Parameterized Complexity of Geometric Covering Problems Having Conflicts ⋆

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

[1] Indian Institute of Technology, Jodhpur, India.
{aritrabanik|sahlotvibha}@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|saket}@imsc.res.in

Abstract. The input for the Geometric Coverage problem consists of a pair $\Sigma = (P, \mathcal{R})$, where P is a set of points in $\mathbb{R}^d$ and $\mathcal{R}$ is a set of subsets of P defined by the intersection of P with some geometric objects in $\mathbb{R}^d$. These coverage problems form special instances of the Set Cover problem which is notoriously hard in several paradigms including approximation and parameterized complexity. Motivated by what are called *choice problems* in geometry, we consider a variation of the Geometric Coverage problem where there are conflicts on the covering objects that precludes some objects from being part of the solution if some others are in the solution.

As our first contribution, we propose two natural models in which the conflict relations are given: (a) by a graph on the covering objects, and (b) by a representable matroid on the covering objects. We consider the parameterized complexity of the problem based on the structure of the conflict relation. Our main result is that as long as the conflict graph has bounded arboricity (that includes all the families of intersection graphs of low density objects in low dimensional Euclidean space), there is a parameterized reduction to the problem without conflicts on the covering objects. This is achieved through a randomization-derandomization trick. As a consequence, we have the following results when the conflict graph has bounded arboricity.

– If the Geometric Coverage problem is fixed parameter tractable (FPT), then so is the conflict free version.
– If the Geometric Coverage problem admits a factor α-approximation, then the conflict free version admits a factor α-approximation algorithm running in FPT time.

As a corollary to our main result we get a plethora of approximation algorithms running in FPT time. Our other results include an FPT algorithm and a W[1]-hardness proof for the conflict-free version of Covering Points by Intervals. The FPT algorithm is for the case when the conflicts are given by a representable matroid, and the W[1]-hardness result

⋆ Supported by Parameterized Approximation, ERC Starting Grant 306992, and Rigorous Theory of Preprocessing, ERC Advanced Investigator Grant 267959.

F. Ellen et al. (Eds.): WADS 2017, LNCS 10389, pp. 61–72, 2017.
DOI: 10.1007/978-3-319-62127-2_6

is for all the families of conflict graphs for which the INDEPENDENT SET problem is W[1]-hard.

1 Introduction, Motivation, Model and Our Results

There are many real life geometric covering problems, for which there exist additional constrains that need to be enforced. In this paper, we attempt to address these problems and hope that this will initiate a new line of research directed at bridging the gap between theory and practice.

To define our model of covering with conflicts, we start by defining the classic covering problem. The input to a covering problem consists of a universe U of size n, a family $\mathcal{F}$ of size m of subsets of U and a positive integer k. Our objective is to check whether there exists a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ of size at most k satisfying some desired properties. If $\mathcal{F}'$ is required to contain all the elements of U, then it corresponds to the classical SET COVER problem and $\mathcal{F}'$ is called a *set cover*. The SET COVER problem is part of Karp's 21 NP-complete problems [11].

We begin the development with a conflict free problem already studied, CONFLICT FREE INTERVAL COVERING, introduced in [1,2,3]. Let P be a set of points on the x-axis, and let $\mathcal{I} = \{I_1, \ldots, I_m\}$ be a set of intervals on the x-axis. Furthermore, let $\mathcal{C} = \{C_1, C_2, \ldots, C_\ell\}$ denote a set of color classes, where each color class C_i consists of a pair of intervals from $\mathcal{I}$. Moreover, for any pair of integers i, j $(1 \le i < j \le \ell)$, $C_i \cap C_j = \emptyset$. We term $\mathcal{C}$ a *matching family*. For a set of intervals $Q \subseteq \mathcal{I}$, Q is *conflict free* if Q contains at most one interval from each color class, i.e. $\forall_{1 \le i \le \ell} |Q \cap C_i| \le 1$. Finally, for an interval $I = [a, b]$ and a point c on x-axis, we say I *covers* p if and only if $a \le c \le b$. Now we are ready to define the problem formally.

RAINBOW COVERING
Input: A set of points P on the x-axis, a set of intervals $\mathcal{I} = \{I_1, \ldots, I_m\}$ on the x-axis and a matching family $\mathcal{C} = \{C_1, C_2, \ldots, C_\ell\}$.
Question: Does there exist a conflict free subset Q of intervals which covers all the points in P?

Our first goal is to define *a model* in which we can express much more generalized version of conflicts beyond the matching family of conflict graphs.

To define our model we revisit SET COVER, as the model is best defined in the most general setting. Recall that the input to a SET COVER consists of a universe U of size n, a family $\mathcal{F}$ of subsets of U of size m. A natural way to model conflict is by using graphs. Formally stated, we have a graph $CG_{\mathcal{F}}$, on the vertex set $\mathcal{F}$ and there is an edge between two sets $F_i, F_j \in \mathcal{F}$ if F_i and F_j are in conflict. We call $CG_{\mathcal{F}}$ a *conflict graph*. Observe that in the RAINBOW COVERING problem, the family $\mathcal{C}$ would corresponds to $CG_{\mathcal{C}}$ with degree at most one. That is, edges of $CG_{\mathcal{C}}$ form a matching. And the question of finding a conflict free subset Q of intervals covering all the points in P becomes a problem of finding a set Q of intervals that covers all the points in P and $CG_{\mathcal{C}}[Q]$ is an independent set. The set cover $\mathcal{F}'$ such that $CG_{\mathcal{F}}[\mathcal{F}']$ is an independent set will be called *conflict free set cover*.

Our Contributions. In this paper we study the following problems in "geometric settings" in the realm of Parameterized Complexity. For more details about parameterized complexity we refer to monographs [4].

> Graphical Conflict Free Set Cover (Graphical CF-SC)
> **Input:** A universe U of size n, a family $\mathcal{F}$ of size m of subsets of U, a conflict graph $CG_{\mathcal{F}}$ and a positive integer k.
> **Parameter:** k
> **Question:** Does there exist a set cover $\mathcal{F}' \subseteq \mathcal{F}$ of size at most k such that $CG_{\mathcal{F}}[\mathcal{F}']$ is an independent set?

Let $(\mathcal{A}, \mathcal{B})$-Set Cover denote a restriction of Set Cover, where every instance $(U, \mathcal{F}, k)$ of Set Cover satisfies the property that $U \subseteq \mathcal{A}$ and $\mathcal{F} \subseteq \mathcal{B}$. For example in this setting, Covering Points by Intervals corresponds to $(\mathcal{A}, \mathcal{B})$-Set Cover where $\mathcal{A}$ is the set of points on x-axis and $\mathcal{B}$ is the set of intervals on x-axis. Given $(\mathcal{A}, \mathcal{B})$-Set Cover, the corresponding Graphical CF-SC corresponds to $(\mathcal{A}, \mathcal{B})$-Graphical CF-SC.

Observe that Graphical CF-SC becomes Set Cover if $CG_{\mathcal{F}}$ is an independent set. As the general Set Cover is hard in the parameterized framework, to design an FPT algorithm for Graphical CF-SC, it is important that the base Set Cover problem is FPT. This restricts us to $(\mathcal{A}, \mathcal{B})$-Set Cover which are either FPT or polynomial time solvable. If we are seeking FPT approximation algorithms then we can also restrict ourselves to $(\mathcal{A}, \mathcal{B})$-Set Cover which has either polynomial time approximation scheme (PTAS), constant factor approximation algorithm or FPT approximation algorithms, even if the problem is not in FPT. For example $(\mathcal{A}, \mathcal{B})$-Set Cover, where $\mathcal{A}$ is set of points in $\mathbb{R}^2$ and $\mathcal{B}$ is a set of unit discs in $\mathbb{R}^2$ is known to be W[1] hard [14] but admits a PTAS [10]. We will call $(\mathcal{A}, \mathcal{B})$-Set Cover *tractable* if it admits one of the following: a polynomial time algorithm, an FPT algorithm, an (E)PTAS, a constant factor approximation algorithm, an FPT approximation algorithm.

The next natural question is if we restrict ourselves to tractable $(\mathcal{A}, \mathcal{B})$-Set Cover, can an arbitrary conflict graph $CG_{\mathcal{F}}$ yield tractable algorithms for the conflict-free versions of $(\mathcal{A}, \mathcal{B})$-Set Cover? To formalize this question, let $\mathcal{G}$ denote a family of graphs. Then, the question is for which family of graphs $\mathcal{G}$, does $(\mathcal{A}, \mathcal{B})$-Graphical CF-SC admit an FPT algorithm or an FPT approximation algorithm when $CG_{\mathcal{F}}$ belongs to $\mathcal{G}$. For example, if $\mathcal{G}$ is the family of *cliques*, then even Graphical CF-SC trivially becomes polynomial time solvable when $CG_{\mathcal{F}}$ belongs to this family of cliques.

A problem that will be central to our study is the following. Let $\mathscr{P}$ and $\mathscr{I}$ denote a set of points and a set of intervals on the x-axis, respectively.

> $(\mathscr{P}, \mathscr{I})$-Graphical CF-SC **Parameter:** k
> **Input:** A set of points $P \subseteq \mathscr{P}$, a set of intervals $\mathcal{I} = \{I_1, \ldots, I_m\} \subseteq \mathscr{I}$, a conflict graph $CG_{\mathcal{I}}$ and a positive integer k.
> **Question:** Does there exist a conflict free set cover of size at most k?

In $(\mathscr{P}, \mathscr{I})$-Graphical CF-SC, when $CG_{\mathcal{I}}$ belongs to the family of matchings then the problem becomes Parameterized Rainbow Covering. This problem was studied in [1] and shown to be NP-complete. In fact, even if we do not care about the size of the conflict free set cover we seek, *just the decision version of a* conflict free set cover set is the same as RAINBOW COVERING, which is known to be NP-complete. Thus, seeking a conflict free set cover can transform a problem from being tractable to intractable.

In order to restrict the family of graphs to which a *conflict graph* belongs, we need to define the notion of *arboricity*. The arboricity of an undirected graph is the minimum number of forests into which its edges can be partitioned. A graph G is said to have *arboricity* d if the edges of G can be partitioned into at most d forests. Let $\mathcal{G}_d$ denote the family of graphs of arboricity d. This family includes the family of intersection graphs of low density objects in low dimensional Euclidean space as explained in [8,9]. Specifically, this includes planar graphs, graphs excluding a fixed graph as a minor, graphs of bounded expansion, and graphs of bounded degeneracy. Har-Peled and Quanrud [8,9] showed that low-density geometric objects form a subclass of the class of graphs that have polynomial expansion, which in turn, is contained in the class of graphs of bounded arboricity. Thus, our restriction of the family of conflict graphs to a family of graphs of bounded arboricity covers a large class of low-density geometric objects.

Theorem 1. *Let* $(\mathcal{A}, \mathcal{B})$-SET COVER *be tractable and let* $\mathcal{G}_d$ *be the family of graphs of arboricity* d*. Then, the corresponding* $(\mathcal{A}, \mathcal{B})$-GRAPHICAL CF-SC *is also tractable if* $CG_{\mathcal{F}}$ *belongs to* $\mathcal{G}_d$*. In particular we obtain following results when* $CG_{\mathcal{F}}$ *belongs to* $\mathcal{G}_d$*:*

- *If* $(\mathcal{A}, \mathcal{B})$-SET COVER *admits an* FPT *algorithm with running time* $\tau(k) \cdot n^{\mathcal{O}(1)}$*, then* $(\mathcal{A}, \mathcal{B})$-GRAPHICAL CF-SC *admits an* FPT *algorithm with running time* $2^{\mathcal{O}(dk)} \cdot \tau(k) \cdot n^{\mathcal{O}(1)}$*.*
- *If* $(\mathcal{A}, \mathcal{B})$-SET COVER *admits a factor* α*-approximation running in time* $n^{\mathcal{O}(1)}$ *then* $(\mathcal{A}, \mathcal{B})$-GRAPHICAL CF-SC *admits a factor* α-FPT*-approximation algorithm running in time* $2^{\mathcal{O}(dk)} \cdot n^{\mathcal{O}(1)}$*.*

The proof of Theorem 1 is essentially a black-box reduction to the non-conflict version of the problem. Thus, Theorem 1 covers a number of conflict-free version of many fundamental geometric coverage problems as illustrated in Table 1. In light of Theorem 1, it is natural to ask whether or not, these problems admit polynomial time *approximation* algorithms. Unfortunately, we cannot expect these problems to admit even a factor $o(n)$-approximation algorithm. This is because for most of these problems even deciding whether there exists a conflict free solution, *with no restriction on the size of the solution*, is NP-complete (for example RAINBOW COVERING is NP-complete [1]). Thus, having an $o(n)$-approximation algorithm would imply a polynomial time algorithm for the decision version of the problem, which we do not expect unless P=NP. Hence, the best we can expect for the $(\mathcal{A}, \mathcal{B})$-GRAPHICAL CF-SC problems is an FPT-approximation algorithm, as for many of them we can neither have an FPT algorithm, nor a polynomial time approximation algorithm.

We complement our algorithmic findings by a hardness reduction. Let $\mathscr{G}$ denote a family of graphs. Let $\mathscr{G}$-INDEPENDENT SET be the problem where the

$(\mathbb{R}^2, \mathcal{A})$-SC	**Complexity of** $(\mathbb{R}^2, \mathcal{A})$-SC	**Complexity of** $(\mathbb{R}^2, \mathcal{A})$-GRAPHICAL CF-SC
Disks/pseudo-disks	PTAS	α-FPT approx., $\forall \alpha > 1$
Fat triangles of same size	$\mathcal{O}(1)$	$\mathcal{O}(1)$-FPT approx.
Fat objects in $\mathbb{R}^2$	$\mathcal{O}(\log^* \mathsf{OPT})$	$\mathcal{O}(\log^* \mathsf{OPT})$-FPT approx.
$\mathcal{O}(1)$ density objects in $\mathbb{R}^2$	PTAS	α-FPT approx., $\forall \alpha > 1$
Objects with polylog density	QPTAS	$2^{\mathcal{O}(k)} n^{\mathcal{O}(\log^* n)}$ time approx., $\forall \alpha > 1$
Objects with density $\mathcal{O}(1)$ in $\mathbb{R}^d$	PTAS	α-FPT approx., $\forall \alpha > 1$
$(\mathcal{A}, \mathcal{B})$-SET COVER where every instance $(U, \mathcal{F})$ has VC dimension d	$\mathcal{O}(d \log(d\mathsf{OPT}))$	$\mathcal{O}(d \log(d\mathsf{OPT}))$-FPT approx.
POINT GUARD ART GALLERY	$\mathcal{O}(\log \mathsf{OPT})$	$\mathcal{O}(\log \mathsf{OPT})$-FPT approx.
TERRAIN GUARDING	PTAS	α-FPT approx., $\forall \alpha > 1$
$(\mathscr{P}, \mathscr{I})$-SET COVER	Polynomial Time	$2^{\mathcal{O}(dk)} \cdot n^{\mathcal{O}(1)}$-FPT algorithm

Table 1. Corollaries of Theorem 1. Here $(\mathbb{R}^2, \mathcal{A})$-SET COVER ($(\mathbb{R}^2, \mathcal{A})$-SC) is a geometric set cover problem where $\mathbb{R}^2$ is a set of points in the plane and the covering objects are specified in the first column. The conflict graph for all the problems is $\mathscr{G}_d$, family of graphs of arboricity d, for some constant d. For the definitions of density and fatness we refer to [8]. The entries in the second column give the approximation ratio of the $(\mathbb{R}^2, \mathcal{A})$-SC problem based on Theorem 1.

input is a graph $G \in \mathscr{G}$ and a positive integer k, and the objective is to decide whether there is a set S of size at least k such that $G[S]$ is an independent set.

Theorem 2. *Let* $\mathscr{G}$ *denote a family of graphs such that* $\mathscr{G}$-INDEPENDENT SET *is* W[1]*-hard. If* $CG_{\mathcal{I}}$ *belongs to* $\mathscr{G}$*, then* $(\mathscr{P}, \mathscr{I})$-GRAPHICAL CF-SC *does not admit an* FPT *algorithm, unless* FPT $=$W[1].

The proof of Theorem 2 is a Turing reduction based on (n, k)*-perfect hash families* [16] that takes time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$. In fact, for any fixed $\mathcal{A}$ and $\mathcal{B}$, one should be able to follow this proof and show W[1]-hardness for $(\mathcal{A}, \mathcal{B})$-GRAPHICAL CF-SC, where $CG_{\mathcal{F}}$ belongs to a graph family $\mathscr{G}$ for which $\mathscr{G}$-INDEPENDENT SET is W[1]-hard. Due to paucity of space the proof of Theorem 2 is deferred to the full version of the paper.

Theorem 1 captures those families of conflict graphs that are "everywhere sparse". However, the $(\mathcal{A}, \mathcal{B})$-GRAPHICAL CF-SC problem is also tractable if the conflict graphs belong to the family of cliques. When the conflict graph belongs to a "dense family" of graphs, we design a general theorem using matroid machinery.

Let $(U, \mathcal{F}, k)$ be an instance of SET COVER. In the matroidal model of representing conflicts, we are given a matroid $M = (E, \mathcal{J})$, where the ground set $E = \mathcal{F}$, and $\mathcal{J}$ is a family of subsets of $\mathcal{F}$ satisfying all the three properties of a matroid. In this paper we assume that $M = (E, \mathcal{J})$ is a *linear or representable matroid*, and the corresponding linear representation is given as part of the input. In the RAINBOW COVERING problem, let $\mathcal{Q}$ denote the family of conflict free subsets of intervals in $\mathcal{I}$. One can define a *partition matroid* on $\mathcal{F}$ such that $\mathcal{J} = \mathcal{Q}$. Thus, the question of finding a conflict free subset of intervals covering all the points in P becomes a problem of finding an independent set in $\mathcal{J}$ that covers all the points in P. The Matroidal Conflict Free Set Cover problem (Matroidal CF-SC, in short) is defined similarly to Graphical CF-SC. In particular, the input consists of a linear matroid $M = (\mathcal{F}, \mathcal{J})$ over the ground set $\mathcal{F}$ such that the set cover $\mathcal{F}' \in \mathcal{J}$.

Theorem 3. $(\mathscr{P}, \mathscr{I})$-Matroidal CF-SC *is* FPT *for all representable matroids $M = (\mathcal{I}, \mathcal{J})$ defined over $\mathcal{I}$. In fact, given a linear representation, the algorithm runs in time $2^{\omega k} \cdot (n+m)^{\mathcal{O}(1)}$. Here, ω is the exponent in the running time of matrix multiplication.*

A graph is called a *cluster graph*, if all its connected components are cliques. Since cluster graphs can be captured by partition matroids, Theorem 3 implies that $(\mathscr{P}, \mathscr{I})$-Matroidal CF-SC is FPT if $CG_{\mathcal{F}}$ is a cluster graph.

Notations. For $t \in \mathbb{N}$, we use $[t]$ as a shorthand for $\{1, 2, \ldots, t\}$. A family of sets $\mathcal{A}$ is called a p-family, if the cardinality of all the sets in $\mathcal{A}$ is p. Given two families of sets $\mathcal{A}$ and $\mathcal{B}$, we define $\mathcal{A} \bullet \mathcal{B} = \{X \cup Y \mid X \in \mathcal{A} \text{ and } Y \in \mathcal{B} \text{ and } X \cap Y = \emptyset\}$. Given a graph G, $V(G)$ and $E(G)$ denote its vertex-set and edge-set, respectively. We borrow notations from the book of Diestel [5] for graph-related notations.

2 FPT Algorithms

In this section we prove Theorems 1 and Theorem 3. The Proof of Theorem 1 is based on a randomization scheme while the proof of Theorem 3 uses the idea of efficient computation of representative families [6].

2.1 FPT **Algorithms for Graphical CF-SC**

Our algorithm for Theorem 1 is essentially a randomized reduction from $(\mathcal{A}, \mathcal{B})$-Graphical CF-SC to $(\mathcal{A}, \mathcal{B})$-Set Cover, when the conflict graph has bounded arboricity. Towards this, we start with a forest decomposition of graphs of bounded arboricity and then apply a randomized process to obtain an instance of $(\mathcal{A}, \mathcal{B})$-Set Cover. However, to design a deterministic algorithm we use the construction of universal sets. For this, we will exploit the following definition and theorem.

Definition 1 ([16]). *An (n,t)-universal set $\mathscr{F}$ is a set of functions from $\{1, \ldots, n\}$ to $\{0, 1\}$, such that for every subset $S \subseteq \{1, \ldots, n\}$, $|S| = t$, the set $\mathscr{F}|_S = \{f|_S \mid f \in \mathscr{F}\}$ is equal to the set 2^S of all the functions from S to $\{0, 1\}$.*

Theorem 4 ([16]). *There is a deterministic algorithm with $\mathcal{O}(2^t t^{\mathcal{O}(\log t)} n \log n)$ run time that constructs an (n,t)-universal set $\mathscr{F}$ such that $|\mathscr{F}| = 2^t t^{\mathcal{O}(\log t)} \log n$.*

Now we are ready to give the proof of Theorem 1[4]

Proof (Proof of Theorem 1). Let $(U, \mathcal{F}, CG_{\mathcal{F}}, k)$ be an instance of $(\mathcal{A}, \mathcal{B})$-Graphical CF-SC, where $CG_{\mathcal{F}}$ belongs to $\mathcal{G}_d$. Our algorithm has the following phases.

[4] The idea used in the proof of Theorem 1 is inspired by a proof used in [13].

Decomposing $CG_{\mathcal{F}}$ into Forests. We apply the known polynomial time algorithm [7] to decompose the graph $CG_{\mathcal{F}}$ into $T_1, \ldots, T_d$ where T_i is a forest in $CG_{\mathcal{F}}$ and $\bigcup_{i=1}^{d} E(T_i) = E(CG_{\mathcal{F}})$. Let v_{root} be a special vertex such that v_{root} does not belong to $V(CG_{\mathcal{F}}) = \mathcal{F}$. Now for every T_i, and for every connected component of T_i, we pick an arbitrary vertex and connect it to v_{root}. Now if we look at the tree induced on $V(T_i) \cup \{v_{\mathsf{root}}\}$ then it is connected and we will denote this tree by T_i'. Furthermore, we will treat each T_i' as a tree rooted at v_{root}. This automatically defines *parent-child* relationship among the vertices of T_i'. This completes the partitioning of the edge set of $CG_{\mathcal{F}}$ into forests.

Step 1: Randomized event and probability of success. Independently color the vertices of $CG_{\mathcal{F}}$ into blue and green uniformly at random. That is, we color the vertices of $CG_{\mathcal{F}}$ blue and green with probability $\frac{1}{2}$. Furthermore, we color $\{v_{\mathsf{root}}\}$ to blue. Let $\mathcal{F}'$ be a conflict free set cover of size at most k. We consider the following event to be *good*.

> Every vertex in $\mathcal{F}'$ is colored green and every *parent* of every vertex in $\mathcal{F}'$ in every tree T_i' is colored blue.

Let S_{parent} denote the set of parents of every vertex in $\mathcal{F}'$ in every tree T_i'. Since, we have at most d trees and the size of $\mathcal{F}'$ is upper bounded by k we have that $|S_{\mathsf{parent}}| \leq kd$. We say that $\mathcal{F}'$ (S_{parent}) is green (blue) to mean that every vertex in $\mathcal{F}'$ (S_{parent}) is colored green (blue). Thus,

$$\begin{aligned}\Pr[\text{good event happens}] &= \Pr[\mathcal{F}' \text{ is green } \wedge S_{\mathsf{parent}} \text{ is blue}] \\ &= \Pr[\mathcal{F}' \text{ is green}] \times \Pr[S_{\mathsf{parent}} \text{ is blue}] \geq \frac{1}{2^{k(d+1)}}.\end{aligned}$$

The second equality follows from the following fact. The set $\mathcal{F}'$ is an independent set in $CG_{\mathcal{F}}$ and $S_{\mathsf{parent}} \subseteq N_{CG_{\mathcal{F}}}(\mathcal{F}') \cup \{v_{\mathsf{root}}\}$. Thus, these sets are pairwise disjoint and hence the events $\mathcal{F}'$ is colored green and S_{parent} is colored blue are independent.

Step 2: A cleaning process. Let $p = \frac{1}{2^{kd}}$. Now we apply a cleaning procedure so that we get a set Z such that $CG_{\mathcal{F}}[Z]$ is an independent set in $CG_{\mathcal{F}}$ and it contains $\mathcal{F}'$. Let $\mathscr{B}$ denote the set of vertices that have been colored blue. We start by deleting every vertex in $\mathscr{B}$. Now for every edge (f_1, f_2) in $CG_{\mathcal{F}}[V(CG_{\mathcal{F}}) \setminus \mathscr{B}]$, we do as follows. We know that (f_1, f_2) belongs to some tree T_i' and thus either f_1 is a *child* of f_2 or vice-versa. If f_1 is a child then we delete f_1, otherwise we delete f_2. Let the resulting set of vertices be Z. By construction Z is an independent set in $CG_{\mathcal{F}}$. Next we show that $\mathcal{F}' \subseteq Z$ with probability $p/2^k$. Clearly, with probability $\frac{1}{2^k}$ we know that no vertex of $\mathcal{F}'$ is colored blue and thus with probability $\frac{1}{2^k}$ we know that $\mathcal{F}' \subseteq V(CG_{\mathcal{F}}) \setminus \mathscr{B}$. Observe that with probability p, we have that all the parents of $\mathcal{F}'$ in any tree T_i' have been colored blue. Thus, a vertex $x \in V(CG_{\mathcal{F}}) \setminus \mathscr{B}$, colored green, can not belong to $\mathcal{F}'$, if it is a child of some vertex in some tree T_i' after deleting the vertices of $\mathscr{B}$. This is the reason when we delete a vertex from an edge (f_1, f_2), we delete the one which is a child in some tree T_i'. Thus, by deleting a vertex that is a child in an edge (f_1, f_2), we do not delete any vertex from $\mathcal{F}'$. This implies that with probability $\frac{1}{2^{k(d+1)}}$, we have that $\mathcal{F}' \subseteq Z$. This completes the proof.

Solving the problem. Let $\mathscr{Q}$ be a parameterized algorithm for $(\mathcal{A},\mathcal{B})$-Set Cover running in time $\tau(k)\cdot n^{\mathcal{O}(1)}$. Recall that $(U,\mathcal{F},CG_{\mathcal{F}},k)$ is an instance of $(\mathcal{A},\mathcal{B})$-Graphical CF-SC. Now to test whether there exists a conflict free set cover $\mathcal{F}'$ of size at most k, we run $\mathscr{Q}$ on (U,Z,k). If the algorithm return Yes, we return the same for $(\mathcal{A},\mathcal{B})$-Graphical CF-SC. Else, we repeat the process by randomly finding another Z^* by following Steps 1 and 2 and then running the algorithm $\mathscr{Q}$ on the instance (U,Z^*,k) and returning the answer accordingly. We repeat the process $2^{k(d+1)}$ time. If we fail to detect whether $(U,\mathcal{F},k,CG_{\mathcal{F}})$ is a Yes instance of $(\mathcal{A},\mathcal{B})$-Graphical CF-SC in $2^{k(d+1)}$ rounds, then we return that the given instance is a No instance. Thus, if $(U,\mathcal{F},k,CG_{\mathcal{F}})$ is No instance of $(\mathcal{A},\mathcal{B})$-Graphical CF-SC, then we always return No. However, if $(U,\mathcal{F},k,CG_{\mathcal{F}})$ is a Yes instance of $(\mathcal{A},\mathcal{B})$-Graphical CF-SC then there exists a set $\mathcal{F}'$, that is a conflict free set cover of size at most k. The probability that we will not find a set Z containing $\mathcal{F}'$ in $q=2^{k(d+1)}$ rounds is upper bounded by $\left(1-\frac{1}{q}\right)^q\leq\frac{1}{e}$. Thus, the probability that we will find a set Z containing $\mathcal{F}'$ in q rounds is at least $1-\frac{1}{e}\geq\frac{1}{2}$. Thus, if the given instance is a Yes instance then the algorithm succeeds with probability at least $\frac{1}{2}$. The running time of the algorithm is upper bounded by $\tau(k)\cdot 2^{k(d+1)}\cdot n^{\mathcal{O}(1)}$.

Derandomizing the algorithm. Now to design our deterministic algorithm all we will need to do is to replace the randomized coloring function with a deterministic coloring function that colors the vertices in $\mathcal{F}'$ green and all the vertices in S_{parent} to blue. To design such a coloring function we set $t=k(d+1)$, and use Theorem 4 to construct an (n,t)-universal set $\mathscr{F}$ such that $|\mathscr{F}|=2^t t^{\mathcal{O}(\log(t))}\log n$. The algorithm to construct $\mathscr{F}$ takes $\mathcal{O}(2^t t^{\mathcal{O}(\log(t))} n\log n)$. Finally, to derandomize our algorithm, rather than randomly coloring vertices with $\{\mathsf{blue},\mathsf{green}\}$, we go through each function f in the family $\mathscr{F}$ and view the vertices that have assigned 0 as blue and others as green. By the properties of (n,t)-universal set we know that there exists a function f that correctly colors the vertices in $\mathcal{F}'$ with 1 and every vertex in S_{parent} with 0. Thus, the set Z_f we will obtain by applying Step 2 will contain the set $\mathcal{F}'$. After this the correctness of the algorithm follows from the correctness of the algorithm $\mathscr{Q}$. Thus, the running time of the algorithm is upper bounded by $\tau(k)\cdot|\mathscr{F}|\cdot n^{\mathcal{O}(1)}=\tau(k)\cdot 2^{k(d+1)+o(kd)}\cdot n^{\mathcal{O}(1)}$. This completes the proof of the first part.

Let $\mathscr{S}$ be a factor α-approximation algorithm for $(\mathcal{A},\mathcal{B})$-Set Cover running in time $n^{\mathcal{O}(1)}$. To obtain the desired FPT approximation algorithm with factor α, we do as follows. We only give the deterministic version of the algorithm based on the uses of universal sets. As before, let $(U,\mathcal{F},CG_{\mathcal{F}},k)$ be an instance of $(\mathcal{A},\mathcal{B})$-Graphical CF-SC, where $CG_{\mathcal{F}}$, belongs to $\mathcal{G}_d$. We again set $t=k(d+1)$, and use Theorem 4 to construct an (n,t)-universal set $\mathscr{F}$ such that $|\mathscr{F}|=2^t t^{\mathcal{O}(\log(t))}\log n$. The algorithm to construct $\mathscr{F}$ takes $\mathcal{O}(2^t t^{\mathcal{O}(\log(t))} n\log n)$. We go through each function f in the family $\mathscr{F}$ and view the vertices that have been assigned 0 as blue and others as green. If there exists a conflict free set cover $\mathcal{F}'$ of size at most k, then by the properties of (n,t)-universal set we know that there exists a function f that correctly color the vertices in $\mathcal{F}'$ with 1 and every vertex in S_{parent} with 0. Thus, the set Z_f we will obtain by applying Step 2, will contain the set $\mathcal{F}'$. Thus, to design the approximation algorithm, for every $f \in \mathscr{F}$, we first construct Z_f. And for each such Z_f we run $\mathscr{S}$ on (U, Z_f, k). This could either return that there is No solution, or returns a solution $\mathcal{F}'$ which is a factor α-approximation to the instance (U, Z_f, k). If for some $f \in \mathscr{F}$, $\mathscr{S}$ returns $\mathcal{F}'$ of size at most αk when run on (U, Z_f, k) then the algorithm returns $\mathcal{F}'$. In all other cases the algorithm returns that the given instance is a No instance. The correctness of the algorithm follows from the properties of universal sets and the correctness of the algorithm $\mathscr{S}$. The running time of the algorithm is upper bounded by: $|\mathscr{F}| \times$ Running time of $\mathscr{S} = 2^{k(d+1)+o(kd)} \cdot n^{\mathcal{O}(1)}$. This completes the proof. □

2.2 FPT Algorithm for $(\mathscr{P}, \mathscr{I})$-Matroidal CF-SC

In this section we will design an FPT algorithm proving Theorem 3. Towards that we need to define some basic notions related to representative families and results regarding their fast and efficient computation. For definitions related to matroids and a broad overview of representative families we refer to [4, Chapter 12].

Definition 2 (q-Representative Family [15,4]). *Given a matroid $M = (E, \mathcal{J})$ and a family $\mathcal{S}$ of subsets of E, we say that a subfamily $\widehat{\mathcal{S}} \subseteq \mathcal{S}$ is q-*representative *for $\mathcal{S}$ if the following holds: for every set $Y \subseteq E$ of size at most q, if there is a set $X \in \mathcal{S}$ disjoint from Y with $X \cup Y \in \mathcal{J}$, then there is a set $\widehat{X} \in \widehat{\mathcal{S}}$ disjoint from Y with $\widehat{X} \cup Y \in \mathcal{J}$. If $\hat{\mathcal{S}} \subseteq \mathcal{S}$ is q-representative for $\mathcal{S}$ we write $\widehat{\mathcal{S}} \subseteq^q_{rep} \mathcal{S}$.*

Lemma 1 ([6]). *Let $M = (E, \mathcal{J})$ be a matroid and $\mathcal{S}$ be a family of subsets of E. If $\mathcal{S}' \subseteq^q_{rep} \mathcal{S}$ and $\widehat{\mathcal{S}} \subseteq^q_{rep} \mathcal{S}'$, then $\widehat{\mathcal{S}} \subseteq^q_{rep} \mathcal{S}$.*

Lemma 2 ([12]). *Let $M = (E, \mathcal{J})$ be a linear matroid of rank n and let $\mathcal{S} = \{S_1, \ldots, S_t\}$ be a p-family of independent sets. Let A be a $n \times |E|$ matrix representing M over a field $\mathbb{F}$, where $\mathbb{F} = \mathbb{F}_{p^\ell}$ or $\mathbb{F}$ is $\mathbb{Q}$. Then there is a deterministic algorithm computing $\widehat{\mathcal{S}} \subseteq^q_{rep} \mathcal{S}$ of size $np\binom{p+q}{p}$ in $\mathcal{O}\left(\binom{p+q}{p} tp^3 n^2 + t\binom{p+q}{q}^{\omega-1} (pn)^{\omega-1}\right) + (n + |E|)^{\mathcal{O}(1)}$ operations over $\mathbb{F}$.*

Now we are ready to prove Theorem 3. Let $(P, \mathcal{I}, k, M = (\mathcal{I}, \mathcal{J}))$ be an instance of $(\mathscr{P}, \mathscr{I})$-Matroidal CF-SC, where P is a set of points on the x-axis, $\mathcal{I} = \{I_1, \ldots, I_m\}$ is a set of intervals on the x-axis and $M = (\mathcal{I}, \mathcal{J})$ is a matroid over the ground set $\mathcal{I}$. The objective is to find a set cover $\mathcal{S} \subseteq \mathcal{I}$ of size at most k such that $\mathcal{S} \in \mathcal{J}$.

To design our algorithm for $(\mathscr{P}, \mathscr{I})$-Matroidal CF-SC, we will use efficient computation of representative families applied on a dynamic programming algorithm. Let $P = \{p_1, \ldots, p_n\}$ denote the set of points sorted from left to right. Next we introduce the notion of family of partial solutions. Let

$$\mathcal{P}^i = \left\{ X \;\middle|\; X \subseteq \mathcal{I}, X \in \mathcal{J},\ |X| \leq k,\ X \text{ covers } p_1, \ldots, p_i \right\}$$

denote the family of subsets of intervals of size at most k that covers first i points and are independent in the matroid $M = (\mathcal{I}, \mathcal{J})$. Furthermore, for every $j \in [k]$, by $\mathcal{P}^{ij}$, we denote the subset of $\mathcal{P}^i$ containing sets of size *exactly* j. Thus,

$$\mathcal{P}^i = \biguplus_{j=1}^{k} \mathcal{P}^{ij}.$$

In this subsection whenever we talk about independent sets, these are independent sets of the matroid $M = (\mathcal{I}, \mathcal{J})$. Furthermore, we *assume that we are given, A_M, the linear representation of M*. Without loss of generality we can assume that A_M is a $n' \times |\mathcal{I}|$ matrix, where $n' \leq |\mathcal{I}|$.

Observe that $(P, \mathcal{I}, k, M = (\mathcal{I}, \mathcal{J}))$ is a Yes instance of $(\mathscr{P}, \mathscr{I})$-MATROIDAL CF-SC if and only if $\mathcal{P}^n$ is non-empty. This implies that $\mathcal{P}^n$ is non-empty if and only if $\widehat{\mathcal{P}}^n \subseteq^0_{rep} \mathcal{P}^n$ is non-empty. We capture this into the following lemma.

Lemma 3. *Let $(P, \mathcal{I}, k, M = (\mathcal{I}, \mathcal{J}))$ be an instance of $(\mathscr{P}, \mathscr{I})$-*MATROIDAL CF-SC*. Then, $(P, \mathcal{I}, k, M = (\mathcal{I}, \mathcal{J}))$ is a* Yes *instance of $(\mathscr{P}, \mathscr{I})$-*MATROIDAL CF-SC *if and only if $\mathcal{P}^n$ is non-empty if and only if $\widehat{\mathcal{P}}^n \subseteq^0_{rep} \mathcal{P}^n$ is non-empty.*

For an ease of presentation by $\mathcal{P}^0$, we denote the set $\{\emptyset\}$. The next lemma provides an efficient computation of the family $\widehat{\mathcal{P}}^i \subseteq^{1\cdots k}_{rep} \mathcal{P}^i$. In particular, for every $1 \leq i \leq n$, we compute

$$\widehat{\mathcal{P}}^i = \bigcup_{j=1}^{k} \left(\widehat{\mathcal{P}}^{ij} \subseteq^{k-j}_{rep} \mathcal{P}^{ij} \right).$$

Lemma 4. *Let $(P, \mathcal{I}, k, M = (\mathcal{I}, \mathcal{J}))$ be an instance of $(\mathscr{P}, \mathscr{I})$-*MATROIDAL CF-SC*. Then for every $1 \leq i \leq n$, a collection of families $\widehat{\mathcal{P}}^i \subseteq^{1\cdots k}_{rep} \mathcal{P}^i$, of size at most $2^k \cdot |\mathcal{I}| \cdot k$ can be found in time $2^{\omega k} \cdot (n + |\mathcal{I}|)^{\mathcal{O}(1)}$.*

Proof. We describe a dynamic programming based algorithm. Let $P = \{p_1, \ldots, p_n\}$ denote the set of points sorted from left to right and $\mathcal{D}$ be a $n+1$-sized array indexed with $\{0, \ldots, n\}$. The entry $\mathcal{D}[i]$ will store a family $\widehat{\mathcal{P}}^i \subseteq^{1\cdots k}_{rep} \mathcal{P}^i$. We fill the entries in the matrix $\mathcal{D}$ in the increasing order of index. For $i = 0$, $\mathcal{D}[i] = \{\emptyset\}$. Let $i \in \{0, 1, \ldots, n\}$ and assume that we have filled all the entries until the row i (i.e, $\mathcal{D}[i]$ will contain a family $\widehat{\mathcal{P}}^i \subseteq^{1\cdots k}_{rep} \mathcal{P}^i$). For any interval $I \in \mathcal{I}$, let ℓ_I be the lowest index in $[n]$ such that p_{ℓ_I} is covered by I. Let $\mathcal{Z}_{i+1}$ denote the set of intervals $I \in \mathcal{I}$ that covers the point p_{i+1}. Now we compute

$$\mathcal{N}^{i+1} = \bigcup_{I \in \mathcal{Z}_{i+1}} (\mathcal{D}[\ell_I - 1] \bullet \{I\}) \cap \mathcal{J} \tag{1}$$

Notice that in the Equation 1, the union is taken over $I \in \mathcal{Z}_{i+1}$. Since for any $I \in \mathcal{Z}_{i+1}$, I covers p_{i+1}, the value $\ell_I - 1$ is strictly less than $i+1$ and hence Equation 1 is well defined. Let $\mathcal{N}^{(i+1)j}$ denote the subset of $\mathcal{N}^{i+1}$ containing subsets of size exactly j.

Claim. $\mathcal{N}^{i+1} \subseteq^{1\cdots k}_{rep} \mathcal{P}^{i+1}$.

Proof. Let $S \in \mathcal{P}^{(i+1)j}$ and Y be a set of size $k-j$ (which is essentially an independent set of M) such that $S \cap Y = \emptyset$ and $S \cup Y \in \mathcal{J}$. We will show that there exists a set $\widehat{S} \in \mathcal{N}^{(i+1)}$ such that $\widehat{S} \cap Y = \emptyset$ and $\widehat{S} \cup Y \in \mathcal{J}$. This will imply the desired result.

Since S covers $\{p_1, \ldots, p_{i+1}\}$, there is an interval J in S which covers p_{i+1}. Since S covers $\{p_1, \ldots, p_{i+1}\}$ and J covers p_{i+1}, the set of intervals $S' = S \setminus \{J\}$ covers $\{p_1, \ldots, p_{i+1}\} \setminus \{p_{\ell_J}, \ldots, p_{i+1}\}$ and J covers $\{p_{\ell_J}, \ldots p_{i+1}\}$. Let $Y' = Y \cup \{J\}$. Notice that $S' \cup Y' = S \cup Y \in \mathcal{J}$, $|S'| = j-1$, $|Y'| = k-j+1$ and S' covers $\{p_1, \ldots, p_{i+1}\} \setminus \{p_{\ell_J}, \ldots, p_{i+1}\}$. This implies that $S' \in \mathcal{P}^{(\ell_J-1)(j-1)}$ and by our assumption that $\mathcal{D}[\ell_J - 1]$ contain $\widehat{\mathcal{P}}^{(\ell_J-1)(j-1)} \subseteq_{rep}^{k-j+1} \mathcal{P}^{(\ell_J-1)(j-1)}$, we have that there exists $S^* \in \mathcal{D}[\ell_J - 1]$ such that $S^* \cap Y' = \emptyset$ and $S^* \cup Y' \in \mathcal{J}$. By Equation 1, $S^* \cup \{J\}$ in $\mathcal{N}^{i+1}$, because $S^* \cup \{J\} \in \mathcal{J}$. Now we set $\widehat{S} = S^* \cup \{J\}$. Observe that $\widehat{S} \cap Y = \emptyset$ and $\widehat{S} \cup Y \in \mathcal{J}$. This completes the proof of the claim. $\square$

We fill the entry for $\mathcal{D}[i+1]$ as follows.

$$\mathcal{D}[i+1] = \bigcup_{j=1}^{k} \left(\widehat{\mathcal{N}}^{(i+1)j} \subseteq_{rep}^{k-j} \mathcal{N}^{(i+1)j} \right) \tag{2}$$

In Equation 2, for every $1 \le j \le k$, $\mathcal{N}^{(i+1)j}$ denote the subset of $\mathcal{N}^{(i+1)}$ containing sets of size *exactly* j and $\widehat{\mathcal{N}}^{(i+1)j}$ can be computed using Lemma 2. Lemma 1 and Claim 2.2 implies that $\mathcal{D}[i+1] \subseteq_{rep}^{1\cdots k} \mathcal{P}^{i+1}$.

Now we analyse the running time of the algorithm. Consider the time to compute $\mathcal{D}[i+1]$. We already have computed the family corresponding to $\mathcal{D}[r]$ for all $r \in [i]$. By Lemma 2, for any $r \in [i]$ and $j \in [k]$, the subset of $\mathcal{D}[r]$ containing sets of size exactly j is upper bounded by $|\mathcal{I}| \cdot k \cdot \binom{k}{j}$. Hence, the cardinality of $\mathcal{N}^{(i+1)j}$ is upper bounded by $|\mathcal{I}|^2 \cdot n \cdot k \cdot \binom{k}{j}$. Thus, by Lemma 2, the time to compute $\widehat{\mathcal{N}}^{(i+1)j} \subseteq_{rep}^{k-j} \mathcal{N}^{(i+1)j}$ is bounded by $\left(\binom{k}{j}^2 + \binom{k}{j}^{\omega}\right)(n+|\mathcal{I}|)^{\mathcal{O}(1)} = \binom{k}{j}^{\omega} \cdot (n+|\mathcal{I}|)^{\mathcal{O}(1)}$ number of operation over the field in which A_M is given and $|\widehat{\mathcal{N}}^{(i+1)j}| \le |\mathcal{I}| \cdot k \cdot \binom{k}{j}$. Hence the total running time to compute $\mathcal{D}[i+1]$ for any $i+1 \in [n]$ is

$$\sum_{j=1}^{k} \binom{k}{j}^{\omega} \cdot (n+|\mathcal{I}|)^{\mathcal{O}(1)}) = 2^{\omega k} \cdot (n+|\mathcal{I}|)^{\mathcal{O}(1)}.$$

By Lemma 2, the cardinality of $\mathcal{D}[i+1]$ is bounded by,

$$|\mathcal{D}[i+1]| = \sum_{j=1}^{k} |\widehat{\mathcal{N}}^{(i+1)j}| \le \sum_{j=1}^{k} |\mathcal{I}| \cdot k \cdot \binom{k}{j} = 2^k |\mathcal{I}| \cdot k.$$

This completes the proof. $\square$

Theorem 3 follows from Lemmata 3 and 4. Now we explain an application of Theorem 3. Consider the problem $(\mathscr{P}, \mathscr{I})$-Graphical CF-SC, where $CG_{\mathcal{I}}$ is a cluster graph. Let $(P, \mathcal{I}, CG_{\mathcal{I}}, k)$ be an instance of $(\mathscr{P}, \mathscr{I})$-Graphical CF-SC.

Let $C_1, \ldots C_t$ be the connected components of $CG_{\mathcal{I}}$, where each C_i is a clique for all $i \in [t]$. In any solution we are allowed to pick at most one vertex (an interval) from C_i for any $i \in [t]$. This information can be encoded using a partition matroid $M = (\mathcal{I} = V(C_1) \uplus \ldots \uplus V(C_t), \mathcal{J})$ where any subset $\mathcal{I}' \subseteq \mathcal{I}$ is independent in M if and only if $|\mathcal{I}' \cap V(C_i)| \leq 1$ for any $i \in [t]$. Moreover, a linear representation of a partition matroid can be found in polynomial time ([15, Proposition 3.5]). As a result, by applying Theorem 3 and Proposition 3.5 of [15], we get the following corollary.

Corollary 1. $(\mathscr{P}, \mathscr{I})$-Graphical CF-SC*, when* $CG_{\mathcal{I}}$ *is a cluster graph, can be solved in time* $2^{\omega k} \cdot (n + |\mathcal{I}|)^{\mathcal{O}(1)}$.

References

1. Arkin, E.M., Banik, A., Carmi, P., Citovsky, G., Katz, M.J., Mitchell, J.S.B., Simakov, M.: Choice is hard. In: ISAAC. pp. 318–328 (2015)
2. Arkin, E.M., Banik, A., Carmi, P., Citovsky, G., Katz, M.J., Mitchell, J.S.B., Simakov, M.: Conflict-free covering. In: CCCG (2015)
3. Banik, A., Panolan, F., Raman, V., Sahlot, V.: Fréchet distance between a line and avatar point set. FSTTCS pp. 32:1–32:14 (2016)
4. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized algorithms. Springer (2015)
5. Diestel, R.: Graph Theory, 4th Edition, Graduate texts in mathematics, vol. 173. Springer (2012)
6. Fomin, F.V., Lokshtanov, D., Panolan, F., Saurabh, S.: Efficient computation of representative families with applications in parameterized and exact algorithms. J. ACM 63(4), 29 (2016)
7. Gabow, H.N., Westermann, H.H.: Forests, frames, and games: Algorithms for matroid sums and applications. Algorithmica 7(5&6), 465–497 (1992)
8. Har-Peled, S., Quanrud, K.: Approximation algorithms for low-density graphs. CoRR abs/1501.00721 (2015)
9. Har-Peled, S., Quanrud, K.: Approximation algorithms for polynomial-expansion and low-density graphs. In: ESA. vol. 9294, pp. 717–728. Springer (2015)
10. Hunt III, H.B., Marathe, M.V., Radhakrishnan, V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E.: NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. J. Algorithms 26(2), 238–274 (1998)
11. Karp, R.M.: Reducibility among combinatorial problems. In: Proceedings of a symposium on the Complexity of Computer Computations. pp. 85–103 (1972)
12. Lokshtanov, D., Misra, P., Panolan, F., Saurabh, S.: Deterministic truncation of linear matroids. In: ICALP, Proceedings, Part I. vol. 9134, pp. 922–934. Springer (2015)
13. Lokshtanov, D., Panolan, F., Saurabh, S., Sharma, R., Zehavi, M.: Covering small independent sets and separators with applications to parameterized algorithms. ArXiv e-prints (May 2017)
14. Marx, D.: Parameterized complexity of independence and domination on geometric graphs. In: IPEC, pp. 154–165. Springer (2006)
15. Marx, D.: A parameterized view on matroid optimization problems. Theor. Comput. Sci. 410(44), 4471–4479 (2009)
16. Naor, M., Schulman, J.L., Srinivasan, A.: Splitters and near-optimal derandomization. In: FOCS. pp. 182–191 (1995)