Satisfiability to Coverage in Presence of Fairness, Matroid, and Global Constraints

Tanmay Inamdar
Indian Institute of Technology Jodhpur, India

Pallavi Jain
Indian Institute of Technology Jodhpur, India

Daniel Lokshtanov
University of California Santa Barbara, CA, USA

Abhishek Sahu
National Institute of Science, Education and Research Bhubaneswar, HBNI, India

Saket Saurabh
The Institute of Mathematical Sciences, HBNI, Chennai, India
University of Bergen, Norway

Anannya Upasana[1]
The Institute of Mathematical Sciences, HBNI, Chennai, India

Abstract

In the MaxSAT with Cardinality Constraint problem (CC-MaxSAT), we are given a CNF-formula Φ, and a positive integer k, and the goal is to find an assignment β with at most k variables set to true (also called a weight k-assignment) such that the number of clauses satisfied by β is maximized. Maximum Coverage can be seen as a special case of CC-MaxSat, where the formula Φ is monotone, i.e., does not contain any negative literals. CC-MaxSat and Maximum Coverage are extremely well-studied problems in the approximation algorithms as well as the parameterized complexity literature.

Our first conceptual contribution is that CC-MaxSat and Maximum Coverage are equivalent to each other in the context of FPT-Approximation parameterized by k (here, the approximation is in terms of the number of clauses satisfied/elements covered). In particular, we give a randomized reduction from CC-MaxSat to Maximum Coverage running in time $\mathcal{O}(1/\epsilon)^k \cdot (m+n)^{\mathcal{O}(1)}$ that preserves the approximation guarantee up to a factor of $(1-\epsilon)$. Furthermore, this reduction also works in the presence of "fairness" constraints on the satisfied clauses, as well as matroid constraints on the set of variables that are assigned true. Here, the "fairness" constraints are modeled by partitioning the clauses of the formula Φ into r different colors, and the goal is to find an assignment that satisfies at least t_j clauses of each color $1 \leq j \leq r$.

Armed with this reduction, we focus on designing FPT-Approximation schemes (FPT-ASes) for Maximum Coverage and its generalizations. Our algorithms are based on a novel combination of a variety of ideas, including a carefully designed probability distribution that exploits sparse coverage functions. These algorithms substantially generalize the results in Jain et al. [SODA 2023] for CC-MaxSat and Maximum Coverage for $K_{d,d}$-free set systems (i.e., no d sets share d elements), as well as a recent FPT-AS for Matroid Constrained Maximum Coverage by Sellier [ESA 2023] for frequency-d set systems.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Partial Vertex Cover, Max SAT, FPT Approximation, Matroids

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.88

[1] corresponding author

51st International Colloquium on Automata, Languages, and Programming (ICALP 2024).
Editors: Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson;
Article No. 88; pp. 88:1–88:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2403.07328

Funding *Tanmay Inamdar*: Part of this work was done when the author was affiliated with the University of Bergen, Norway and was funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819416).
Pallavi Jain: The author acknowledges the support from SERB-SUPRA (grant number SPR/2021/000860) and ITJ Seed Grant (grant number I/SEED/PJ/20210119).
Daniel Lokshtanov: The author is supported by NSF award CCF-2008838.
Saket Saurabh: The author is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819416); and he also acknowledges the support of Swarnajayanti Fellowship grant DST/SJF/MSA-01/2017-18.

1 Introduction

Two problems that have gained considerable attention from the perspective of Parameterized Approximation [11] are the classical MaxSAT with cardinality constraint (CC-MaxSat) problem and its monotone version, the Maximum Coverage problem. In the CC-MaxSat problem, we are given a CNF-formula Φ over m clauses and n variables, and a positive integer k, and the objective is to find a weight k assignment that maximizes the number of satisfied clauses. We use $\mathsf{var}(\Phi)$ and $\mathsf{cla}(\Phi)$ to denote the set of variables and clauses in Φ, respectively. An *assignment* to a CNF-formula Φ is a function $\beta : \mathsf{var}(\Phi) \to \{0, 1\}$. The *weight* of an assignment β is the number of variables that have been assigned 1.

The classical Maximum Coverage problem is a special case of the CC-MaxSat problem. Indeed, it is a monotone variant of CC-MaxSat, where negated literals are not allowed. An input to the Maximum Coverage problem consists of a family of m sets, $\mathcal{F}$, over a universe U of size n, and an integer k, and the goal is to find a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ of size k such that the number of elements *covered* (belongs to some set in $\mathcal{F}'$) by $\mathcal{F}'$ is maximized. Observe that when the goal is to cover every element in U, the Maximum Coverage problem corresponds to Set Cover. A natural question that has guided research on these problems is whether CC-MaxSat or Maximum Coverage admits an algorithm with running time $f(k)n^{\mathcal{O}(1)}$? That is, whether CC-MaxSat or Maximum Coverage is fixed parameter tractable (FPT) with solution size k? Unfortunately, these problems are W[2]-hard [9]. That is, we do not expect these problems to admit an algorithm with running time $f(k)n^{\mathcal{O}(1)}$. This negative result sets the platform for studying these problems from the viewpoint of Parameterized Approximation [11]. It is well known that both CC-MaxSat and Maximum Coverage admit a polynomial time $(1 - \frac{1}{e})$-approximation algorithm [26], which is in fact optimal. [10]. So, in the realm of Parameterized Approximation, we ask does there exist an $\epsilon > 0$, such that CC-MaxSat or Maximum Coverage admits an approximation algorithm with factor $(1 - \frac{1}{e} + \epsilon)$ and runs in time $f(k, \epsilon)n^{\mathcal{O}(1)}$. While there has been a lot of work on Maximum Coverage [17, 20, 25, 15, 24], Jain et al. [17] studied CC-MaxSat and designed a standalone algorithm for the problem. Our first result, a bit of a surprise to us, shows that in the world of Parameterized Approximation, CC-MaxSat and Maximum Coverage are *"equivalent"*.

▶ **Theorem 1.1** (Informal). *Let $\epsilon > 0$. There is a polynomial time randomized algorithm that given an instance (Φ, k) of CC-MaxSat produces an instance $(U, \mathcal{F}, k)$ of* Maximum Coverage *such that the following holds with probability $\frac{1}{2}(\frac{\epsilon}{2})^k$. Given a $(1 - \epsilon)\mathsf{OPT}_{\mathsf{cov}}$ solution to $(U, \mathcal{F}, k)$ we can obtain a $(1-\epsilon)\mathsf{OPT}_{\mathsf{sat}}$ solution to (Φ, k) in polynomial time. Here, $\mathsf{OPT}_{\mathsf{cov}}$ ($\mathsf{OPT}_{\mathsf{sat}}$) denotes the value of the maximum number of covered elements (satisfied clauses) by a k-sized family of subsets (weight k assignment).*

Theorem 1.1 allows us to focus on Maximum Coverage, rather than CC-MaxSat, at the expense of $\epsilon^{-\mathcal{O}(k)}$ in the running time. Further, there is no assumption on the input formulas in Theorem 1.1. This reduction immediately implies faster algorithms for CC-MaxSat by utilizing the known good algorithms for Maximum Coverage [17, 20, 25, 15, 24]. The Maximum Coverage problem has been generalized in several directions by adding either fairness constraints or asking our solution to be an independent set of a matroid. In what follows, we take a closer look at progresses on Maximum Coverage and its generalizations and then design algorithms that generalize and unify all the known results for CC-MaxSat and Maximum Coverage.

1.1 Tractability Boundaries for Maximum Coverage

Cohen-Addad et al. [8] studied Maximum Coverage and showed that there is no $\epsilon > 0$, such that Maximum Coverage admits an approximation algorithm with factor $(1 - \frac{1}{e} + \epsilon)$ and runs in time $f(k, \epsilon)(m + n)^{\mathcal{O}(1)}$ [2]. Later, this was also studied by Manurangsi [20], who obtained the following strengthening over [8]: for any constant $\epsilon > 0$ and any function h, assuming Gap-ETH, no $h(k)(n + m)^{o(k)}$ time algorithm can approximate Maximum Coverage with n elements and m sets to within a factor $(1 - \frac{1}{e} + \epsilon)$, even with a promise that there exist k sets that fully cover the whole universe. This negative result sets the contour for possible positive results. In particular, if we hope for an FPT algorithm that improves over a factor $(1 - \frac{1}{e})$ then we must assume some additional structure on the input families. This automatically leads to the families wherein each set has bounded size, or each element appears in bounded sets which was considered earlier.

Skowron and Faliszewski [25] showed that, if we are working on set families, such that each element in U appears in at most p sets, then there exists an algorithm, that given an $\epsilon > 0$, runs in time $(\frac{p}{\epsilon})^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$ and returns a subfamily $\mathcal{F}'$ of size k that is a $(1-\epsilon)$-approximation. These kind of FPT-approximation algorithms are called FPT-approximation Schemes (FPT-ASes). For $p = 2$, Manurangsi [20] independently obtained a similar result. Jain et al. [17] generalized these two settings by looking at $K_{d,d}$-free set systems (i.e., no d sets share d elements). They also considered $K_{d,d}$-free formulas (that is, the clause-variable incidence bipartite graph of the formula excludes $K_{d,d}$ as an induced subgraph). They showed that for every $\epsilon > 0$, there exists an algorithm for $K_{d,d}$-free formulas with approximation ratio $(1 - \epsilon)$ and running in time $2^{\mathcal{O}((\frac{dk}{\epsilon})^d)}(n + m)^{\mathcal{O}(1)}$. For, Maximum Coverage on $K_{d,d}$-free set families, they obtain an FPT-AS with running time $(\frac{dk}{\epsilon})^{\mathcal{O}(dk)} n^{\mathcal{O}(1)}$. Using these results together with Theorem 1.1 we get the following.

▶ **Corollary 1.2.** *Let $\epsilon > 0$. Then, CC-MaxSat admits a randomized* FPT-AS *with running time $(\frac{dk}{\epsilon})^{\mathcal{O}(dk)} n^{\mathcal{O}(1)}$ on $K_{d,d}$-free formulas. Furthermore, if the size of clauses is bounded by p or every variable appears in at most p clauses then CC-MaxSat admits randomized* FPT-AS *with running time $(\frac{p}{\epsilon})^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$. Both results hold with constant probability.*

We note that the deterministic version of the first result in the corollary was given in [21] in a recent independent work.

Corollary 1.2 follows by utilizing Theorem 1.1 and repurposing the known results about Maximum Coverage ([17, 5, 25, 20]). We will return to the case of $K_{d,d}$-free set systems later. Apart from extending the classes of set families where Maximum Coverage admits FPT-ASes, the study on the Maximum Coverage problem has been extended in many directions.

[2] Throughout the paper, the approximation factor will refer to the number of elements covered/number of satisfied clauses, unless explicitly stated otherwise

1.1.1 Matroid Constraints

Note that Maximum Coverage is a special case of submodular function maximization subject to a cardinality constraint. In the latter problem, we are given (an oracle access to) a submodular function $f : 2^V \to \mathbb{R}_{\geq 0}$ [3], and the goal is to find a subset $U \subseteq V$ that maximizes $f(U)$ over all subsets of size at most k. Indeed, coverage functions are submodular and monotone (i.e., adding more sets cannot decrease the number of elements covered). There has been a plethora of work on monotone submodular maximization subject to cardinality constraints, starting from Wolsey [27]. In a further generalization, we are interested in monotone submodular maximization subject to a matroid constraint – in this setting, we are given a matroid $\mathcal{M} = (U, \mathcal{I})$ [4] via an *independence oracle*, i.e., an algorithm that answers queries of the form "Is $P \in \mathcal{I}$?" for any $P \subseteq U$ in one step, and we want to find an independent set $S \in I$ that maximizes $f(S)$. Note here that a uniform matroid of rank k [5] exactly captures the cardinality constraint. Calinescu et al. [6] gave an optimal $(1 - 1/e)$-approximation.

More recently, Huang and Sellier [15] and Sellier [24] studied the problem of maximizing a coverage function subject to a matroid constraint, called Matroid Constrained Maximum Coverage. In this problem, which we call M-MaxCov (M for "matroid" constraint), we are given a set system $(U, \mathcal{F})$ and a matroid $\mathcal{M} = (\mathcal{F}, I)$ of rank k, and the goal is to find a subset $\mathcal{F}' \subseteq \mathcal{F}$ such that $\mathcal{F}' \in I$ and $\mathcal{F}'$ maximizes the number of elements covered. Note that M-MaxCov is a generalization of Maximum Coverage. In the latter paper, Sellier [24] designed an FPT-AS for M-MaxCov, running in time $(d/\epsilon)^{\mathcal{O}(k)} \cdot (m+n)^{\mathcal{O}(1)}$ for frequency-d set systems. Note that this result generalizes that of [25, 20] from a uniform matroid consraint to an arbitrary matroid constraint of rank k.

Analogous to M-MaxCov, one can define a matroid constrained version of CC-MaxSat, called M-MaxSAT. In this problem, we are given a CNF-SAT formula Φ and a matroid $\mathcal{M}$ of rank k on the set of variables. The goal is to find an assignment that satisfies the maximum number of clauses, with the restriction that, the set of variables assigned 1 must be an independent set in $\mathcal{M}$. Note that M-MaxSAT generalizes M-MaxCov as well as CC-MaxSat. We obtain the following result for M-MaxSAT, by combining the results on a variant of Theorem 1.1 with the corresponding result on M-MaxCov.

▶ **Theorem 1.3.** *There exists an* FPT-AS *for M-MaxSAT parameterized by k, d, and ϵ, on d-CNF formulas, where k denotes the rank of the given matroid.*

1.1.2 Fairness or Multiple Coverage Constraints

Now we consider an orthogonal generalization of Maximum Coverage. Note that an optimal solution for Maximum Coverage may leave many elements uncovered. However, such a solution may be deemed *unfair* if the elements are divided into multiple colors (representing, say, people of different demographic groups), and the set uncovered elements are biased against a specific color. To address these constraints, the following generalization of Maximum Coverage, which we call F-MaxCov (F stands for "fair"), has been studied in the literature. Here, we are given a set system $(U, \mathcal{F})$, a coloring function $\chi : U \to [r]$, a

---

[3] $f : 2^V \to \mathbb{R}$ is submodular if it satisfies $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ for all $A, B \subseteq V$

[4] Recall that a matroid is a pair $\mathcal{M} = (U, \mathcal{I})$, where U is the ground set, and $\mathcal{I}$ is a family of subsets of U satisfying the following three axioms: (i) $\emptyset \in \mathcal{I}$, (ii) If $A \in \mathcal{I}$, then $B \in \mathcal{I}$ for all subsets $B \subseteq A$, and (iii) for any $A, B \in \mathcal{I}$ with $|B| > |A|$, then there exists an element $e \in B \setminus A$ such that $A \cup \{e\} \in \mathcal{I}$.

[5] Rank of a matroid is equal to the maximum size of any independent set in the matroid.

coverage requirement function $t : [r] \rightarrow \mathbb{N}$, and an integer k; and the goal is to find a subset $\mathcal{F}' \subseteq \mathcal{F}$ of size at most k such that, for each $i \in [r]$, the union of elements in $\mathcal{F}'$ is at least $t(i)$ (or t_i).

Since F-MaxCov is a generalisation of Maximum Coverage, it inherits all the lower bounds known for Maximum Coverage. Furthermore, we can mimic the algorithm for Maximum Coverage (Partial Set Cover) parameterized by t (where you want to cover at least t elements with k sets) [5] to obtain an algorithm for Partition Maximum Coverage parameterized by $\sum_{j\in[r]} t_j$. However, the problem is NP-hard even when $t_j \leq 1$, $j \in [r]$, via a simple reduction from Set Cover.

F-MaxCov has been studied under multiple names in the approximation algorithms literature; however much of the focus has been on approximating the *size* of the solution, rather than the coverage. Notable exception include Chekuri et al. [7] who gave a "bicriteria" approximation, that outputs a solution of size at most $\mathcal{O}(\log r/\epsilon)$ times the optimal size, and covers at least $(1 - 1/e - \epsilon)$ fraction of the required coverage of each color. Very recently, Bandyapadhyay et al. [3] recently designed an FPT-AS for F-MaxCov for the set systems of frequency 2, running in time $2^{\mathcal{O}(\frac{rk^2 \log k}{\epsilon})} \cdot (m+n)^{\mathcal{O}(1)}$. We obtain the following result on F-MaxCov.

▶ **Theorem 1.4.** *There exists a randomized* FPT-AS *for F-MaxCov running in time* $\left(dr\left(\frac{\log k}{\epsilon}\right)^r\right)^{\mathcal{O}(k)} \cdot (m+n)^{\mathcal{O}(1)}$*, on set systems with frequency bounded by* d.

Note that this result generalizes the result of [3] to frequency-d set systems, and in the case of $d = 2$, our running time is faster than that of [3] (albeit our algorithm is randomized).

One can also define *fair* version of CC-MaxSat in an analogous way, which we call F-MaxSAT. In this problem, we are given a CNF-formula Φ, a coloring function $\chi : \mathsf{cla}(\Phi) \rightarrow [r]$, a coverage demand function $t : [r] \rightarrow \mathbb{N}$, and an integer k. The goal is to find a weight-k assignment that satisfies at least $t(j)$ (also denoted as t_j) clauses of each color $j \in [r]$. By combining Theorem 1.4 with a slightly more general version of the reduction theorem (Theorem 1.1) also yields FPT-AS for F-MaxSAT with a similar running time.

1.2 Our New Problem: Combining Matroid and Fairness Constraints

As discussed in the previous subsections, Maximum Coverage has been generalized in two orthogonal directions, namely, matroid constraints on the sets chosen in the solution, and fairness constraints on the elements covered by the solution. Although the corresponding variants of CC-MaxSat have not been studied in the literature, we mentioned that our techniques readily imply FPT-ASes for these problems for many "sparse" formulas. Given this, the following natural question arises.

> Can we find good approximations for the variants of CC-MaxSat (resp. Maximum Coverage) that combines the two orthogonal generalizations, namely, matroid constraint on the variables assigned 1, and fairness constraints on the satisfied clauses (resp. matroid constraint on the sets chosen in the solution, and fairness constraints on the elements covered)?

In the following, we formally define the common generalization of M-MaxSAT and F-MaxSAT, which we call (M, F)-MaxSAT.

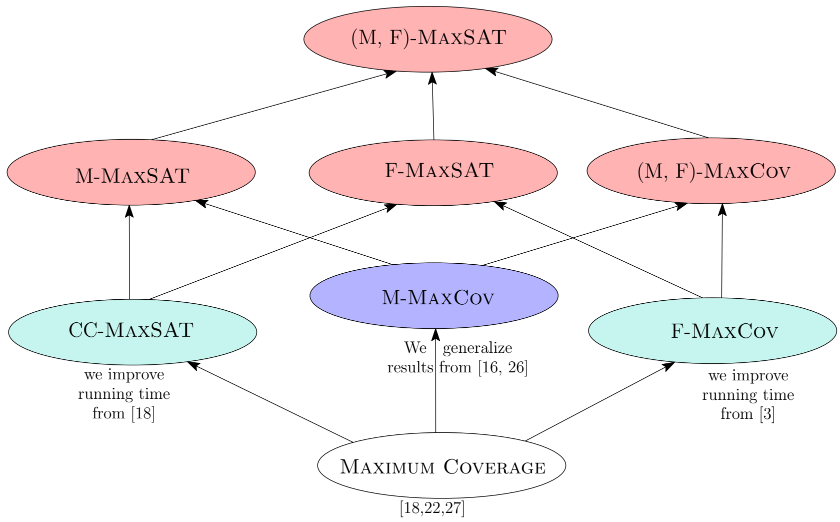

Figure 1 If there is an arrow of the form $A \to B$, then problem B generalizes problem A. FPT-ASes for the problems in red bubbles are not known in the literature, and we study in this paper. For all the other problems FPT-ASes are known in the literature for some cases. This paper improves the results in cyan and blue.

(M, F)-MaxSAT
Input. A CNF-SAT formula Φ where the clauses $\mathsf{cla}(\Phi)$ of Φ are partitioned into r colors. Each color $j \in [r]$ has an associated demand t_j. Additionally, we are provided the independence oracle to a matroid $\mathcal{M} = (\mathsf{var}(\Phi), I)$ of rank k.
Question. Does there exist an assignment $\Psi : \mathsf{var}(\Phi) \to \{0,1\}$, such that
- The number of clauses satisfied by Ψ of color j is at least t_j, for each $j \in [r]$,
- The set of variables assigned 1 must be independent in $\mathcal{M}$, i.e., $\Psi^{-1}(1) \in I$.

In the special case where the CNF-SAT formula is monotone (i.e., does not contain negated literals), we obtain (M, F)-MaxCov, which generalizes all the variants of Maximum Coverage discussed earlier. We obtain the following result for (M, F)-MaxCov.

▶ **Theorem 1.5.** *There exists a randomized* FPT-AS *for (M, F)-MaxCov on set systems with maximum frequency d, that runs in time* $\left(\frac{d \log k}{\epsilon}\right)^{\mathcal{O}(kr)} \cdot (m+n)^{\mathcal{O}(1)}$ *and returns a* $(1-\epsilon)$*-approximation with at least a constant probability.*

Finally, by reducing (M, F)-MaxSAT on d-CNF formulas to (M, F)-MaxCov with frequency d set systems, using the randomized reduction, and then using the results of Theorem 1.5, we obtain our most general result, as follows.

▶ **Theorem 1.6.** *There exists a randomized* FPT-AS *for (M, F)-MaxSAT on d-CNF formulas, that runs in time* $\left(\frac{d \log k}{\epsilon}\right)^{\mathcal{O}(kr)} \cdot (m+n)^{\mathcal{O}(1)}$ *and returns a* $(1-\epsilon)$*-approximation with at least a constant probability.*

We give a summary of how the various problems are related to each other, and a comparison of our results with the literature in Figure 1.

1.3 Related Results

Max k-VC or Partial Vertex Cover has been extensively studied in Parameterized Complexity. In this problem we are given a graph and the task is to select a subset of k vertices covering as many of the edges as possible. The problem is known to be approximable within 0.929 and is hard to approximate within 0.944, assuming UGC [20]. Max k-VC is known to be W[1]-hard [14], parameterized by k, but admits FPT algorithms on planar graphs, graphs of bounded degeneracy, $K_{d,d}$-free graphs, and bipartite graphs, parameterized by k [1, 13, 18]. Indeed, it is among the first problems to admit FPT-AS [23, 20, 25]. It is also known to have "lossy kernels" [23, 19], a lossy version of classical kernelization.

Bera et al. [4] considered the special case of Partition Vertex Cover, where the set of edges of a graph are divided into r colors, and we want to find a subset of vertices that covers at least a certain number of edges from each color class. For this problem, they gave a polynomial-time $\mathcal{O}(\log r)$-approximation algorithm. Hung and Kao [16] generalized this to F-MaxCov, and gave a $\mathcal{O}(d \log r)$-approximation, where each element of the universe is contained in at most d sets (i.e., d is the maximum *frequency*). Bandyapadhyay et al. [2] studied this problem under the name of Fair Covering, and designed a $\mathcal{O}(d)$-approximation, but their running time is XP in the number of colors. Chekuri et al. [7] designed a general framework for F-MaxCov, yielding tight approximation guarantees for a variety of set systems satisfying certain property; in particular, they improve the approximation guarantee for frequency-d set systems to $\mathcal{O}(d + \log r)$, which is tight in polynomial time.

2 Overview of Our Results and Techniques

2.1 Reduction from CC-MaxSat to Maximum Coverage: An overview of Theorem 1.1

This theorem is essentially a randomized *approximation-preserving reduction* from CC-MaxSat to Maximum Coverage. Given an instance $\mathcal{I} = (\Phi, k)$ of CC-MaxSat, we first compute a random assignment Ψ that assigns a variable independently to be 1 with probability $p = \epsilon/2$ and 0 with probability $1 - p$. Let V^* be the set of at most k variables set to be 1 by an optimal assignment Ψ^*. It is straightforward to see that, the probability that all the variables in V^* are set to be 1 by the random assignment Ψ is p^k – we say that this is the good event $\mathcal{G}$. Now, consider a clause that is satisfied negatively by Ψ^*, i.e., a clause C that contains a negative literal $\neg x$ and $\Psi^*(x) = 0$. It is also easy to see that, conditioned on the good event $\mathcal{G}$, the probability that such a clause C is also satisfied negatively by Ψ is at least $1 - p$. Thus, the expected number of clauses that are satisfied negatively by Ψ, conditioned on $\mathcal{G}$, is at least $1 - p$ times the number of clauses satisfied negatively by Ψ^*. Markov's inequality implies that, with probability at least 1/2, the actual number of such clauses is close to its expected value. Thus, conditioned on $\mathcal{G}$, and the previous event, we can focus on the positively satisfied clauses (note that the probability that both of these events occur is at least $1/2 \cdot (\epsilon/2)^k$. To this end, we can eliminate all the negatively satisfied clauses, and we can also prune the remaining clauses by eliminating any negative literals and the variables that are set to 0 by Ψ. Thus, all the remaining clauses only contain positive literals, which can be seen as an instance $\mathcal{I}'$ of Maximum Coverage. Furthermore, conditioned on $\mathcal{G}$, the variables set to 1 by Φ^* correspond to a family $\mathcal{F}^*$ of size k, and the elements covered by $\mathcal{F}^*$ correspond to the set of clauses satisfied only positively by Φ^*. Thus, if we find a $(1 - \epsilon)$-approximate solution to $\mathcal{I}'$, and set the corresponding variables to 1, and the rest of the variables to 0, then we get a weight-k assignment that satisfies at least $(1 - \epsilon) \cdot \mathsf{OPT}_{sat}$ clauses. Note that this reduction, combined with the algorithm of [17] gives the proof of Corollary 1.2.

Furthermore, this reduction is robust enough that it can accommodate the fairness constraints on the clauses, as defined above. To be precise, one can give a similar reduction from $\mathcal{C}$-MaxSAT to $\mathcal{C}$-MaxCov, where $\mathcal{C} \in \{\text{M, F, (M, F)}\}$ – note that when we have fairness constraints, the success probability now becomes $(r/\epsilon)^{\mathcal{O}(k)}$. Essentially, these reductions translate a constraint on the variables set to 1 (for CC-MaxSat and variants), to the corresponding family of sets (for Maximum Coverage and variants). Thus, at the expense of a multiplicative $(r/\epsilon)^{\mathcal{O}(k)}$ factor in the running time, we can focus on Maximum Coverage and its variants, which is what we do in this section, as well as in the paper. As a warm-up, we start in Section 2.2 with the vanilla Maximum Coverage on frequency-d set systems (where our algorithms *do not* improve over the known algorithms in the literature), and give a complete formal proof. Then, we will gradually introduce the ideas required to handle fairness (Section 2.3) and matroid (Section 2.4) constraints – first separately, and then together. Finally, in Section 2.5, we briefly discuss the ideas required to these results to $K_{d,d}$-free set systems and multiple matroid constraints.

2.2 Deterministic and Randomized Branching using a Largest Set

To introduce our ideas in a clean and gradual way, we start with the simplest setting of Maximum Coverage where the maximum frequency of the elements is bounded by d. Recall that we are given an instance $(U, \mathcal{F}, k)$ and the goal is to find a sub-family of $\mathcal{F}$ of size k that covers the maximum number of elements. For any sub-family $\mathcal{R} \subseteq \mathcal{F}$, let $U(\mathcal{R})$ denote the subset of elements covered by $\mathcal{R}$, and $\mathsf{OPT}_k(\mathcal{R})$ denote the maximum number of elements that can be covered by a subset of $\mathcal{R}$ of size k. Further, for a set $S \in \mathcal{F}$, we denote by $\mathcal{F} - S$, the family obtained by removing S, as well as the elements of S from each of the remaining sets. Our approach is inspired by the approaches of Skowron and Faliszewski [25] and Manurangsi [20] who show that $\mathcal{O}(kd/\epsilon)$ sets of the largest size is guaranteed to contain a $(1-\epsilon)$-approximate solution. This naturally begs the question, "*why not start by adding the largest set into the solution?*" (in a sense, the following presentation is closer in spirit to Jain et al. [17].) Let us inspect this question more closely. Let L be a largest set in $\mathcal{F}$. By looking at the contribution of coverage of each set in an optimal solution, say $\mathcal{O}$, we can easily see that $|L| \geq \frac{\mathsf{OPT}_k(\mathcal{F})}{k}$. We say that a set $S \in \mathcal{F}$ is *heavy* w.r.t. L if $|L \cap S| \geq \frac{\epsilon|L|}{k}$ (note that L is heavy w.r.t. itself). However, since the frequency of each element is bounded by d, each element in L can appear in at most d (in fact, $d-1$) sets $L \cap S$ for different $R \in \mathcal{F}$. This implies that at most $\frac{kd}{\epsilon}$ sets in $\mathcal{F}$ are heavy w.r.t. L.

Algorithm. Our algorithm simply branches on the sets in $\mathcal{H}(L)$, which is the family of heavy sets w.r.t. L. Specifically, in the branch corresponding to a heavy set $S \in \mathcal{H}(L)$, we include it in the solution, and recursively call the algorithm on the residual instance $(U \setminus S, \mathcal{F} - S, k-1)$. If any of the sets in $\mathcal{O}$ is heavy w.r.t. L, then in the branch corresponding to such a set yields an approximate solution via induction. The main idea is that, if no set in $\mathcal{O}$ is heavy w.r.t. L, then the branch corresponding to L yields a good solution. This is justified as the effect of any of the $k-1$ sets in $\mathcal{O}$ is too small. For the sake of clarity, we formally analyze this algorithm below via induction.

We want to show that, for a given input $(U', \mathcal{F}', k')$ the recursive algorithm returns a family $\mathcal{R} \subseteq \mathcal{F}'$ of size k' such that $|U'(\mathcal{R})| \geq (1-\epsilon) \cdot \mathsf{OPT}_{k'}(\mathcal{F}')$. The base case for $k' = 0$ is trivial, since the algorithm returns an empty set. Suppose that the claim is true for all inputs with budget $k-1$, and we want to prove it for $(U, \mathcal{F}, k)$. Let $\mathcal{O}$ denote an optimal solution of size k with $|\mathsf{OPT}_k(\mathcal{F})| = |U(\mathcal{O})|$.

Approximation Ratio in the Easy Case. If $\mathcal{O} \cap \mathcal{H}(L) \neq \emptyset$, then there exists a branch corresponding to a set $S \in \mathcal{O} \cap \mathcal{H}(L)$. This is the *easy case* (for the analysis). In this case, $\mathsf{OPT}_{k-1}(\mathcal{F} - S) = \mathsf{OPT}_k(\mathcal{F}) - |S|$, and hence the approximation ratio is:

$$\begin{aligned}\frac{|S| + (1-\epsilon)\mathsf{OPT}_{k-1}(\mathcal{F} \setminus S)}{\mathsf{OPT}_k(\mathcal{F})} &= \frac{|S| + (1-\epsilon)(\mathsf{OPT}_k(\mathcal{F}) - |S|)}{\mathsf{OPT}_k(\mathcal{F})} \\ &\geq \frac{(1-\epsilon)(\mathsf{OPT}_k(\mathcal{F})}{\mathsf{OPT}_k(\mathcal{F})} = (1-\epsilon)\end{aligned}$$

Approximation Ratio in the Hard Case. The more hard case (for analysis) is when $\mathcal{O} \cap \mathcal{H}(S) = \emptyset$. In this case, we argue that the branch that includes the element L is good enough. As in the easy case, we first lower bound the value of $\mathsf{OPT}_{k-1}(\mathcal{F} - L)$. By counting the unique contributions to the solution, there exists a *light* set $S_l \in \mathcal{O}$ such that for $\mathcal{O}' = \mathcal{O} \setminus \{S_l\}$, it holds $|U'(\mathcal{O}')| \geq \frac{k-1}{k} \cdot \mathsf{OPT}_k(\mathcal{F})$. Because no set in $\mathcal{O}$ is heavy w.r.t. L, it follows that for each $R \in \mathcal{O}'$, it holds that $|R \cap L| < \frac{\epsilon|L|}{k}$, and therefore by counting it holds that $|U(\mathcal{O}') \cap L| < \epsilon \cdot |L|$. Therefore,

$$\mathsf{OPT}_{k-1}(\mathcal{F} \setminus L) \geq |U'(\mathcal{O}') \setminus L| \geq |U(\mathcal{O}')| - |U(\mathcal{O}') \cap L| \geq \frac{k-1}{k} \cdot \mathsf{OPT}_k(\mathcal{F}) - \epsilon \cdot |L|.$$

Therefore, the approximation ratio of the branch that includes L is as follows.

$$\begin{aligned}\frac{|L| + (1-\epsilon)\mathsf{OPT}_{k-1}(\mathcal{F} - L)}{\mathsf{OPT}_k(\mathcal{F})} &= \frac{|L| + (1-\epsilon)\left(\frac{k-1}{k} \cdot \mathsf{OPT}_k(\mathcal{F}) - \epsilon \cdot |L|\right)}{\mathsf{OPT}_k(\mathcal{F})} \\ &\geq \frac{|L| + (1-\epsilon)\left(\frac{k-1}{k} \cdot \mathsf{OPT}_k(\mathcal{F})\right) - \epsilon|L|}{\mathsf{OPT}_k(\mathcal{F})} \\ &= \frac{(1-\epsilon)|L| + (1-\epsilon)\left(\frac{k-1}{k} \cdot \mathsf{OPT}_k(\mathcal{F})\right)}{\mathsf{OPT}_k(\mathcal{F})} \\ &\geq \frac{(1-\epsilon)(\mathsf{OPT}_k(\mathcal{F})}{\mathsf{OPT}_k(\mathcal{F})} = (1-\epsilon)\end{aligned}$$

The second last inequality holds from the fact that $|L| \geq \frac{\mathsf{OPT}_k(\mathcal{F})}{k}$.

This leads to a deterministic $(1-\epsilon)$-approximation algorithm with running time $((\frac{kd}{\epsilon})^k) \cdot (n+m)^{\mathcal{O}(1)}$.

Insight into the probabilistic branching

A closer inspection of the analysis reveals that the reason L may not a good choice is that the sets of $\mathcal{O}$ *together* cover more than a certain threshold fraction of elements covered by L. We utilize this idea through a smoothening process that captures the effect of the size of the intersection of a set S with L in a more nuanced manner. Let us define the weight $h_L(S)$ of a set $S \in \mathcal{F} \setminus \{L\}$ as $h_L(S) = \frac{|S \cap L|}{|L|}$. Our algorithm now instead does "randomized branching", i.e., it samples one set to be included in the solution according to some probability distribution, and then continues recursively. Note that the single run of the algorithm finishes in polynomial time. The probability distribution used by the algorithm is as follows: the set L is sampled with probability $1/2$, and any other set $S \in \mathcal{F} \setminus \{L\}$ is sampled with probability proportional to its weight $h(S)$ (the constant of proportionality is chosen such that this is a valid probability distribution, that is, the probabilities sum up to 1). In particular, observe that $\sum_{S \in \mathcal{F} \setminus \{L\}} h_L(S) \leq \frac{d|L|}{|L|} = d$. Thus, the probability of selecting S is at least $\frac{h_L(S)}{2d}$. Note that due to the way the weights $h_L(S)$ are defined, the sets with a large intersection with L have a greater chance of being sampled, as compared to the sets with a small intersection with L.

We will show that the algorithm returns a $(1-\epsilon)$-approximate solution with probability at least $(\frac{\epsilon}{2d})^k$, and runs in polynomial time. This implies that by repeating the algorithm $(\frac{2d}{\epsilon})^k$ times, we obtain a $(1-\epsilon)$-approximation with probability at least a positive constant. This leads to a randomized algorithm with running time $\mathcal{O}^*((\frac{2d}{\epsilon})^k)$.

The proof is again by induction. We want to show that, for any input $(U', \mathcal{F}', k')$, the algorithm returns a solution $\mathcal{R} \subseteq \mathcal{F}'$ of size k' such that $|U'(\mathcal{R})| \geq (1-\epsilon) \cdot \mathsf{OPT}_{k'}(\mathcal{F}')$, with probability at least $(\frac{\epsilon}{2d})^{k'}$. We reuse much of the notation from the previous analysis. Let $\mathcal{O}$ be an optimal solution of size k. First, the case when $L \in \mathcal{O}$, since our algorithm samples and includes L in the solution with probability $1/2$. Then, conditioned on this event (that is, L being sampled), the approximation ratio analysis proceeds similarly to the *easy case* of the previous analysis. By induction, the recursive algorithm returns a $(1-\epsilon)$-approximate solution with probability at least $(\frac{\epsilon}{2d})^{k-1}$. Thus, overall, the algorithm returns a $(1-\epsilon)$-approximation with probability at least $\frac{1}{2}(\frac{\epsilon}{2d})^{k-1} \geq (\frac{\epsilon}{2d})^k$.

Now suppose $L \notin \mathcal{O}$. As before, let $S_l \in \mathcal{O}$ be a light set as defined earlier, and $\mathcal{O}' = \mathcal{O} \backslash \{S_l\}$. We analyze by considering the following two cases: either (i) $|U(\mathcal{O}') \cap L| \leq \epsilon \cdot |L|$, or (ii) $|U(\mathcal{O}') \cap L| > \epsilon \cdot |L|$.

In case (i), we are effectively in the same situation as the *hard case* of the previous analysis – as before, the algorithm samples L with probability at least $1/2$, and as argued in the *hard case*, conditioned on the previous event, the branch corresponding to L returns a $(1-\epsilon)$-approximate solution, but now with probability at least $(\frac{\epsilon}{2d})^{k-1}$ by induction. Therefore, we obtain an $(1-\epsilon)$-approximate solution with probability at least $\frac{1}{2}(\frac{\epsilon}{2d})^{k-1} \geq (\frac{\epsilon}{2d})^k$.

In case (ii), we have that $|U(\mathcal{O}') \cap L| > \epsilon \cdot |L|$. Notice that,

$$\sum_{S \in \mathcal{O}'} |S \cap L| > \epsilon.$$

This implies that

$$\sum_{S \in \mathcal{O}'} h_L(S) = \sum_{S \in \mathcal{O}'} \frac{|S \cap L|}{|L|} \geq |U(\mathcal{O}') \cap L| > \epsilon |L|.$$

Therefore, the total weight of the sets in $\mathcal{O}'$ is at least ϵ. Therefore, the probability that the algorithm samples a set from $\mathcal{O}'$ is at least $\frac{\epsilon}{2d}$. Conditioned on this event, the approximation ratio analysis now proceeds as in the *easy case*, and the algorithm returns a $(1-\epsilon)$-approximate solution with probability at least $\frac{\epsilon}{2d}(\frac{\epsilon}{2d})^{k-1} = (\frac{\epsilon}{2d})^k$.

2.3 Handling fairness constraints via the Bucketing trick

The aforementioned idea of prioritizing the largest set L, or sets that are heavy w.r.t. it, fails to generalize when we have multiple coverage constraints in F-MaxCov. This is simply because there is no notion of "the largest set" even when we want to cover elements of *two* different colors, each with different coverage requirements. To handle such multiple coverage constraints, our idea is to use multidimensional-knapsack-style bucketing technique to group the sets of $\mathcal{F}$ into *approximate equivalence classes*, called *bags* for short. At a high level, all the vertices belonging to a particular bag contain *approximately equal* (i.e., within a factor of $(1+\epsilon)$) number of elements of *all* of the r colors. Thus, in isolation, any two sets L_1 and L_2 belonging to the same bag are interchangeable, since we tolerate an ϵ-factor loss in the coverage. Since the total number of bags can be shown to be $\left(\frac{\log k}{\epsilon}\right)^{\mathcal{O}(rk)}$, and hence we can "guess" a bag $\mathcal{B}$ containing a set from a solution $\mathcal{O}$. However, due to different amount of overlap with an optimal solution $\mathcal{O}$, two sets $L_1, L_2 \in \mathcal{B}$ may not be interchangeable w.r.t. $\mathcal{O}$. That is, if $L_1 \in \mathcal{O}$, then $\mathcal{O} \setminus \{L_1\} \cup \{L_2\}$ may not be a good solution. However, assuming that we have correctly guessed a bag that intersects with $\mathcal{O}$, we can then select a set $L \in \mathcal{B}$, and define the heavy sets (for deterministic algorithm) or weights $h_L(\cdot)$ (for the randomized algorithm) w.r.t. L. Note that, since we have multiple coverage constraints, we cannot simply look at the total size of the intersection $|S \cap L|$. Instead, we need to tweak the notion of *heaviness* that takes into account the number of elements of each color in the intersection $S \cap L$. To summarize, we need two additional ideas to handle multiple colors in F-MaxCov: (1) "guessing" over buckets, and (2) a suitable generalization of the notion of heaviness. Modulo this, the rest of the analysis is again similar to the *easy* and *hard* cases as before. Using these ideas, we can prove the first part of Theorem 1.4.

2.4 Handling Matroid Constraints

First, we consider M-MaxCov (note that this is an orthogonal generalization of Maximum Coverage, without multiple coverage constraints), where the solution is required to be an independent set in the given matroid $\mathcal{M} = (\mathcal{F}, I)$ of rank k. We assume that we are given an *oracle access* to $\mathcal{M}$ in the form of an algorithm that answers the queries of the form "Is $\mathcal{R}$ an independent set?" for any subset $\mathcal{R} \subseteq \mathcal{F}$. Let us revisit the initial deterministic FPT-AS for Maximum Coverage and try to generalize it to M-MaxCov. Recall that this algorithm branches on each set $S \in \mathcal{H}(L)$, where L is a largest set. The analysis of easy case goes through even in presence of the matroid constraint, since we branch on a set from an optimal solution $\mathcal{O}$. However, in the hard case, our analysis replaces a $S_l \in \mathcal{O}$ with L, and argues that the branch corresponding to L returns a $(1-\epsilon)$-approximate solution. However, this does not work for M-MaxCov, since $(\mathcal{O} \setminus \{S_l\}) \cup \{L\}$ may not be an independent set in $\mathcal{M}$. To summarize, although L handles the coverage constraints (approximately), it may fail to handle the matroid constraint. In fact, it may just so happen that L is not a good set *at all*, in the sense that, for *any* set $S \in \mathcal{O}$, $(\mathcal{O} \setminus \{S\}) \cup \{L\}$ is not independent in $\mathcal{M}$, which is crucial for the induction to go through.

To solve this issue, we resort to the bucketing idea as in the fair coverage case (thus, the subsequent arguments generalize to (M, F)-MaxCov in a straightforward manner; although let us stick to the special case of M-MaxCov for now). Indeed, branching w.r.t. the largest set L is an overkill – it suffices to pin down a bag $\mathcal{B}$ containing a set in $\mathcal{O}$ (it does not even have to be the largest set), by "guessing" from $\mathcal{O}\left(\frac{\log k}{\epsilon}\right)$ bags. However, we again cannot select an arbitrary set $S \in \mathcal{B}$ and define heavy sets w.r.t. S, precisely due to the matroid compatibility issues mentioned earlier. Therefore, we resort to the idea of *representative sets* from matroid theory [22] [6] Assuming our guess for $\mathcal{B}$ is correct, there exists some $S \in \mathcal{B} \cap \mathcal{O}$. However, we cannot further "guess" S, since the size of the bag may be too large. Instead, we compute a inclusion-wise maximal independent set $\mathcal{B}' \subseteq \mathcal{B}$. Note that the size of $\mathcal{B}'$ is at most k, and it can be computed using polynomially many queries to the independence oracle. However, it may very well happen that $S \notin \mathcal{B}'$. Nevertheless, using matroid properties, we can argue that, there exists a set $S' \in \mathcal{B}'$, such that $(\mathcal{O} \setminus \{S\}) \cup \{S'\}$ is an independent set. Thus, $\mathcal{B}'$ is a *representative set* of $\mathcal{B}$. Furthermore, since both S and S' come from the same bag, they cover approximately the same number of elements. Thus, our modified deterministic

[6] Although this is a powerful hammer in its full generality – which we do use to handle multiple matroid constraints – our specialized setting lets us use much simpler arguments to handle single matroid constraint in M-MaxCov/(M, F)-MaxCov.

algorithm works as follows. First, it computes maximal independent set $\mathcal{B}' \subseteq \mathcal{B}$, and for each $S' \in \mathcal{B}'$, it computes the heavy family $\mathcal{H}(S')$. Then, it branches over all sets in $\bigcup_{S' \in \mathcal{B}'} \mathcal{H}(S')$. If one of the branches corresponds to branching on a set from $\mathcal{O}$, then the analysis is similar to the easy case. Otherwise, we know that $(\mathcal{O} \setminus \{S\}) \cup \{S'\}$ is an $(1-\epsilon)$-approximate solution that is also an independent set. Therefore, the branch corresponding to S' yields the required $(1-\epsilon)$-approximation. We can improve the running time via doing a randomized branching in two steps: first we pick a set $S'' \in \mathcal{B}'$ uniformly at random, and then we perform the probabilistic branching using the weights $h_{S''}(\cdot)$.

Both deterministic and randomized variants incur a further multiplicative overhead of $(\frac{k \log k}{\epsilon})^k$ due to first guessing a bag, and then computing a representative set $\mathcal{B}' \subseteq \mathcal{B}$ of the bag, and thus do not improve over the results of Sellier [24] in terms of running time for M-MaxCov. However, this idea naturally generalizes to (M, F)-MaxCov, with the appropriate modifications in bucketing (as mentioned in the previous paragraph) to handle the multiple coverage requirements of different colors. This leads to the proof of Theorem 1.5.

2.5 Further Extensions

The ideas mentioned in the previous subsections can be extended to even more general settings in a couple of ways. First, we describe how to extend the ideas from frequency-d set systems for Maximum Coverage to $K_{d,d}$-free set systems, i.e., set system $(U, \mathcal{F})$, where no d sets in $\mathcal{F}$ contain d elements of U in common. Note that frequency-d set systems are $K_{d+1,d+1}$-free. Then, we describe how the linear algebraic toolkit of *representative sets* can be used to handle multiple (linear) matroid constraints.

$K_{d,d}$-free Set Systems

Next, we consider $K_{d,d}$-free set systems $(U, \mathcal{F})$, where no d sets of $\mathcal{F}$ contain d common elements of U. To design the FPT-AS on $K_{d,d}$-free set systems, we combine the bucketing idea along with the combinatorial properties of $K_{d,d}$-free graphs to bound the number of heavy neighbors of a set. To this end, however, we need to modify the precise definition of *heaviness* (as in Jain et al. [17]). This leads to a somewhat cumbersome branching algorithm that handles colors differently based on their coverage requirement. For colors with small coverage requirement, we highlight the covered vertices using the standard technique of *label coding*[7]. Now, vertices in a bag cover the elements with the same label and for colors with high coverage requirements, the sizes of the sets in the same bag are "almost" the same. Then, we pick an *arbitrary* set L from the bag $\mathcal{B}$, and we branch on (suitably defined) *heavy* sets w.r.t. S. Since the number of heavy sets is bounded by a function of k, d, and ϵ, this leads to a deterministic version of Theorem 1.4 to $K_{d,d}$-free set systems. Note that since frequency-d set systems is a special case of this, this implies a deterministic FPT-AS in this case; however with a much worse running time compared to Theorem 1.4. This leads to the proof of Theorem 1.5.

[7] The technique is better known as *color coding*. However, this creates an unfortunate clash of terminology – these *colors* have nothing to do with the original colors corresponding to coverage constraints. Bandyapadhyay et al. [3] instead use the term "label coding", and we also adopt the same terminology

Handling Multiple Matroid Constraints

Our results on M-MaxCov and (M, F)-MaxCov can be generalized to handle multiple matroid constraints on the solution, in the case when the matroids are *linear* or representable[8]. In this more general problem, we are given q linear matroids $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_q$, where $\mathcal{M}_i = (\mathcal{F}, I_i)$, each of rank at most k, and the solution S is required to be independent in all q matroids, i.e., $S \in \bigcap_{i\in[q]} I_i$. In this case, we can use linear algebraic tools ([22, 12]) to compute a representative set of size qk instead of k, and the computation requires $2^{\mathcal{O}(qk)} \cdot n^{\mathcal{O}(1)}$ time. Thus, FPT-ASes for this problem now have a factor of q in the exponent.

Note that our FPT-AS improves upon the polynomial-time approximation guarantee of $1 - 1/e$ of Calinescu et al. [6] for monotone submodular maximization subject to a matroid constraint, in the special case of $K_{d,d}$-free coverage functions. To the best of our knowledge, this is the largest class of monotone submodular functions and matroid constraints for which the lower bound of $1 - 1/e$ can be overcome, even in FPT time. Further, the analogous results to (M, F)-MaxSAT generalize these results to maximization of non-monotone/non-submodular functions.

3 Preliminaries

For a positive integer q, let $[q] := \{1, 2, \ldots, q\}$.

Convention. We consistently use index j to refer to a color from the range $[r]$, and may often write "for a color j" instead of "for a color $j \in [r]$". Finally, for a coverage requirement function t (resp. variations such as $t', \widetilde{t}$), and a color j, we shorten $t(j)$ to t_j (resp. $t'_j, \widetilde{t}_j$).

4 Reduction from (M, F)-MaxSAT to (M, F)-MaxCov

In this section, we begin with a polynomial time approximate preserving randomized reduction from F-MaxSAT to F-MaxCov. The success probability of the reduction is $\mathcal{O}((\epsilon/r)^k)$. Recall that in the F-MaxSAT problem, given a CNF-formula Φ with $\chi : \mathsf{cla}(\Phi) \to [r]$, a coverage demand function $t : [r] \to \mathbb{N}$ and an integer k, the goal is to find an assignment of weight at most k that satisfies at least $t(i)$ (also denoted as t_i) clauses of color class i (an assignment Ψ satisfying these properties is called *optimal weight k assignment*).

We begin with some basic definitions. Let Φ be a CNF-formula. By $\mathsf{var}(\Phi)$ and $\mathsf{cla}(\Phi)$, we denote the set of variables and clauses in the formula Φ, respectively. An assignment to a CNF-formula Φ is a function $\Psi : \mathsf{var}(\Phi) \to \{0, 1\}$. The weight of an assignment is the number of variables that have been assigned 1. By $T(\Psi)$ and $F(\Psi)$, we denote the set of variables assigned 1 and 0 by the assignment Ψ, respectively. For a clause $c \in \mathsf{cla}(\Phi)$, $\mathsf{var}(c)$ is the set of variables that occur in the clause c as a positive or negative literal. Similarly, for a set of clauses $C \in \mathsf{cla}(\Phi)$, $\mathsf{var}(C)$ is the set of variables that occur as a positive or negative literal in a clause $c \in C$.

Our reduction (Algorithm 1) takes an instance (Φ, χ, t, k) of F-MaxSAT as input. It constructs a random assignment Ψ by setting each variable to 1 with probability p and 0 with probability $1 - p$. It constructs a new formula by first removing the set of clauses that are

[8] A matroid $\mathcal{M} = (E, \mathcal{I})$ is representable over a field $\mathbb{F}$ if there exists a matrix M such that there exists a bijection between E and the columns on M with the property that, a subset $E' \subseteq E$ is independent in $\mathcal{M}$ iff the corresponding set of columns are linearly independent over $\mathbb{F}$.

Algorithm 1 Reduction Algorithm($\mathcal{I} = (\Phi, \chi, t, k)$ of F-MaxSAT).

1: Construct a random assignment Ψ as follows. For each variable $x \in \mathsf{var}(\Phi)$, independently set $\Psi(x)$ to 1 with probability p and 0 with probability $1-p$. ▷ We will later set $p = \frac{\epsilon}{2r}$.
2: Construct a new formula Φ' as follows:
   - Let $N \subseteq \mathsf{cla}(\Phi)$ be the set of clauses that are satisfied negatively by Ψ. Then, $\mathsf{cla}(\Phi') = \mathsf{cla}(\Phi) \setminus N$.
   - For each $c \in \mathsf{cla}(\Phi')$, remove all the variables in c that occur either as a negative literal or set to 0 by Ψ.
   - For each $c \in \mathsf{cla}(\Phi')$, add $\mathsf{var}(c)$ to $\mathsf{var}(\Phi')$.
3: Construct an instance $\mathcal{J}_\Psi = (\mathcal{U}, \mathcal{F}, \chi', t', k')$ of F-MaxCov as follows:
   - Set $\mathcal{U} = \mathsf{cla}(\Phi')$.
   - For each $v \in \mathsf{var}(\Phi')$, add a set f_v to $\mathcal{F}$ where $f_v = \{c \in \mathsf{cla}(\Phi') : v \in \mathsf{var}(c)\}$.
   - For each $c \in \mathsf{cla}(\Phi')$, if the corresponding element in $\mathcal{U}$ is e_c, set $\chi'(e_c) = \chi(c)$.
   - Set $t'(i) = t(i) - \frac{|N \cap \chi^{-1}(i)|}{1-\epsilon}$ for each $i \in [r]$.
   - Set $k' = k$.

satisfied negatively by Ψ, followed by removing negative literals from the remaining clauses. It reduces the formula to an instance $(\mathcal{U}, \mathcal{F}, \chi', t', k')$ of F-MaxCov as described in Step 3 of Algorithm 1. Clearly, this reduction takes polynomial time.

Next, we prove the correctness of our reduction. For a Yes-instance $\mathcal{I}$ of F-MaxSAT, let $\Psi^\star$ be an optimal weight k assignment. Let $N^\star$ be the set of clauses satisfied negatively by $\Psi^\star$, i.e., every clause in $N^\star$ contains a negative literal that is set to 1, and let $P^\star$ be the set of clauses satisfied only positively by $\Psi^\star$, i.e., every clause in $P^\star$ contains a positive literal that is set to 1 and no negative literal in this clause is set to 1. By $N_i^\star$, we mean the set of clauses in color class i satisfied negatively by $\Psi^\star$ and by $P_i^\star$, we mean the set of clauses in color class i satisfied only positively by $\Psi^\star$. We call a random assignment, constructed in Algorithm 1, *good* if each variable in $T(\Psi^\star)$ (positive variables under $\Psi^\star$) is assigned 1 by Ψ, i.e., $T(\Psi^\star) \subseteq T(\Psi)$, which occurs with probability at least p^k. For a good assignment Ψ, let N_i denote the set of clauses in color class i satisfied negatively by Ψ and P_i denote the set of clauses in color class i satisfied only positively by Ψ. We say that an event $\mathcal{G}$ is *good* if a good assignment Ψ is generated in Algorithm 1. We begin with the following claim.

▷ Claim 4.1. Given a Yes-instance $\mathcal{I}$ of F-MaxSAT, with probability at least 1/2, a good assignment Ψ satisfies at least $(1-\epsilon)|N_i^\star|$ clauses negatively, for each $i \in [r]$.

Proof. Let (Φ, χ, t, k) be a Yes-instance of F-MaxSAT. Let Ψ be a good assignment, which occurs with probability at least p^k. We show that Ψ satisfies at least $(1-\epsilon)|N^\star|$ clauses negatively, with probability at least 1/2. Let X_i be the number of clauses in $N_i^\star$ that are satisfied negatively by Ψ. We define an indicator random variable x_j, for each $j \in [|N_i^\star|]$, as follows.

$$x_j = \begin{cases} 1 & \text{clause } c_j \in N_i^\star \text{ is satisfied negatively by } \Psi \\ 0 & \text{otherwise} \end{cases}$$

$$\mathsf{Pr}(x_j|\mathcal{G}) = \mathsf{Pr}(\text{clause } c_j \in N_i^\star \text{ is satisfied negatively by } \Psi|\mathcal{G}) \geq (1-p)$$

$$\mathbb{E}[X_i|\mathcal{G}] = \sum_{j \in [|N_i^\star|]} x_j \times \mathsf{Pr}(x_j|\mathcal{G}) \geq (1-p)|N_i^\star|$$

Let $Y_i = |N_i^\star| - |N_i|$, where N_i is the set of clauses satisfied negatively by Ψ. Note that,

$$Y_i = |N_i^\star| - |N_i| \leq |N_i^\star \setminus N_i| = |N_i^\star| - X_i$$

Thus,

$$\mathbb{E}[Y_i|\mathcal{G}] \leq |N_i^\star| - \mathbb{E}[X_i|\mathcal{G}] \leq p|N_i^\star|$$

Since $Y_i \geq 0$, we can use Markov's inequality and get

$$\mathsf{Pr}(Y_i \geq 2rp|N_i^\star||\mathcal{G}) \leq \frac{\mathbb{E}[Y_i]}{2rp|N_i^\star|} \leq \frac{1}{2r}$$

Since $Y_i = |N_i^\star| - |N_i|$, we get

$$\mathsf{Pr}(|N_i| \leq (1-2rp)|N_i^\star||\mathcal{G}) \leq \frac{1}{2r}$$

By union bound,

$$\mathsf{Pr}\left(\exists i \in [r], |N_i| \leq (1-2rp)|N_i^\star||\mathcal{G}\right) \leq \sum_{i\in[r]} \mathsf{Pr}(|N_i| \leq (1-2rp)|N_i^\star||\mathcal{G}) \leq \frac{1}{2}$$

This implies that

$$\mathsf{Pr}\left(\forall i \in [r], |N_i| > (1-2rp)|N_i^\star||\mathcal{G}\right) \geq \frac{1}{2}$$

Setting $p = \frac{\epsilon}{2r}$ gives us the required result, i.e., with probability at least 1/2, for all colors $i \in [r]$, $|N_i| > (1-\epsilon)|N_i^\star|$. ◁

▶ **Lemma 4.2.** *If $\mathcal{I} = (\Phi, \chi, t, k)$ is a yes-instance of F-MaxSAT, then with probability at least $(\frac{\epsilon}{2r})^k$, the reduced instance $\mathcal{J}_\Psi = (\mathcal{U}, \mathcal{F}, \chi', t', k')$ is a yes-instance of F-MaxCov.*

Proof. Let $\mathcal{I}$ be a Yes-instance of F-MaxSAT and let $\Psi^\star$ be an optimal weight k assignment. Further, let $N^\star$ be the set of clauses satisfied negatively by $\Psi^\star$, i.e., every clause in $N^\star$ contains a negative literal that is set to 1, and let $P^\star$ be the set of clauses satisfied only positively by $\Psi^\star$, i.e., every clause in $P^\star$ contains a positive literal that is set to 1 and no negative literal in this clause is set to 1. Then, there exists a set $V_{P^\star} \subseteq \mathsf{var}(P^\star)$ of size at most k that satisfies all the clauses in $P^\star$ positively, i.e., for each clause in $P^\star$, there is a variable in $V_{P^\star}$ that occurs as a positive literal in it and is assigned 1 under $\Psi^\star$. Let Ψ be a good assignment which is generated with probability at least $(\frac{\epsilon}{2r})^k$. Since Ψ is a good assignment, $T(\Psi^\star) \subseteq T(\Psi)$ and $F(\Psi) \subseteq F(\Psi^\star)$. Hence, $P^\star \subseteq P$. Thus, $P^\star \subseteq \mathcal{U}$ and for each variable in $\mathsf{var}(P^\star)$, we have a set in the family $\mathcal{F}$. Let $Z = \{f_v \in \mathcal{F} : v \in V_{P^\star}\}$. We claim that Z is a solution to $\mathcal{J}_\Psi$. Clearly, Z covers at least $|P_i^\star|$ elements in $\mathcal{U}$, for each $i \in [r]$. We claim that $t_i' \leq |P_i^\star|$. Since $\Psi^\star$ is a solution to $\mathcal{I}$, it satisfies t_i clauses for each $i \in [r]$. Since $t_i = |P_i^\star| + |N_i^\star|$, due to Claim 4.1, we know that $t_i \leq |P_i^\star| + \frac{|N_i|}{1-\epsilon}$. Thus, $|P_i^\star| \geq t_i - \frac{|N_i|}{1-\epsilon}$. Since $t_i' = t_i - \frac{|N_i|}{1-\epsilon}$, $t_i' \leq |P_i^\star|$. This completes the proof. ◀

▶ **Lemma 4.3.** *Assume that $\mathcal{I} = (\Phi, \chi, t, k)$ is a yes-instance of F-MaxSAT. If there exists $(1-\epsilon)$-approximate solution for $\mathcal{J}_\Psi = (\mathcal{U}, \mathcal{F}, \chi', t', k')$, where Ψ is a good assignment, then there exists $(1-\epsilon)$-approximate solution for $\mathcal{I}$ with probability at least $\frac{1}{2}$.*

Proof. Let $\Psi^\star$ be an optimal assignment. Due to Claim 4.1, Ψ satisfies at least $(1-\epsilon)|N_i^\star|$ clauses negatively, for each $i \in [r]$, with probability at least $1/2$. Let S be a $(1-\epsilon)$-approximate solution to $\mathcal{J}_\Psi$. We construct an assignment σ as follows: if $f_x \in S$, then $\sigma(x) = 1$, otherwise 0. We claim that σ is a $(1-\epsilon)$-approximate solution to $\mathcal{I}$. Due to the construction of $\mathcal{J}_\Psi$, note that $\mathcal{F}$ does not contain a set corresponding to the variable that is set to 0 by Ψ. Thus, if $\Psi(x) = 0$, then $\sigma(x) = 0$. Hence, σ satisfies at least $(1-\epsilon)|N_i^\star|$ clauses negatively, for each $i \in [r]$, with probability at least $1/2$. Next, we argue that σ satisfies at least $(1-\epsilon)|P_i^\star|$ clauses only positively, for each $i \in [r]$. Since S is a $(1-\epsilon)$-approximate solution to $\mathcal{J}_\Psi$, for each $i \in [r]$, S covers at least $(1-\epsilon)t_i'$ elements. Recall that $\mathcal{U}$ contains an element corresponding to each clause in $\cup_{i\in[r]} P_i$. Thus, σ satisfies at least $(1-\epsilon)t_i'$ clauses only positively for each $i \in [r]$. Recall that $t_i' = t_i - \frac{|N_i|}{1-\epsilon}$. Thus, $|P_i| + |N_i| \geq (1-\epsilon)(t_i - \frac{|N_i|}{1-\epsilon}) + |N_i| = (1-\epsilon)t_i$. Hence, σ is a factor $(1-\epsilon)$-approximate solution for $\mathcal{I}$. ◀

Due to Lemma 4.2 and 4.3, we have the following result.

▶ **Theorem 4.4.** *There exists a polynomial time randomized algorithm that given a* Yes*-instance $\mathcal{I}$ of F-MaxSAT generates a* Yes*-instance $\mathcal{J}$ of F-MaxCov with probability at least $(\frac{\epsilon}{2r})^k$. Furthermore, given a factor $(1-\epsilon)$-approximate solution of $\mathcal{J}$, it can be extended to a $(1-\epsilon)$-approximate solution of $\mathcal{I}$ with probability at least $1/2$.*

Note that if the variable-clause incidence graph of the input formula belongs to a subgraph closed family $\mathcal{H}$, then the incidence graph of the resulting instance of F-MaxCov will also belong to $\mathcal{H}$. Thus, due to Theorem 4.4 and Theorem 1.2 in [17], we have the following result, which is an improvement over Theorem 1.1 in [17].

▶ **Theorem 4.5.** *There is a randomized algorithm that given a* Yes*-instance $\mathcal{I}$ of* CC-Max-SAT*, where the variable-clause incidence graph is $K_{d,d}$-free, returns a factor $(1-\epsilon)$-approximate solution with probability at least $(1-\frac{1}{e})$, and runs in time $(\frac{rdk}{\epsilon})^{\mathcal{O}(dk)}(n+m)^{\mathcal{O}(1)}$.*

The $(\frac{r}{\epsilon})^{\mathcal{O}(k)}$ factor in the running time of the algorithm in Theorem 4.5 comes by repeating the algorithm in Theorem 4.4, followed by Theorem 1.2 in [17], independently $(\frac{r}{\epsilon})^{\mathcal{O}(k)}$ many times. This also boosts the success probabilty to at least $(1-\frac{1}{e})$.

▶ Remark 4.6. Note that the reduction from F-MaxSAT to F-MaxCov also works in presence of matroid constraint(s) on the set of variables assigned 1. Recall that in the former (resp. latter) problem, we are given a matroid $\mathcal{M}$ on the set of variables (resp. sets), and the set of at most k variables assigned 1 (resp. at most k sets chosen in the solution) is required to be an independent set in $\mathcal{M}$. This follows from the fact that the randomized algorithm preserves the optimal independent set in the set cover instance with good probability.

5 Conclusion

In this paper, we designed FPT-approximation schemes for (M, F)-MaxSAT, which is a generalization of the CC-MaxSat problem with fairness and matroid constraints. In particular, we designed FPT-AS for the classes of formulas where the maximum frequency of a variable in the clause is bounded by d, and more generally, for $K_{d,d}$-free formulas. Our algorithm for F-MaxCov on the set systems of frequency bounded by d is substantially faster compared to the recent result of Bandyapadhyay et al. [3], even for the special case of $d = 2$. We use a novel combination of the bucketing trick and a carefully designed probability distribution in order to obtain this faster FPT-AS.

Our work naturally leads to the following intriguing questions. Firstly, our approximation-preserving reduction from CC-MaxSat (and variants) to Maximum Coverage (and variants) is inherently randomized. Is it possible to derandomize this reduction? A similar question of derandomization is also interesting for our aforementioned algorithm for F-MaxCov on bounded-frequency set systems. In this case, can we design an FPT-AS for the problem running in time single-exponential in k?

References

1 Omid Amini, Fedor V. Fomin, and Saket Saurabh. Implicit branching and parameterized partial cover problems. *J. Comput. Syst. Sci.*, 77(6):1159–1171, 2011. `doi:10.1016/j.jcss.2010.12.002`.

2 Sayan Bandyapadhyay, Aritra Banik, and Sujoy Bhore. On colorful vertex and edge cover problems. *Algorithmica*, pages 1–12, 2023.

3 Sayan Bandyapadhyay, Zachary Friggstad, and Ramin Mousavi. A parameterized approximation scheme for generalized partial vertex cover. In Pat Morin and Subhash Suri, editors, *WADS 2023*, pages 93–105, 2023.

4 Suman Kalyan Bera, Shalmoli Gupta, Amit Kumar, and Sambuddha Roy. Approximation algorithms for the partition vertex cover problem. *Theor. Comput. Sci.*, 555:2–8, 2014. `doi:10.1016/j.tcs.2014.04.006`.

5 Markus Bläser. Computing small partial coverings. *Inf. Process. Lett.*, 85(6):327–331, 2003. `doi:10.1016/S0020-0190(02)00434-9`.

6 Gruia Călinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011. `doi:10.1137/080733991`.

7 Chandra Chekuri, Tanmay Inamdar, Kent Quanrud, Kasturi R. Varadarajan, and Zhao Zhang. Algorithms for covering multiple submodular constraints and applications. *J. Comb. Optim.*, 44(2):979–1010, 2022. `doi:10.1007/s10878-022-00874-x`.

8 Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. Tight FPT approximations for k-median and k-means. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *ICALP 2019*, pages 42:1–42:14, 2019.

9 M. Cygan, F. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms.* Springer, 2015.

10 Uriel Feige. A threshold of ln n for approximating set cover (preliminary version). In Gary L. Miller, editor, *STOCS, 1996*, pages 314–318. ACM, 1996. `doi:10.1145/237814.237977`.

11 Andreas Emil Feldmann, Karthik C. S., Euiwoong Lee, and Pasin Manurangsi. A survey on approximation in parameterized complexity: Hardness and algorithms. *Algorithms*, 13(6):146, 2020. `doi:10.3390/a13060146`.

12 F. V. Fomin, D. Lokshtanov, and S. Saurabh. Efficient computation of representative sets with applications in parameterized and exact algorithms. In *proceedings of SODA*, pages 142–151, 2014.

13 Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, and Saket Saurabh. Subexponential algorithms for partial cover problems. *Inf. Process. Lett.*, 111(16):814–818, 2011. `doi:10.1016/j.ipl.2011.05.016`.

14 Jiong Guo, Rolf Niedermeier, and Sebastian Wernicke. Parameterized complexity of vertex cover variants. *Theory Comput. Syst.*, 41(3):501–520, 2007. `doi:10.1007/s00224-007-1309-3`.

15 Chien-Chung Huang and François Sellier. Matroid-constrained maximum vertex cover: Approximate kernels and streaming algorithms. In Artur Czumaj and Qin Xin, editors, *SWAT 2022*, pages 27:1–27:15, 2022. `doi:10.4230/LIPIcs.SWAT.2022.27`.

16 Eunpyeong Hung and Mong-Jen Kao. Approximation algorithm for vertex cover with multiple covering constraints. *Algorithmica*, 84(1):1–12, 2022. `doi:10.1007/s00453-021-00885-w`.

17 Pallavi Jain, Lawqueen Kanesh, Fahad Panolan, Souvik Saha, Abhishek Sahu, Saket Saurabh, and Anannya Upasana. Parameterized approximation scheme for biclique-free max k-weight SAT and max coverage. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3713–3733. SIAM, 2023. `doi:10.1137/1.9781611977554.ch143`.

18 Tomohiro Koana, Christian Komusiewicz, André Nichterlein, and Frank Sommer. Covering many (or few) edges with k vertices in sparse graphs. In Petra Berenbrink and Benjamin Monmege, editors, *STACS 2022*, pages 42:1–42:18, 2022. doi:10.4230/LIPIcs.STACS.2022.42.

19 Daniel Lokshtanov, Fahad Panolan, M. S. Ramanujan, and Saket Saurabh. Lossy kernelization. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *STOC 2017*, pages 224–237. ACM, 2017. doi:10.1145/3055399.3055456.

20 Pasin Manurangsi. A note on max k-vertex cover: Faster fpt-as, smaller approximate kernel and improved approximation. In Jeremy T. Fineman and Michael Mitzenmacher, editors, *SOSA 2019*, pages 15:1–15:21, 2019.

21 Pasin Manurangsi. Improved FPT approximation scheme and approximate kernel for biclique-free max k-weight SAT: greedy strikes back. *CoRR*, abs/2403.06335, 2024. doi:10.48550/ARXIV.2403.06335.

22 Dániel Marx. A parameterized view on matroid optimization problems. In Hajo Broersma, Stefan S. Dantchev, Matthew Johnson, and Stefan Szeider, editors, *ACiD 2006*, volume 7 of *Texts in Algorithmics*, page 158. King's College, London, 2006.

23 Dániel Marx. Parameterized complexity and approximation algorithms. *Comput. J.*, 51(1):60–78, 2008.

24 François Sellier. Parameterized matroid-constrained maximum coverage. In Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman, editors, *ESA 2023*, volume 274 of *LIPIcs*, pages 94:1–94:16, 2023.

25 Piotr Skowron and Piotr Faliszewski. Chamberlin-courant rule with approval ballots: Approximating the maxcover problem with bounded frequencies in FPT time. *J. Artif. Intell. Res.*, 60:687–716, 2017. doi:10.1613/jair.5628.

26 Maxim Sviridenko. Best possible approximation algorithm for MAX SAT with cardinality constraint. *Algorithmica*, 30(3):398–405, 2001. doi:10.1007/s00453-001-0019-5.

27 Laurence A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Comb.*, 2(4):385–393, 1982. doi:10.1007/BF02579435.