question by answering it in the affirmative when the input points lie on a circle (which would be a special case[2] of points in convex position). In particular, we show that when points lie on a circle, both RBS and APRBS admit exact polynomial-time algorithms. Interestingly, the RBS problem is significantly simpler in this special case compared to its axis-parallel counterpart. For the latter, the size of the optimal solution is captured by a structural parameter of a graph that is naturally associated with the point set. Our proof is algorithmic and can be used to solve the associated computational question.

Further, we introduce a natural variant of APRBS, which is the (p, q)-APRBS problem. Here, as before, we are given a set of red and blue points, and the question is if there is a set of at most p horizontal lines and at most q vertical lines that separate R from B. We show that this problem is W[2]-hard when parameterized by p alone. Finally, we also show by a simple observation

*IIT Gandhinagar, neeldhara.m@iitgn.ac.in

†IIT Gandhinagar, mittal_harshil@iitgn.ac.in

‡IIT Gandhinagar, aditi.sethia@iitgn.ac.in

[1] This is based on the idea that lines separating R from B must stab all rectangles formed by red and blue points at the corners.

[2] We speculate that these ideas would also be relevant for the more general scenario of points in convex position. While the algorithm for RBS in fact works as-is for points in convex position, the details for the axis-parallel variant are less obvious.

that the $2^{O(|B|)}$ algorithm for APRBS from [5] can be improved to $2^{O(k \log |B|)}$.

The rest of the paper is organized as follows. We formally define the problems that we address in Section 2 and focus on the case of points on a circle in Section 3 for both RBS and APRBS. We describe the W[2]-hardness result for (p, q)-APRBS parameterized by p in Section 4. Due to lack of space, we make our remarks about the improved algorithm in the full version of the paper [11].

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. Given a set of points $R \cup B$ in the plane, R is said to be separated from B by a collection of lines L if every straight-line segment with one endpoint in R and the other one in B is intersected by at least one line in L. We adopt the convention of requiring a "strict" separation, which is to say that no point in $R \cup B$ is on a separating line. We let $n := |R \cup B|$, $r = |R|$ and $b = |B|$. The computational problems that we study are the following.

Red-Blue Separation. (RBS) Given a set R of red points and a set B of blue points in the plane and a positive integer k as input, determine if there exists a set of at most k lines that separate R from B.

Axis-Parallel Red-Blue Separation. (APRBS) Given a set R of red points and a set B of blue points in the plane and a positive integer k as input, determine if there exists a set of at most k axis-parallel lines that separate R from B.

(p,q)-Axis-Parallel Red-Blue Separation. ((p,q)-APRBS) Given a set R of red points and a set B of blue points in the plane and positive integers p and q as input, determine if there exists a set of at most p horizontal and q vertical lines that separate R from B.

3 Points on a Circle

In this section, we focus on the special case when all the points lie on a circle C. Let $P = (R \cup B)$ denote a set of n points on a circle, with r red points and b blue points. As usual, R (respectively, B) denotes the set of red (respectively, blue) points. Without loss of generality, we assume that all points of P lie on an unit circle centered at the origin. Fix an order σ on $R \cup B$ based on the order of their appearance on the circle, starting at $(1, 0)$ and moving around the circle counterclockwise. We let r_i and b_j denote the i^{th} red and the j^{th} blue point that we encounter in this order. For a point p on the circle, we let $\mathsf{col}(p)$ denote the color of the point p.

We call a maximal sequence of monochromatic points in σ a *chunk*. Let $\mathcal{C}_P$ denote the set of chunks of P. In the order of their appearance on the circle, we denote the individual chunks by $C_1, \ldots, C_w$. The color of a chunk is the color of any point belonging to it. We refer to a chunk consisting of red (blue) points as a red (blue) chunk. We overload notation and let $\mathsf{col} : \mathcal{C}_P \rightarrow \{R, B\}$ be a function that returns the color of a chunk. The arc of C starting at the last point on the r^{th} chunk and the first point on the $(r + 1)^{th}$ chunk is called a *switch*. Let $\mathcal{S}_P$ denote the set of switches of P. Note that any instance with w chunks has w switches. We denote the switches by $S_1, \ldots, S_w$ in the order of their appearance on the circle. Also note that w must always be even, and that there are $\frac{w}{2}$ red chunks and $\frac{w}{2}$ blue chunks.

We say that a switch S is *stabbed* by a line ℓ if $\ell \cap S \neq \emptyset$. We first make the following observation.

Proposition 3.1 *Let $P = (R \cup B)$ be a red-blue point set on a circle. If L is a set of lines that separates R from B, then every switch must be stabbed by some line in L.*

Proof. Suppose that there exist a switch $S_i \in \mathcal{S}_P$ that is not stabbed by any line from the set L. Note that S_i separates the chunks C_i and C_{i+1}. Without loss of generality, suppose col(C_i)= R and col(C_{i+1})=B. Since S_i is not stabbed by any line from L, the last point of C_i and first point of C_{i+1} are not separated, which leads to the contradiction of the fact that L separates R from B. Therefore, every switch must be stabbed by some line in L. □

Based on Proposition 3.1, we have that in an instance with w switches, $\frac{w}{2}$ is a lower bound on the optimum, since any line can stab at most two chunks. In the next subsection, we show that this can always be achieved by a set of general lines. For axis-parallel lines, we strengthen the lower bound further using an auxiliary graph structure on the switches, and demonstrate an algorithmic argument to match the stronger lower bound.

3.1 The Case of General Lines

With arbitrary lines, our strategy is simple: we "protect" each monochromatic chunk of a fixed color with a single line passing through it's adjacent switches. In particular, consider any chunk C_i such that $\mathsf{col}(C_i) = B$. Fix an arbitrary point p_i in the switch[3] S_{i-1} and another point q_i in the switch S_i. Let ℓ_i be the line passing through p_i and q_i and let $L := \{\ell_i \mid \mathsf{col}(C_i) = B\}$. In other words, L is the set of lines thus defined based on blue chunks. Note that there are $\frac{w}{2}$ lines in this solution, since we introduce one line for each blue chunk. Moreover, it is also easy to check that these lines sep-

[3] If $i = 1$, then we let $S_{-1} := S_w$.

arate R from B, since every blue chunk belongs to a separate cell of this configuration.

3.2 The Case of Axis-Parallel Lines

When we are restricted to axis-parallel lines in the solution, then the strategy described in the previous subsection would fail since the lines that are described need not be axis-parallel. A similar strategy does give us a simple 2-approximation, which we describe informally. Observe that each monochromatic chunk can be protected by a "wedge" consisting of a pair of axis-parallel lines. Indeed, consider the points p_i and q_i defined as before, and let T be the unique rectangle whose sides are axis-parallel and which has p_i and q_i as diagonally opposite corner points. Clearly, one of the other two corner points c lies inside C. We can now choose the two axis-parallel lines that contain the edges of the rectangle which intersect at c, and we have a wedge-like structure that protects the chunk (depending on the length of the chunk, note that the points of the chunk may be distributed over multiple cells). This gives us a solution with w lines, and is therefore a two-approximate solution.

We now demonstrate a stronger lower bound for the setting of axis-parallel lines. To this end, we introduce some terminology and define an auxiliary graph based on the point set P. We say that a pair of switches *face each other* if there exists a horizontal or vertical line that stab both of them. A switch which faces at least one other switch is said to be *nice*, a pair of switches facing each other is called a *nice pair*, and a switch that is not nice is said to be *isolated*. We define a graph based on P that has a vertex for every switch, and an edge between every pair of vertices corresponding to switches that are nice pairs. Formally, for a red-blue point set $P = (R \cup B)$ with w switches $S_1, \ldots, S_w$, we define the graph $G_P = (V_P, E_P)$ as follows: $V_P = \{v_j \mid 1 \leq j \leq w\}$ and $E_P = \{(v_i, v_j) \mid (S_i, S_j) \text{ is a nice pair}\}$.

Observe that every isolated switch of P corresponds to an isolated vertex of G_P. Recall that an *edge cover* of a graph G is a set of edges such that every vertex of the graph is incident to at least one edge of the set. Note that a minimum-sized edge cover can be found by greedily extending a maximum matching of a graph G. We use the abbreviation MEC to refer to a minimum edge cover. Let $I_P \subseteq V_P$ be the set of isolated vertices of G_P and let $H_P := G_P \setminus I_P$. We define $\kappa(G_P) := |I_P| + \mathsf{MEC}(H_P)$, where $\mathsf{MEC}(G)$ denotes the size of a minimum edge cover of the graph G. Our first claim is that any instance $P = (R \cup B)$ of APRBS requires at least $\kappa(G_P)$ lines to separate R from B. Next, we will show that this bound is tight.

Before stating the claims formally, we make some remarks about the bound. Note that this coincides with the bound obtained as a consequence of Proposition 3.1 when G_P has a perfect matching. Further, the bound is w when G_P is the empty graph, or equivalently, when every switch is an isolated switch. In this scenario, note that the approach described for the two-approximate solution will, in fact, yield an optimal solution. The intuition for the bound in the generic case is the association between lines and edges in a MEC: indeed, every edge e in G_P corresponds to a family of lines that stab switches corresponding to the endpoints of e. Our goal is to show that we can pick one line corresponding to each edge in the MEC and one line for each isolated switch in such a way that we separate R from B. However, it is easy to come up with examples where this does not happen, and indeed, the argument for the upper bound follows by making a bounded number of modifications to the set of lines that was proposed with guidance from the MEC. On the other hand, this association runs both ways, so we can recover subset of edges from any collection of lines separating R from B, using that stab two switches. If a solution uses fewer than $\kappa(G_P)$ lines, the hope is that the edges recovered lead us to an edge cover that has fewer edges than the MEC, which would be a contradiction. We now formalize both sides of this argument. We begin with the lower bound.

Lemma 3.1 *Let $P = (R \cup B)$ be a red-blue point set on a circle. Let L be a set of k axis-parallel lines that separate R from B. Then $k \geq \kappa(G_P)$.*

Proof. Consider any solution L that uses k axis-parallel lines. By Proposition 3.1, we know that every switch must be stabbed by some line from L. In particular, suppose there are α lines in L that stab a pair of (nice) switches, and β lines that stab one switch. Clearly, $\beta \geq |I_P|$, the number of isolated vertices in G_P.

Let X be the set of non-isolated vertices in G_P which are not covered by the edges corresponding to the α lines stabbing pairs of nice switches. Now, note that the switches corresponding to these non-isolated vertices in X must be stabbed by one of the β lines. So, $\beta \geq |X| + |I_P|$. Recall that $H_P = G_P \setminus I_P$ and $\mathsf{MEC}(H_P)$ covers every non-isolated vertex of G_P. Observe that $\mathsf{MEC}(H_P) \leq \alpha + |X|$, since the edges corresponding to the α lines stabbing pairs of switches can be extended by a collection of $|X|$ edges, one each for each non-isolated vertex that is not accounted for so far, to obtain a MEC for H_P. Adding both the inequalities above, we get:

$$\alpha + \beta + |X| \geq |X| + |I_P| + \mathsf{MEC}(H_P)$$
$$\Rightarrow \alpha + \beta \geq |I_P| + \mathsf{MEC}(H_P); \Rightarrow k \geq \kappa(G_P),$$

as desired. □

We now turn to the upper bound. In this section, we state some claims without proofs due to lack of space and refer the reader to the full version of the paper for the detailed arguments.

Lemma 3.2 *Let $P = (R \cup B)$ be a red-blue point set on a circle. There exists a collection of at most $\kappa(G_P)$ lines that separate R from B.*

The proof of the upper bound is algorithmic, and we demonstrate it with a series of claims. To begin with, let $F_P \subseteq E_P$ be a MEC of H_P and let $t := |I_P|$. We define a set of lines L_0 as follows. For every edge $e = (v_i, v_j) \in F_P$, let ℓ_e be an arbitrary axis-parallel line passing through the switches S_i and S_j. For every isolated switch S_r, let ℓ_r be an arbitrary axis-parallel line stabbing S_r. Now define L_0 as the collection of all of these lines, i.e:

$$L_0 = \{\ell_e \mid e \in F_P\} \cup \{\ell_r \mid v_r \in I_P\}.$$

Note that $|L_0| = \kappa(G_P)$. If L_0 separates R from B, then we are done. Otherwise, we will obtain another set of axis-parallel lines that "dominates" L_0 in that it has the same size as L_0, separates all pairs of points separated by L_0 and at least one additional pair. To formalize this, we introduce the notion of a strictly dominating solution. For a set of lines L, let $\mathsf{sep}(L) \subseteq R \times B$ denote the set of red-blue point pairs that are separated by L. Given two collections of axis-parallel lines L and $L^\star$, we say that $L^\star$ *strictly* dominates L if $|L^\star| \leq |L|$ and $\mathsf{sep}(L) \subsetneq \mathsf{sep}(L^\star)$. We will now show that there exists a sequence of sets of axis-parallel lines $L_0, L_1, \ldots, L_g$ such that L_i strictly dominates L_{i-1} for all $1 \leq i \leq g$ and L_g separates R from B. Note that the number of steps is bounded by rb. Throughout, we will maintain the invariant that every switch is stabbed by at least one line. Note that this is true, in particular, for L_0.

Claim 3.1 *Every switch is stabbed by at least one line from L_0.*

For a collection of axis-parallel lines L, we say that a cell of L is *corrupt* if it is non-monochromatic, that is, if it contains at least one red point and at least one blue point. Note that L_0 contains at least one corrupt cell, otherwise we would be done. We consider all the possible ways in which a cell can intersect the circle underlying our point set.

Claim 3.2 *Let $\mathcal{R}$ be an axis-parallel rectangle and let C be a circle centered at the origin. Then $\mathcal{R} \cap C$ is either empty or consists of at most four disjoint arcs of C.*

Next, we note that any corrupt cell must contain at least two disjoint arcs of the circle.

Claim 3.3 *Let L be a set of lines that stabs every switch at least once, and let $\mathcal{R}$ be a corrupt cell of L. Then $\mathcal{R} \cap C$ contains at least two disjoint arcs of the circle C.*

We say that a cell $\mathcal{R}$ is *large* if $\mathcal{R} \cap C$ contains three or four disjoint arcs of C. We note that any instance can have at most one large cell.

We are now ready to describe the high-level strategy for obtaining a sequence of strictly dominating solutions. It turns out that if a corrupt cell consists of exactly two disjoint arcs, then depending on the "location" of the cell, there is a simple strategy that allows us to clean up the cell by flipping the orientation of one of the lines in the solution. In particular, and informally speaking, the strategy works for corrupt cells that are "above" ("below") the origin if all cells above it are monochromatic, or corrupt cells "to the left" ("to the right") the origin if all cells before (after) it are monochromatic. This gives us a natural sweeping strategy to clean up corrupt cells from four directions, while potentially getting stuck at a large cell "at the center". When the large cell is the only corrupt one, it turns out that there are a fixed number of configurations it can have when considered along with its surrounding cells, and for each of these cases, we suggest an explicit strategy to clean up the large cell to arrive at a solution with no corrupt cells at all. We now formalize this argument.

Let L denote the current solution: to begin with, L is L_0, and we describe a process to obtain a solution L' that strictly dominates L if L is not already a valid solution. Note that L divides the plane into vertical and horizontal strips, which we will refer to as the rows and columns of the solution. Also, we call a cell of this configuration *empty* if it does not contain any points of P. We first focus on corrupt cells that are *not* large. Consider a cell $\mathcal{R}$ whose intersection with C consists of exactly two disjoint arcs, say A_1 and A_2. Note that A_1 and A_2 lie in distinct quadrants. We call $\mathcal{R}$ a horizontal cell if these arcs lie in the first and second or the third and fourth quadrants; and we call $\mathcal{R}$ a vertical cell if these arcs lie in the first and fourth or the second and third quadrants. Note that the remaining possibilities do not arise with cells that are not large. We refer the reader to Figure 1 in the full version of the paper for a visual representation of these cases.

Consider the corrupt horizontal cell whose center has the largest y-coordinate in absolute value. This is either the top-most corrupt cell above the x-axis (Case 1) or the bottom-most corrupt cell below the x-axis (Case 2). If there are no corrupt horizontal cells, then consider the corrupt vertical cell whose center has the largest x-coordinate in absolute value. This is either the left-most corrupt cell to the left of the y-axis (Case 3) or the right-most corrupt cell to the right of the y-axis (Case 4).

Let us consider Case 1. Here, observe that any row above the row containing the cell $\mathcal{R}$ consists of at most one non-empty cell and that all such cells are monochromatic by the choice of $\mathcal{R}$. Now, if the cell above $\mathcal{R}$ is monochromatic red and the arc in the first (second) quadrant consists of red points, then the top line of $\mathcal{R}$ can be flipped to a vertical line about the top-left (top-right) corner of the cell $\mathcal{R}$. It is easy to check that this solution strictly dominates L. The case when the cell above $\mathcal{R}$ is monochromatic blue can be argued similarly. We refer the reader to Figure 2 in the full version of the paper for an illustration of the switching strategies in these scenarios.

Case 2 is similar to Case 1 except that we argue relative to the cells below $\mathcal{R}$ rather than above it. In Cases 3 and 4, we flip vertical lines to a horizontal orientation, and the argument is based on monochromatic cells that lie to the left and right of $\mathcal{R}$, respectively. All the details are analogous to the case that we have discussed. Therefore, as long as the current solution has a corrupt cell that is not large, this discussion enables us to find a strictly dominating solution.

Now, the only case that remains is the situation where we have exactly one corrupt cell which is large. For a large cell we have four surrounding monochromatic or empty cells. The three or four arcs contained inside the large cell may also have red or blue points in different configurations. It turns out that each of these cases admits a new solution which makes all cells monochromatic. This can be established by inspection, and we refer the interested reader to the supplementary material that goes over all the individual cases[4]. Meanwhile, we refer the reader to Figure 3 in the full version of the paper for a stereotypical case and the corresponding strategy. Based on this discussion, we conclude with the formal statement of the main result of this section.

Theorem 3.1 *RBS and APRBS can be solved in polynomial time when the input points lie on a circle.*

4 W-hardness of (p,q)-Separation

In this section, we focus on the (p, q)-APRBS problem. Before describing our result, we briefly comment on the relationship between this problem parameterized by only the budget for horizontal lines (or vertical lines, by symmetry) and APRBS parameterized by the size of the entire solution. If APRBS had turned out to be W[1]-hard or W[2]-hard parameterized by k, then it would imply that (p, q)-APRBS is unlikely to be FPT parameterized by either p or q, since such an algorithm can be used as a black box to resolve the former question with only a polynomial overhead (guess p, q such that $p + q = k$). On the other hand, if (p, q)-APRBS turns out to be FPT parameterized by either p or q, then this would imply that APRBS is also FPT for the same reason. We show that (p, q)-APRBS is W[2]-hard when parameterized by p, the number of horizontal lines used in the solution. Therefore, our observation here establishes the hardness of the problem for a smaller parameter, and it does not have any direct implications for APRBS. Our result is also not implied by what is known about APRBS, since it turns out that the problem is FPT parameterized by k.

We reduce from the Colorful Red-Blue Dominating Set (C-RBDS) problem, which is defined as follows. The input is a bipartite graph $G = (R, B, E(G))$ along with a partition of R into k parts $R_1 \uplus \cdots \uplus R_k$. The question is if there exists a subset $S \subseteq R$ such that $|R_i \cap S| = 1$ for all $1 \leq i \leq k$ and that dominates every vertex in B; in other words, for all $v \in B$, there exists a $u \in S$ such that $(u, v) \in E(G)$. Such a set is called a colorful red-blue dominating set for the graph G. This problem is well-known to be W[2]-hard when parameterized by k [2]. Our reduction is inspired by the reduction from SAT used to show the hardness of the problem of separating n points from each other [1]. One aspect that is specific to our setting is ensuring that the budget for lines in one orientation is controlled as a function of the parameter.

Theorem 4.1 *(p, q)-APRBS is W[2]-hard when parameterized by p.*

Proof. Let $G = (R = R_1 \uplus \cdots \uplus R_k, B); k)$ be an instance of C-RBDS. Without loss of generality, we assume that every vertex $v \in B$ has the same degree d and that d is even[5]. We may also assume that all R_i's have the same number of vertices (padding R_i with $\max_{j=1}^{k}\{|R_j|\} - |R_i|$ dummy isolated vertices if required). We let $|R_1| = \cdots = |R_k| = m$ and $n := |B|$. We also assume that k is even, again without loss of generality. Finally, we impose an arbitrary but fixed ordering on each R_i and on the sets $N(v)$ (neighbours of v in G) for every $v \in B$.

It will be convenient to think of the point set of the reduced instance as lying within a sufficiently large bounding box, say $\mathcal{B}$. To describe the placement of the points, we impose an uniform $(k + 2) \times (n + 1)$ grid on $\mathcal{B}$, which divides $\mathcal{B}$ into $k + 2$ horizontal regions $H_0, H_1, \ldots, H_k, H_{k+1}$ (labeled from bottom to top) and $(n + 1)$ vertical regions $V_0, V_1, \ldots, V_n$ (labeled from left to right) which we call *tracks*. Each horizontal track H_i for $i \in [k]$ is divided further into $m + 2$ horizontal strips

[4]Please see the full version of this work for more details [11].

[5]When this is not the case, let $\Delta := \max_{v \in B}\{d(v)\}$. We may introduce an additional "dummy color" class R_0 with a forced dummy vertex (for example, by adding a d-star whose center is in B and whose leaves are in R_0), and for every vertex $v \in B$ we may introduce $\Delta - d(v)$ new pendant red neighbors of v in R_0. If d happens to be odd, use $\Delta + 1$ in this process instead of Δ to ensure that d is even.

and each vertical track V_j for $j \in [n]$ is divided further into $2d$ vertical strips. Within a horizontal track, the first and last horizontal strips are called *buffer zones*. Further, when we refer to the p^{th} horizontal strip within any horizontal track, the buffer zones are not counted. We refer the reader to Figure 4 in the full version of this paper for a visual representation of the reduced instance.

For $i \in [k]$, $j \in [n]$, $\alpha \in [m]$, and $\beta \in [2d]$, we refer to the intersection of the α^{th} horizonal strip in H_i and the β^{th} vertical strip in V_j as $Z_{ij}[\alpha, \beta]$. We note that two points that share the same x-coordinate (y-coordinate) have to be separated by a vertical (horizontal) line. We now describe three sets of points that we need to add: the first will lead us to a choice of a vertex from each R_i, the second set encodes the structure of the graph, and the third set ensures that the chosen set is indeed a dominating set by forcing the use of a budget in a certain way.

Selectors. Consider the first vertical track. Here, for any even (odd) $i \in [k]$, we add a red (blue) point to the top buffer zone and a blue (red) point to the bottom buffer zone of the i^{th} track. These $2k$ points are called the *selectors*. We ensure that all selectors have the same x-coordinate. Intuitively, the selector points ensure that any valid solution is required to use at least one horizontal line drawn in each of the k horizontal tracks — and the budget will eventually ensure that any valid solution uses exactly one. Which horizontal strip these lines end up in will act as a signal for our choice of vertices in the dominating set in the reverse direction.

Functional Points. Next, consider any vertex $v_j \in B$. For every $u \in N(B)$, we add a pair of red and blue points in $Z_{ij}[\alpha, 2\beta]$ if u is the α^{th} vertex of R_i and is the β^{th} neighbor of v_j. These points are added to the bottom-left and top-right corners of the box. If β is odd (even)[6], then the bottom-left corner gets a blue (red) point and the opposite corner gets the red (blue) point. These pairs of points will be referred to as the *functional pairs*. The functional pairs encode the structure of the graph, and we would like to ensure that the responsibility of separating at least one functional pair in each vertical track falls on a horizontal line used to separate the selectors. We force this by choosing an appropriately small budget for vertical lines, which ensures that not all separations can be accounted for using vertical lines. However, we still need to control how the vertical budget is utilized across different tracks. To this end, we introduce a special gadget that forces the use of a certain number of vertical lines in each vertical track.

[6] The organization of colors based on the parity of the columns in the case of functional pairs and rows in the case of selectors is to ensure that there are no additional separation requirements other than the ones that we desire to encode.

Guards. In the horizontal track H_0, we place d points, all with the same choice of y-coordinate which is arbitrary but fixed. Within the j^{th} vertical track, x-coordinate of the s^{th} point is chosen so that the point lies in the middle of the $(2s)^{th}$ vertical strip of V_j. The color of the first vertex in the track V_i is blue if i is odd and red if i is even. This ensures that for $2 \leq i \leq n$, the first point in the i^{th} track has the same color as the last point of the $(i-1)^{th}$ track. The colors of the remaining points are chosen so that consecutive points within the same track have distinct colors. Equivalently, the s^{th} guard vertex in the i^{th} track is blue (red) if s and i are both odd (even), and is red (blue) if s is odd (even) and i is even (odd). We refer to these points as *guards*.

We briefly discuss the role of the guard vertices: we note that the guards can be separated from each other by $(d-1)$ vertical lines, and since all guards have the same y-coordinate, this is the only way to separate them. However, there is no set of $(d-1)$ lines that can separate all the guards *and* all the associated functional pairs in any vertical track. The budget for the vertical lines will be such that we can only afford to separate the guards as we are required to do, and we will be forced to separate at least one functional pair using a horizontal line, which will essentially ensure that the selectors have chosen vertices corresponding to a dominating set.

We let $p := k + 2$ and $q := (d-1)n + 1$. This mostly completes the description of the construction. Due to lack of space, we defer the argument for equivalence and some minor details in the construction to the full version of the paper [11]. □

5 Concluding Remarks

We showed that RBS and APRBS are polynomial-time solvable when points lie on a circle. Further, we introduced a natural variant that separates out the budget for horizontal and vertical lines in the axis-parallel variant, and demonstrated that (p, q)-APRBS is W[2]-hard when parameterized by p. We expect a natural adaptation of these arguments to work for points in convex position as well. In the general setting, since APRBS is FPT when parameterized by k [8], the question of whether the problem admits a polynomial kernel would be natural to explore further. Our W[1]-hardness reduction for (p, q)-APRBS may provide some starting points towards an answer in the negative — in its present form the parameter k of the reduced instance depends on k, d, and n. APRBS would not admit a polynomial kernel (under standard complexity-theoretic assumptions) if this dependence can be reduced to k and n only [3].

References

[1] G. Călinescu, A. Dumitrescu, H. J. Karloff, and P. Wan. Separating points by axis-parallel lines. *International Journal of Computational Geometry and Applications*, 15(6):575–590, 2005.

[2] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 1st edition, 2015.

[3] M. Dom, D. Lokshtanov, and S. Saurabh. Kernelization lower bounds through colors and ids. *ACM Transactions Algorithms*, 11(2):13:1–13:20, 2014.

[4] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence.*, pages 1022–1029. Morgan Kaufmann, 1993.

[5] P. Giannopoulos, E. Bonnet, and M. Lampis. On the parameterized complexity of red-blue points separation. *Journal of Computational Geometry*, 10(1):181–206, 2019.

[6] S. Har-Peled and M. Jones. On separating points by lines. *Discret. Comput. Geom.*, 63(3):705–730, 2020.

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

[8] S. Kratsch, T. Masarík, I. Muzi, M. Pilipczuk, and M. Sorge. Optimal discretization is fixed-parameter tractable. *CoRR*, abs/2003.02475, 2020.

[9] J. Kujala and T. Elomaa. Improved algorithms for univariate discretization of continuous features. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Database (PKDD)*, volume 4702 of *Lecture Notes in Computer Science*, pages 188–199. Springer, 2007.

[10] N. Megiddo. On the complexity of polyhedral separability. *Discrete Computational Geometry*, 3:325–337, 1988.

[11] N. Misra, H. Mittal, and A. Sethia. Red-blue point separation for points on a circle, arXiv/2005.06046, 2020.