Algorithmica (2020) 82:1–19
https://doi.org/10.1007/s00453-019-00600-w

ORIGINAL RESEARCH

Parameterized Complexity of Geometric Covering Problems Having Conflicts

Aritra Banik[1] · Fahad Panolan[2] · Venkatesh Raman[3] · Vibha Sahlot[4] · Saket Saurabh[3]

Received: 15 April 2018 / Accepted: 13 June 2019 / Published online: 11 July 2019

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 $\mathbb{R}^d$. 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. Our main result is that as long as the conflict graph has bounded arboricity there is a parameterized reduction to the conflict-free version. As a consequence, we have the following results when the conflict graph has bounded arboricity. (1) If the Geometric Coverage problem is fixed parameter tractable (FPT), then so is the conflict free version. (2) 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 that run in FPT time. Our other results include an FPT algorithm and a hardness result for conflict-free version of Covering Points by Intervals. The FPT algorithm is for the case when the conflicts are given by a representable matroid. We prove that conflict-free version of Covering Points by Intervals does not admit an FPT algorithm, unless FPT =W[1], for the family of conflict graphs for which the Independent Set problem is W[1]-hard.

Keywords Conflict problems · Geometric Coverage · Parameterized Complexity · Matroids

A preliminary version of this paper appeared in the proceedings of WADS 2017.

Extended author information available on the last page of the article

1 Introduction

In any VLSI design, floorplanning is a process which includes geometric layout design of the final integrated circuit. The basic problem is, given a set of geometric modules/blocks/shapes of arbitrary sizes, we have to place them on a plane within a rectangle of minimum area without any overlaps. The problem of packing rectangles in a minimum area is known to be NP-hard, and there exists a PTAS to solve this problem [9]. An issue of practical concern in VLSI chip design is chip overheating. Efforts have been made to address the overheating problem by incorporating thermal constraints where the objective is to distribute heat more evenly over the chips and reduce hot spots [30].

Consider the problem of covering a set of cell phone users by a minimum number of mobile towers. In view of the adverse effect of radio waves, it might not be wise to place two towers in close proximity of each other. That is, the placement of a tower at one point should preclude the placement of towers in all nearby points.

There are many similar 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, Set Cover. The input to Set Cover consists of a universe U of size n, a family $\mathcal{F}$ of size m of subsets of U and a positive integer k, and the objective is to check whether there exists a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ of size at most k such that $\mathcal{F}'$ is required to contain all the elements of U. The set $\mathcal{F}'$ is called a *set cover*. Set Cover is one of the Karp's 21 NP-complete problems [26]. It is one of the central problems in all the paradigms that have been established to cope with NP-hardness, including Approximation Algorithms, Randomized Algorithms and Parameterized Complexity.

Now we consider Conflict Free Interval Covering (also referred to as Rainbow Covering in the literature), described in [1,2,6]. Let P be a set of points on the x-axis, and $\mathcal{I} = \{I_1, \ldots, I_m\}$ be a set of intervals on the x-axis. Also, 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 $(1 \leq i < j \leq \ell)$, $C_i \cap C_j = \emptyset$. 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. Finally, we say $I = [a, b]$ *covers* a point p if and only if $a \leq c \leq b$. Now we are ready to define the problem formally.

Conflict Free Interval 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?

The set $\mathcal{C}$ represents *conflicts* between pairs of intervals. If $\mathcal{C}$ is an empty set, that is, if there are *no conflicts* among the intervals, then the problem (known as Covering Points by Intervals) is polynomial time solvable [17, p. 240]. On the other hand if $\mathcal{C}$ is *non-empty*, then Arkin et al. [1] showed that Conflict Free Interval Covering is NP-complete.

Our first goal is to define *a model* in which we can express a much more generalized version of conflicts beyond the matching family of conflict graphs. We use SET COVER to define our model. 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 represent a conflict. We call $CG_{\mathcal{F}}$ a *conflict graph.* Observe that in CONFLICT FREE INTERVAL COVERING, the family $\mathcal{C}$ would correspond to $CG_{\mathcal{C}}$ with degree at most one. 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 $CG_{\mathcal{C}}[Q]$ contains only isolated vertices. The set cover $\mathcal{F}'$ such that the vertex subset $\mathcal{F}' \subseteq V(CG_{\mathcal{F}})$ is an independent set in $CG_{\mathcal{F}}$ is called *conflict free set cover*.

Our Model In this paper we study the following problems in "geometric setting" in the realm of Parameterized Complexity.

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: Is there a conflict free set cover $\mathcal{F}' \subseteq \mathcal{F}$ of size at most k?

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, 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. We refer to Sect. 2 for the definition of FPT, FPT-approximation, W[1]-hardness and other related notions. For more details about parameterized complexity, refer to monographs [12,15].

Observe that GRAPHICAL CF- SC reduces to SET COVER if $CG_{\mathcal{F}}$ is an edgeless graph. 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 is either FPT or polynomial time solvable. If we want 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. 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? 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 $\mathcal{G}$.

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

($\mathscr{P}$, $\mathscr{I}$)-Graphical CF- SC
Input: A set of points $P \subseteq \mathscr{P}$, a set of intervals $\mathcal{I} = \{I_1, \dots, I_m\} \subseteq \mathscr{I}$, a conflict graph $CG_{\mathcal{I}}$ and a positive integer k.
Parameter: 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 reduces to Parameterized Conflict Free Interval Covering. 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 Conflict Free Interval Covering, which is known to be NP-complete [1]. Thus, seeking a conflict free set cover can transform a problem from being tractable to intractable.

We also use matroidal machinery to design algorithms for restricted classes of Graphical CF- SC. 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. See Sect. 3.2.1 for the definition of matroid and related concepts. In the Conflict Free Interval 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 $\mathcal{F}' \in \mathcal{J}$.

Our Results In order to restrict the family of graphs to which a *conflict graph* belongs, we need to define the notion of *arboricity*. 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 [24,25]. Specifically, this includes planar graphs, graphs excluding a fixed graph as a minor, graphs of bounded expansion, and graphs of bounded degeneracy. In most applications, conflict graphs themselves belong to a family of geometric graphs. Har-Peled and Quanrud [24,25] 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*, where* d *is a constant. 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 following hardness result. Let $\mathscr{G}$ denote a family of graphs. Let $\mathscr{G}$-Independent Set be the problem where the input is a graph $G \in \mathscr{G}$ and a positive integer k and the objective is to decide whether there exists a set S of size at least k such that S is an independent set in G.

Table 1 Corollaries of Theorem 1

$(\mathbb{R}^2, \mathcal{A})$-SC	Complexity of $(\mathbb{R}^2, \mathcal{A})$-SC	Complexity of $(\mathbb{R}^2, \mathcal{A})$-Graphical CF-SC
Disks/pseudo-disks	PTAS [35]	α-FPT approx., $\forall \alpha > 1$
Fat triangles of same size	$\mathcal{O}(1)$ [10]	$\mathcal{O}(1)$-FPT approx.
Fat objects in $\mathbb{R}^2$	$\mathcal{O}(\log^* \mathsf{OPT})$ [5]	$\mathcal{O}(\log^* \mathsf{OPT})$-FPT approx.
$\mathcal{O}(1)$ density objects in $\mathbb{R}^2$	PTAS [25]	α-FPT approx., $\forall \alpha > 1$
Objects with polylog density	QPTAS [25]	$2^{\mathcal{O}(k)} n^{\mathcal{O}(\log^* n)}$ time approx., $\forall \alpha > 1$
Objects with density $\mathcal{O}(1)$ in $\mathbb{R}^d$	PTAS [25]	α-FPT approx., $\forall \alpha > 1$
$(\mathcal{A}, \mathcal{B})$-Set Cover where every instance $(U, \mathcal{F})$ has VC dimension c	$\mathcal{O}(c \log(c\mathsf{OPT}))$ [8]	$\mathcal{O}(c \log(c\mathsf{OPT}))$-FPT approx.
Point Guard Art Gallery	$\mathcal{O}(\log \mathsf{OPT})$ [7]	$\mathcal{O}(\log \mathsf{OPT})$-FPT approx.
Terrain Guarding	PTAS [28]	α-FPT approx., $\forall \alpha > 1$
$(\mathscr{P}, \mathscr{I})$-Set Cover	Polynomial time [17]	$2^{\mathcal{O}(dk)} \cdot n^{\mathcal{O}(1)}$-FPT algorithm

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. Density and fatness are defined in Sect. 2 for completeness. The entries in the second column give the approximation ratio of the $(\mathbb{R}^2, \mathcal{A})$-SC problem based on Theorem 1

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* [36] 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.

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 as follows.

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 $(\mathscr{P}, \mathscr{I})$-Graphical CF- SC is FPT if $CG_{\mathcal{F}}$ belongs to the family of cluster graphs. The proof of Theorem 2 is given in Sect. 4.

Related Work To the best of our knowledge, "choice" problems of this kind were first studied by Gabow et al. [22]. The input to the problem is a directed acyclic graph G, two vertices s and t and k other pairs of vertices. Thus, $|V(G)| = 2k + 2$. The objective is to decide whether there exists a path from s to t such that at most one vertex from each pair can be chosen. Problems of the same flavor in geometric settings were considered by Arkin and Hassin [4]. The problem they considered is: Given a set V, a collection of (possibly nondisjoint) subsets of V and a real matrix describing distances between elements of V, a cover is a subset of V containing at least one representative from each subset. The multiple-choice minimum-diameter problem is to select a cover of minimum diameter. They also considered multiple-choice dispersion problem, which asks to maximize the minimum distance between any pair of elements in the cover. The problems were proven to be NP-hard. Another such problem considered by Arkin et al. [3] is given a set of pairs of points in the plane, we want to color one point red and another point blue in each pair so as to optimize the radii of the minimum enclosing ball of the red points and the minimum enclosing ball of the blue points. Consuegra and Narasimhan [11] also considered various geometric problems such as minimum maximum gap problem, convex hull, line segment intersection, minimum spanning Euclidean tree and others in "multiple-choice" setting.

In the parameterized setting, Set Cover, parameterized by k, is W[2]-hard [16]. In literature, various special cases of Set Cover have been studied. A few examples are instances with sets of bounded size [18], sets with bounded intersection [29,38], and instances where the bipartite incidence graph corresponding to the set family has bounded treewidth or excludes some graph H as a minor [13,20]. Apart from these results, there have also been extended study on different parameterizations of Set Cover. A special case of Set Cover which is central to the topic of this paper is the one where the *sets in the family correspond to some geometric object*. In the simplest geometric variant of Set Cover, called Point Line Cover, the elements of U are points in $\mathbb{R}^2$ and each set contains a maximal number of collinear points. This version of the problem is FPT and in fact has a polynomial kernel [29]. Moreover, the size of these kernels have been proved to be tight, under standard assumptions, in [27]. When we take the sets to be the space bounded by unit squares, Set Cover is W[1]-hard [33]. On the other hand when surfaces of hyperspheres are sets then the problem is FPT [29]. These geometric results motivate a systematic study of the Parameterized Complexity of Graphical CF- SC and Matroidal CF- SC when the input set families as well as conflict graphs are taken from geometric settings.

2 Preliminaries

For a positive integer t, we use $[t]$ as a shorthand for $\{1, 2, \ldots, t\}$. Given a function $f : D \to R$ and a subset $D' \subseteq D$, let $f|_{D'}$ denote the restriction of the function f to the domain D' and we say that $f|_{D'}$ is injective if for any $x, y \in D'$, $f(x) \neq f(y)$. 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 $\mathcal{A} \bullet \mathcal{B} = \{X \cup Y \mid X \in \mathcal{A} \text{ and } Y \in \mathcal{B} \text{ and } X \cap Y = \emptyset\}$. Throughout the paper we use ω to denote the exponent in the running time of matrix multiplication, the current best known bound for ω is < 2.373 [39]. We use e to denote the base of natural logarithm.

We use standard notation and terminology from the book of Diestel [14] for graph-related terms which are not explicitly defined here. Given a graph G, $V(G)$ and $E(G)$ denote its vertex-set and edge-set, respectively.

Low Density Graphs and Cluster Graphs A graph G is called a cluster graph, if each connected component of G is a clique.

Definition 1 ([25]) A set of objects $\mathcal{A}$ in $\mathbb{R}^d$ (not necessarily convex or connected) has density ρ if any object f (not necessarily in $\mathcal{A}$) intersects at most ρ objects in $\mathcal{A}$ with diameter greater than or equal to the diameter of f. The minimum such quantity is called the density of $\mathcal{A}$. If ρ is a constant, then $\mathcal{A}$ has low density.

Definition 2 ([25]) For $\alpha > 0$, an object $g \subseteq \mathbb{R}^d$ is α-fat if for any ball b with a center inside g, that does not contain g in its interior, we have $vol(b \cap g) \geq \alpha \cdot vol(b)$. A set $\mathcal{A}$ of objects is fat if all its members are α-fat for some constant α.

PTAS A *polynomial time approximation scheme* (PTAS) for a minimization problem is an algorithm $\mathscr{A}$ that takes an input instance, a constant $\epsilon > 0$, and returns a solution SOL such that $\mathsf{SOL} \leq (1+\epsilon)\mathsf{OPT}$, where OPT is the optimal value, and the running time of $\mathscr{A}$ is $n^{\mathcal{O}(f(\frac{1}{\epsilon}))}$, for some computable function f depending only on ϵ.

Parameterized Complexity The goal of Parameterized Complexity is to find ways of solving NP-hard problems more efficiently than brute force: here the aim is to restrict the combinatorial explosion to a parameter that is hopefully much smaller than the input size. Formally, a *parameterization* of a problem is assigning a positive integer parameter k to each input instance and we say that a parameterized problem is *fixed-parameter tractable* (FPT) if there is an algorithm that solves the problem in time $f(k)\cdot|I|^{\mathcal{O}(1)}$, where $|I|$ is the size of the input and f is an arbitrary computable function depending only on the parameter k. Such an algorithm is called an FPT algorithm and such a running time is called FPT running time. There is also an accompanying theory of parameterized intractability using which one can identify parameterized problems that are unlikely to admit FPT algorithms. These are essentially proved by showing that the problem is W-hard. We will also need the concept of FPT approximation algorithms. Towards this, we define the notion of parameterized minimization problem as defined in [32].

Definition 3 ([32]) A parameterized minimization problem Π is a computable function

$$\Pi \ : \ \Sigma^* \times \mathbb{N} \times \Sigma^* \to \mathbb{R} \cup \{\pm\infty\}.$$

The *instances* of a parameterized minimization problem Π are pairs $(I,k) \in \Sigma^* \times \mathbb{N}$, and a *solution* to (I,k) is simply a string $s \in \Sigma^*$, such that $|s| \leq |I|+k$. The *value* of the solution s is $\Pi(I,k,s)$. Just as for "classical" optimization problems the instances of Π are given as input, and the algorithmic task is to find a solution with the best possible value, where best means minimum for minimization problems.

Definition 4 ([32]) For a parameterized minimization problem Π, the *optimum value* of an instance $(I,k) \in \Sigma^* \times \mathbb{N}$ is $OPT_\Pi(I,k) = \min_{\substack{s\in\Sigma^* \\ |s|\leq|I|+k}} \Pi(I,k,s)$. For an instance (I,k) of a parameterized minimization problem Π, an *optimal solution* is a solution s such that $\Pi(I,k,s) = OPT_\Pi(I,k)$.

Definition 5 ([32]) Let $\alpha \geq 1$ be constant. A fixed parameter tractable α-approximation algorithm for a parameterized minimization problem Π is an algorithm that takes as input an instance (I,k), runs in time $f(k)|I|^{\mathcal{O}(1)}$, and outputs a solution s such that $\Pi(I,k,s) \leq \alpha \cdot OPT(I,k)$.

Note that Definition 5 only defines constant factor FPT-approximation algorithms. The definition can in a natural way be extended to approximation algorithms whose approximation ratio depends on the parameter k, on the instance I, or on both. The parameterized minimization version of Set Cover (SC)[1] can be defined as follows.

$$SC(((U,\mathcal{F}),k),\mathcal{F}') = \begin{cases} \infty & \text{if } \mathcal{F}' \text{ is not a set cover of } (U,\mathcal{F}) \\ \min\{|\mathcal{F}'|, k+1\} & \text{otherwise} \end{cases}$$

One can similarly define parameterized minimization version of other problems such as Graphical CF- SC and Matroidal CF- SC. For more background, the reader is referred to the following books [12,15,19].

[1] We refer readers to page number 15, paragraph titled "**Capping the objective function at** $k+1$" in [32] for the explanation of capping the objective function to $k+1$ in the parameterized approximation.

3 Fixed Parameter Algorithms: Proofs of Theorems 1 and 3

In this section we prove Theorems 1 and 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 [21].

3.1 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. Towards that we will exploit the following definition and theorem.

Definition 6 ([36]) 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 ([36]) *There is a deterministic algorithm with running time $\mathcal{O}(2^t t^{\mathcal{O}(\log t)} n \log n)$ 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. For an ease of presentation we restate Theorem 1.

Theorem 1 *Let $(\mathcal{A},\mathcal{B})$-*Set Cover *be tractable and let $\mathcal{G}_d$ be the family of graphs of arboricity d, where d is a constant. 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 (or PTAS) running in time $n^{\mathcal{O}(1)}$ (or $n^{f(\epsilon)}$) then $(\mathcal{A},\mathcal{B})$-*Graphical CF- SC *admits a factor α* FPT *approximation algorithm (or* FPT *PTAS) running in time $2^{\mathcal{O}(dk)} \cdot n^{\mathcal{O}(1)}$ (or $2^{\mathcal{O}(dk)} \cdot n^{f(\epsilon)}$).*

Proof 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.

Decomposing $CG_{\mathcal{F}}$ into Forests We apply the known polynomial time algorithm [23] 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: Random Coloring 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.

Step 2: A Cleaning Process Now we apply a cleaning procedure so that we get a set Z such that 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 $CG_{\mathcal{F}}[V(CG_{\mathcal{F}})\backslash\mathscr{B}]$, 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 at least $\frac{1}{2^{k(d+1)}}$. Towards that we say that the following event is *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 textsfblue.

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,

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

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.

Assume that the good event happened. Then, $S_{\mathsf{parent}} \cap Z = \emptyset$, and $\mathcal{F}' \subseteq V(CG_{\mathcal{F}})\backslash\mathscr{B}$. Moreover, since $S_{\mathsf{parent}} \subseteq \mathscr{B}$, a vertex $x \in V(CG_{\mathcal{F}})\backslash\mathscr{B}$, colored green, can not belongs to $\mathcal{F}'$, if it is a child of some vertex in some tree T_i' after deleting the vertices of $\mathscr{B}$. 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 at least $\frac{1}{2^{k(d+1)}}$, we have that $\mathcal{F}' \subseteq Z$.

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} 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 $\mathcal{O}(2^t t^{\mathcal{O}(\log(t))} n \log n)$ time. Finally, to derandomize our algorithm, rather than randomly coloring vertices with {blue, 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 sets 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 $\mathcal{F}^\star$ of size at most k, then by the properties of (n, t)-universal set we know that (a) there exists a function f that correctly colors the vertices in $\mathcal{F}^\star$ with 1 and every vertex in S_{parent} with 0. Thus, the set Z_f we will obtain by applying the Step 2, will contain the set $\mathcal{F}^\star$. 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. In other words, our algorithm returns No if the following happens: For every f, $\mathscr{S}$ either returns that (U, Z_f, k) is a No instance or the size of the solution, $\mathcal{F}'$, returned by $\mathscr{S}$ when run on (U, Z_f, k), is more than αk. The correctness of the algorithm follows from the properties of universal sets (i.e., the statement (a)) 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 of the theorem. □

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

In this section we will design an FPT algorithm proving Theorem 3. We start with restating the statement.

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.*

Towards that we need to define some basic notions related to representative families and results regarding their fast and efficient computation.

3.2.1 Matroids and Representative Families

In this subsection we give definitions related to matroids and representative families. For a broader overview on matroids we refer to [37], see also [12, Chapter 12].

Definition 7 A pair $M = (E, \mathcal{J})$, where E is a ground set and $\mathcal{J}$ is a family of subsets (called independent sets) of E, is a *matroid* if it satisfies the following conditions:

(I1) $\emptyset \in \mathcal{J}$.
(I2) If $A' \subseteq A$ and $A \in \mathcal{J}$ then $A' \in \mathcal{J}$.
(I3) If $A, B \in \mathcal{J}$ and $|A| < |B|$, then there is $e \in (B \setminus A)$ such that $A \cup \{e\} \in \mathcal{J}$.

An inclusion wise maximal set of $\mathcal{J}$ is called a *basis* of the matroid. Using axiom (I3) it is easy to show that all the bases of a matroid have the same size. This size is called the *rank* of the matroid M, and is denoted by $\mathsf{rank}(M)$.

Linear Matroids Let A be a matrix over an arbitrary field $\mathbb{F}$ and let E be the set of columns of A. For A, we define matroid $M = (E, \mathcal{J})$ as follows. A set $X \subseteq E$ is independent (that is $X \in \mathcal{J}$) if the corresponding columns are linearly independent over $\mathbb{F}$. The matroids that can be defined by such a construction are called *linear matroids*, and if a matroid can be defined by a matrix A over a field $\mathbb{F}$, then we say that the matroid is representable over $\mathbb{F}$. That is, a matroid $M = (E, \mathcal{J})$ of rank d is representable over a field $\mathbb{F}$ if there exist vectors in $\mathbb{F}^d$ corresponding to the elements such that linearly independent sets of vectors correspond to independent sets of the matroid. A matroid $M = (E, \mathcal{J})$ is called *representable* or *linear* if it is representable over some field $\mathbb{F}$.

Partition Matroids A partition matroid $M = (E, \mathcal{J})$ is defined by a ground set E being partitioned into (disjoint) sets $E_1, \ldots, E_\ell$ and by ℓ non-negative integers $k_1, \ldots, k_\ell$. A set $X \subseteq E$ is independent if and only if $|X \cap E_i| \leq k_i$ for all $i \in \{1, \ldots, \ell\}$.

Proposition 1 ([34, Proposition 3.5]) *A representation over a field of size $\mathcal{O}(|E|)$ of a partition matroid can be constructed in polynomial time.*

Representative Families Now we define the notion of q-representative family and state the results about its efficient computation.

Definition 8 (*q-Representative Family* [34]) 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 $\widehat{\mathcal{S}} \subseteq_{rep}^{q} \mathcal{S}$.

In other words if some independent set in $\mathcal{S}$ can be extended to a larger independent set by adding q new elements, then there is a set in $\widehat{\mathcal{S}}$ that can be extended by the same q elements.

Lemma 1 ([21]) *Let $M = (E, \mathcal{J})$ be a matroid and $\mathcal{S}$ be a family of subsets of E. If $\mathcal{S}' \subseteq_{rep}^{q} \mathcal{S}$ and $\widehat{\mathcal{S}} \subseteq_{rep}^{q} \mathcal{S}'$, then $\widehat{\mathcal{S}} \subseteq_{rep}^{q} \mathcal{S}$.*

Theorem 5 ([21,31]) *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 $\mathbb{F}$ is $\mathbb{Q}$ or $\mathbb{F} = \mathbb{F}_{(\widehat{p})^\ell}$ for a prime $\widehat{p}$ and $\ell \in \mathbb{N}$. Then, there is a deterministic algorithm computing $\widehat{\mathcal{S}} \subseteq_{rep}^{q} \mathcal{S}$ of size $np\binom{p+q}{p}$ in $\mathcal{O}\left(\binom{p+q}{p}tp^3n^2 + t\binom{p+q}{q}^{\omega-1}(pn)^{\omega-1}\right) + (n + |E|)^{\mathcal{O}(1)}$ operations over $\mathbb{F}$.*

We also use the notion of $[k]$-representative families for our algorithm, which is defined as follows.

Definition 9 (*$[k]$-Representative Family*) Let $M = (E, \mathcal{J})$ be a matroid, and let $\mathcal{S}$ be a family of subsets of E such that for all $S \in \mathcal{S}$, $|S| \leq k$. For any $i \in [k]$, $\mathcal{S}^i$ denote the subset of $\mathcal{S}$ exactly containing all the sets in $\mathcal{S}$ of size exactly i. We say that a subfamily $\widehat{\mathcal{S}} \subseteq \mathcal{S}$ is a *$[k]$-representative* for $\mathcal{S}$ if the following holds. For any $i \in [k]$, $\widehat{\mathcal{S}^i} = \widehat{\mathcal{S}} \cap \{A \in \widehat{\mathcal{S}} \colon |A| = i\}$ is a $(k-i)$-representative for $\mathcal{S}^i$ (i.e., $\widehat{\mathcal{S}^i} \subseteq_{rep}^{k-j} \mathcal{S}^i$). If $\widehat{\mathcal{S}} \subseteq \mathcal{S}$ is $[k]$-representative for $\mathcal{S}$, we write $\widehat{\mathcal{S}} \subseteq_{rep}^{[k]} \mathcal{S}$ or $\widehat{\mathcal{S}} \subseteq_{rep}^{1\ldots k} \mathcal{S}$.

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

Now we have gathered all the tools required 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 the first i points and are independent in the matroid $M = (\mathcal{I}, \mathcal{J})$. Furthermore, for every $1 \leq j \leq k$, 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, a 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. Also, observe 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 2 *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 main step of the algorithm is the computation of $\widehat{\mathcal{P}^i} \subseteq^{1\cdots k}_{rep} \mathcal{P}^i$ for all $i \in [n]$. Recall from the definition of $[k]$-representative family that $\widehat{\mathcal{P}^i}$ is a union of $\widehat{\mathcal{P}}^{i1}, \ldots, \widehat{\mathcal{P}}^{ik}$ where $\widehat{\mathcal{P}}^{ij} \subseteq^{k-j}_{rep} \mathcal{P}^{ij}$ for all $1 \leq j \leq k$. In particular, for every $1 \leq i \leq n$, we compute

$$\widehat{\mathcal{P}^i} = \bigcup_{j=1}^{k} \widehat{\mathcal{P}}^{ij}$$

where $\widehat{\mathcal{P}}^{ij} \subseteq^{k-j}_{rep} \mathcal{P}^{ij}$ for all $1 \leq j \leq k$. The following observation follows from the definition of representative families.

Observation 1 *Let $i \in [n]$ and $\widehat{\mathcal{P}^i} \subseteq^{1\cdots k}_{rep} \mathcal{P}^i$. Then, $\widehat{\mathcal{P}^i}$ is a 0-representative for $\mathcal{P}^i$ (i.e., $\widehat{\mathcal{P}^i} \subseteq^0_{rep} \mathcal{P}^i$).*

Proof If $\mathcal{P}^i = \emptyset$, then $\widehat{\mathcal{P}^i} = \emptyset$, and the proof of the observation is trivial. Now on, we assume that $\mathcal{P}^i \neq \emptyset$. Then, $\widehat{\mathcal{P}^i}$ is a 0-representative for $\mathcal{P}^i$ if and only if $\widehat{\mathcal{P}^i} \neq \emptyset$. Towards proving $\widehat{\mathcal{P}^i} \neq \emptyset$, consider a set $X \in \mathcal{P}^i$. Let $j = |X|$. By the definition of $\mathcal{P}^i$, we have that $1 \leq j \leq k$ and $X \in \mathcal{P}^{ij}$. Since $\widehat{\mathcal{P}^i} \cap \{Y \in \widehat{\mathcal{P}^i} : |Y| = j\}$ is a

$(k-j)$-representative for $\mathcal{P}^{ij}$ (because $\widehat{\mathcal{P}}^i \subseteq_{rep}^{1\cdots k} \mathcal{P}^i$), we have that $\widehat{\mathcal{P}}^i$ is nonempty. This completes the proof of the observation. □

Lemma 3 *Let $(P, \mathcal{I}, k, M = (\mathcal{I}, \mathcal{J}))$ be an instance of $(\mathscr{P}, \mathscr{I})$-*Matroidal CF-SC. *Then, a collection $\{\widehat{\mathcal{P}}^i \subseteq_{rep}^{1\cdots k} \mathcal{P}^i : 1 \le i \le n\}$ of families where each family is 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 $\widehat{\mathcal{P}}^i \subseteq_{rep}^{1\cdots k} \mathcal{P}^i$. We fill the entries in the array $\mathcal{D}$ 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 $\widehat{\mathcal{P}}^i \subseteq_{rep}^{1\cdots k} \mathcal{P}^i$). 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 Eq. 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 Eq. 1 is well defined. Moreover, any set in the family $\mathcal{N}^{i+1}$ covers the points $p_1, \ldots, p_{i+1}$. Let $\mathcal{N}^{(i+1)j}$ denote the subfamily of $\mathcal{N}^{i+1}$ containing the sets of size exactly j. □

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

Proof Let $S \in \mathcal{P}^{(i+1)j}$ and Y be a set of size at most $k-j$ (which is essentially an independent set in M) such that $S \cap Y = \emptyset$ and $S \cup Y \in \mathcal{J}$. We will show that there exists a set $\widehat{S} \in \mathcal{N}^{(i+1)j}$ 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 $S' \in \mathcal{P}^{(\ell_J - 1)(j-1)}$. By our assumption, $\mathcal{D}[\ell_J - 1] = \widehat{\mathcal{P}}^{(\ell_I - 1)} \subseteq_{rep}^{1\cdots k} \mathcal{P}^{(\ell_I - 1)}$. Let $\widehat{\mathcal{P}}^{(\ell_J - 1)(j-1)} = \{X \in \widehat{\mathcal{P}}^{(\ell_J - 1)} : |X| = j-1\}$. Since $\widehat{\mathcal{P}}^{(\ell_J - 1)(j-1)} \subseteq_{rep}^{k-j+1} \mathcal{P}^{(\ell_J - 1)(j-1)}$, $S' \in \mathcal{P}^{(\ell_J - 1)(j-1)}$, and $S' \cup Y' \in \mathcal{J}$, we have that there exists $S^* \in \widehat{\mathcal{P}}^{(\ell_J - 1)(j-1)} \subseteq \mathcal{D}[\ell_J - 1]$ such that $S^* \cap Y' = \emptyset$ and $S^* \cup Y' \in \mathcal{J}$. By Eq. 1, $\widehat{S} = S^* \cup \{J\}$ is in $\mathcal{N}^{i+1}$, because $S^* \cup \{J\} \in \mathcal{J}$. Since $|S^*| = j-1$, we have that $|\widehat{S}| = j$. Observe that $\widehat{S} \in \mathcal{N}^{(i+1)j}$, $\widehat{S} \cap Y = \emptyset$, and $\widehat{S} \cup Y \in \mathcal{J}$. This completes the proof of the claim. □

For every $1 \le j \le k$, we compute $\widehat{\mathcal{N}}^{(i+1)j} \subseteq_{rep}^{k-j} \mathcal{N}^{(i+1)j}$ using Theorem 5. Then, we fill the entry for $\mathcal{D}[i+1]$ as follows.

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

Lemma 1 and Claim 3.2.2 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 Theorem 5, 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 Theorem 5, 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 operations over the field in which A_M is given and $|\widehat{\mathcal{N}}^{(i+1)j}| \leq |\mathcal{I}| \cdot k \cdot \binom{k}{j}$. 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 Theorem 5, the cardinality of $\mathcal{D}[i+1]$ is bounded by,

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

This completes the proof. □

Theorem 3 follows from Observation 1 and Lemmata 2 and 3. 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 $i \in [t]$. As a result, by applying Theorem 3 along with Proposition 1, 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* $2^{\omega k} \cdot (n+|\mathcal{I}|)^{\mathcal{O}(1)}$.

This corollary also gives an FPT algorithm for the Conflict Free Set Cover with the running time $2^{\omega k} \cdot (n+|\mathcal{I}|)^{\mathcal{O}(1)}$. However, Banik et. al [6] have already considered the Conflict Free Set Cover from parameterized complexity perspective, and gave an FPT algorithm with the running time $2^k \cdot (n+|\mathcal{I}|)^{\mathcal{O}(1)}$.

4 Hardness

In this section we prove the following theorem.

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].

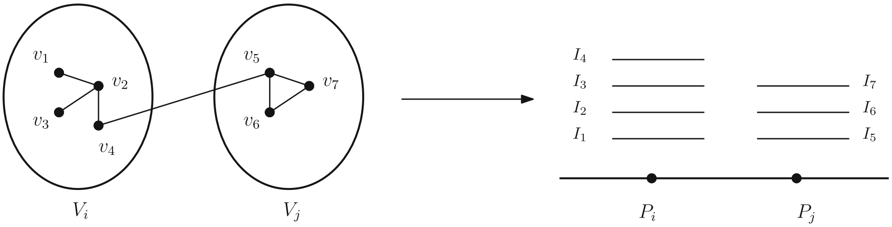

Fig. 1 Here V_i and V_j are the set of vertices in G which are colored by i and j, respectively

Towards proving Theorem 2, we give a Turing reduction from $\mathscr{G}$-Independent Set to $(\mathscr{P}, \mathscr{I})$-Graphical CF- SC where $CG_{\mathcal{I}}$ belongs to $\mathscr{G}$. To give such a reduction we need the notion of (n, k)-*perfect hash families*. Towards that, we first define (n, k)-perfect hash family and state a theorem about efficient computation of these objects.

Theorem 6 ([36]) *There is a deterministic algorithm with running time* $e^k k^{\mathcal{O}(\log k)} n \log n$ *that constructs an* (n, k)*-perfect hash family* $\mathscr{F}$ *of cardinality at most* $e^k k^{\mathcal{O}(\log k)} \log n$.

Proof of Theorem 2 We give a Turing reduction from $\mathscr{G}$-Independent Set. Let (G, k) be an instance of $\mathscr{G}$-Independent Set and $n = |V(G)|$. We first apply Theorem 6 and construct a (n, k)-perfect hash family $\mathscr{F}$. Then, we construct $|\mathscr{F}|$ many instances of $(\mathscr{P}, \mathscr{I})$-Graphical CF- SC such that the parameter value in each instance is k and (G, k) is a Yes instance of $\mathscr{G}$-Independent Set if and only if at least one instance constructed is a Yes instance of $(\mathscr{P}, \mathscr{I})$-Graphical CF- SC. Moreover, the running time of our reduction will be $e^k k^{\mathcal{O}(\log k)} n \log n$.

Now we give details about the reduction. As mentioned earlier let (G, k) be the given instance of $\mathscr{G}$-Independent Set and $n = |V(G)|$. For each $f \in \mathscr{F}$ we create an instance $(P, \mathcal{I}, G, k)$. Notice that in our reduction the conflict graph is G itself. Now we create a set of points $P = \{p_1 = (1, 0), p_2 = (2, 0), \ldots, p_k = (k, 0)\}$. We can think of f as a coloring function on $V(G)$, where each vertex in v gets a color from $[k]$. For each vertex $v \in V(G)$ we create an interval $I_v = [f(v) - 0.5, f(v) + 0.5]$. Notice that any interval I_v covers only the point $p_{f(v)}$. See Fig. 1 for an illustration.

Now we prove that (G, k) is a Yes instance of $\mathscr{G}$-Independent Set if and only if at least one instance constructed is a Yes instance of $(\mathscr{P}, \mathscr{I})$-Graphical CF- SC. Suppose (G, k) is a Yes instance of $\mathscr{G}$-Independent Set and S is an independent set of size k in G. By the property of (n, k)-perfect hash family, there is a function f such that $f|_S$ is injective. Now consider the instance $(P, \mathcal{I}, G, k)$ created for f. The set of intervals $\{I_v : v \in S\}$ covers P and S is independent in G. Hence $(P, \mathcal{I}, G, k)$ is a Yes instance. Now suppose there is an instance $(P, \mathcal{I}, G, k)$ created for a function $f \in \mathscr{F}$, such that $(P, \mathcal{I}, G, k)$ is a Yes instance. This implies that there is a k sized independent set in G and hence (G, k) is a Yes instance of $\mathscr{G}$-Independent Set.

Because of Theorem 6, the running time of our reduction is $e^k k^{\mathcal{O}(\log k)} n \log n$. This completes the proof of the theorem. □

Acknowledgements Funding was provided by FP7 Ideas: European Research Council (Grant No. 306992).

References

1. Arkin, E.M., Banik, A., Carmi, P., Citovsky, G., Katz, M.J., Mitchell, J.S.B., Simakov, M.: Choice is hard. In: Proc. 26th Internat. Sympos. Algorithms and Computation, ISAAC 2015, pp. 318–328 (2015). https://doi.org/10.1007/978-3-662-48971-0_28
2. Arkin, E.M., Banik, A., Carmi, P., Citovsky, G., Katz, M.J., Mitchell, J.S.B., Simakov, M.: Conflict-free covering. In: Proc. 27th Canadian Conf. on Comput. Geom., CCCG 2015 (2015)
3. Arkin, E.M., Díaz-Báñez, J.M., Hurtado, F., Kumar, P., Mitchell, J.S.B., Palop, B., Pérez-Lantero, P., Saumell, M., Silveira, R.I.: Bichromatic 2-center of pairs of points. Comput. Geom. **48**(2), 94–107 (2015). https://doi.org/10.1016/j.comgeo.2014.08.004
4. Arkin, E.M., Hassin, R.: Minimum-diameter covering problems. Networks **36**(3), 147–155 (2000)
5. Aronov, B., de Berg, M., Ezra, E., Sharir, M.: Improved bounds for the union of locally fat objects in the plane. SIAM J. Comput. **43**(2), 543–572 (2014)
6. Banik, A., Panolan, F., Raman, V., Sahlot, V.: Fréchet distance between a line and avatar point set. Algorithmica **80**(9), 2616–2636 (2018). https://doi.org/10.1007/s00453-017-0352-y
7. Bonnet, É., Miltzow, T.: An approximation algorithm for the art gallery problem. In: 33rd International Symposium on Computational Geometry, SoCG 2017, 4–7 July 2017, Brisbane, Australia, pp. 20:1–20:15 (2017). https://doi.org/10.4230/LIPIcs.SoCG.2017.20
8. Brönnimann, H., Goodrich, M.T.: Almost optimal set covers in finite VC-dimension. Discrete Comput. Geom. **14**(4), 463–479 (1995)
9. Chan, T.M.: Polynomial-time approximation schemes for packing and piercing fat objects. J. Algorithms **46**(2), 178–189 (2003)
10. Clarkson, K.L., Varadarajan, K.R.: Improved approximation algorithms for geometric set cover. Discrete Comput. Geom. **37**(1), 43–58 (2007)
11. Consuegra, M.E., Narasimhan, G.: Geometric avatar problems. In: IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2013, LIPIcs, vol. 24, pp. 389–400 (2013). https://doi.org/10.4230/LIPIcs.FSTTCS.2013.389
12. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer, Berlin (2015)
13. Demaine, E.D., Fomin, F.V., Hajiaghayi, M.T., Thilikos, D.M.: Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs. J. ACM **52**(6), 866–893 (2005). https://doi.org/10.1145/1101821.1101823
14. Diestel, R.: Graph Theory. Graduate Texts in Mathematics, vol. 173, 4th edn. Springer, Berlin (2012)
15. Downey, R., Fellows, M.: Fundamentals of Parameterized Complexity. Springer, Berlin (2013)
16. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer, Berlin (1999)
17. Edmonds, J.: How to Think About Algorithms. Cambridge University Press, New York (2008)
18. Fellows, M.R., Knauer, C., Nishimura, N., Ragde, P., Rosamond, F.A., Stege, U., Thilikos, D.M., Whitesides, S.: Faster fixed-parameter tractable algorithms for matching and packing problems. Algorithmica **52**(2), 167–176 (2008). https://doi.org/10.1007/s00453-007-9146-y
19. Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. An EATCS Series. Springer, Berlin (2006)
20. Fomin, F.V., Gaspers, S., Saurabh, S., Stepanov, A.A.: On two techniques of combining branching and treewidth. Algorithmica **54**(2), 181–207 (2009). https://doi.org/10.1007/s00453-007-9133-3
21. 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)
22. Gabow, H.N., Maheshwari, S.N., Osterweil, L.J.: On two problems in the generation of program test paths. IEEE Trans. Softw. Eng. **2**(3), 227–231 (1976)
23. Gabow, H.N., Westermann, H.H.: Forests, frames, and games: algorithms for matroid sums and applications. Algorithmica **7**(5&6), 465–497 (1992)
24. Har-Peled, S., Quanrud, K.: Approximation algorithms for low-density graphs (2015). CoRR arXiv:1501.00721
25. Har-Peled, S., Quanrud, K.: Approximation algorithms for polynomial-expansion and low-density graphs. In: Algorithms—ESA 2015—23rd Annual European Symposium, Patras, Greece, 14–16 September 2015, Proceedings, vol. 9294, pp. 717–728. Springer (2015)
26. Karp, R.M.: Reducibility among combinatorial problems. In: Proceedings of a symposium on the Complexity of Computer Computations, pp. 85–103 (1972)

27. Kratsch, S., Philip, G., Ray, S.: Point line cover: the easy kernel is essentially tight. ACM Trans. Algorithms **12**(3), 40:1–40:16 (2016). https://doi.org/10.1145/2832912
28. Krohn, E., Gibson, M., Kanade, G., Varadarajan, K.R.: Guarding terrains via local search. JoCG **5**(1), 168–178 (2014)
29. Langerman, S., Morin, P.: Covering things with things. Discrete Comput. Geom. **33**(4), 717–729 (2005)
30. Liu, C., Veeraraghavan, K., Iyengar, V.: Thermal-aware test scheduling and hot spot temperature minimization for core-based systems. In: 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05), pp. 552–560. IEEE (2005)
31. Lokshtanov, D., Misra, P., Panolan, F., Saurabh, S.: Deterministic truncation of linear matroids. ACM Trans. Algorithms **14**(2), 14:1–14:20 (2018). https://doi.org/10.1145/3170444
32. Lokshtanov, D., Panolan, F., Ramanujan, M.S., Saurabh, S.: Lossy kernelization. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, 19-23 June 2017, pp. 224–237 (2017). https://doi.org/10.1145/3055399.3055456
33. Marx, D.: Efficient approximation schemes for geometric problems? In: ESA, pp. 448–459. Springer (2005)
34. Marx, D.: A parameterized view on matroid optimization problems. Theor. Comput. Sci. **410**(44), 4471–4479 (2009)
35. Mustafa, N.H., Raman, R., Ray, S.: Settling the APX-hardness status for geometric set cover. In: 55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, pp. 541–550. IEEE Computer Society (2014)
36. Naor, M., Schulman, J.L., Srinivasan, A.: Splitters and near-optimal derandomization. In: FOCS, pp. 182–191 (1995)
37. Oxley, J.G.: Matroid Theory, vol. 3. Oxford University Press, Oxford (2006)
38. Raman, V., Saurabh, S.: Short cycles make W-hard problems hard: FPT algorithms for W-hard problems in graphs with no short cycles. Algorithmica **52**(2), 203–225 (2008)
39. Williams, V.V.: Multiplying matrices faster than Coppersmith-Winograd. pp. 887–898. ACM (2012)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Aritra Banik[1] · Fahad Panolan[2] · Venkatesh Raman[3] · Vibha Sahlot[4] · Saket Saurabh[3]

✉ Saket Saurabh
saket@imsc.res.in

Aritra Banik
aritrabanik@gmail.com

Fahad Panolan
fahad.panolan@ii.uib.no

Venkatesh Raman
vraman@imsc.res.in

Vibha Sahlot
sahlotvibha@gmail.com

[1] National Institute of Science Education and Research, Bhubaneswar, India

[2] Department of Informatics, University of Bergen, Bergen, Norway

[3] The Institute of Mathematical Sciences, HBNI, Chennai, India

[4] Indian Institute of Technology, Jodhpur, India