Fréchet Distance Between a Line and Avatar Point Set*

Aritra Banik[1], Fahad Panolan[2], Venkatesh Raman[3], and Vibha Sahlot[4]

1 **Indian Institute of Technology, Jodhpur, India**
`aritrabanik@gmail.com`
2 **Department of Informatics, University of Bergen, Norway**
`fahad.panolan@ii.uib.no`
3 **The Institute of Mathematical Sciences, HBNI, Chennai, India**
`vraman@imsc.res.in`
4 **Indian Institute of Technology, Jodhpur, India**
`sahlotvibha@gmail.com`

Abstract

Frèchet distance is an important geometric measure that captures the distance between two curves or more generally point sets. In this paper, we consider a natural variant of Fréchet distance problem with multiple choice, provide an approximation algorithm and address its parameterized and kernelization complexity. A multiple choice problem consists of a set of color classes $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_n\}$, where each class Q_i consists of a pair of points $Q_i = \{q_i, \bar{q}_i\}$. We call a subset $A \subset \{q_i, \bar{q}_i : 1 \leq i \leq n\}$ conflict free if A contains at most one point from each color class. The standard objective in multiple choice problem is to select a conflict free subset that optimizes a given function.

Given a line segment ℓ and set $\mathcal{Q}$ of a pair of points in $\mathbb{R}^2$, our objective is to find a conflict free subset that minimizes the Fréchet distance between ℓ and the point set, where the minimum is taken over all possible conflict free subsets. We first show that this problem is NP-hard, and provide a 3-approximation algorithm. Then we develop a simple randomized FPT algorithm which is later derandomized using universal family of sets. We believe that this technique can be of independent interest, and can be used to solve other parameterized multiple choice problems. The randomized algorithm runs in $O(2^k n \log^2 n)$ time, and the derandomized deterministic algorithm runs in $O(2^k k^{O(\log k)} n \log^2 n)$ time, where k, the parameter, is the number of elements in the conflict free subset solution. Finally we present a simple branching algorithm for the problem running in $O(2^k n^2 \log n)$ time. We also show that the problem is unlikely to have a polynomial sized kernel under standard complexity theoretic assumption.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Fréchet Distance, Avatar Problems, Multiple Choice, FPT

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2016.32

1 Introduction

The Fréchet distance measures similarity between two curves by considering an ordering of the points along the two curves. An intuitive definition of the Fréchet distance is to imagine

* The research of the second author leading to these results received funding from the European Research Council (ERC) via grants Rigorous Theory of Preprocessing, reference 267959.

© Aritra Banik, Fahad Panolan, Venkatesh Raman, and Vibha Sahlot;
licensed under Creative Commons License CC-BY

that a dog and its handler are walking on their respective curves. Both can control their speed but can only go forward. The Fréchet distance of these two curves is the minimum length of any leash necessary for the handler and the dog to move from the starting points of the two curves to their respective endpoints [6].

Eiter and Mannila [14] introduced discrete Fréchet distance. Intuitively, the discrete Fréchet distance replaces the dog and its owner by a pair of frogs that can only reside on any of the n and m specific pebbles on the curves A and B respectively. These frogs hop from a pebble to the next without backtracking. Formally let $A = \{a_1, a_2, \dots a_n\}$ and $B = \{b_1, b_2, \dots b_m\}$ be a sequence of points. For any $r \in \mathbb{R}$ we define the graph G_r with vertices $A \times B$ and there exists an edge between (a_i, b_j) and (a_{i+1}, b_j) if $d(a_{i+1}, b_j) < r$ and there exists an edge between (a_i, b_j) and (a_i, b_{j+1}) if $d(a_i, b_{j+1}) < r$. Discrete Fréchet distance between A and B is the infimum value of r such that in G_r there is a path between (a_1, b_1) and (a_n, b_m).

In this paper we introduce a semi-discrete Fréchet distance which is, given a continuous curve S and a set of points P, the minimum length of a leash that simultaneously allows the owner to walk on S continuously and the frog to have discrete jumps from one point to another in P without backtracking. Hence the leash is allowed to switch discretely when frog jumps from one point to another. We assume that S is a line segment. Our main point of consideration is the multiple choice problem in this setting. Here instead of a set of points P, we are given a set of pair of points $\mathcal{Q}$ in $\mathbb{R}^2$ such that at most one point is selected from each pair so that the length of leash needed is minimized.

These problems are motivated by 2D curve fitting and object construction from noisy data which can further be used in computer vision for data comparison and biomolecules structure comparison. Here the "resemblance" corresponds to minimizing the semi-discrete Fréchet distance. For example, given a noisy data with/without multiple choice constraints, we may construct a curve/object resembling the standard curve/object and may find the resemblance parameter (specified by semi-discrete Fréchet distance).

Related Work. Fréchet distance problem has been extensively studied in the literature. Alt et al. [3] presented an algorithm to compute the Fréchet distance between two polygonal curves of n and m vertices in time $O(nm \log^2(nm))$. The discrete Fréchet distance can be computed in $O(mn)$ time by a straightforward dynamic programming algorithm. Agarwal et al. [1] presented a sub-quadratic algorithm for computing the discrete Fréchet distance between two sequences of points in the plane.

The following problem has been recently addressed by Shahbaz [19]. Given a point set S and a polygonal curve P in $\mathbb{R}^d (d > 2)$, find a polygonal curve Q, with its vertices chosen from S, such that the Fréchet distance between P and Q is minimum with the relaxation that not all points in S need to be chosen, and a point in S can appear more than once as a vertex in Q. They show that a curve minimizing the Fréchet distance can be computed in $O(nk^2 \log(nk))$ time where n and k represent the sizes of P and S respectively. In a recent paper [9] Consuegra and Narasimhan introduce the concept of Avatar problems that deal with situations where each entity has multiple copies or "avatars" and the solutions are constrained to use exactly one of the avatars. Further study of the problems of same flavor can be found in [5, 4]. An Avatar problem consists of a set of color classes $\mathcal{Q} = \{Q_1, Q_2, \dots, Q_n\}$, where each color class Q_i consists of a pair of points $Q_i = \{q_i, \bar{q}_i\}$ (in general $k \geq 2$ points can be in each class). We call a subset $A \subset \{q_i, \bar{q}_i : 1 \leq i \leq n\}$ conflict free if A contains at most one point from each color class.

Problems we address. We formally define the problems considered in this paper starting with the Semi-discrete Fréchet Distance problem.

> SEMI-DISCRETE FRÉCHET DISTANCE
> **Input:** A set of points $P = \{p_1, p_2, \dots, p_n\}$ and a line segment ℓ in $\mathbb{R}^2$.
> **Question:** Find a sequence of points $\lambda^* = \{q_1, q_2 \dots q_k\}$ where $q_i \in P$, which minimizes Fréchet distance with ℓ where the minimum is taken over all the sequence of points in P. We denote the minimum distance by $d^F(P, \ell)$.

Next we consider the following problems involving choices.

> CONFLICT-FREE FRÉCHET DISTANCE
> **Input:** A set $\mathcal{Q}$ of pairs of points, and a line segment ℓ in $\mathbb{R}^2$.
> **Question:** Find a conflict free subset of points $P^* \subset \bigcup_{i=1}^{n} Q_i$ which minimizes $d^F(P^*, \ell)$

The natural decision version of this problem is as follows.

> CONFLICT-FREE FRÉCHET DISTANCE (DECISION VERSION)
> **Input:** A set $\mathcal{Q}$ of pairs of points, a line segment ℓ in $\mathbb{R}^2$, and $d \in \mathbb{R}$.
> **Question:** Is there a conflict free set of points $P^* \subset \bigcup_{i=1}^{n} Q_i$ such that $d^F(P^*, \ell) \leq d$.

The natural parameterized version of the problem is

> PARAMETERIZED CONFLICT-FREE FRÉCHET DISTANCE **Parameter:** k
> **Input:** A set $\mathcal{Q}$ of pairs of points, a line segment ℓ in $\mathbb{R}^2$, $d \in \mathbb{R}$, and $k \in \mathbb{N} \cup \{0\}$.
> **Question:** Is there a conflict free subset of points P^* of cardinality at most k such that $d^F(P^*, \ell) \leq d$.

We also consider parameterized version of "minimum maxGap" introduced in [9]. Here given a set of points $x_1, \dots, x_n$ on a line, *maxGap* is the largest gap between consecutive points in the sorted order. The problem is as follows.

> PARAMETERIZED MINIMUM MAXGAP **Parameter:** k
> **Input:** A set $\mathcal{Q}$ of pairs of points on a line L, two points p_s and p_e on L, $d \in \mathbb{R}$, and $k \in \mathbb{N} \cup \{0\}$.
> **Question:** Is there a conflict free subset of points P^* of cardinality at most k between p_s and p_e such that the minimum *maxGap* of $P^* \cup \{p_s, p_e\}$ is at most d.

Our Results and the organization of the paper. In Section 2 we prove that CONFLICT-FREE FRÉCHET DISTANCE (DECISION VERSION) is NP-Complete. In Section 3 we show that SEMI-DISCRETE FRÉCHET DISTANCE is solvable in $O(n \log n)$ time. In Section 4 we provide a constant factor approximation algorithm for CONFLICT-FREE FRÉCHET DISTANCE. In Section 5 we consider the parameterized complexity of the problem, i.e, PARAMETERIZED CONFLICT-FREE FRÉCHET DISTANCE. In parameterized complexity, algorithm runtimes are measured in terms of input length and a parameter, which is expected to be small. More precisely, a parameterized problem is *fixed-parameter tractable (FPT)* if an instance (I, k) can be solved in time $f(k) \cdot |I|^{O(1)}$ for some function f. Another major research field in parameterized complexity is kernelization. A parameterized problem is said to admit a *polynomial kernel* if any instance (I, k) can be reduced to an equivalent instance (I', k'), in polynomial time, with $|I'|$ and k' bounded by a polynomial in k. There is also a lower bound framework for kernelization which allows us to rule out the existence of polynomial kernels for some problems under standard complexity-theoretic assumptions [8, 11, 15] . For more details about parameterized complexity we refer to monographs [12, 10].

We begin with a simple randomized FPT algorithm and provide a method to derandomize the algorithm using universal sets. In Section 5.2 we give another FPT algorithm using branching. Finally in Section 5.3 we show that the problem is unlikely to have a polynomial sized kernel using OR-composition.

2 Hardness of Conflict-free Fréchet distance Problem

In this section we show that CONFLICT-FREE FRÉCHET DISTANCE (DECISION VERSION) is NP-complete by giving a reduction from Rainbow covering problem mentioned in [4]. Suppose we are given a set $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ where each P_i contains a pair of intervals $\{I_i, \overline{I_i}\}$ such that each interval is a finite continuous subset of the x-axis. A set of intervals $Q \subseteq \bigcup_{i=1}^{n} P_i$ is a rainbow, if it contains at most one interval from each interval pair. An interval is said to cover a point if the point lies inside the interval. The formal definition of Rainbow covering problem is as follows.

> RAINBOW COVERING
> **Input:** A set of pairs of intervals $\mathcal{P}$ and a set of points $S = \{s_1, s_2, \ldots, s_n\}$ on x-axis.
> **Question:** Does there exist a rainbow Q such that each point in S is covered by at least one interval in Q.

RAINBOW COVERING is known to be NP-complete [4]. We introduce an intermediate problem called RAINBOW LINE COVER and show it NP-complete using a reduction from RAINBOW COVERING. Then we give a reduction from RAINBOW LINE COVER to CONFLICT-FREE FRÉCHET DISTANCE (DECISION VERSION).

> RAINBOW LINE COVER
> **Input:** Set $\mathcal{P}' = \{P'_1, P'_2, \ldots, P'_m\}$ where each P'_i contains a pair of left open intervals $\{I_i, \overline{I_i}\}$ and a line segment on x-axis, $\ell^{in} = [x_1, x_2]$.
> **Question:** Is there a rainbow Q^{in} such that it covers line segment ℓ^{in}.

▶ **Lemma 1.** RAINBOW LINE COVER *is NP-hard.*

Proof. The proof is by a polynomial time reduction from RAINBOW COVERING. Let $(\mathcal{P}, S)$ be an instance of RAINBOW COVERING. Without loss of generality, let $s_1, s_2, \ldots, s_n$ be the arrangement of points from S in increasing order on x-axis according to their x-coordinates and each interval from $\mathcal{P}$ covers at least one point in S. We will create an instance $(\mathcal{P}', \ell^{in})$ of RAINBOW LINE COVER as follows. Each interval in $\mathcal{P}$ will be extended and the pairing in the new set $\mathcal{P}'$ is same as the old one. Now for each interval $I_j = [a_j, b_j]$ covering s_1, i.e, the first point in S, consider the interval formed via extending it by a small distance $\delta \in \mathbb{R}$ on left such that it is open at the extended point. Denote it as $I_j^{in} = (a_j - \delta, b_j]$. For each remaining intervals $I_i = [a_i, b_i]$, consider the point s in S such that s is strictly to the left of a_i and is closest to it. Extend I_i to the left such that it is open ended at that point to make I_i^{in}. For example, if $s = (c, 0)$ then $I_i^{in} = (c, b_i]$. Now suppose $s_1 = (a_1, 0)$ and $s_n = (a_n, 0)$, then ℓ^{in} is the line segment on x-axis is $[a_1, a_n]$.

▶ **Claim 2.** *There exists a rainbow from $\mathcal{P}$ of size d, covering S, if and only if there exists a rainbow from $\mathcal{P}'$ of size d covering ℓ^{in}.*

Proof. Let Q be a rainbow from $\mathcal{P}$ covering S. Let Q^{in} be the set of intervals constructed from Q in the reduction. We claim that Q^{in} is a rainbow covering ℓ^{in}. Since Q is a rainbow,

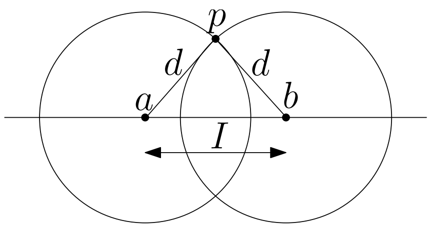

Figure 1 Reduction from Rainbow line cover problem.

Q^{in} is also also a rainbow. Since all all the points in S is covered by Q and each interval of Q is extended left by non-zero distance, S is covered by Q^{in}. Let $q \notin S$ be a point in ℓ^{in}. Let s be the point in S to the right of q and closest to q. By construction any interval covering s, also covers q. Hence Q^{in} covers ℓ^{in}.

Similarly if there is rainbow Q^{in} covering ℓ^{in} then there exists a rainbow Q such that it covers S. Here Q will be the set of intervals that were used to construct intervals in Q^{in}. Since Q^{in} is a rainbow, Q is also a rainbow. Since Q^{in} covers $S \subseteq \ell^{in}$ and the intervals in Q^{in} are obtained by extending openly to nearest left point from S, Q covers S. ◀

This completes the proof. ◀

▶ **Theorem 3.** CONFLICT-FREE FRÉCHET DISTANCE (DECISION VERSION) *is* NP*-complete.*

Proof. Given a sequence of at most n points as witness, we can check in polynomial time whether the the points in the sequence is conflict free and Fréchet distance is at most d, thus the problem is in NP.

To prove NP-hardness we give a polynomial time reduction from RAINBOW LINE COVER. Let $(\mathcal{P}', \ell)$ be an instance of RAINBOW LINE COVER, where $|\mathcal{P}'| = n$. From $\mathcal{P}'$ we create a set of pairs of points $\mathcal{Q}$. For each pair $P_i \in \mathcal{P}'$ we create a pair of points $Q_i \in \mathcal{Q}$. To do this, for each interval $I = (a, b] \in P_i$ (similarly $\overline{I}$) we create a point p (similarly $\overline{p}$) as follows. If $a < x_1$, then prune the interval such that $a = x_1$. Similarly if $b > x_2$ then make $b = x_2$. Let the length of an interval $I_i = (a_i, b_i]$ be $b_i - a_i$ and len be the largest length among the length of all the intervals in $\mathcal{P}'$. Define $d = len + 1$. Now for each interval $I = (a, b] \in \bigcup_{P \in \mathcal{P}'} P$, we create a point p. Consider two disks $D(a)$ and $D(b)$ of radius d centred at $(a, 0)$ and $(b, 0)$ respectively. Let p be the intersection point above x-axis between $D(a)$ and $D(b)$.

The set Q is the set of pairs of points created as above, one for each $P \in \mathcal{P}'$. The pair $(\mathcal{Q}, \ell)$ is the output of the reduction. Clearly, the reduction takes polynomial time.

▶ **Claim 4.** *There is a rainbow covering for $(\mathcal{P}', \ell)$ if and only if the conflict free Fréchet distance between $\mathcal{Q}$ and ℓ is at most d.*

Proof. Consider a rainbow covering R for $(\mathcal{P}', \ell)$. Now consider the set S constructed from intervals in rainbow R. Since R is a rainbow, S is conflict free. Also as the intervals were covering ℓ, each point on ℓ has a point in S which is at maximum distance of d. Hence the Fréchet distance between $\mathcal{Q}$ and ℓ is at most d.

For the reverse direction, assume the Fréchet distance between $\mathcal{Q}$ and ℓ is d. That is, there exists a sequence $T = (p_1, p_2, \ldots, p_k)$ of conflict free points from $\mathcal{Q}$, which should be traversed in order to attain Fréchet distance d. Here p_i is point in $\mathcal{Q}$ for all $1 \le i \le k$. Assume $\ell = [x_1, x_2]$ and $y_i = [a_i, a_{i+1}]$ be the interval on ℓ nearest to point p_i for all points $p_i \in T$. We have $a_1 = x_1$ and $a_{k+1} = x_2$. Also for all points $z \in y_i, d(z, p_i) \le d$ where $1 \le i \le k$. Thus by construction, corresponding to each y_i, we have an interval $I_i \in \mathcal{P}'$ such that $y_i \subseteq I_i$. Let the set of such intervals be R. Since T is conflict free, R is a rainbow. Also as Fréchet distance is d, $\ell = \bigcup_{i=1}^{k} y_i$. Hence R also covers ℓ. Therefore R is a rainbow covering of ℓ. ◀

This completes the proof of the theorem. ◀

3 Polynomial Algorithm for Semi-discrete Fréchet distance problem

In this section we first prove that Semi-discrete Fréchet Distance problem can be solved in $O(n \log n)$ time. Without loss of generality, assume that the line segment ℓ coincides with the X-axis and has end points $(x_1,0)$ and $(x_2,0)$. Take any point $p_i \in P$ where $p_i = (a_i, b_i)$ and let x be a variable depicting the position of a point on line segment ℓ with $x_1 \leq x \leq x_2$. Then the function $f_i(x)$ representing the distance between the point p_i and x is $f_{p_i}(x) = \sqrt{(x-a_i)^2 + {b_i}^2}$.

For each point $p_i \in P$ we can find out the function $f_i(x)$, where each such function represents one sided hyperbola lying above the X-axis and in interval between x_1 and x_2. Let the lower envelop of such functions defined in the domain $[x_1, x_2]$ be $\Gamma(P)$. Let d^* be the maximum perpendicular distance between $\Gamma(P)$ and ℓ. Then we can see that

▶ **Observation 5.** *d^* is the minimum Fréchet distance between ℓ and P.*

Note that two hyperbolas will intersect at at most one point. To see this, note that solving the two equations $f_{p_i}(x) = \sqrt{(x-a_i)^2 + {b_i}^2}$ and $f_{p_j}(x) = \sqrt{(x-a_j)^2 + {b_j}^2}$ gives only one solution. Thus each hyperbola can appear in the lower envelop at most once.

Before proceeding further let us have a look at Davenport–Schinzel sequence. Davenport–Schinzel sequences were introduced by H. Davenport and A. Schinzel in the 1960s.

▶ **Definition 6.** For two positive integers n and s, a finite sequence $U =< u_1, u_2, u_3, \ldots, u_m >$ is said to be a Davenport–Schinzel sequence of order s (denoted as DS(n, s)-sequence) if it satisfies the following properties:

1. $1 \leq u_i \leq n$ for each $i \leq m$.
2. $u_i \neq u_{i+1}$for each $i < m$.
3. If x and y are two distinct values in the sequence U, then U does not contain a subsequence $\ldots x \ldots y \ldots x \ldots y \ldots$ consisting of $s + 2$ values alternating between x and y.

▶ **Theorem 7** ([17, 7, 2])**.** *The lower envelope of a set $\mathcal{F}$ of n continuous, totally defined, univariate functions, each pair of whose graphs intersects in at most s points, can be constructed in an appropriate model of computation, in $O(\lambda_s(n) \log n)$ time where $\lambda_s(n)$ is the Davenport–Schinzel sequence of order s including n distinct values.*

Since $\lambda_1(n) = n$, by substituting $s = 1$ in Theorem 7, we get,

▶ **Theorem 8.** Semi-discrete Fréchet Distance *problem can be solved in $O(n \log n)$ time.*

4 Approximation algorithm for Conflict-free Fréchet distance problem

In this section we present an approximation algorithm for Conflict-free Fréchet Distance. Let us first define some terminology. As before, assume that the line segment ℓ coincides with the X-axis and has end points $(x_1,0)$ and $(x_2,0)$. For any point set A, denote Semi-discrete Fréchet distance between A and line-segment ℓ by $d^F(A, \ell)$. Also let $\Gamma(A)$ be

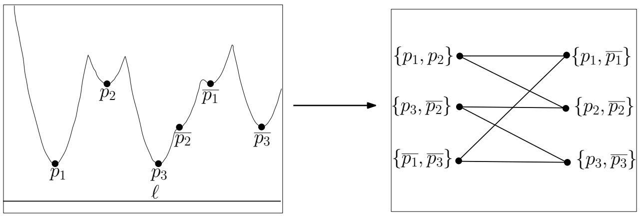

Figure 2 Creating the bipartite graph from the lower envelope.

the lower envelope of the functions $f_{p_i}(x) = \sqrt{(x-a)^2 + b^2}$ for all $p_i = (a, b) \in A$ where $x_1 \leq x \leq x_2$. Now let us start our discussion with the following observation about the Semi-discrete Fréchet distance.

▶ **Observation 9.** *For any set of points A and B where $A \subseteq B$, $d^F(A, \ell) \geq d^F(B, \ell)$.*

Proof. Let $C \subseteq A$ be the set of points that achieves $d^F(A, \ell) = d$. Since $A \subseteq B$, we have that $C \subset B$ and hence $d^F(B, \ell) \leq d$. ◀

Let $(\mathcal{Q} = \{Q_1, Q_2 \ldots Q_n\}, \ell)$ be the input instance of Conflict-free Fréchet Distance, where $Q_i = \{q_i, \overline{q_i}\}$. Let $Q = \bigcup_{i=1}^{n} Q_i$. By Theorem 7 we can find $d^F(Q, \ell)$ in $O(n \log n)$ time. Among all conflict free subsets of Q, assume P^{opt} is a subset that minimizes the Semi-discrete Fréchet distance and let $d^{opt} = d^F(P^{opt}, \ell)$. If $\Gamma(Q)$ contains at most one of f_{q_i} or $f_{\overline{q_i}}$ for each $Q_i = \{q_i, \overline{q_i}\}$, then $d^{opt} = d^F(Q, \ell)$. As $P^{opt} \subseteq Q$, from Observation 9 we have following lemma.

▶ **Lemma 10.** $d^{opt} \geq d^F(Q, \ell)$.

Suppose the set of points for which the corresponding $f_{q_i}(x)$ are in $\Gamma(Q)$ be P'. Observe that if P' does not contain points from the same pair, then $d^F(P', \ell)$ is the conflict free Semi-discrete Fréchet distance and we have $d^{opt} = d^F(Q, \ell) = d^F(P', \ell)$. If not, then our objective is to choose a conflict free subset P'' of P' such that $d^F(P'', \ell) \leq 3d^F(P', \ell)$. First, for all the points $q_i \in P'$ such that $\overline{q_i} \notin P'$, we include q_i in P''. For the rest of the points, let $P_{pair} = \{p_1, p_2, \ldots p_{2k}\}$ be the sorted order of points along x-axis where each $p_i = q_j$ or $\overline{q_j}$ for some j. Now from P_{pair} , we create bags $B_1, B_2, \ldots, B_k$ where $B_i = \{p_{2i-1}, p_{2i}\}$. We construct a bipartite graph $G = (U, V, E)$ where $U = \{B_1, B_2, \ldots, B_k\}$ and V is set of all k pairs $Q_i = \{q_i, \overline{q_i}\}$ such that both q_i and $\overline{q_i}$ are in P_{pair}. We add an edge $e_{ij} = (B_i, Q_j)$, if $B_i \cap Q_j \neq \emptyset$. For an example, see Figure 2.

Now we have the following lemma.

▶ **Lemma 11.** *$G = (U, V, E)$ contains a perfect matching M.*

Proof. Each vertex in U and V has degree at least 1 and at most 2. Also if vertex B_i in U has degree one, then the vertex Q_j to which it is connected in V also has degree one (as it implies that both $B_i = Q_j = \{q_i, \overline{q_i}\}$). Thus every subset W of U has a set of neighbours $N_G(W)$ such that $|W| \leq |N_G(W)|$ (here the neighbours of W is the set of vertices in V to which vertices in W are connected). Hence by Hall's marriage theorem [16], G has a perfect matching M. ◀

Let M be a perfect matching in G. Now for each edge (B_i, Q_j) selected in matching M, if $|B_i \cap Q_j| = 1$ then include $|B_i \cap Q_j|$ in P'', else if $|B_i \cap Q_j| = 2$ then we include one arbitrary point of $B_i \cap Q_j$ in P''. Observe that from each pair of points in P_{pair}, only one point is selected. Thus P'' is conflict free. Now we have following lemma.

▶ **Lemma 12.** $d^F(P^{''}, \ell) \leq 3d^F(Q, \ell)$.

Proof. Since $d^F(Q, \ell) = d^F(P^{'}, \ell)$, it is enough to show that $d^F(P^{''}, \ell) \leq 3d^F(P^{'}, \ell)$. Let π be the sorted order of points in $P^{'}$ along x-axis. We first prove the follwing claim.

▶ **Claim 13.** *For any point $s \in P^{'}$, at least one among s, its predecessor in π and its successor in π, is in P''.*

Proof. We claim that (i) no three consecutive points from π can be in $P' \setminus P''$. For any three consecutive points q_1, q_2, q_3, either one of them does not belong to P_{pair} and thus belongs to P'' or one among $\{q_1, q_2\}$ and $\{q_2, q_3\}$ belongs to P_{pair}. From the construction of P'', we include one among q_1, q_2, q_3, in P''. Now we claim that (ii) at least one among the first two points in π is in P''. Let s_1 and s_2 be the first two points in π. If $\{s_1, s_2\} \not\subseteq P_{pair}$, then $P'' \cap \{s_1, s_2\} \neq \emptyset$. Otherwise $B_1 = \{s_1, s_2\}$ and by the construction of P'', we have that $P'' \cap \{s_1, s_2\} \neq \emptyset$. Similarly we can prove that (iii) at least one among the last two points in π is in P''.

The claim follows from the statements (i),(ii) and (iii). ◀

Let $d = d^F(P^{'}, \ell)$. Now we prove that $d^F(P^{''}, \ell) \leq 3d$. Towards that it is enough to prove that for any point on ℓ, there is a point in $P^{''}$, which is at a distance at most $3d$. For any two points x, y, we use $d(x, y)$ to denote the distance between x and y. Let z be a point in ℓ. Since $d = d^F(P^{'}, \ell)$, there is a point s in $P^{'}$ such that $d(z, s) \leq d$. Now we show that there is a point $s' \in P''$ such that $d(z, s') \leq 3d$. If $s \in P''$, then we set $s' = s$. Otherwise, by Claim 13, either its successor or its predecessor in π belongs to P''. Let s' be a point in P'' which is either successor of s or predecessor of s. Since the $d = d^F(P^{'}, \ell)$, there is a point t on ℓ such that $d(t, s) \leq d$ and $d(t, s') \leq d$. Now we have that $d(z, s) \leq d$, $d(s, t) \leq d$ and $d(t, s') \leq d$. Hence by triangular inequality, we get $d(z, s') \leq 3d$. This completes the proof of the lemma. ◀

▶ **Theorem 14.** *There is a 3-approximation algorithm for* Conflict-free Fréchet Distance.

5 Fixed Parameter Tractable Algorithms

Here we give two FPT algorithms for Parameterized Conflict-free Fréchet distance Problem. The first algorithm is based on randomization and the second is based on branching.

5.1 Randomized algorithm

We give a randomized FPT algorithm which succeeds with a constant success probability. It uses the following problem for which there is a simple greedy algorithm running in time $O(n \log n)$; the algorithm is very similar to that of the Interval Point Cover [13].

Interval Line Cover
Input: A line segment ℓ and a set Q of n intervals on ℓ.
Question: Find a minimum cardinality subset $Q' \subseteq Q$ such that the intervals in Q' cover all the points in the line segment ℓ.

▶ **Theorem 15.** *There is a randomized algorithm for* Parameterized Conflict-free Fréchet Distance *running in time $O(2^k n \log n)$ which outputs* No *for all* No*-instances and outputs* Yes *for all* Yes*-instances with constant probability.*

Proof. Let $|\mathcal{Q}| = n$. The algorithm work as follows. It creates a set S of n points through the following random process. For each $\{q_i, \overline{q}_i\} \in \mathcal{Q}$, it uniformly at random picks one point from $\{q_i, \overline{q}_i\}$ and adds to the set S. Then for each point $p \in S$, the algorithm then computes an interval on ℓ as follows. Draw a circle C_p of radius d with p as the centre. The interval $[a_p, b_p]$ on ℓ is the interval on ℓ covered by the circle C_p. Now run the $O(n \log n)$ algorithm for Interval Line Cover on instance $(\ell, \{[a_p, b_p] \mid p \in S\})$. for the problem If this algorithm returns a solution of size at most k, then our algorithm outputs Yes.

Now we show that if the input instance is an Yes instance, then our algorithm outputs Yes with probability $\frac{1}{2^k}$. Let P^* be a conflict free subset of points of cardinality k such that $d^F(P^*, \ell) \leq d$. Notice that for each $p_i \in P^*$, there is point $\overline{p}_i \notin P^*$ such that $\{p_i, \overline{p}_i\} \in \mathcal{Q}$ and with probability $1/2$ we have added p_i to S. This implies that $\Pr(S = P^*) = \frac{1}{2^k}$. Since each point on ℓ is at a distance at most d to some point P^*, when $S = P^*$, the algorithm of Interval Line Cover outputs Yes Since $\Pr(S = P^*) = \frac{1}{2^k}$ our algorithm output Yes with probability at least $\frac{1}{2^k}$. Suppose input is a No-instance. Then for each conflict free point set P^* of size at most k, $d^F(P^*, \ell) > d$. Also note that the set S we constructed is a conflict free set. Since $d^F(P^*, \ell) > d$, we need more than k intervals from $\{[a_p, b_p] \mid p \in S\}$ to cover ℓ. This implies that the algorithm of Interval Line Cover will return a set of size more than k, and so our algorithm will output No.

We can boost the success probability to a constant by running our algorithm 2^k times. For an Yes instance the algorithm will fail in all 2^k run is at most $(1 - \frac{1}{2^k})^{2^k} \leq \frac{1}{e}$. Since we are running the algorithm of Interval Line Cover 2^k time, the running time mentioned in the theorem follows. ◀

Derandomization

Here, we define matching universal sets. Then we give a derandomization of algorithm for the problem. First we define some notations. For $n \in \mathbb{N}$, let $[n] = \{1, \ldots, n\}$. For a set U, $\binom{U}{k}$ denotes the family of subsets of U, where each subset is of size exactly k.

Matching universal sets for a family of disjoint pairs. Here we define a restricted version of universal sets (defined below) which we call matching universal sets and it is defined for a family of disjoint pairs. We give an efficient construction of these objects by reducing to universal sets. We use it to derandomize our algorithm given in the section. We believe that these objects will add to the list of tools used to derandomize algorithms and will be of independent interest.

▶ **Definition 16** ((n, k)-universal sets [18])**.** Let U be a set of size n. A family of subsets $\mathcal{F}$ of A is called (n, k)-universal sets for U, if for any $A, B \subseteq U$ such that $A \cap B = \emptyset, |A \cup B| = k$, there is a set $F \in \mathcal{F}$ such that $A \subseteq F$ and $F \cap B = \emptyset$

▶ **Lemma 17** ([18])**.** *There is a deterministic algorithm which constructs an (n, k)-universal family of sets of cardinality $2^k k^{O(\log k)} \log n$ in time $2^k k^{O(\log k)} n \log n$.*

▶ **Definition 18.** Let $U = \{a_i, b_i \mid i \in [n]\}$ be a $2n$ sized set and $\mathcal{S} = \{\{a_i, b_i\} \mid i \in [n]\}$ be a family of pairwise disjoint subsets of U. A family of subsets $\mathcal{F}$ of U is called an (n, k)-matching universal family for $\mathcal{S}$, if for each $I \in \binom{[n]}{k}$, and $S \in \binom{U}{k}$ such that $|S \cap \{a_j, b_j\}| = 1$ for all $j \in I$, we have a set $F \in \mathcal{F}$ such that $S \subseteq F$ and $F \cap (\{a_j, b_j \mid j \in I\} \setminus S) = \emptyset$.

Now we use Lemma 17, to get an efficient construction of (n, k)-matching universal sets.

▶ **Theorem 19.** *Given a $2n$ sized set $U = \{a_i, b_i \mid i \in [n]\}$ and a family $\mathcal{S} = \{\{a_i, b_i\} \mid i \in [n]\}$ of pairwise disjoint subsets of U, there is a deterministic algorithm which constructs an (n, k)-matching universal family of cardinality $2^k k^{O(\log k)} \log n$ in time $2^k k^{O(\log k)} n \log n$.*

Proof. Let $U' = \{e_1, \dots, e_n\}$ be a set of size n, where each e_i represents the set $\{a_i, b_i\}$. Now our algorithm first constructs an (n, k)-universal family $\mathcal{F}'$ for the set U' using Lemma 17. Now the algorithm constructs an (n, k)-matching universal sets $\mathcal{F}$ for $\mathcal{S}$ from the family $\mathcal{F}'$ as follows. For each set $F' \in \mathcal{F}'$, it creates a set $F \subseteq U$ of size n and adds to $\mathcal{F}$: for each $e_i \in U'$, if $e_i \in F'$, then it adds a_i to F, otherwise it adds b_i to F.

Notice that $|\mathcal{F}| = |\mathcal{F}'|$, and hence the cardinality of (n, k)-matching universal family mentioned in the theorem follows. Since the algorithm mentioned in Lemma 17 takes time $2^k k^{O(\log k)} n \log n$ and construction of F from F' takes time $O(n)$, the running time of our algorithm is $2^k k^{O(\log k)} n \log n$.

Now we show that $\mathcal{F}$ is indeed an (n, k)-matching universal family for $\mathcal{S}$. Consider a set $I \in \binom{[n]}{k}$ and $S \in \binom{U}{k}$ such that $|S \cap \{a_j, b_j\}| = 1$ for all $j \in I$. Let $A' = S \cap \{a_j \mid j \in I\}$, $B' = \{a_j \mid j \in I\} \setminus A'$ and $C = \{b_j \mid a_j \in B'\}$. Notice that $S = A' \cup C$, $A' \cap B' = \emptyset$ and since $|I| = k$, we have that $|A' \cup B'| = k$. Let $A = \{e_j \mid a_j \in A'\}$ and $B = \{e_j \mid a_j \in B'\}$. Since $A' \cap B' = \emptyset$ and $|A' \cup B'| = k$ we have that $A \cap B = \emptyset$ and $|A \cup B| = k$. By the definition of (n, k)-universal family, we know that there is a set $F' \in \mathcal{F}'$ such that $A \subseteq F'$ and $F' \cap B = \emptyset$. Now consider the set F created corresponding to F'. Since for each $e_j \in A$, $e_j \in F'$, we have that $a_j \in F$. Since for each $e_{j'} \in B$, $e_{j'} \notin F'$, we have that $b_{j'} \in F$. This implies that $A' \subseteq F$ and $C \subseteq F$, and hence $A \cup C = S \subseteq F$. Since $|F \cap \{a_i, b_j\}| = 1$ for all $i \in [n]$ and $S \subseteq F$, we have that $F \cap (\{a_j, b_j \mid j \in I\} \setminus S) = \emptyset$. This completes the proof of the lemma. ◀

Instead of creating the set S by the random process, we can use (n, k)-matching universal family $\mathcal{F}$ for $\mathcal{Q}$ to get a deterministic algorithm. That is for each $S \in \mathcal{F}$, run the algorithm for Interval Line Cover on the input created using ℓ and S as above, and output Yes, if at least once the algorithm for Interval Line Cover returns a solution of size at most k. The correctness of the algorithm follows from the definition of (n, k)-matching universal family. By Theorem 19, the running time to construct $\mathcal{F}$ is $2^k k^{O(\log k)} n \log n$ and $|\mathcal{F}| = 2^k k^{O(\log k)} \log n$. Hence our deterministic algorithm will run in time $2^k k^{O(\log k)} n \log^2 n$. This gives us the following theorem.

▶ **Theorem 20.** *There is a deterministic algorithm for* Parameterized Conflict-free Fréchet Distance *running in time $O(2^k k^{O(\log k)} n \log^2 n)$.*

Note: This technique is especially interesting because the same technique can be used to provide FPT algorithms for similar class of problems. Consider a generalized multiple choice problem $\mathcal{P}(\mathcal{Q}, c)$ where we are given a set $\mathcal{Q}$ with n color classes where each color class contains c objects. The objective is to select minimum number of objects taken at most one from each color class to satisfy certain conditions. If there exists a polynomial time algorithm for $\mathcal{P}(\mathcal{Q}, 1)$ then the same technique gives a randomized c^k algorithm.

5.2 Branching algorithm

For this algorithm, we will consider the more general problem which is the parameterized version of Rainbow Covering.

PARAMETERIZED RAINBOW COVERING	**Parameter:** k

Input: A set of n pairs of intervals $\mathcal{P}$ and a set of points S on X-axis, and $k \in \mathbb{N} \cup \{0\}$.
Question: Is there a rainbow Q of cardinality at most k such that each point in S is covered by at least one interval in Q.

We now give an algorithm based on branching for this problem. The algorithm can be modified to solve PARAMETERIZED CONFLICT-FREE FRÉCHET DISTANCE.

Let $S = \{s_1, s_2, \dots, s_n\}$. Without loss of generality, assume that $s_1, s_2, \dots, s_n$ are sorted in ascending order of their x-coordinates. Now for each interval $I_i \in P_i$ where $P_i \in \mathcal{P}$, assume that the interval is starting not before s_1 and ending not beyond s_n. If not, trim such intervals such that they satisfy above criteria. Also initialize an integer variable $k' = k$.

In the first step, consider the intervals covering s_1. Let the sorted order of these intervals according to their length in descending order be $I_{c_1} = (I_1, I_2, \dots, I_q)$ (here the length of interval $I = [a, b]$ is calculated as $b - a$ where we have $b > a$). Let $s_i \in S$ be the first point right to I_1. If $q = 1$, then choose I_1 in solution, delete I_1,$\bar{I_1}$,s_1, $I_2, \dots, I_q$ and all points covered by I_1. Else if $q > 1$ then we have the following lemma.

▶ **Lemma 21.** *There exists an optimal solution that contains I_1 or $\bar{I_1}$.*

Proof. Suppose the lemma is false. Then we have some other I_j covering s_1. But $I_j \subseteq I_1$ and also $\bar{I_1}$ is not in solution. So we can choose I_1 and delete I_j in our new optimal solution. ◀

Thus we can either choose I_1 in optimal solution or may choose $\bar{I_1}$ in it. If I_1 is chosen then delete I_1, $\bar{I_1}$, s_1, $I_2, \dots, I_q$ and all points covered by I_1. If I_1 is not chosen then put $\bar{I_1}$ in solution, and delete I_1,$\bar{I_1}$, all intervals I_i such that $I_i \subseteq \bar{I_1}$ and all points covered by $\bar{I_1}$. At the end of the first step, put $k' = k' - 1$. For the second step, start with s_i if I_1 is chosen in the previous step. Else consider s_1 again with branching on I_2. Repeat the same procedure till either all points are covered or $k' = 0$. Now if atleast one branch of these $\mathcal{O}(2^k)$ choices covers all the points then accept else reject. The time complexity of this algorithm will be $\mathcal{O}(2^k n^2 \log n)$. Hence we have following theorem.

▶ **Theorem 22.** *There is branching algorithm for* PARAMETERIZED RAINBOW COVERING *running in time $\mathcal{O}(2^k n^2 \log n)$. Similarly, there is a branching algorithm for* PARAMETERIZED CONFLICT-FREE FRÉCHET DISTANCE *with runtime $\mathcal{O}(2^k n^2 \log n)$.*

We observe that the branching algorithm can be used to obtain FPT algorithm for the PARAMETERIZED MINIMUM MAXGAP. Outline of algorithm is as follows. Start from the first point p_s. Take the farthest point from p_s having distance less than d. Let the point chosen be p_i. Then we claim that there exists an optimal solution which contains either p_i or $\bar{p_i}$. So branch on p_i.

5.3 Kernel Lower bound

In this subsection we show that PARAMETRIZED RAINBOW COVERING does not admit a polynomial kernel unless co-NP $\subseteq$ NP/poly. Towards that we first explain one of the tools to prove such a lower bound– called composition.

▶ **Definition 23** (Composition [8]). A composition algorithm (also called OR-composition algorithm) for a parameterized problem $\Pi \subseteq \Sigma^* \times \mathbb{N}$ is an algorithm that receives as input a sequence $((x_1, k), \dots, (x_t, k))$, with $(x_i, k) \in \Sigma^* \times \mathbb{N}$ for each $1 \le i \le t$, uses time polynomial in $\sum_{i=1}^{t} |x_i| + k$, and outputs $(y, k') \in \Sigma^* \times \mathbb{N}$ with (a) $(y, k') \in \Pi \iff (x_i, k) \in \Pi$ for some $1 \le i \le t$ and (b) k' is polynomial in k. A parameterized problem is compositional (or OR-compositional) if there is a composition algorithm for it.

It is *unlikely* that an NP-complete problem has both a composition algorithm and a polynomial kernel as suggested by the following theorem.

▶ **Theorem 24** ([8, 15]). *Let* Π *be a compositional parameterized problem whose unparameterized version* $\tilde{\Pi}$ *is NP-complete. Then, if* Π *has a polynomial kernel then* co-NP $\subseteq$ NP/poly*.*

Towards getting a composition for Parametrized Rainbow Covering, we first show how we can compose two instances and then we use this to get a composition algorithm. Next we have the following lemma.

▶ **Lemma 25.** *There is a polynomial time algorithm which takes two instances* $((\mathcal{P}_1, S_1), k)$ *and* $((\mathcal{P}_2, S_2), k)$ *of* Parametrized Rainbow Covering *as input and outputs an instance* $((\mathcal{P}, S), k+1)$ *such that* $((\mathcal{P}, S), k+1)$ *is an* Yes*-instance of* Parametrized Rainbow Covering *if and only if at least one among* $((\mathcal{P}_1, S_1), k)$ *and* $((\mathcal{P}_2, S_2), k)$ *is a* Yes*-instance of* Parametrized Rainbow Covering*.*

Proof. Let $S_1 = \{s_1, \dots, s_n\}$, and $S_2 = \{s'_1, \dots, s'_n\}$. Without loss of generality assume that $s_1 < s_2 < \dots < s_n$ and $s'_1 < s'_2 < \dots < s'_n$. Without loss of generality we can assume that for any interval J which is part of any pair in $\mathcal{P}_1$ and for any interval J' which is part of any pair in $\mathcal{P}_2$, J is contained in $[s_1, s_n]$ and J' is contained in $[s'_1, s'_n]$. Now we create a set of points $S' = \{s_n + 1 + s'_i \mid i \in [n]\}$, and a pair of intervals $(I, \overline{I}) = ([s_1, s_n], [s_n + 1 + s'_1, s_n + 1 + s'_n])$. Now we shift each interval of the instance $((\mathcal{P}_2, S_2), k)$ by $s_n + 1$. For any interval $J = [a, b]$ and $c \in \mathbb{R}$ we use $c + J$ to denote the interval $[c + a, c + b]$. Let $S = S_1 \cup S'$ and $\mathcal{P} = \mathcal{P}_1 \cup \{(s_n + 1 + J, s_n + 1 + \overline{J}) \mid (J, \overline{J}) \in \mathcal{P}_2\} \cup \{(I, \overline{I})\}$. Our algorithm will output $((\mathcal{P}, S), k+1)$.

Now we need to show the correctness of the algorithm. Suppose $((\mathcal{P}, S), k+1)$ is a Yes-instance of Parametrized Rainbow Covering and let $\mathcal{I}$ be a solution of size $k+1$. We know that at most one of I and $\overline{I}$ belong to $\mathcal{I}$. Hence, if $I \notin \mathcal{I}$, then $\mathcal{I} \setminus \{I\}$ covers all the points in S_1. From the construction of $\mathcal{P}$, we have that all the intervals which intersects $[s_1, s_n]$ are from $\{J, \overline{J} \mid (J, \overline{J}) \in \mathcal{P}_1\}$. This implies that $\mathcal{I} \cap \{J, \overline{J} \mid (J, \overline{J}) \in \mathcal{P}_1\}$ covers all the points in S_1 and $\mathcal{I} \cap \{J, \overline{J} \mid (J, \overline{J}) \in \mathcal{P}_1\}$ is a set of conflict free intervals from $\mathcal{P}_1$. This implies that $((\mathcal{P}_1, S_1), k)$ is a Yes-instance of Parametrized Rainbow Covering. When $\overline{I} \notin \mathcal{I}$, by similar arguments we can show that $((\mathcal{P}_2, S_2), k)$ is a Yes-instance of Parametrized Rainbow Covering.

Suppose one among $((\mathcal{P}_1, S_1), k)$ and $((\mathcal{P}_2, S_2), k)$ is a Yes-instance of Parametrized Rainbow Covering. Assume $((\mathcal{P}_1, S_1), k)$ is a Yes-instance and let $\mathcal{I}$ be a solution of size k for it. Then $\mathcal{I} \cup \{\overline{I}\}$ is a set of conflict free intervals and these intervals cover all the points in S. The case when $((\mathcal{P}_2, S_2), k)$ is a Yes-instance can be proved by similar arguments. ◀

▶ **Lemma 26.** Parametrized Rainbow Covering *is compositional.*

Proof. Let $((\mathcal{P}_1, S_1), k), \dots, (\mathcal{P}_t, S_t), k)$ be the input of the composition algorithm. If $t > 2^k$, then the composition algorithm solves each instance separately using Theorem 22 and outputs a trivial Yes instance if at least one of the given instances is a Yes instance and outputs a trivial No instance otherwise. In this case the running time of the algorithm is bounded by $t^2 n^{O(1)}$ and hence it is a polynomial time algorithm.

So now we can assume that $t \leq 2^k$. Without loss of generality assume that $t = 2^\ell$, where $\ell \leq k$. If t is not a power of 2, we can add dummy No instances to make the total number of instances a power of 2. Now we design a recursive algorithm to get a desired output. The pseudocode is mentioned in Algorithm 1.

By induction on ℓ we show that the parameter in the output instance is $k + \ell$. The base case is when $\ell = 1$, and the statement is true by Lemma 25. Now consider the induction

Algorithm 1: Composition algorithm with inputs $((\mathcal{P}_1, S_1), k), \ldots, ((\mathcal{P}_{2^\ell}, S_{2^\ell}), k)$

1 **if** $\ell = 1$ **then**
2 Run the algorithm mentioned in Lemma 25 and return the result
3 $((\mathcal{P}'_1, S'_1), k') :=$ Algorithm 1($((\mathcal{P}_1, S_1), k), \ldots, ((\mathcal{P}_{2^{\ell-1}}, S_{2^{\ell-1}}), k)$)
4 $((\mathcal{P}'_2, S'_2), k') :=$ Algorithm 1($(\mathcal{P}_{2^{\ell-1}}, S_{2^{\ell-1}}), k), \ldots, ((\mathcal{P}_{2^\ell}, S_{2^\ell}), k)$)
5 Run algorithm mentioned in Lemma 25 on $((\mathcal{P}'_1, S'_1), k')$ and $((\mathcal{P}'_1, S'_1), k')$, and return the result

step. For the two instances created by recursively calling Algorithm 1 on $2^{\ell-1}$ instances, the parameters are $k + \ell - 1$ each, by induction hypothesis. Hence, in Step 5, by Lemma 25, the parameter in the output instance is $k + \ell$. This implies that the parameter in the output instance is $k + \ell \leq 2k$.

Again by induction on ℓ, we can show that the output instance of Algorithm 1 is a Yes instance if and only if at least one of the input instances is a Yes instance. For the base case when $\ell = 1$, the statement is true by Lemma 25. Now consider the induction step. Suppose that there is a Yes instance in the input. Then by induction hypothesis, at least one the instances created in Step 3 or Step 4 is a Yes instance. Then, by Lemma 25, in Step 5, Algorithm 1 will output a Yes instance. Now suppose Algorithm 1 output a Yes instance. Then, by Lemma 25, one of the instances created in Step 3 or Step 4 is a Yes instance. Hence, by induction hypothesis, at least one of the input instances is a Yes instance. ◀

By Theorem 24 and Lemma 26, we get the following theorem.

▶ **Theorem 27.** Parametrized Rainbow Covering *does not admit a polynomial kernel unless co-NP ⊆ NP/poly.*

Acknowledgements. The first author thanks Esther M. Arkin, Paz Carmi, Gui Citovsky, Matthew J. Katz and Joseph S. B. Mitchell for helpful discussion regarding approximation algorithms for multiple choice problems. The authors are grateful to Saket Saurabh for many valuable discussions.

References

1 Pankaj K. Agarwal, Rinat Ben Avraham, Haim Kaplan, and Micha Sharir. Computing the discrete fréchet distance in subquadratic time. *SIAM J. Comput.*, 43(2):429–449, 2014. doi:10.1137/130920526.

2 Pankaj K. Agarwal and Micha Sharir. Davenport-schinzel sequences and their geometric applications, 1998.

3 Helmut Alt and Michael Godau. Computing the fréchet distance between two polygonal curves. *Int. J. Comput. Geometry Appl.*, 5:75–91, 1995. doi:10.1142/S0218195995000064.

4 Esther M. Arkin, Aritra Banik, Paz Carmi, Gui Citovsky, Matthew J. Katz, Joseph S. B. Mitchell, and Marina Simakov. Choice is hard. In *Algorithms and Computation – 26th International Symposium, ISAAC 2015, Nagoya, Japan, December 9-11, 2015, Proceedings*, pages 318–328, 2015. doi:10.1007/978-3-662-48971-0_28.

5 Esther M. Arkin, Aritra Banik, Paz Carmi, Gui Citovsky, Matthew J. Katz, Joseph S. B. Mitchell, and Marina Simakov. Conflict-free covering. In *Proceedings of the 27th Canadian Conference on Computational Geometry, CCCG 2015, Kingston, Ontario, Canada, August 10-12, 2015*, 2015. URL: http://research.cs.queensu.ca/cccg2015/CCCG15-papers/28.pdf.

6 Boris Aronov, Sariel Har-Peled, Christian Knauer, Yusu Wang, and Carola Wenk. Fréchet distance for curves, revisited. In *Algorithms – ESA 2006, 14th Annual European Symposium, Zurich, Switzerland, September 11-13, 2006, Proceedings*, pages 52–63, 2006. doi:10.1007/11841036_8.

7 Mikhail J. Atallah. Some dynamic computational geometry problems. *Computers & Mathematics with Applications*, 11(12):1171–1181, 1985. doi:10.1016/0898-1221(85)90105-1.

8 Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009. doi:10.1016/j.jcss.2009.04.001.

9 Mario E. Consuegra and Giri Narasimhan. Geometric Avatar Problems. In Anil Seth and Nisheeth K. Vishnoi, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2013)*, volume 24 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 389–400, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.FSTTCS.2013.389.

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

11 Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In *STOC*, pages 251–260, 2010. doi:10.1145/1806689.1806725.

12 R. Downey and M. Fellows. *Fundamentals of parameterized complexity*. Springer, 2013.

13 Jeff Edmonds. *How to Think About Algorithms*. Cambridge University Press, New York, NY, USA, 2008.

14 Thomas Eiter and Heikki Mannila. Computing discrete fréchet distance. Technical report, Citeseer, 1994.

15 Lance Fortnow and Rahul Santhanam. Infeasibility of instance compression and succinct PCPs for NP. In *STOC*, pages 133–142, 2008. doi:10.1145/1374376.1374398.

16 P. Hall. On representatives of subsets. *Journal of the London Mathematical Society*, s1-10(1):26–30, 1935. URL: http://jlms.oxfordjournals.org/content/s1-10/1/26.short, arXiv:http://jlms.oxfordjournals.org/content/s1-10/1/26.full.pdf+html, doi:10.1112/jlms/s1-10.37.26.

17 John Hershberger. Finding the upper envelope of n line segments in $o(n \log n)$ time. *Information Processing Letters*, 33(4):169–174, 1989. doi:10.1016/0020-0190(89)90136-1.

18 Moni Naor, Leonard J. Schulman, and Aravind Srinivasan. Splitters and near-optimal derandomization. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, 23-25 October 1995*, pages 182–191, 1995. doi:10.1109/SFCS.1995.492475.

19 Kaveh Shahbaz. Applied similarity problems using fréchet distance. *CoRR*, abs/1307.6628, 2013. URL: http://arxiv.org/abs/1307.6628.