3

Parameterized Complexity of Conflicts as Matching

In this chapter, we study the problems where the conflict graph is a matching. We start with semi-discrete Fréchet distance which is based on the following intuition as mentioned in Chapter 1. Consider a girl walking her dog on a leash. Given a continuous curve S and a set of points P, semi-discrete Fréchet distance is the minimum length of a leash that simultaneously allows the girl 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. Here we consider the case when S is a line segment. We denote it by ℓ. We take the cardinality of P to be n. More formally, let α is a continuous, non-decreasing, surjection from $[0,1]$ to ℓ. Also assume β be any function from $[0,1]$ to P such that there exist disjoint subdivisions of $[0,1]$ into a set of intervals $\lambda_1, \lambda_2, ..\lambda_k$ for some $k \in \mathbb{N}, k \leq n$. We have $\bigcup_{i=1}^{k} \lambda_i = [0,1]$. For any two points $t_1, t_2 \in [0,1]$, $\beta(t_1) = \beta(t_2)$ if and only if t_1 and t_2 belongs to same interval. Let $d(a,b)$ for two points a and b is Euclidean distance between a and b. Semi-discrete Fréchet distance between ℓ and P is defined as:

$$d^F(\ell, P) = \underset{\alpha,\beta}{inf} \max_{t\in[0,1]} \{d(\alpha(t), \beta(t))\}$$

We define the problem Semi-discrete Fréchet Distance as follows.

Semi-discrete Fréchet Distance **Input:** A set of points P and a line-segment ℓ in $\mathbb{R}^2$. **Question:** Find $d^F(P, \ell)$.

Along with Fréchet distance, Hausdorff distance [44] is another popular measure for closeness of two set of points A and B. Intuitively, two sets of points are close if every point in either set has a point close to it in another set. In other words, it is the greatest of all the distances from a point in either set to the closest point in the other set. Formally, Hausdorff distance between A and B can be defined as

$$\mathscr{H}(A,B) = \max\{\sup_{a\in A} \inf_{b\in B} d(a,b), \sup_{b\in B} \inf_{a\in A} d(a,b)\}$$

We define semi-discrete Hausdorff distance between a point set $P = \{p_1, p_2, \dots p_n\}$ and a line-segment ℓ in $\mathbb{R}^2$ as

$$d^H(P, \ell) = \min_{P' \subseteq P, P' \neq \emptyset} \{\mathscr{H}(\ell, P')\}$$

Formally, we may define the problem Semi-discrete Hausdorff Distance as

Semi-discrete Hausdorff Distance **Input:** A set of points P and a line-segment ℓ in $\mathbb{R}^2$. **Question:** Find $d^H(P, \ell)$.

On similar lines we define conflict-free semi-discrete Hausdorff distance between a set of pairs of points $\mathscr{Q}$ and a line-segment ℓ to be minimum semi-discrete Hausdorff distance between ℓ and any conflict-free choice of points from $\mathscr{Q}$.

Now we have a simple observation which follows from the definitions of semi-discrete Fréchet distance and semi-discrete Hausdorff distance.

Observation 3.0.1. *Semi-discrete Fréchet distance and semi-discrete Hausdorff distance are equal.*

Proof. Consider a set of points $P = \{p_1, p_2, \ldots, p_n\}$ and a line-segment ℓ, both lying in the Euclidean plane. Let $d = d^F(P, \ell)$. We need to show that $d^H(P, \ell) = d$.

Since $d = d^F(P, \ell)$, there exist two points $p_i, p_j \in P$ and $l_k \in \ell$ such that $d(p_i, l_k) = d(p_j, l_k) = d$ and there does not exist any point $p \in P$ with $d(p, l_k) < d$. Thus $\inf_{p_j \in P} d(p_j, l_k) = d$. Hence, $d^H(P, \ell) \geq d$. Let $P' = \{p_1, p_2, \ldots, p_q\} \subseteq P$ be the minimum cardinality set such that $d^F(P', \ell) = d$. Hence, for all $p_i \in P'$, $\inf_{l' \in \ell} d(p_i, l') \leq d$. Also, since $d = d^F(P', \ell)$, for all points $l' \in \ell$, there exist a point $p' \in P'$ such that $d(l', p') \leq d$. Thus, for all $l' \in \ell$, $\inf_{p' \in P'} d(p', l') \leq d$. Hence, $\mathscr{H}(P', \ell) \leq d$. So, we have $d^H(P, \ell) = d$ and thus $d^H(P, \ell) = d^F(P, \ell)$. $\square$

Thus all the results we present here for semi-discrete Fréchet distance also hold for semi-discrete Hausdorff distance as both are equivalent measures of distance between a line and a set of points.

Now we recall the decision version and parameterized version of Semi-discrete Fréchet distance problems in conflict settings,

> CONFLICT-FREE FRÉCHET DISTANCE (DECISION VERSION)
> **Input:** A set $\mathscr{Q}$ 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 $P^* \subset \bigcup_{Q \in \mathscr{Q}} Q$ such that $d^F(P^*, \ell) \leq d$.

> PARAMETERIZED CONFLICT-FREE FRÉCHET DISTANCE **Parameter:** k
> **Input:** A set $\mathscr{Q}$ of pairs of points, a line-segment ℓ in $\mathbb{R}^2$, $d \in \mathbb{R}$, and $k \in \mathbb{N}$.
> **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" problem. Recall that here, given a set of points $x_1, \ldots, x_n$ on a line, $maxGap$ is the largest gap between consecutive points in the sorted order. The "gap" refers to Euclidean distance between two points. The problem is as follows.

> PARAMETERIZED MINIMUM MAXGAP **Parameter:** k
> **Input:** A set $\mathscr{Q}$ 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.

Results in this Chapter We first give polynomial time algorithm for SEMI-DISCRETE FRÉCHET DISTANCE in 3.1. Next, in section 3.2 we prove that CONFLICT-FREE FRÉCHET DISTANCE (DECISION VERSION) is NP-Complete. Later in section 3.3, we use two approaches to give FPT algorithm for these problems. Finally in Section 3.4 we show that the problem is unlikely to have a polynomial sized kernel using OR-composition.

3.1 POLYNOMIAL TIME ALGORITHM FOR SEMI-DISCRETE FRÉCHET DISTANCE

In this section we prove that SEMI-DISCRETE FRÉCHET DISTANCE problem is solvable 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 envelope 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 3.1.1. *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 while solving the two equations $f_{p_i}(x) = \sqrt{(x-a_i)^2 + {b_i}^2}$ and $f_{p_j}(x) = \sqrt{(x-a_j)^2 + {b_j}^2}$, the terms $(x-a_i)^2, b_i^2, (x-a_j)^2, b_j^2$ are all positive and thus using distance property, gives only one solution. Thus each hyperbola can appear in the lower envelope 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 3.1.1. *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 3.1.2 ([45] [7] [1])**.** *The lower envelope of a set $\mathscr{F}$ 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 3.1.2, we get,

Theorem 3.1.3. SEMI-DISCRETE FRÉCHET DISTANCE *problem can be solved in $O(n\log n)$ time.*

3.2 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 [3].

Let us begin our discussion by defining Rainbow covering problem. Suppose that we are given a set $\mathscr{P} = \{P_1, P_2, \ldots, P_n\}$ 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 is a conflict-free choice. An interval $[a,b]$ is said to cover a point c if $c \in [a,b]$. The formal definition of the Rainbow covering problem is as follows.

RAINBOW COVERING
Input: A set of pairs of intervals $\mathscr{P}$ and a set of points S 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 [3]. We introduce an intermediate problem called RAINBOW LINE COVER and show that it is NP-complete using a polynomial time many to one reduction from RAINBOW COVERING. Then we give a polynomial time many to one reduction from RAINBOW LINE COVER to CONFLICT-FREE FRÉCHET DISTANCE (DECISION VERSION).

RAINBOW LINE COVER
Input: Set $\mathscr{P}' = \{P'_1, P'_2, \ldots, P'_m\}$ where each P'_i contains a pair of intervals $\{I_j^{in}, \overline{I_j}^{in}\}$ 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 3.2.1. RAINBOW LINE COVER *is* NP-*hard.*

Proof. The proof is by a polynomial time reduction from RAINBOW COVERING. Let $(\mathscr{P}, S)$ be an instance of RAINBOW COVERING and let $S = \{s_1, \ldots, s_n\}$. 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 $\mathscr{P}$ covers at least one point in S. Also, all the intervals are pruned to lie between s_1 and s_n. We create an instance $(\mathscr{P}', \ell^{in})$ of RAINBOW LINE COVER as follows.

For each interval present in $\mathscr{P}$, we create an interval in $\mathscr{P}'$ and the pairs in the new set $\mathscr{P}'$ corresponds to the respective pairs in $\mathscr{P}$. For convenience we assume that $s_0 = s_1$ and $s_{n+1} = s_n$. For an interval I, we construct another interval I' as follows. Let $\{s_i, \ldots, s_j\} \subseteq S$ be the set of points in S covered by I. Then we set $I' = [s_i - |\frac{s_i - s_{i-1}}{2}|, s_j + |\frac{s_{j+1} - s_j}{2}|]$. Now suppose $s_1 = (a_1, 0)$ and $s_n = (a_n, 0)$, then ℓ^{in} is the line-segment on X-axis is $[a_1, a_n]$. Clearly the construction of $(\mathscr{P}', \ell^{in})$ takes polynomial time. Towards the correctness we prove the following claim.

Claim 3.2.1. *There exists a rainbow from $\mathscr{P}$ of size d, covering S, if and only if there exists a rainbow from $\mathscr{P}'$ of size d covering ℓ^{in}.*

Proof. Let $Q \subseteq \bigcup_{i=1}^{n} P_i$ be a rainbow covering S. Observe that corresponding to each interval in $\mathscr{P}$, there is an interval in $\mathscr{P}'$. Let the set of intervals corresponding to Q in $\mathscr{P}'$ be Q^{in}. Since Q is a rainbow, Q^{in} is also a rainbow. Now it suffices to show that every point $q \in \ell^{in}$ is covered by intervals in Q^{in}.

For any point $s \in S$, we also use s to denote the x-coordinate of the point s (note that the y-coordinate is 0). By construction, if a point $s_i \in S$ is covered by an interval $I \in Q$, then s_i is covered by $I' \in Q^{in}$. As Q covers all the points in S, Q^{in} covers S. Now we show that every point in ℓ^{in} which is not in S is also covered by Q^{in}. By the construction of ℓ^{in}, for point z in $\ell^{in} \setminus S$, there exists $s_k, s_{k+1} \in S$ such that $z \in [s_k, s_{k+1}]$. Suppose that there is an interval $I \in Q$ which covers s_k and s_{k+1}. Then the corresponding interval I' covers all the points in $[s_k, s_{k+1}]$ and $I' \in Q^{in}$. Suppose there is no interval in Q, which covers both s_k and s_{k+1}. Also, since Q covers S, there are intervals I_1 and I_2 in Q such that I_1 covers s_k, but not s_{k+1} and, I_2 covers s_{k+1}, but not s_k. By construction $I'_1 = [a, s_k + |\frac{s_{k+1} - s_k}{2}|]$ and $I'_2 = [s_{k+1} - |\frac{s_{k+1} - s_k}{2}|, b]$ for some $a \leq s_k$ and some $b \geq s_{k+1}$. This implies that I'_1 and I'_2 together cover $[s_k, s_{k+1}]$. Moreover $I'_1, I'_2 \in Q^{in}$. Thus Q^{in} covers $[s_k, s_{k+1}]$.

Similarly if there is a 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. For any interval I and the corresponding interval I', $I \cap S = I' \cap S$, by our construction. Thus, since Q^{in} covers S, Q also covers S.

□

This concludes the proof of the lemma. □

Now we have the following theorem.

Theorem 3.2.1. CONFLICT-FREE FRÉCHET DISTANCE (DECISION VERSION) *is* NP*-complete.*

Proof. Given a sorted sequence of points $Q = \{q'_1, q'_2, \dots q'_j\}$ as witness, we can check in linear time whether the points in the sequence are conflict-free. To check whether Fréchet distance is at most d given Q, we use the Theorem 3.1.3 (in Section 3.1) which takes $\mathcal{O}(n \log n)$ time. Thus the problem is in NP.

To prove NP-hardness we give a polynomial time reduction from RAINBOW LINE COVER. Let $(\mathscr{P}', \ell)$ be an instance of RAINBOW LINE COVER, where $|\mathscr{P}'| = n$ and $\ell = [x_1, x_2]$. From $\mathscr{P}'$ we create a set of pairs of points $\mathscr{Q}$ such that for each interval pair $\{I, \bar{I}\} = P \in \mathscr{P}$ we construct a corresponding pair of points $\{q, \bar{q}\} = Q \in \mathscr{Q}$.

To do this, for interval $I = [a, b]$ 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 any interval $I_i = [a_i, b_i]$ be $len_{I_i} = b_i - a_i$ and $d = \{\max_{I_j \in \bigcup_{i=1}^{n} \{I_i, \bar{I}_i\}} len_{I_j}\} + 1$.

Now for interval $I = [a, b]$, draw two circles $C(a)$ and $C(b)$ with centre at a and b, respectively, of radius d each. Let $C(a) \cap C(b) = \{(x, y_1), (x, y_2)\}$. Take $q = (x, y)$ where $y \geq 0$ and $y \in \{y_1, y_2\}$. See Figure 3.1 for an illustration. Let q' be the point constructed for $\bar{I}$. Add $\{q, q'\}$ to $\mathscr{Q}$. Now, $(\mathscr{Q}, \ell)$ is the output of the reduction. Clearly, the reduction takes polynomial time. The correctness of the reduction is proved in the following claim.

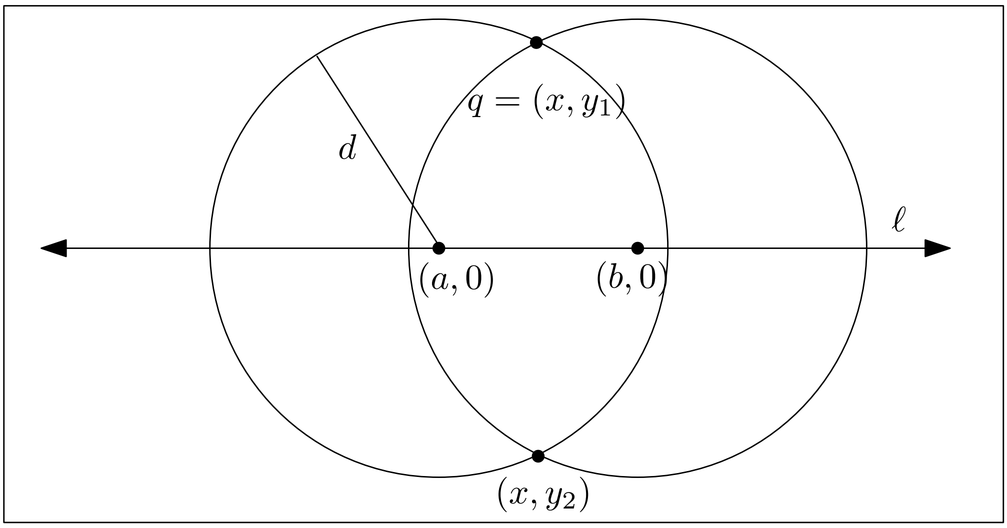

Figure 3.1 : Reduction from RAINBOW LINE COVER PROBLEM

Claim 3.2.2. *There is a rainbow covering for $(\mathscr{P}', \ell)$ if and only if the conflict-free Fréchet distance between $\mathscr{Q}$ and ℓ is at most d.*

Proof. Consider a rainbow covering R for $(\mathscr{P}',\ell)$. Now consider the set S constructed from intervals in the rainbow R. Since R is a rainbow, S is conflict-free. Consider any point z in ℓ and let $I \in R$ be the interval covering z. Let q be the point created for I. Then $d(q,z) \leq d$ (see Figure 3.1). 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 $\mathscr{Q}$ and ℓ is at most d.

For the reverse direction, assume that the conflict-free Fréchet distance between $\mathscr{Q}$ and ℓ is d. Hence, there exists a rainbow $T = \{q_1,q_2,\dots,q_k\} \subset \bigcup_{1=1}^{n} Q_i$ where $Q_i \in \mathscr{Q}$ such that $d^F(T,\ell) \leq d$. Corresponding to each $q \in T$, there is an interval I_q in some pair in $\mathscr{P}'$. For each point q, the set of points in ℓ which are at a distance at most d are in the interval I_q. Moreover each point in I_q is at a distance at most d from q (see Figure 3.1). This implies that $\mathscr{I} = \{I_q \mid q \in T\}$ covers ℓ, because $d^F(T,\ell) \leq d$. Since T is conflict-free, R is a rainbow. Therefore, $\mathscr{I}$ is a rainbow covering for ℓ. □

This concludes the proof of the theorem. □

3.3 FIXED PARAMETER TRACTABLE ALGORITHMS

Here, we give two FPT algorithms for PARAMETERIZED CONFLICT-FREE FRÉCHET DISTANCE PROBLEM. Our first FPT algorithm is based on randomization and the second is based on branching.

3.3.1 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 $\mathscr{O}(n \log n)$; the algorithm is very similar to that of INTERVAL POINT COVER [27].

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 3.3.1. *There is a randomized algorithm for* PARAMETERIZED CONFLICT-FREE FRÉCHET DISTANCE *running in time $\mathscr{O}(2^k n \log n)$ which outputs* NO *for all* NO*-instances and outputs* YES *for all* YES*-instances with constant probability.*

Proof. Let $|\mathscr{Q}| = n$. The algorithm works as follows. It creates a set S of n points through the following random process. For each $\{q_i,\overline{q}_i\} \in \mathscr{Q}$, uniformly at random it 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 $\mathscr{O}(n \log n)$ algorithm for INTERVAL LINE COVER on the instance $(\ell,\{[a_p,b_p] \mid p \in S\})$. 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 $\{p_i,\overline{p}_i\} \in \mathscr{Q}$ 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. $\square$

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 3.3.1 ((n,k)-universal sets [70])**.** *Let U be a set of size n. A family of subsets $\mathscr{F}$ of A is called the* (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 $F \in \mathscr{F}$ such that $A \subseteq F$ and $F \cap B = \emptyset$*

Lemma 3.3.1 ([70])**.** *There is a deterministic algorithm which constructs an (n,k)-universal family of sets of cardinality $2^k k^{\mathscr{O}(\log k)} \log n$ in time $2^k k^{\mathscr{O}(\log k)} n \log n$.*

Definition 3.3.2. *Let $U = \{a_i, b_i \mid i \in [n]\}$ be a $2n$ sized set and $\mathscr{S} = \{\{a_i,b_i\} \mid i \in [n]\}$. A family of subsets $\mathscr{F}$ of U is called an (n,k)-*matching universal family for *$\mathscr{S}$, 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 $F \in \mathscr{F}$ such that $S \subseteq F$ and $F \cap (\{a_j,b_j \mid j \in I\} \setminus S) = \emptyset$.*

Now we use Lemma 3.3.1, to get an efficient construction of (n,k)-matching universal sets.

Theorem 3.3.2. *Given a $2n$ sized set $U = \{a_i,b_i \mid i \in [n]\}$ and the family $\mathscr{S} = \{\{a_i,b_i\} \mid i \in [n]\}$, there is a deterministic algorithm which constructs an (n,k)-matching universal family of cardinality $2^k k^{\mathscr{O}(\log k)} \log n$ in time $2^k k^{\mathscr{O}(\log k)} n \log n$.*

Proof. Let $U' = \{e_1,\ldots,e_n\}$ be a set of size n, where each e_i represents the set $\{a_i,b_i\}$. Next our algorithm first constructs an (n,k)-universal family $\mathscr{F}'$ for the set U' using Lemma 3.3.1. Now the algorithm constructs an (n,k)-matching universal sets $\mathscr{F}$ for $\mathscr{S}$ from the family $\mathscr{F}'$ as follows. For each set $F' \in \mathscr{F}'$, it creates a set $F \subseteq U$ of size n and adds to $\mathscr{F}$: 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 $|\mathscr{F}| = |\mathscr{F}'|$, and hence the cardinality of (n,k)-matching universal family mentioned in the theorem follows. Since the algorithm mentioned in Lemma 3.3.1 takes time $2^k k^{\mathscr{O}(\log k)} n \log n$ and construction of F from F' takes time $\mathscr{O}(n)$, and the running time of our algorithm is $2^k k^{\mathscr{O}(\log k)} n \log n$.

Now we show that $\mathscr{F}$ is indeed an (n,k)-matching universal family for $\mathscr{S}$. 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 $F' \in \mathscr{F}'$ 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.

$\square$

Now, in the proof of Theorem 3.3.1, instead of creating the set S by the random process, we use the sets in a (n,k)-matching universal family $\mathscr{F}$ for $\mathscr{Q}$ to get a deterministic algorithm. That is for each $S \in \mathscr{F}$ obtained using Theorem 3.3.2, we run the algorithm for INTERVAL LINE COVER on the input created using ℓ and S as above. We 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 3.3.2, the running time to construct $\mathscr{F}$ is $2^k k^{\mathscr{O}(\log k)} n \log n$ and $|\mathscr{F}| = 2^k k^{\mathscr{O}(\log k)} \log n$. Hence our deterministic algorithm will run in time $2^k k^{\mathscr{O}(\log k)} n \log^2 n$. This gives us the following theorem.

Theorem 3.3.3. *There exists a deterministic algorithm for* PARAMETERIZED CONFLICT-FREE FRÉCHET DISTANCE *running in time $2^k k^{\mathscr{O}(\log k)} n \log^2 n$.*

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 $\mathscr{P}(\mathscr{Q}, c)$ where we are given a set $\mathscr{Q}$ 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 $\mathscr{P}(\mathscr{Q}, 1)$ then the same technique gives a randomized $\mathscr{O}(c^k n^y)$ algorithm where y is a constant.

3.3.2 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 $\mathscr{P}$, 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?	

Recall that a subset $C \subset \bigcup_{P \in \mathscr{P}} P$ is called a rainbow if and only if it is a conflict-free set. We now give an algorithm based on branching for this problem. The algorithm can be modified to solve PARAMETERIZED CONFLICT-FREE FRÉCHET DISTANCE. In order to solve PARAMETERIZED CONFLICT-FREE FRÉCHET DISTANCE, we reduce PARAMETERIZED CONFLICT-FREE FRÉCHET DISTANCE to PARAMETERIZED RAINBOW COVERING in linear time. For each point $p_i \in \bigcup_{i=1}^{n} Q_i$ where $Q_i \in \mathscr{Q}$, draw a disk $D(p_i)$ of radius d. Now, $I_i = \ell \cap D(p_i)$ represents the interval corresponding to point p_i. Thus, we have a pair of intervals corresponding to each pair of points given in $\mathscr{Q}$. This set of pair of intervals corresponds to $\mathscr{P}$ in PARAMETERIZED RAINBOW COVERING. Let, the union of all the intervals in $\mathscr{P}$ covers ℓ. If not, return NO. For an interval $I = [a,b]$, we call the points $(a,0),(b,0)$ as end-points of I. The set of end-points of all intervals in $\bigcup_{i=1}^{n} P_i$ where $P_i \in \mathscr{P}$ represents S in PARAMETERIZED RAINBOW COVERING. Now, a solution to PARAMETERIZED CONFLICT-FREE FRÉCHET DISTANCE is also a solution to PARAMETERIZED RAINBOW COVERING.

Let $S = \{s_1, s_2, \ldots, s_n\}$. Without loss of generality, assume that $s_1, s_2, \ldots, s_n$ are sorted in ascending order of their x-coordinates. Now for each interval $I_i \in P_i$ where $P_i \in \mathscr{P}$, 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 $I_{c_1} = (I_1, I_2, \ldots, I_q)$ (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 and all points covered by I_1. Else if $q > 1$ then we have the following lemma.

Lemma 3.3.2. *If the given instance has a solution of size at most k, then there exists an optimal solution containing either 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 we 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, and all intervals contained in I_1. If I_1 is not chosen, then place $\bar{I_1}$ 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 at least 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 $\mathcal{O}(2^k n \log n)$. Hence we have the following theorem.

Theorem 3.3.4. *There exists a branching algorithm for* PARAMETERIZED RAINBOW COVERING *running in time* $\mathcal{O}(2^k n \log n)$. *Similarly, there is a branching algorithm for* PARAMETERIZED CONFLICT-FREE FRÉCHET DISTANCE *with runtime* $\mathcal{O}(2^k n \log n)$.

We also consider parameterized version of "minimum maxGap" problem given 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.

We observe that the branching algorithm can be used to obtain FPT algorithm for the PARAMETERIZED MINIMUM MAXGAP. Thus we have the following theorem.

Theorem 3.3.5. *There is branching algorithm for* PARAMETERIZED MINIMUM MAXGAP *running in time* $\mathcal{O}(2^k n \log n)$.

Proof. Let $k' = k+1$. 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. We make the following claim.

Claim 3.3.1. *If the given instance has a solution of size at most k, then there exists an optimal solution containing either p_i or $\bar{p}_i$.*

Proof. Suppose the claim is not true. Then we have some other point p_j in our solution such that $d(p_s, p_j) < d(p_s, p_i)$. Hence, we can remove p_j from our solution and add p_i to our new optimal solution which is a contradiction.

□

Now branch on p_i and reduce k' by 1. If p_i is in solution, then we delete $\bar{p}_i$. We again start from p_i and choose the point that is farthest from p_i and has distance less than d. If $\bar{p}_i$ is in solution, then we delete p_i. We again start from p_s and choose the point that is farthest from p_s and has distance less than d. We continue with the branching till either we reach p_e or $k' = 0$. So, if at least one branch of these $\mathcal{O}(2^k)$ reaches p_e then accept, else reject. The time complexity of this algorithm is $\mathcal{O}(2^k n \log n)$. This completes the proof of the theorem.

□

3.4 KERNEL LOWER BOUND

We recall that a parameterized problem is a language $\Pi \subseteq \Sigma^* \times \mathbb{N}$ where Σ is an alphabet over which the language is defined and the natural number associated is called parameter. We say Π admits a *polynomial kernel* if any instance (I,k) can be reduced to an equivalent instance (I',k'), in polynomial time with respect to $|I|$ and k, such that $|I'|+k'$ is bounded by a polynomial function in k. 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 3.4.1 (Composition [10]). *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 $((x_1,k),\ldots,(x_t,k))$, 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 (as it is considered unlikely that co-NP $\subseteq$ NP/poly.

Theorem 3.4.1 ([10, 34]). *Let Π be a compositional parameterized problem whose unparameterized version $\tilde{\Pi}$ is NP-complete. 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 3.4.1. *There is a polynomial time algorithm which takes two instances $((\mathscr{P}_1,S_1),k)$ and $((\mathscr{P}_2,S_2),k)$ of* PARAMETRIZED RAINBOW COVERING *as input and outputs an instance $((\mathscr{P},S),k+1)$ such that $((\mathscr{P},S),k+1)$ is an* YES*-instance of* PARAMETRIZED RAINBOW COVERING *if and only if at least one among $((\mathscr{P}_1,S_1),k)$ and $((\mathscr{P}_2,S_2),k)$ is a* YES*-instance of* PARAMETRIZED RAINBOW COVERING.

Proof. Let $S_1 = \{s_1,\ldots,s_n\}$, and $S_2 = \{s'_1,\ldots,s'_n\}$. Without loss of generality assume that $s_1 < s_2 < \ldots < s_n$ and $s'_1 < s'_2 < \ldots < s'_n$. Without loss of generality we can assume that for any interval J which is part of any pair in $\mathscr{P}_1$ and for any interval J' which is part of any pair in $\mathscr{P}_2$, 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 $(I,\bar{I}) = ([s_1,s_n],[s_n+1+s'_1,s_n+1+s'_n])$. Now we shift each interval of the instance $((\mathscr{P}_2,S_2),k)$ 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 $\mathscr{P} = \mathscr{P}_1 \cup \{(s_n+1+J, s_n+1+\bar{J}) \mid (J,\bar{J}) \in \mathscr{P}_2\} \cup \{(I,\bar{I})\}$. Our algorithm will output $((\mathscr{P},S),k+1)$.

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

Suppose one among $((\mathscr{P}_1,S_1),k)$ and $((\mathscr{P}_2,S_2),k)$ is a YES-instance of PARAMETRIZED RAINBOW COVERING. Assume $((\mathscr{P}_1,S_1),k)$ is a YES-instance and let $\mathscr{I}$ be a solution of size k for it. Then $\mathscr{I} \cup \{\bar{I}\}$ is a set of conflict-free intervals and these intervals cover all the points in S. The case when $((\mathscr{P}_2,S_2),k)$ is a YES-instance can be proved by similar arguments. □

Lemma 3.4.2. PARAMETRIZED RAINBOW COVERING *is compositional.*

Proof. Let $((\mathscr{P}_1, S_1), k), \ldots, (\mathscr{P}_t, S_t), k)$ be the input of the composition algorithm. If $t > 2^k$, then the composition algorithm solves each instance separately using Theorem 3.3.4 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 $t^2 n^{\mathscr{O}(1)}$ 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.

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

```
1 if ℓ = 1 then
2     Run the algorithm mentioned in Lemma 3.4.1 and return the result
3 ((P'_1, S'_1), k') := Algorithm 1( ((P_1, S_1), k), ..., ((P_{2^{ℓ-1}}, S_{2^{ℓ-1}}), k) )
4 ((P'_2, S'_2), k') := Algorithm 1( (P_{2^{ℓ-1}+1}, S_{2^{ℓ-1}+1}), k), ..., ((P_{2^ℓ}, S_{2^ℓ}), k) )
5 Run algorithm mentioned in Lemma 3.4.1 on ((P'_1, S'_1), k') and ((P'_1, S'_1), k'), and return the result
```

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 3.4.1. Now consider the induction 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 3.4.1, 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 3.4.1. 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 3.4.1, in Step 5, Algorithm 1 will output a YES instance. Now suppose Algorithm 1 output a YES instance. Then, by Lemma 3.4.1, 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 3.4.1 and Lemma 3.4.2, we get the following theorem.

Theorem 3.4.2. PARAMETRIZED RAINBOW COVERING *does not admit a polynomial kernel unless* co-NP $\subseteq$ NP/poly.