Lower Bounds for Matroid Optimization Problems with a Linear Constraint

Ilan Doron-Arad
Computer Science Department, Technion, Haifa, Israel

Ariel Kulik
Computer Science Department, Technion, Haifa, Israel

Hadas Shachnai
Computer Science Department, Technion, Haifa, Israel

Abstract

We study a family of *matroid optimization problems with a linear constraint* (MOL). In these problems, we seek a subset of elements which optimizes (i.e., maximizes or minimizes) a linear objective function subject to (i) a matroid independent set, or a matroid basis constraint, (ii) additional linear constraint. A notable member in this family is BUDGETED MATROID INDEPENDENT SET (BM), which can be viewed as classic 0/1-KNAPSACK with a matroid constraint. While special cases of BM, such as KNAPSACK WITH CARDINALITY CONSTRAINT and MULTIPLE-CHOICE KNAPSACK, admit a *fully polynomial-time approximation scheme* (Fully PTAS), the best known result for BM on a general matroid is an *Efficient* PTAS. Prior to this work, the existence of a Fully PTAS for BM, and more generally, for any problem in the family of MOL problems, has been open.

In this paper, we answer this question negatively by showing that none of the (non-trivial) problems in this family admits a Fully PTAS. This resolves the complexity status of several well studied problems. Our main result is obtained by showing first that EXACT WEIGHT MATROID BASIS (EMB) does not admit a pseudo-polynomial time algorithm. This distinguishes EMB from the special cases of k-SUBSET SUM and EMB on a linear matroid, which are solvable in pseudo-polynomial time. We then obtain unconditional hardness results for the family of MOL problems in the oracle model (even if randomization is allowed), and show that the same results hold when the matroids are encoded as part of the input, assuming P $\neq$ NP. For the hardness proof of EMB, we introduce the Π-*matroid* family. This intricate subclass of matroids, which exploits the interaction between a weight function and the matroid constraint, may find use in tackling other matroid optimization problems.

2012 ACM Subject Classification Theory of computation

Keywords and phrases matroid optimization, budgeted problems, knapsack, approximation schemes

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.56

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://doi.org/10.48550/arXiv.2307.07773 [20]

Funding *Ariel Kulik*: This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 852780-ERC (SUBMODULAR).

1 Introduction

Matroids are simple combinatorial structures, providing a unified abstraction for independence systems such as linear independence in a vector space, or cycle-free subsets of edges in a given graph. A *matroid* is a set system $(E, \mathcal{I})$, where E is a finite set and $\mathcal{I} \subseteq 2^E$ are the *independent sets (IS)* such that (i) $\emptyset \in \mathcal{I}$, (ii) for all $A \in \mathcal{I}$ and $B \subseteq A$ it holds that $B \in \mathcal{I}$, and (iii) for all $A, B \in \mathcal{I}$ where $|A| > |B|$, there is $e \in A \setminus B$ such that $B \cup \{e\} \in \mathcal{I}$.[1]

[1] Properties (ii) and (iii) are known, respectively, as *hereditary property*, and *exchange property*.

While serving as a generic abstraction for numerous applications, matroids possess useful combinatorial properties that allow the development of efficient algorithms. These algorithms include such canonical results as the classic greedy approach for finding a maximum weight independent set of a matroid (see, e.g., [13]), Edmond's algorithm for matroid partitioning [21], and Lawler's algorithm for matroid intersection [31]. In all of the above, polynomial running time is enabled due to the structure of the problem – a single objective function with a matroid constraint. However, in many natural applications, there is an added *linear* constraint.

Consider, for example, the problem of finding a maximum independent set in a matroid subject to a budget constraint. Formally, we are given a set of elements E, a membership oracle for a collection of independent sets $\mathcal{I} \subseteq 2^E$ of a matroid $(E, \mathcal{I})$, a budget $L > 0$, a weight function $w : E \to \mathbb{R}_{\geq 0}$, and a value function $v : E \to \mathbb{R}_{\geq 0}$. A *solution* for the problem is an independent set $S \in \mathcal{I}$ of total weight at most L, i.e., $w(S) \leq L$.[2] The *value* of a solution S is given by $v(S)$, and the objective is to find a solution of maximum value. This problem, known as BUDGETED MATROID INDEPENDENT SET (BM), is a generalization of the classic 0/1-KNAPSACK, which is NP-hard and therefore unlikely to admit an exact polynomial-time algorithm. Thus, obtaining efficient approximations has been a main focus in the study of BM.

For an instance I of an optimization problem $\mathcal{G}$, let $\text{OPT}_{\mathcal{G}}(I)$ be the value of an optimal solution for I. For some $\rho \geq 1$, a *ρ-approximate solution* S for I is a solution of value $v \geq \frac{\text{OPT}_{\mathcal{G}}(I)}{\rho}$ ($v \leq \rho \cdot \text{OPT}_{\mathcal{G}}(I)$) if $\mathcal{G}$ is a maximization (minimization) problem. We say that $\mathcal{A}$ is a *randomized ρ-approximation algorithm* for $\mathcal{G}$ if given an instance I of $\mathcal{G}$ $\mathcal{A}$ returns with probability at least $\frac{1}{2}$ a ρ-approximate solution for I – if a solution exists. If no solution exists – $\mathcal{A}$ returns that I does not have a solution.

Let $|I|$ be the encoding size of an instance I of a problem $\mathcal{G}$. A (randomized) *polynomial-time approximation scheme* (PTAS) for $\mathcal{G}$ is a family of algorithms $(A_\varepsilon)_{\varepsilon>0}$ such that, for any $\varepsilon > 0$, A_ε is a (randomized) polynomial-time $(1+\varepsilon)$-approximation algorithm for $\mathcal{G}$. A (randomized) *Efficient PTAS* (EPTAS) is a (randomized) PTAS $(A_\varepsilon)_{\varepsilon>0}$ with running time of the form $f\left(\frac{1}{\varepsilon}\right) \cdot |I|^{O(1)}$, where f is an arbitrary computable function. The strong dependence of run-times on the error parameter, $\varepsilon > 0$, often renders the above schemes highly impractical. This led to the study of the following more desirable class of schemes. A (randomized) approximation scheme $(A_\varepsilon)_{\varepsilon>0}$ is a (randomized) *Fully PTAS* (FPTAS) if the running time of A_ε is of the form $\left(\frac{|I|}{\varepsilon}\right)^{O(1)}$.[3]

In the past decades, BM was shown to admit a PTAS [2, 12, 25], and more recently an Efficient PTAS [18, 17]. As the special case of 0/1-KNAPSACK admits a Fully PTAS, it is natural to explore the existence of a Fully PTAS for BM. There are known Fully PTASs for BM on restricted families of matroids. This includes KNAPSACK WITH A CARDINALITY CONSTRAINT [7], MULTIPLE-CHOICE KNAPSACK [32], and BM WITH LAMINAR MATROID CONSTRAINT [19]. However, the question whether BM admits a Fully PTAS on general matroids remained open.

In this paper, we resolve this question negatively for BM and other fundamental matroid optimization problems with a linear constraint.

[2] For every set X, a function $f : X \to \mathbb{R}_{\geq 0}$ and $Y \subseteq X$ we define $f(Y) = \sum_{e \in Y} f(e)$.

[3] To better distinguish between EPTAS and FPTAS, we use throughout the paper Efficient PTAS and Fully PTAS.

1.1 Our Results

For a matroid $\mathcal{M} = (E, \mathcal{I})$ we define $\text{IS}(\mathcal{M}) = \mathcal{I}$ and $\text{bases}(\mathcal{M}) = \{S \in \mathcal{I} \mid |S| = \mathsf{rank}(\mathcal{M})\}$, where $\mathsf{rank}(\mathcal{M}) = \max_{T \in \mathcal{I}} |T|$ is the *rank* of $\mathcal{M}$, i.e., the maximum cardinality of an independent set. We study a family of *matroid optimization problems with a linear constraint* (MOL). Problems in this family are characterized by three parameters:

(i) The optimization objective opt – either the operator "max" or "min".

(ii) A *matroid feasibility constraint* $\mathcal{F}$ – either the independent sets of a matroid, or the set of *bases* of a matroid. The feasibility constraint is $\mathcal{F} \in \{\text{IS}, \text{bases}\}$.

(iii) A relation $\lhd$ – realized by one of the relations "$\geq$" or "$\leq$".

Let $\mathcal{P} = \{\max, \min\} \times \{\text{bases}, \text{IS}\} \times \{\leq, \geq\}$ be the set of *parameters* for MOL problems. Based on the set of parameters $\mathcal{P}$, we define for every triplet a problem in the MOL family. For $P \in \mathcal{P}$ where $P = (\text{opt}, \mathcal{F}, \lhd)$, define the P-MATROID OPTIMIZATION WITH A LINEAR CONSTRAINT (P-MOL) problem as follows. An instance is a tuple $I = (E, \mathcal{I}, v, w, L)$ such that $\mathcal{M} = (E, \mathcal{I})$ is a matroid, $v : E \to \mathbb{R}_{\geq 0}$ is the objective function, $w : E \to \mathbb{R}_{\geq 0}$ is a weight function, and $L \in \mathbb{R}_{\geq 0}$ is a *bound* for the linear constraint. A *solution* of I is $S \subseteq E$ which satisfies the matroid feasibility constraint $S \in \mathcal{F}(\mathcal{M})$ and the linear constraint $w(S) \lhd L$. The goal is to optimize (i.e., maximize or minimize) the value $v(S)$. Thus, we can formulate a P-MOL optimization problem as

$$\text{opt } v(S) \text{ s.t. } S \in \mathcal{F}(\mathcal{M}), \;\; w(S) \lhd L. \tag{1}$$

Observe that $(\max, \text{IS}, \leq)$-MOL is the BM problem. That is, given a BM instance (equivalently, $(\max, \text{IS}, \leq)$-MOL instance) $I = (E, \mathcal{I}, v, w, L)$, the goal is to find an independent set $S \in \mathcal{I}$ of maximum total value $v(S)$ such that $w(S) \leq L$. Other notable examples for MOL problems are CONSTRAINED MINIMUM BASIS OF A MATROID (CMB) [26], which can be cast as $(\min, \text{bases}, \leq)$-MOL, and KNAPSACK COVER WITH MATROID CONSTRAINT (KCM) [8] formalized by $(\min, \text{IS}, \geq)$-MOL.[4]

We note that (1) does not refer to the *representation* of the instance I. We consider two possible representations. For any $P \in \mathcal{P}$, in an instance $(E, \mathcal{I}, v, w, L)$ of *oracle* P-MOL, the arguments E, v, w, L are given as the input, and the independent sets $\mathcal{I}$ are accessed via a *membership oracle*, which determines whether a given set $S \subseteq E$ belongs to $\mathcal{I}$ in a single query. Thus, the independent sets are not considered in the encoding size of the instance. The term *running time* for problems involving oracles refers to the sum of the number of queries to the oracle and the number of basic operations. Previous works on MOL problems often consider membership oracles [12, 2, 8, 18, 17]. As hardness with oracles does not necessarily imply hardness in non-oracle models (see, e.g., [6, 9]), in Section 4 we show lower bounds for variants of MOL problems in which the independent sets are encoded as part of the input.

Clearly, the problem $(\min, \text{IS}, \leq)$-MOL is trivial since the empty set achieves the optimal objective value. However, for any other $P \in \mathcal{P}$, solving the P-MOL problem is challenging. The *non-trivial* MOL problems are all the MOL problems excluding $(\min, \text{IS}, \leq)$-MOL. That is, P-MOL is non-trivial if $P \in \mathcal{Q}$ where $\mathcal{Q} = \mathcal{P} \setminus \{(\min, \text{IS}, \leq)\}$. Observe that non-trivial MOL problems are NP-hard (e.g., 0/1-KNAPSACK is a special case of $(\max, \text{IS}, \leq)$-MOL); however, all previously studied MOL problems admit approximation schemes.

[4] CMB, KCM, and other MOL problems may not have a solution; however, we can decide in polynomial time if a solution exists, and our definition of approximation algorithms captures instances with no solution.

For certain special cases of MOL problems, e.g., BM with simple matroid constraints, the existence of a Fully PTAS is known for decades [38, 7]. However, for MOL problems with arbitrary matroid constraints, the best known results are Efficient PTAS. While matroids form an important generalization of well known basic constraints, the complexity of the corresponding MOL problems remained open. Specifically, prior to this work, the existence of a MOL problem which does not admit a Fully PTAS was open.

Our main result is that none of the (non-trivial) oracle matroid optimization problem with a linear constraint admits a Fully PTAS, even if randomization is allowed. This unconditioned hardness result is established by deriving a lower bound on the minimum number of queries to the membership oracle.

▶ **Theorem 1.** *For every $P \in \mathcal{Q}$ there is no randomized* Fully PTAS *for oracle P*-MOL.

Table 1 Implications of our results for previously studied MOL problems. All of our bounds hold for randomized algorithms.

Problem	Previous Results	This Paper
Budgeted Matroid Independent Set	Efficient PTAS [18]	**No Fully PTAS**
Budgeted Matroid Intersection	Efficient PTAS [17]	**No Fully PTAS**
Constrained Minimum Basis of Matroid	Efficient PTAS [26]	**No Fully PTAS**
Knapsack Cover with a Matroid	PTAS [8]	**No Fully PTAS**

Theorem 1 conclusively distinguishes MOL problems with arbitrary matroids, such as BM, from special cases with simpler matroid constraints. Furthermore, it shows that existing Efficient PTAS [26, 18, 17] for MOL problems on general matroids are the best possible. Notable implications of our results are given in Table 1, and consequences of our lower bounds for a set of previously studied problems [5, 26, 2, 12, 25, 8, 18, 17, 16] are given in Section 1.3. By resolving the complexity status of MOL problems on general matroids, our results promote future research to design (or rule out) Fully PTAS for MOL problems on restricted matroid classes (see Section 5).

To prove Theorem 1, we turn our attention to the following problem.

▶ **Definition 2.** *An instance of* Exact Matroid Basis (EMB) *is $I = (E, \mathcal{I}, c, T)$, where $(E, \mathcal{I})$ is a matroid, $c : E \to \mathbb{N}$ is a weight function, and $T \in \mathbb{N}$ is a target value. A* solution *is a basis S of $(E, \mathcal{I})$ such that $c(S) = T$. The goal is to decide if there is a solution.*

Similar to MOL problems, EMB does not specify the input. In an instance $(E, \mathcal{I}, c, T)$ of oracle-EMB, E, c, T are explicitly given, and the independent sets $\mathcal{I}$ are accessed via a membership oracle. An instance I of a decision problem $\mathcal{D}$ is a "yes"-instance if the correct answer for I is "yes"; otherwise, I is a "no"-instance. We say that $\mathcal{A}$ is a *randomized algorithm* for a decision problem $\mathcal{D}$ if, given a "yes"-instance I of $\mathcal{D}$, $\mathcal{A}$ returns "yes" with probability at least $\frac{1}{2}$; for a "no"-instance, $\mathcal{A}$ returns "no" with probability 1. The next result rules out a pseudo-polynomial time algorithm for oracle-EMB, thus distinguishing the problem from the special cases of k-subset sum and EMB on linear matroids, which admit a pseudo-polynomial time algorithm [5].

▶ **Theorem 3.** *For any* oracle-EMB *instance $I = (E, \mathcal{I}, c, T)$, there is no randomized algorithm for* oracle-EMB *that runs in time $(n \cdot (T + 2) \cdot m)^{O(1)}$, where $n = |E| + 1$ and $m = c(E) + 1$.*

1.2 Technical Contribution

We derive our results by introducing Π-*matroids*. This new family of *paving* matroids carefully exploits a simple weight function to define a matroid that successfully *hides* a specific property Π within its independent sets (see Section 2).[5] Using Π-matroids, we define oracle-EMB instances whose solutions must satisfy the property Π. This shows the unconditional hardness of oracle-EMB, as Π can be discovered only via an exponential number of queries to the membership oracle. Our hardness results for MOL problems (as stated in Theorem 1) are derived via reduction from oracle-EMB.

Despite the abundance of lower bounds for matroid problems [33, 28, 29, 39], as well as for knapsack problems [10, 30, 14, 3], we are not aware of lower bounds that leverage the interaction between the matroid constraint and the additional linear constraint required for deriving our new lower bound for EMB, and consequently for MOL problems. Indeed, if the matroid constraint is removed from (1) (equivalently, $\mathcal{F}(\mathcal{M}) = 2^E$), MOL problems become variants of classic 0/1-KNAPSACK, which admits a Fully PTAS. Alternatively, if the linear constraint imposed by w, L is removed, then we have the polynomially solvable maximum/minimum weight matroid independent set problem. This distinguishes our construction from existing lower bounds for matroid problems, as even previous constructions of *paving* matroids (e.g., [28]) cannot be easily adapted to tackle both the knapsack constraint along with the matroid constraint. Π-matroids may be useful for deriving lower bounds for other problems (see Section 5).

Our unconditional lower bounds apply in the *oracle* model, where the independent sets of the given matroid can be accessed only via a membership oracle. One may question the validity of the bounds for variants of the problems where the matroid is encoded as part of the input. Indeed, in some scenarios, the use of oracles makes problems harder [6, 9]. Thus, we complement our results by showing that the same lower bounds hold under the standard complexity assumption P $\neq$ NP, even if the matroid is encoded as part of the instance and membership can be decided in polynomial time. We accomplish this by designing the family of SAT-*matroids* – a counterpart of the Π-matroid family whose members can be efficiently encoded. This construction can be used to obtain hardness results for other matroid problems in non-oracle models, based on existing analogous lower bounds in the oracle model (e.g., [28]). We elaborate on that in Section 4.

1.3 Implications of Our Results and Prior Work

Below we describe in further detail the implications of our results, and discuss previous work on MOL problems. In the following problems, general matroids are assumed to be accessed via membership oracles.

Exact Matroid Basis (EMB). This is a generalization of the k-SUBSET SUM problem (where $(E, \mathcal{I})$ is a uniform matroid).[6] Thus, EMB is unlikely to be solvable in polynomial time. Instead, we seek a pseudo-polynomial time algorithm whose running time has polynomial dependence on the encoding size of the instance and the target value T. Indeed, the special case of EMB in which the matroid is *representable* (or, linear) admits such a pseudo-polynomial time algorithm [5]. Since the 1990s, it has been an open question whether the result of Camerini et al. [5] can be extended to general matroids. Theorem 3 resolves this question, ruling out the existence of a pseudo-polynomial time algorithm for EMB.

[5] We note that paving matroids have been used in earlier work, e.g., to show intractability of the matroid matching problem in the oracle model [33, 28].

[6] In a uniform matroid, $\mathcal{I} = \{S \subseteq E \mid |S| \leq k\}$.

Budgeted Matroid Independent Set (BM). This problem is cast as $(\max, \text{IS}, \leq)$-MOL. BM is a natural generalization of the classic 0/1-KNAPSACK problem, for which a Fully PTAS has been known since the 1970s [32]. As mentioned above, a Fully PTAS is known also for other special cases of BM. A PTAS for BM was first given in [2] as a special case of BUDGETED MATROID INTERSECTION (BMI). In this generalization of BM, we are given *two* matroids $(E, \mathcal{I}_1)$ and $(E, \mathcal{I}_2)$, and a solution has to be an independent set of *both* matroids. A PTAS for BM also follows from the results of [25, 12] which present PTASs for multi-budgeted variants of BM. An Efficient PTAS for BM was recently given in [18] and for BMI in [17]. The existence of a Fully PTAS for BM was posed as a central open question in [2, 18, 17]. We answer this question negatively, as formalized in Theorem 1, giving a tight lower bound for BM and BMI.

Constrained Minimum Basis of a Matroid (CMB). This problem can be cast as the matroid optimization problem $(\min, \text{bases}, \leq)$-MOL. The CONSTRAINED MINIMUM SPANNING TREE (CST) problem is the special case of CMB in which the matroid $(E, \mathcal{I})$ is graphical [36, 1, 26, 27], namely, there is a graph $G = (V, E)$ such that the independent sets $\mathcal{I}$ are cycle-free subsets of edges in G. A PTAS for CST was given by Ravi and Goemans [36]. This result was improved to an Efficient PTAS by Hassin and Levin [26]. A *bicriteria* FPTAS, which violates the budget constraint by a factor of $(1+\varepsilon)$, was presented in [27]. The authors of [26] mention that their result actually gives an Efficient PTAS for CMB. The existence of a Fully PTAS for CMB remained an open question. Theorem 1 shows that the Efficient PTAS for CMB cannot be improved.

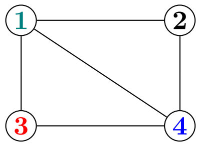
Knapsack Cover with a Matroid (KCM). As a final implication, Theorem 1 rules out the existence of a Fully PTAS for a *coverage* variant of 0/1-KNAPSACK, formulated as $(\min, \text{IS}, \geq)$-MOL. In [8], Chakaravarthy et al. presented a PTAS for KCM using integrality properties of a linear programming formulation of KCM. Moreover, for the special case of KCM with a *partition matroid*, they give a Fully PTAS based on dynamic programming. The existence of a Fully PTAS for KCM on a general matroid was posed in [8] as an open question. Theorem 1 answers this question negatively. Our initial study indicates that an Efficient PTAS for KCM can potentially be obtained by adapting the approach of Hassin and Levin [26] to the setting of KCM. This suggests that our lower bound cannot be strengthened.

1.4 Organization

In Section 2 we introduce the Π-matroid family and give the proof of Theorem 3. In Section 3 we prove Theorem 1, and in Section 4 we show that similar lower bounds hold in the standard computational model. We conclude in Section 5 with a summary and directions for future work. Due to space constraints, some of the proofs are given in the full version of the paper [20].

2 The Hardness of ORACLE EXACT MATROID BASIS

In this section, we prove Theorem 3. We use in the proof the family of Π-*matroids*. For any $m \in \mathbb{N}$, let $[m] = \{1, \ldots, m\}$. A member in the Π-matroid family is given by four arguments: $n, k, \alpha \in \mathbb{N}_{>0}$, and $\Pi \subseteq 2^{[n]}$. The first argument, $n \in \mathbb{N}_{>0}$, is the number of elements, and the ground set is $[n]$. The second argument, $k \in [n]$, is the rank of the matroid. The third argument, $\alpha \in \mathbb{N}_{>0}$, is a target value, that is usually equal to $\mathsf{sum}(S)$ for some $S \subseteq [n]$, where $\mathsf{sum}(S) = \sum_{i \in S} i$. The last argument is a family of subsets $\Pi \subseteq 2^{[n]}$.

A graph G 

$\Pi = \{\text{independent sets of } G\}$

$$\mathcal{J}_{4,2} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}\}, \quad \mathcal{K}_{4,2,5} = \{\{2,4\}, \{2,1\}, \{4,3\}, \{3,1\}\}, \quad \mathcal{L}_{4,2,5}(\Pi) = \{\{2,3\}\}$$

Figure 1 The independent sets of the Π-matroid $M_{n,k,\alpha}(\Pi)$, with parameters $n = 4$, $k = 2$, and $\alpha = 5$. The secret family Π contains all independent sets in the graph G, where $\{2, 3\}$ is the only independent set in G with k elements.

The set Π is called the *secret family* because finding $S \in \Pi$ is possible only via repeated queries to the membership oracle of the matroid. Since Π can have an arbitrary structure, this may require exhaustive enumeration.

▶ **Definition 4.** *Let $n, k, \alpha \in \mathbb{N}_{>0}$. For some $\Pi \subseteq 2^{[n]}$, define the* **Π-matroid** *on n, k, and α as $M_{n,k,\alpha}(\Pi) = ([n], \mathcal{I}_{n,k,\alpha}(\Pi))$, where*

$$\mathcal{I}_{n,k,\alpha}(\Pi) = \mathcal{J}_{n,k} \cup \mathcal{K}_{n,k,\alpha} \cup \mathcal{L}_{n,k,\alpha}(\Pi)$$

and $\mathcal{J}_{n,k}, \mathcal{K}_{n,k,\alpha}, \mathcal{L}_{n,k,\alpha}(\Pi)$ are defined as follows.

$$\begin{aligned} \mathcal{J}_{n,k} &= \big\{S \subseteq [n] \bigm| |S| < k\big\} \\ \mathcal{K}_{n,k,\alpha} &= \big\{S \subseteq [n] \bigm| |S| = k, \mathsf{sum}(S) \neq \alpha\big\} \\ \mathcal{L}_{n,k,\alpha}(\Pi) &= \big\{S \subseteq [n] \bigm| |S| = k, \mathsf{sum}(S) = \alpha, S \in \Pi\big\}. \end{aligned} \tag{2}$$

In words, $\mathcal{J}_{n,k}$ contains all subsets of strictly less than k elements; $\mathcal{K}_{n,k,\alpha}$ contains all subsets of cardinality k whose total sum is not α. Finally, $\mathcal{L}_{n,k,\alpha}(\Pi)$ contains all subsets of cardinality k and total sum α which also belong to Π. See Figure 1 for an example of a member of the Π-matroid family. Using a simple argument, we show that the set system in Definition 4 is indeed a matroid. For the sets $\mathcal{I}_{n,k,\alpha}(\Pi)$, $\mathcal{J}_{n,k}$, $\mathcal{K}_{n,k,\alpha}$ and $\mathcal{L}_{n,k,\alpha}(\Pi)$ defined in Definition 4, we often omit the subscripts n, k, α and n, k when the values of n, k, α are known by context. For simplicity, for any set A and an element a, let $A + a$, $A - a$ be $A \cup \{a\}$ and $A \setminus \{a\}$, respectively.

▶ **Lemma 5.** *For every $n, k, \alpha \in \mathbb{N}_{>0}$, and $\Pi \subseteq 2^{[n]}$ it holds that $M_{n,k,\alpha}(\Pi)$ is a matroid.*[7]

Proof. We first note that $\emptyset \in \mathcal{J}$ since $0 < k$; therefore, $\emptyset \in \mathcal{I}(\Pi)$. For the hereditary property, let $A \in \mathcal{I}(\Pi)$. For all $B \subset A$ it holds that $|B| < k$; thus, $B \in \mathcal{J}$ and it follows that $B \in \mathcal{I}(\Pi)$. For the exchange property, let $A, B \in \mathcal{I}(\Pi)$ such that $|A| > |B|$. We consider the following cases.

1. $|B| < k - 1$. Then, for all $e \in A \setminus B$ it holds that $|B + e| < k - 1 + 1 = k$. Hence, $B + e \in \mathcal{J}$ and it follows that $B + e \in \mathcal{I}(\Pi)$. Note that there is such $e \in A \setminus B$ because $|A| > |B|$.
2. $|B| = k - 1$ and $|A| = k$. We consider two subcases.
   a. $B \subseteq A$. Then, as $|A| > |B|$ there is $e \in A \setminus B$. Hence, $B + e = A$ (because $|B| = k - 1$ and $|A| = k$). As $A \in \mathcal{I}(\Pi)$, it follows that $B + e \in \mathcal{I}(\Pi)$.

[7] We remark that the lemma can be proved using Theorem 5.3.5 in [23]. We give the proof for completeness.

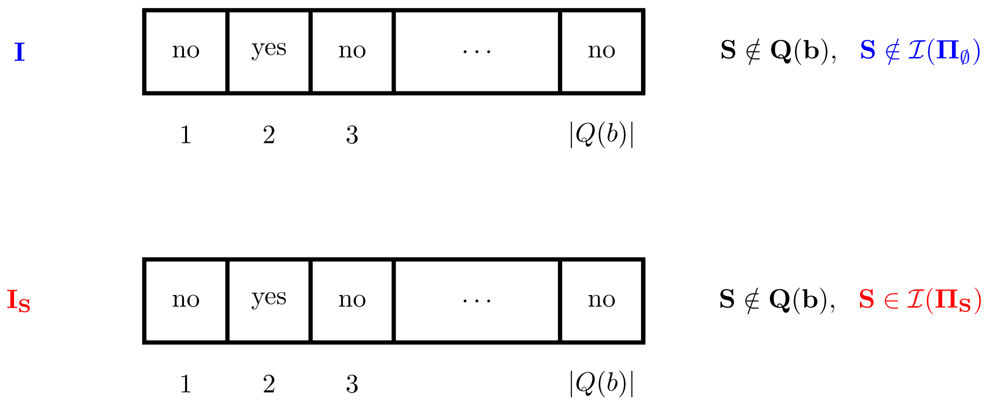


Figure 2 An illustration of the proof of Theorem 3. The figure presents the sequences of queries to the membership oracles by the algorithm on the instances I and I_S for a string of bits b, such that $S \notin Q(b)$. The label "yes" ("no") indicates that the queried set is (not) independent in the matroid. The only query that distinguishes between I and I_S is on the set S, which is not queried; thus, the algorithm returns the same output for I and I_S.

b. $B \not\subseteq A$. Then, as $|B| = k - 1$ and $|A| = k$ it follows that $|B \cap A| < |B| = k - 1$. Thus, $|A \setminus B| = |A| - |A \cap B| > k - (k-1) = 1$. Hence, there are $e, f \in A \setminus B$ such that $e \neq f$. It follows that there is $g \in \{e, f\}$ such that $\mathsf{sum}(B + g) = \mathsf{sum}(B) + g \neq \alpha$. We conclude from (2) that $B + g \in \mathcal{K}$, implying that $B + g \in \mathcal{I}(\Pi)$. ◀

Observe that $\mathcal{K}_{n,k,\alpha} \cup \mathcal{L}_{n,k,\alpha}(\Pi)$ is the set of bases of the matroid. Moreover, for any arguments n, k, α and Π, the cardinality of every dependent set $S \in 2^{[n]} \setminus \mathcal{I}_{n,k,\alpha}(\Pi)$ is at least the rank of the Π-matroid $M_{n,k,\alpha}(\Pi) = ([n], \mathcal{I}_{n,k,\alpha}(\Pi))$. Such matroids are known as *paving matroids* (see, e.g., [34, 35]). Using Π-matroids, we define the following collection of oracle-EMB instances. In these instances, the matroid is a Π-matroid, where Π is some unknown fixed family of subsets of the ground set.

▶ **Definition 6.** *For every $n, k, \alpha \in \mathbb{N}_{>0}$, and $\Pi \subseteq 2^{[n]}$ define the* induced *oracle*-EMB *instance of n, k, α, Π, denoted $I_{n,k,\alpha}(\Pi)$, as follows. Let $\mathsf{id}_n : [n] \to [n]$, where $\mathsf{id}_n(i) = i$ $\forall i \in [n]$. Then, $I_{n,k,\alpha}(\Pi) = ([n], \mathcal{I}_{n,k,\alpha}(\Pi), \mathsf{id}_n, \alpha)$.*

Observe that the above is indeed an oracle-EMB instance if the independent sets of the given matroid are accessible via a membership oracle. The following is an easy consequence of Definition 6.

▶ **Observation 7.** *For every $n, k, \alpha \in \mathbb{N}_{>0}$ and $\Pi \subseteq 2^{[n]}$, it holds that $I_{n,k,\alpha}(\Pi)$ is an* oracle-EMB *"yes"-instance if and only if there is $S \in \Pi$ such that $|S| = k$ and $\mathsf{sum}(S) = \mathsf{id}_n(S) = \alpha$.*

By Observation 7, an algorithm that finds an independent set of $\mathcal{M}_{n,k,\alpha}(\Pi)$ satisfying $|S| = k$ and $\mathsf{sum}(S) = \alpha$, in fact outputs a subset $S \in \Pi$. As the input for an induced oracle-EMB instance $I_{n,k,\alpha}(\Pi)$ does not contain an explicit encoding of Π, finding $S \in \Pi$ requires a sequence of queries to the membership oracle of $M_{n,k,\alpha}(\Pi)$. Roughly speaking, to decide $I_{n,k,\alpha}(\Pi)$, an algorithm for oracle-EMB must iterate over all (exponentially many) subsets $S \subseteq [n]$ such that $|S| = k$ and $\mathsf{sum}(S) = \alpha$. This is the intuition behind the proof of the next result.

▶ **Theorem 3.** *For any* oracle-EMB *instance $I = (E, \mathcal{I}, c, T)$, there is no randomized algorithm for* ORACLE-EMB *that runs in time $(n \cdot (T + 2) \cdot m)^{O(1)}$, where $n = |E| + 1$ and $m = c(E) + 1$.*

Proof. Assume towards contradiction that there exist a constant $d \in \mathbb{N}$ and a randomized algorithm $\mathcal{A}$ that decides every oracle-EMB instance $(E, \mathcal{I}, c, T)$ in time $((T+2) \cdot n \cdot m)^d$, where $n = |E| + 1$ and $m = c(E) + 1$. For every $n, k, \alpha \in \mathbb{N}_{>0}$, consider the set of all subsets of $[n]$ with cardinality k and sum α, i.e.,

$$\mathcal{F}_{n,k,\alpha} = \left\{ S \subseteq [n] \;\middle|\; |S| = k, \mathsf{sum}(S) = \alpha \right\}.$$

To reach a contradiction, we construct an induced oracle-EMB instance on which $\mathcal{A}$ does not compute the proper output with sufficiently high probability. The parameters of the instance are extracted from the following combinatorial claim.

▷ Claim 8. There are $\tilde{n} \in \mathbb{N}_{>0}$, $\tilde{k} \in [\tilde{n}]$, and $\tilde{\alpha} \in \left[\tilde{n}^2\right]$ such that $|\mathcal{F}_{\tilde{n},\tilde{k},\tilde{\alpha}}| > 2 \cdot \left(12 \cdot \tilde{n}^5\right)^d$.

Proof. Since d is a constant, there is $\tilde{n} \in \mathbb{N}_{>0}$ such that

$$\left(12 \cdot \tilde{n}^5\right)^d < \frac{2^{\tilde{n}} - 1}{2 \cdot \tilde{n}^3}. \tag{3}$$

Fix $\tilde{n} \in \mathbb{N}_{>0}$ satisfying (3). Recall that $\sum_{k \in \{0,1,\ldots,\tilde{n}\}} \binom{\tilde{n}}{k} = 2^{\tilde{n}}$ from a basic property of the Pascal triangle; therefore, $\sum_{k \in \{1,\ldots,\tilde{n}\}} \binom{\tilde{n}}{k} = 2^{\tilde{n}} - 1$. Thus, there is $\tilde{k} \in \{1, \ldots, \tilde{n}\}$, such that

$$\binom{\tilde{n}}{\tilde{k}} \geq \frac{2^{\tilde{n}} - 1}{\tilde{n}}. \tag{4}$$

Fix $\tilde{k} \in [\tilde{n}]$ satisfying (4). Observe that for each $S \in 2^{[\tilde{n}]}$ satisfying $|S| = \tilde{k} > 0$, it holds that $1 \leq \mathsf{sum}(S) \leq |S| \cdot \max_{i \in [\tilde{n}]} i = \tilde{n}^2$. Thus, there are $\tilde{n}^2$ possibilities for $\alpha \in \left[\tilde{n}^2\right]$ satisfying $\alpha = \mathsf{sum}(S)$, for some $S \in 2^{[\tilde{n}]}$ such that $|S| = \tilde{k}$. Moreover, there are $\binom{\tilde{n}}{\tilde{k}}$ subsets of $[\tilde{n}]$ of cardinality $\tilde{k}$. By the pigeonhole principle, there is $\tilde{\alpha} \in \left[\tilde{n}^2\right]$ such that $|\mathcal{F}_{\tilde{n},\tilde{k},\tilde{\alpha}}| \geq \frac{\binom{\tilde{n}}{\tilde{k}}}{\tilde{n}^2}$. Thus,

$$|\mathcal{F}_{\tilde{n},\tilde{k},\tilde{\alpha}}| \geq \frac{\binom{\tilde{n}}{\tilde{k}}}{\tilde{n}^2} \geq \frac{2^{\tilde{n}} - 1}{\tilde{n} \cdot \tilde{n}^2} > 2 \cdot \left(12 \cdot \tilde{n}^5\right)^d.$$

The second inequality follows from (4), and the third inequality holds by (3). ◁

Let $\tilde{n} \in \mathbb{N}_{>0}$, $\tilde{k} \in [\tilde{n}]$, and $\tilde{\alpha} \in \left[\tilde{n}^2\right]$ satisfying the conditions of Claim 8. Define t to be the maximum running time of $\mathcal{A}$ on an induced EMB instance $I_{\tilde{n},\tilde{k},\tilde{\alpha}}(\Pi)$ over all $\Pi \in 2^{[\tilde{n}]}$.

▷ Claim 9. $t \leq \left(12 \cdot \tilde{n}^5\right)^d$.

Proof. Let $T = \tilde{\alpha}$, $n = \tilde{n} + 1$, and $m = \mathsf{id}_{\tilde{n}}([\tilde{n}]) + 1$. By the running time guarantee of $\mathcal{A}$, it follows that $t \leq (n \cdot (T+2) \cdot m)^d$. It remains to bound $n \cdot (T+2) \cdot m$. Since $\mathsf{id}_{\tilde{n}}([\tilde{n}]) = \mathsf{sum}([\tilde{n}])$ and $\tilde{\alpha} \leq \tilde{n}^2$,

$$n \cdot (T+2) \cdot m \leq (\tilde{n} + 1) \cdot \left(\tilde{n}^2 + 2\right) (\mathsf{sum}([\tilde{n}]) + 1) \leq 2\tilde{n} \cdot 3\tilde{n}^2 \cdot \left(\frac{\tilde{n} \cdot (\tilde{n} + 1)}{2} + 1\right) \leq 12 \cdot \tilde{n}^5.$$

The second inequality follows from the sum of the terms of an arithmetic sequence. By the above and the running time guarantee of $\mathcal{A}$, it follows that $t \leq (n \cdot (T+2) \cdot m)^d \leq \left(12 \cdot \tilde{n}^5\right)^d$. ◁

Given an induced oracle-EMB instance $I_{\tilde{n},\tilde{k},\tilde{\alpha}}(\Pi)$, for some $\Pi \subseteq [\tilde{n}]$, the randomized algorithm $\mathcal{A}$ generates a random string of bits $\bar{b} \in \{0,1\}^t$ and performs a sequence of queries to the membership oracle of $M_{\tilde{n},\tilde{k},\tilde{\alpha}}$, based on $\bar{b}, \tilde{n}, \tilde{k}, \tilde{\alpha}$, and the results of the previous queries. Then, the algorithm decides the given instance based on the queries.

Let $\Pi_\emptyset = \emptyset$, and consider the induced oracle-EMB instance $I = I_{\tilde{n},\tilde{k},\tilde{\alpha}}(\Pi_0)$. Given a string of bits $b \in \{0,1\}^t$, let $Q(b) \subseteq 2^{[\tilde{n}]}$ be the set of all subsets $S \subseteq [\tilde{n}]$ queried by $\mathcal{A}$ on the instance I and the bit-string b (on the membership oracle of $M_{\tilde{n},\tilde{k},\tilde{\alpha}}(\Pi_\emptyset)$). For clarity, we use $\bar{b}$ for a random string, and b for a realization of $\bar{b}$ to a specific string. Note that $Q(b)$ is a set, since the algorithm is deterministic for every $b \in \{0,1\}^t$; conversely, $Q(\bar{b})$ is a random set for a random string $\bar{b} \in \{0,1\}^t$. As the running time of $\mathcal{A}$ on I is bounded by t, it holds that $|Q(b)| \leq t$ for every $b \in \{0,1\}^t$. Let

$$R(I) = \left\{ S \in \mathcal{F}_{\tilde{n},\tilde{k},\tilde{\alpha}} \;\middle|\; \Pr\left(S \in Q(\bar{b})\right) \geq \frac{1}{2} \right\}$$

be all sets in $\mathcal{F}_{\tilde{n},\tilde{k},\tilde{\alpha}}$ that are queried by $\mathcal{A}$ with probability at least $\frac{1}{2}$.

▷ Claim 10. $|R(I)| < |\mathcal{F}_{\tilde{n},\tilde{k},\tilde{\alpha}}|$.

Proof. By the definition of $R(I)$, it holds that

$$\begin{aligned}|R(I)| &= \left| \left\{ S \in \mathcal{F}_{\tilde{n},\tilde{k},\tilde{\alpha}} \;\middle|\; \Pr\left(S \in Q(\bar{b})\right) \geq \frac{1}{2} \right\} \right| \\ &\leq 2 \cdot \sum_{S \in \mathcal{F}_{\tilde{n},\tilde{k},\tilde{\alpha}}} \Pr\left(S \in Q(\bar{b})\right) \\ &= 2 \cdot \sum_{S \in \mathcal{F}_{\tilde{n},\tilde{k},\tilde{\alpha}}} \sum_{b \in \{0,1\}^t} \Pr\left(S \in Q(b)\right) \cdot \Pr(\bar{b} = b).\end{aligned}$$

Thus, by changing the order of summation, we have

$$|R(I)| \leq 2 \cdot \sum_{b \in \{0,1\}^t} \Pr(\bar{b} = b) \cdot \sum_{S \in \mathcal{F}_{\tilde{n},\tilde{k},\tilde{\alpha}}} \Pr\left(S \in Q(b)\right) \leq 2 \cdot \sum_{b \in \{0,1\}^t} \Pr(\bar{b} = b) \cdot |Q(b)|.$$

Since $|Q(b)| \leq t$ for all $b \in \{0,1\}^t$, by the above we have

$$|R(I)| \leq 2t \cdot \sum_{b \in \{0,1\}^t} \Pr(\bar{b} = b) = 2 \cdot t \leq 2 \cdot \left(12 \cdot \tilde{n}^5\right)^d < |\mathcal{F}_{\tilde{n},\tilde{k},\tilde{\alpha}}|.$$

The second inequality follows from Claim 9. The last inequality holds by Claim 8. ◁

By Claim 10, there exists $S \in F_{\tilde{n},\tilde{k},\tilde{\alpha}} \setminus R(I)$. Consider the induced oracle-EMB instance $I_S = I_{\tilde{n},\tilde{k},\tilde{\alpha}}(\Pi_S)$ where $\Pi_S = \{S\}$, and let $B = \{b \in \{0,1\}^t \mid S \notin Q(b)\}$ be all strings for which S is not queried by $\mathcal{A}$ on the instance I. Observe that for all $b \in B$ it holds that the answers to all queries for $T \in Q(b)$ are the same for both oracles (of $M_{\tilde{n},\tilde{k},\tilde{\alpha}}(\Pi_\emptyset)$ and $M_{\tilde{n},\tilde{k},\tilde{\alpha}}(\Pi_S)$). Moreover, the decision on which set to query next depends only on $b, \tilde{n}, \tilde{k}, \tilde{\alpha}$, and the answers to previous queries.

Hence, for all $b \in B$, the executions of $\mathcal{A}$ on the instances I and I_S are identical. Since $\Pi_\emptyset = \emptyset$, by Observation 7, I is a "no"-instance for oracle-EMB; thus, $\mathcal{A}$ returns that I_S is a "no"-instance for every $b \in B$. However, since $\mathsf{id}_n(S) = \mathsf{sum}(S) = \alpha$, $|S| = k$, and $S \in \Pi_S$, it follows that I_S is a "yes"-instance by Observation 7. Therefore, $\mathcal{A}$ does not decide I_S correctly for all $b \in B$. We give an illustration in Figure 2. Since $S \notin R(I)$, it holds that $\Pr\left(\bar{b} \in B\right) = \Pr\left(S \notin Q(\bar{b})\right) > \frac{1}{2}$; thus, with probability greater than $\frac{1}{2}$, $\mathcal{A}$ does not decide correctly the instance I_S. A contradiction to the correctness of $\mathcal{A}$ as a randomized algorithm for oracle-EMB. The statement of the theorem follows. ◀

3 Hardness of Matroid Optimization with a Linear Constraint

In this section we use Theorem 3 to prove Theorem 1. We apply the following reductions from EMB to MOL problems. Recall that $\mathcal{Q} = \mathcal{P} \setminus \{(\min, \text{IS}, \leq)\}$ is the set of parameters for non-trivial MOL problems. Given a P-MOL problem for some $P \in \mathcal{Q}$, and an EMB instance I, the reduction returns an instance $R_P(I)$ of the P-MOL problem. Note that the reduction is purely mathematical, and does not specify the encoding of the instance. This will be useful for obtaining our hardness results in non-oracle computational models (see Section 4).

Given $(\text{opt}, \mathcal{F}, \triangleleft), (\text{opt}', \mathcal{F}', \triangleleft') \in \mathcal{Q}$, we use the notation (opt is opt′), ($\mathcal{F}$ is $\mathcal{F}'$), ($\triangleleft$ is $\triangleleft'$) to denote the boolean expressions of equality between parameters of a MOL problem. For example, (opt is opt′) is *true* if and only if either opt, opt′ are both max, or opt, opt′ are both min.

▶ **Definition 11.** *Given an* EMB *instance* $I = (E, \mathcal{I}, c, T)$ *and* $P \in \mathcal{Q}$ *where* $P = (\text{opt}, \mathcal{F}, \triangleleft)$, *define the* reduced P-MOL instance *of* I, *denoted by* $R_P(I) = (E, \mathcal{I}, v_I, w_{I,P}, L_{I,P})$, *as follows.*

1. *Define the auxiliary variable*

$$d(P) = \begin{cases} 0 & \text{if } \Big((\text{opt is } \max) \text{ and } (\triangleleft \text{ is } \leq)\Big) \text{ or } \Big((\text{opt is } \min) \text{ and } (\triangleleft \text{ is } \geq)\Big) \\ 1 & \text{otherwise.} \end{cases}$$

   For example, if $P = (\max, \text{IS}, \leq)$ *then* $d(P) = 0$, *and if* $P' = (\max, \text{IS}, \geq)$ *then* $d(P') = 1$.
2. *Let* $H_I = 2 \cdot \max\{1, c(E)\}$.
3. *For all* $e \in E$ *let* $v_I(e) = H_I + c(e)$.
4. *For all* $e \in E$ *let* $w_{I,P}(e) = H_I + c(e) \cdot (-1)^{d(P)}$.
5. *Let* $k_I = \max_{S \in \mathcal{I}} |S|$ *be the rank of* $(E, \mathcal{I})$.
6. *Define* $L_{I,P} = k_I \cdot H_I + T \cdot (-1)^{d(P)}$.

Now, for every EMB instance $I = (E, \mathcal{I}, c, T)$, define the *error parameter* of I as

$$\varepsilon_I = \frac{1}{8 \cdot (|E| + 1) \cdot (T + 1) \cdot (c(E) + 1)}. \tag{5}$$

Indeed, since the value selected for of the error parameter is sufficiently small, we can use a $(1 + \varepsilon_I)$-approximation for $R_P(I)$ to decide an EMB instance I.

▶ **Theorem 12.** *Given an instance* $I = (E, \mathcal{I}, c, T)$ *of* EMB, *and* $P \in \mathcal{Q}$ *with* $P = (\text{opt}, \mathcal{F}, \triangleleft)$, *the following holds.*

1. *If there is a solution* S *for* $R_P(I)$ *such that* $v_I(S) = k_I \cdot H_I + T$, *then* I *is a "yes"-instance.*
2. *If* I *is a "yes"-instance then: (i)* $R_P(I)$ *has a solution, and (ii) every* $(1+\varepsilon_I)$*-approximate solution* S *for* $R_P(I)$ *satisfies* $v_I(S) = k_I \cdot H_I + T$.

Using Theorem 12 and an assumed randomized Fully PTAS for the P-MOL problem, we can decide EMB in time which contradicts Theorem 3. This gives the proof of Theorem 1. We first prove Theorem 1, and later give the proof of Theorem 12.

▶ **Theorem 1.** *For every* $P \in \mathcal{Q}$ *there is no randomized* Fully PTAS *for oracle* P-MOL.

Proof. Assume towards a contradiction that there is a randomized Fully PTAS $\mathcal{A}$ for oracle P-MOL. We use $\mathcal{A}$ to decide oracle-EMB. Let $I = (E, \mathcal{I}, c, T)$ be an oracle-EMB instance, and consider the following randomized algorithm $\mathcal{B}$ that decides I.

1. Construct the oracle P-MOL instance $R_P(I)$ with the membership oracle of $(E, \mathcal{I})$.
2. Execute $\mathcal{A}$ with the input $R_P(I)$ and ε_I.
3. If $\mathcal{A}$ returns that $R_P(I)$ does not have a solution $-$ Return "no" on I.
4. Otherwise, let $S \leftarrow \mathcal{A}(R_P(I), \varepsilon_I)$ be the solution returned by $\mathcal{A}$.
5. Return "yes" on I if and only if $v_I(S) = H_I \cdot k_I + T$.

Let $n = |E| + 1$ and $m = c(E) + 1$. Note that $R_P(I)$ can be naively constructed from I in time $(n \cdot (T+2) \cdot m)^{O(1)}$ using Definition 11. As $\mathcal{A}$ is a randomized Fully PTAS for oracle P-MOL, and by the selection of the error parameter (5), the running time of $\mathcal{B}$ on I is $(n \cdot (T+2) \cdot m)^{O(1)}$. We now show correctness.

- If $\mathcal{B}$ returns "yes" on I, then Step 4 of the algorithm computes a solution S for $R_p(I)$ satisfying $v_I(S) = H_I \cdot k_I + T$. Thus, by Theorem 12, I is a "yes" instance.
- If I is a "yes" instance then $R_P(I)$ has a solution by Theorem 12. As $\mathcal{A}$ is a randomized Fully PTAS, with probability at least $\frac{1}{2}$ $\mathcal{A}$ returns a $(1+\varepsilon_I)$-approximate solution S for $R_P(I)$ in Step 4. By Theorem 12, $v_I(S) = H_I \cdot k_I + T$ (with probability at least $\frac{1}{2}$). Thus, $\mathcal{B}$ returns "yes" on I with probability at least $\frac{1}{2}$.

Hence, $\mathcal{B}$ is a randomized algorithm which decides the oracle-EMB instance I in time $(n \cdot (T+2) \cdot m)^{O(1)}$. This is a contradiction to Theorem 3. ◀

In the remainder of this section we prove Theorem 12. We start with some basic properties of the reduction outlined in Definition 11.

▶ **Lemma 13.** *Given an instance $I = (E, \mathcal{I}, c, T)$ of* EMB*, let $P \in \mathcal{Q}$ and consider a solution S for $R_P(I)$ satisfying $v_I(S) = k_I \cdot H_I + T$. Then, S is a solution for I.*

Proof. Let $\mathcal{M} = (E, \mathcal{I})$. As S is a solution for $R_P(I)$, it holds that $S \in \text{IS}(\mathcal{M})$; thus, $|S| \leq k_I$. Assume towards contradiction that $|S| < k_I$. Then,

$$v_I(S) = |S| \cdot H_I + c(S) \leq (k_I - 1) \cdot H_I + c(S) \leq (k_I - 1) \cdot H_I + c(E) < k_I \cdot H_I \leq k_I \cdot H_I + T.$$

We reach a contradiction since $v_I(S) = k_I \cdot H_I + T$; thus, $|S| = k_I$, and

$$k_I \cdot H_I + T = v_I(S) = |S| \cdot H_I + c(S) = k_I \cdot H_I + c(S). \tag{6}$$

As $|S| = k_I$, we have that S is a basis of $\mathcal{M}$, and by (6), $c(S) = T$. Hence, S is a solution for I. ◀

The next result is the converse of the statement in Lemma 13.

▶ **Lemma 14.** *Let S be a solution for a given* EMB *instance $I = (E, \mathcal{I}, c, T)$, and let $P \in \mathcal{Q}$ where $P = (\text{opt}, \mathcal{F}, \triangleleft)$. Then, S is a solution for $R_P(I)$ of value $v_I(S) = k_I \cdot H_I + T$.*

Proof. Let $\mathcal{M} = (E, \mathcal{I})$. Since S is a solution for I we have $S \in \text{bases}(\mathcal{M})$; thus, $S \in \mathcal{F}(\mathcal{M})$. Then,

$$w_{I,P}(S) = |S| \cdot H_I + c(S) \cdot (-1)^{d(P)} = k_I \cdot H_I + T \cdot (-1)^{d(P)} = L_{I,P}.$$

The second equality holds since S is a solution for I; thus, $|S| = k_I$ (as S is a basis of $\mathcal{M}$), and $c(S) = T$. We conclude that S is a solution for $R_P(I)$. Finally, note that S satisfies

$$v_I(S) = |S| \cdot H_I + c(S) = k_I \cdot H_I + T$$ ◀

The next claim gives an upper bound on the optimal value for maximization MOL problems. We then derive an analogous lower bound for minimization (non-trivial) MOL problems.

▶ **Lemma 15.** *Let* $I = (E, \mathcal{I}, c, T)$ *be an* EMB *instance and* $P \in \mathcal{Q}$*, where* $P = (\text{opt}, \mathcal{F}, \triangleleft)$ *and* (opt is max)*. Then, for every solution* S *of* $R_P(I)$ *it holds that* $v_I(S) \leq k_I \cdot H_I + T$*.*

Proof. Let S be an optimal solution for $R_P(I)$. Thus, $|S| \leq k_I$ as $S \in \mathcal{I}$. If $|S| < k_I$ then

$$v_I(S) \leq (k_I - 1) \cdot H_I + c(S) \leq (k_I - 1) \cdot H_I + c(E) < k_I \cdot H_I \leq k_I \cdot H_I + T.$$

Otherwise, $|S| = k_I$. Consider the two cases for $\triangleleft$.

1. $(\triangleleft \text{ is } \leq)$. Then, $d(P) = 0$ (see Definition 11); thus, since S is a solution for $R_P(I)$:

$$v_I(S) = w_{I,P}(S) \leq L_{I,P} = k_I \cdot H_I + T.$$

2. $(\triangleleft \text{ is } \geq)$. Then, $d(P) = 1$. As S is a solution for $R_P(I)$,

$$k_I \cdot H_I - c(S) = |S| \cdot H_I - c(S) = w_{I,P}(S) \geq L_{I,P} = k_I \cdot H_I - T. \tag{7}$$

By (7), it follows that $c(S) \leq T$; thus, $v_I(S) = k_I \cdot H_I + c(S) \leq k_I \cdot H_I + T$.

In all the above cases, we have that $v_I(S) \leq k_I \cdot H_I + T$, implying the statement of the lemma. ◀

Now, for minimization problems we have the next result.

▶ **Lemma 16.** *Let* $I = (E, \mathcal{I}, c, T)$ *be an* EMB *instance, and* $P \in \mathcal{Q}$*, where* $P = (\text{opt}, \mathcal{F}, \triangleleft)$ *and* (opt is min)*. Then, for every solution* S *of* $R_P(I)$*, it holds that* $v_I(S) \geq k_I \cdot H_I + T$*.*

Using Lemmas 13–16, we can now prove Theorem 12.

▶ **Theorem 12.** *Given an instance* $I = (E, \mathcal{I}, c, T)$ *of* EMB*, and* $P \in \mathcal{Q}$ *with* $P = (\text{opt}, \mathcal{F}, \triangleleft)$*, the following holds.*

1. *If there is a solution* S *for* $R_P(I)$ *such that* $v_I(S) = k_I \cdot H_I + T$*, then* I *is a "yes"-instance.*
2. *If* I *is a "yes"-instance then:* (i) $R_P(I)$ *has a solution, and* (ii) *every* $(1+\varepsilon_I)$*-approximate solution* S *for* $R_P(I)$ *satisfies* $v_I(S) = k_I \cdot H_I + T$*.*

Proof. We note that Property 1 follows directly from Lemma 13. For Property 2, assume that I is a "yes"-instance, then by Lemma 14, there is a solution D for $R_P(I)$ such that $v_I(D) = H_I \cdot k_I + T$. It remains to show Property 2. (ii). Let S be a $(1 + \varepsilon_I)$-approximate solution for $R_P(I)$. We distinguish between two cases.

1. (opt is max). Then, by Lemma 15,

$$0 \leq H_I \cdot k_I + T - v_I(S).$$

Moreover, since S is a $(1 + \varepsilon_I)$-approximate solution for $R_P(I)$, and D is a solution for $R_P(I)$,

$$H_I \cdot k_I + T - v_I(S) \leq H_I \cdot k_I + T - \frac{v_I(D)}{(1+\varepsilon_I)} = \frac{\varepsilon_I \cdot (H_I \cdot k_I + T)}{(1+\varepsilon_I)} \leq \varepsilon_I \cdot (H_I \cdot k_I + T)\,.$$

By the above, it follows that

$$|v_I(S) - (H_I \cdot k_I + T)| \leq \varepsilon_I \cdot (H_I \cdot k_I + T)\,.$$

2. (opt is min). This case is analogous to the above. By Lemma 16,

$$0 \leq v_I(S) - (H_I \cdot k_I + T).$$

Since S is a $(1+\varepsilon_I)$-approximate solution for $R_P(I)$, and D is a solution for $R_P(I)$,

$$v_I(S) - (H_I \cdot k_I + T) \leq (1+\varepsilon_I) \cdot v_I(D) - (H_I \cdot k_I + T) = \varepsilon_I \cdot (H_I \cdot k_I + T).$$

By the above,

$$|v_I(S) - (H_I \cdot k_I + T)| \leq \varepsilon_I \cdot (H_I \cdot k_I + T).$$

Thus in both cases it holds that,

$$|v_I(S) - (H_I \cdot k_I + T)| \leq \varepsilon_I \cdot (H_I \cdot k_I + T). \tag{8}$$

Let $n = |E| + 1$ and $m = c(E) + 1$. Then, by the selection of ε_I in (5),

$$\varepsilon_I \cdot (H_I \cdot k_I + T) = \frac{H_I \cdot k_I + T}{8 \cdot n \cdot (T+1) \cdot m} \leq \frac{2 \cdot m \cdot n + T}{8 \cdot n \cdot (T+1) \cdot m} < \frac{4 \cdot m \cdot n \cdot (T+1)}{8 \cdot n \cdot (T+1) \cdot m} = \frac{1}{2}. \tag{9}$$

The first inequality holds since $k_I \leq |E|$ and $H_I \leq 2 \cdot m$. Therefore, by (8) and (9),

$$|v_I(S) - (H_I \cdot k_I + T)| \leq \varepsilon_I \cdot (H_I \cdot k_I + T) < \frac{1}{2}. \tag{10}$$

Since $v_I(S) \in \mathbb{N}$ by Definition 11, it follows from (10) that $v_I(S) = H_I \cdot k_I + T$. This gives the statement of the theorem. ◀

4 Lower Bounds in the Standard Computational Model

Our hardness result in Section 2 shows that ORACLE EXACT MATROID BASIS (EMB) is hard, leading to the unconditional lower bounds for all non-trivial oracle MOL problems in Section 3. Nonetheless, these hardness results consider matroids with general membership oracles, and do not give a lower bound for matroids that can be efficiently encoded. This is particularly important, as in some settings oracle models differ from non-oracle models w.r.t complexity [6, 9]. Moreover, some matroids show up in problems that can be encoded efficiently. This includes partition matroids, graphic matroids, linear matroids, etc (see, e.g., [37] for a survey on various families of matroids). Next, we formally define an efficient encoding of matroids.

▶ **Definition 17.** *A function $f : \{0,1\}^* \to 2^{\mathbb{N}} \times 2^{2^{\mathbb{N}}}$ is called* matroid decoder *if for every $I \in \{0,1\}^*$ it holds that $f(I) = \left(E_{f(I)}, \mathcal{I}_{f(I)}\right)$ is a matroid, and the following holds.*
1. *There is an algorithm that given $I \in \{0,1\}^*$ returns $E_{f(I)}$ in time $|I|^{O(1)}$.*
2. *There is an algorithm that given $I \in \{0,1\}^*$ and $S \subseteq E_{f(I)}$ decides if $S \in \mathcal{I}_{f(I)}$ in time $|I|^{O(1)}$.*

There is a simple matroid decoder that can decode every matroid $(E, \mathcal{I})$ (such that $E \subseteq \mathbb{N}$), in which the encoding I explicitly lists $\mathcal{I}$. However, using such a matroid decoder, the encoding size of a matroid might be very large, up to $|I| = \Omega\left(2^{|E|}\right)$, while we often seek algorithms with running times polynomial in $|E|$. One way to overcome this difficulty is via the oracle model considered in previous sections. However, our results in this model may suggest that the hardness of EMB and MOL problems is due to the intrinsic hardness of the oracle model. Yet, there are families of matroids with very efficient encoding. For

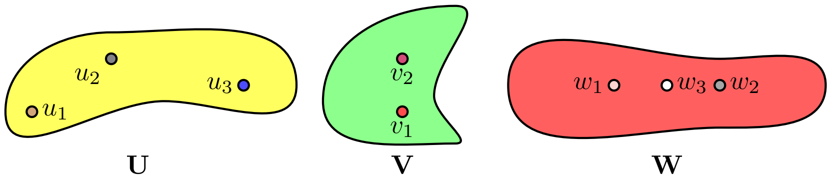


Figure 3 An example of a *partition matroid* $(E, \mathcal{I})$, which can be efficiently encoded. The ground set is $E = \{u_1, u_2, u_3, v_1, v_2, w_1, w_2, w_3\}$, partitioned into three sets: U, V, W. The independent sets are all subsets of E containing at most one element from U, V, and W; that is, $\mathcal{I} = \{S \subseteq E \mid \forall X \in \{U, V, W\} : |S \cap X| \leq 1\}$. A simple efficient encoding of $(E, \mathcal{I})$ is $I = (E, U, V, W)$. Membership can be decided efficiently given I, by checking the feasibility of a given set S w.r.t. U, V and W.

example, a *uniform matroid* $(E, \mathcal{I})$, where $\mathcal{I} = \{S \subseteq E \mid |S| \leq k\}$, can be efficiently encoded using $I = (E, k) \in \{0, 1\}^*$. Clearly, the time to decide membership of a given subset $S \subseteq E$ depends only on $|S|$ and k. Another example is given in Figure 3.

We start with a definition of an encoded variant of EMB. We technically define a different problem for every decoder f. The definition of the problem is analogous to the versions of EMB considered earlier in the paper, besides that the matroid is given via an arbitrary bit-string $I \in \{0, 1\}^*$, that a matroid decoder f decodes into a matroid $f(I)$.

f-DECODED EXACT MATROID BASIS (**f-decoded EMB**)
Decoder $f : \{0, 1\}^* \to 2^{\mathbb{N}} \times 2^{2^{\mathbb{N}}}$ is a matroid decoder.
Instance (I, c, T), where $I \in \{0, 1\}^*$, $c : E_{f(I)} \to \mathbb{N}$, $T \in \mathbb{N}$.
Solution A basis S of the matroid $f(I)$ such that $c(S) = T$.
Objective Decide if there is a solution.

▶ **Definition 18.** *The f-DECODED EXACT MATROID BASIS (f-decoded EMB) problem is defined as follows.*

- ***Decoder:*** $f : \{0, 1\}^* \to 2^{\mathbb{N}} \times 2^{2^{\mathbb{N}}}$ *is a matroid decoder.*
- ***Instance:*** (I, c, T)*, where* $I \in \{0, 1\}^*$, $c : E_{f(I)} \to \mathbb{N}$, $T \in \mathbb{N}$.
- ***Solution:*** *A basis S of the matroid $f(I)$ such that $c(S) = T$.*
- ***Objective:*** *Decide if there is a solution.*

As a simple example, consider the f_{u}-decoded EMB problem, for a specific matroid decoder f_{u} that decodes uniform matroids. The matroid decoder f_{u} interprets every $I \in \{0, 1\}^*$ as $I = (E, k)$ where E is a set (of numbers) and $k \in \mathbb{N}$, and returns the uniform matroid $f_{\mathsf{u}}(I) = (E, \mathcal{I})$ such that $\mathcal{I} = \{S \subseteq E \mid |S| \leq k\}$; clearly, f_{u} is a matroid decoder. Thus, an instance of f_{u}-decoded EMB is a tuple $U = ((E, k), c, T)$ and a solution of U is $S \subseteq E$ such that $|S| = k$ and $c(S) = T$; the goal, as before, is to decide if there is a solution. This problem is commonly known as the k-SUBSET SUM.

Recall that Theorem 3 asserts that oracle-EMB does not admit a pseudo-polynomial time algorithm. However, this does not rule out that hypothetically, for every matroid decoder f there is a pseudo-polynomial time algorithm for f-decoded EMB. The next result excludes this option.

▶ **Theorem 19.** *Assuming* P $\neq$ NP*, there is a matroid decoder f such that there is no algorithm for f-*decoded EMB *that for any f-*decoded EMB *instance $U = (I, c, T)$, where $n = |E_{f(I)}| + 1$ and $m = c\left(E_{f(I)}\right) + 1$, runs in time $(n \cdot (T + 2) \cdot m)^{O(1)}$.*

The proof of Theorem 19 is given towards the end of this section. Analogously to our hardness result for oracle MOL problems, we use Theorem 19 to give a hardness result for an encoded version of MOL problems. For every matroid decoder f and every $P \in \mathcal{P}$, we define a variant of the P-MATROID OPTIMIZATION WITH A LINEAR CONSTRAINT (P-MOL) problem in which the matroid is given via an arbitrary bit-string which a matroid decoder f decodes into a matroid. Formally, let $P \in \mathcal{P}$, where $P = (\text{opt}, \mathcal{F}, \triangleleft)$, be the parameters of the P-MOL problem. For a matroid decoder f, we define the f-decoded P-MOL problem as follows.

▶ **Definition 20.** *The f-DECODED P-MATROID OPTIMIZATION WITH A LINEAR CONSTRAINT (f-decoded P-MOL) problem is defined as follows.*

- ***Decoder:*** $f : \{0,1\}^* \to 2^{\mathbb{N}} \times 2^{2^{\mathbb{N}}}$ *is a matroid decoder.*
- ***Instance:*** (I, v, w, L)*, where* $I \in \{0,1\}^*$, $v : E_{f(I)} \to \mathbb{R}_{\geq 0}$, $w : E_{f(I)} \to \mathbb{R}_{\geq 0}$, $L \in \mathbb{R}_{\geq 0}$.
- ***Solution:*** *A basis S of the matroid $f(I)$ such that $c(S) = T$.*
- ***Objective:*** $\text{opt } v(S)$ *s.t.* $S \in \mathcal{F}(f(I))$, $w(S) \triangleleft L$.

For example, consider the encoded version of the P-MOL for $P = (\max, \text{IS}, \leq)$ with the matroid decoder f_{u} that decodes uniform matroids. An instance of the f_{u}-decoded P-MOL problem is a tuple $U = (I, v, w, L)$ where $I = (E, k)$ is a bit-string used for extracting the uniform matroid $f_{\mathsf{u}}(I) = (E, \mathcal{I})$ such that $\mathcal{I} = \{S \subseteq E \mid |S| \leq k\}$, v is the value function, w is the weight function, and L is the bound. A *solution* of U is $S \subseteq E$ such that $|S| \leq k$ and $w(S) \leq L$; the goal is to find a solution S of maximum value $v(S)$. This problem is widely known as KNAPSACK WITH CARDINALITY CONSTRAINT.

Recall that $\mathcal{Q} = \mathcal{P} \setminus \{(\min, \text{IS}, \leq)\}$ is the set of parameters for non-trivial MOL problems. Using the hardness of f-decoded, for some matroid decoder f (details on f are given towards the end of the section), we show the hardness of the f-decoded variant of all non-trivial MOL problems.

▶ **Theorem 21.** *Assuming* P $\neq$ NP*, for any $P \in \mathcal{Q}$ there is a matroid decoder f such that there is no* Fully PTAS *for f-decoded P*-MOL.

In the remainder of this section, we prove Theorem 19 and Theorem 21. The matroid decoder used in our proofs decodes a subclass of the Π-matroid family (see Section 2), in which the secret family Π consists of the solutions for a BOOLEAN SATISFIABILITY PROBLEM (SAT) instance.

In a SAT instance $A = (V, \bar{V}, \mathcal{C})$ with $n \in \mathbb{N}$ variables (in a slightly simplified notation), we are given a set $V = \{v_1, \ldots, v_n\}$ of variables, their negations $\bar{V} = \{\bar{v}_1, \ldots, \bar{v}_n\}$, and a set $\mathcal{C} \subseteq 2^{V \cup \bar{V}}$ of clauses. The goal is to decide if there is a set $S \subseteq [n]$ satisfying that for all $C \in \mathcal{C}$ there is $i \in [n]$ such that one of the following holds.

- $v_i \in C$ and $i \in S$.
- $\bar{v}_i \in C$ and $i \notin S$.

Such a set S is called a *solution* of A; let $\mathcal{S}(A)$ be the set of solutions of a SAT instance A. In addition, let $n(A) = n$ be the number of variables in the instance A. The family of SAT-matroids is the subfamily of Π-matroids where $\Pi = \mathcal{S}(A)$ for some SAT instance A (for the notation and definition of Π-matroids, see Definition 4). Specifically,

▶ **Definition 22.** *Let A be a* SAT *instance, $k \in [n(A)]$, and $\alpha \in \left[n(A)^2\right]$. Define the* **SAT-matroid** *on A, k, α as* $M_{n(A),k,\alpha}(\mathcal{S}(A)) = \left([n(A)], \mathcal{I}_{n(A),k,\alpha}(\mathcal{S}(A))\right)$.

We show below that SAT-matroids can be encoded efficiently. For every $I \in \{0,1\}^*$, we interpret I as $I = (A, k, \alpha)$, where A is a SAT instance, $k \in [n(A)]$, and $\alpha \in \left[n(A)^2\right]$; w.l.o.g., we may assume that every $I = (A, k, \alpha) \in \{0,1\}^*$ can be interpreted in that manner; moreover, we can assume that $n(A) \leq |A|$, where $|A|$ is the encoding size of A. The following definition defines a matroid decoder that decodes SAT-matroids from a bit-string.

▶ **Definition 23.** *Define the* SAT-decoder *as the function* $f_{\text{SAT}} : \{0,1\}^* \to 2^{\mathbb{N}} \times 2^{2^{\mathbb{N}}}$ *such that for all* $I = (A, k, \alpha) \in \{0,1\}^*$ *it holds that* $f_{\text{SAT}}(I) = M_{n(A),k,\alpha}(\mathcal{S}(A))$.

In the next result, we show that the SAT-decoder f_{SAT} is indeed a matroid decoder.

▶ **Lemma 24.** f_{SAT} *is a matroid decoder such that* $|I| = |A|^{O(1)}$ *for every* $I = (A, k, \alpha) \in \{0,1\}^*$.

Recall the sets $\mathcal{J}_{n,k}$, $\mathcal{K}_{n,k,\alpha}$, $\mathcal{L}_{n,k,\alpha}(\Pi)$ and $\mathcal{I}_{n,k,\alpha}(\Pi)$ were defined in Definition 4. We will use an algorithm for f-decoded EMB to obtain an algorithm for SAT. To this end, consider the following family of *structured* f_{SAT}-decoded EMB instances.

▶ **Definition 25.** *An* f_{SAT}-DECODED EMB *instance* (I, c, T)*, where* $I = (A, k, \alpha)$ *is called* structured *if* $\alpha \in \left[n(A)^2\right]$*,* $T = \alpha$*, and* $c : [n(A)] \to \mathbb{N}$ *such that for all* $i \in [n(A)]$ *it holds that* $c(i) = i$

The next observation immediately follows from the definition of structured f_{SAT}-decoded EMB instances and SAT-matroids.

▶ **Observation 26.** *for any structured* f_{SAT}-DECODED EMB *instance* $U = (I, c, T)$*, where* $I = (A, k, \alpha)$*, and* $S \subseteq [n(A)]$*, it holds that:* S *is a solution for* U *if and only if* $S \in \mathcal{S}(A)$*,* $|S| = k$*, and* $\mathsf{sum}(S) = c(S) = \alpha$.

We show that given a polynomial algorithm that decides structured f_{SAT}-decoded EMB instances, we can decide SAT. This result easily imply Theorem 19 as shown afterwards.

▶ **Lemma 27.** *Assuming* P $\neq$ NP*, there no algorithm that decides every* f_{SAT}*-decoded* EMB *structured instance* (I, c, T) *in time* $|I|^{O(1)}$.

From the above result, the hardness of the more general f_{SAT}-decoded EMB easily follows. As an immediate corollary, Theorem 28 gives the proof of Theorem 19.

▶ **Theorem 28.** *Assuming* P $\neq$ NP*, there is no algorithm for* f_{SAT}*-decoded* EMB *that runs in time* $(n \cdot (T+2) \cdot m)^{O(1)}$*, for any* f_{SAT}*-decoded* EMB *instance* $U = (I, c, T)$ *where* $n = \left|E_{f_{\text{SAT}}(I)}\right| + 1$ *and* $m = c\left(E_{f_{\text{SAT}}(I)}\right) + 1$.

Finally, we show that the variants of non-trivial matroid optimization with a linear constraint (MOL) problems, in which the decoding is performed by the SAT-decoder f_{SAT}, do not admit Fully PTAS under the standard assumption P $\neq$ NP. The proof is similar to the proof of Theorem 1 in Section 3. Lemma 29 directly gives the proof of Theorem 21.

▶ **Lemma 29.** *Assuming* P $\neq$ NP*, for any* $P \in \mathcal{Q}$ *there is no* Fully PTAS *for* f_{SAT}-decoded P-MOL.

5 Discussion

In this paper, we derive lower bounds for a family of matroid optimization problems with a linear constraint. We show that none of the (non-trivial) members of this family admits a Fully PTAS. In particular, this rules out a Fully PTAS for well studied problems such as BUDGETED MATROID INDEPENDENT SET, CONSTRAINED MINIMUM BASIS OF A MATROID, and KNAPSACK COVER WITH A MATROID. As BM and CMB admit an Efficient PTAS, our lower bounds resolve the complexity status of these problems, which has been open also for the generalization of BUDGETED MATROID INTERSECTION [12, 2, 17]. Our preliminary study shows that using the techniques of [26], we may be able to derive Efficient PTAS for *all* MOL problems. This would imply that Theorem 1 gives a tight lower bound for the entire MOL family. We leave the details for future work.

A key result of this paper is that EXACT MATROID BASIS (EMB) does not admit a pseudo-polynomial time algorithm, unlike the known special cases of k-SUBSET SUM and EMB on a linear matroid. Our proofs can be used to obtain lower bounds for other problems. For example, the hardness result for EMB can be adapted to yield lower bounds for related *parameterized* problems [22, 16]. Moreover, the proof of Theorem 1 can be modified to show that an Efficient PTAS for a non-trivial MOL problem with running time $f\left(\frac{1}{\varepsilon}\right) \cdot \text{poly}(n)$ must satisfy $f\left(\frac{1}{\varepsilon}\right) = \Omega\left(2^{\varepsilon^{-\frac{1}{4}}}\right)$. We leave these generalizations of our results to a later version of this paper.

Our results build on the Π-matroid family introduced in this paper. Such matroids exploit the interaction between a weight function and the underlying matroid constraint of the given problem. Aside from the implications of our results for previously studied problems, the new subclass of Π-matroids may enable to derive lower bounds for other problems. For example, consider the generalization of BM where the objective function is submodular and monotone. This is known as monotone submodular maximization with a knapsack and a matroid constraint [11]. Indeed, if the knapsack constraint is removed, there is a tight $\left(1 - \frac{1}{e}\right)$-approximation for the problem [4]. The same bound holds if we relax the matroid constraint [40]. However, the best known approximation for the problem with a knapsack and a matroid constraint is $\left(1 - \frac{1}{e} - \varepsilon\right)$ [11]. This setting resembles the status of MOL problems prior to our work, where removing either the linear or the matroid constraint induces a substantially easier problem. The potential use of Π-matroid variants to rule out a $\left(1 - \frac{1}{e}\right)$-approximation for the above problem remains an interesting open question.

In the context of solving *configuration* LPs for packing problems with a matroid constraint (e.g., [24, 15]), our lower bound implies that an FPTAS for an LP in this class cannot be obtained using the standard ellipsoid method.

We show unconditional hardness results in the oracle model (even if randomization is allowed), and give analogous lower bounds where the matroids are encoded as part of the input, assuming P $\neq$ NP. Our construction in Section 4 can be used to derive hardness results for other matroid problems in non-oracle models. Specifically, we can obtain in the standard computational model hardness results analogous to those in the oracle model of [28]. This includes a proof that it is NP-hard to decide if a given matroid is uniform, analogous to the unconditional hardness result in the oracle model of [28]. We leave these results for future work.

Our lower bounds for MOL problems on general matroids call for a more comprehensive study of these problems on restricted classes of matroids. We note the existence of Fully PTASs for MOL problems on some restricted matroid classes, e.g., BM on a laminar matroid or KCM on a partition matroid. The question whether (non-trivial) MOL problems admit

Fully PTAS on broader matroid classes, such as graphical matroids or linear matroids, remains open. In particular, it would be interesting to obtain a Fully PTAS for CONSTRAINED MINIMUM SPANNING TREE [26] and BM on a linear matroid – or show that one does not exist.

References

1. Kim Allan Andersen, Kurt Jörnsten, and Mikael Lind. On bicriterion minimal spanning trees: An approximation. *Computers & Operations Research*, 23(12):1171–1182, 1996.
2. André Berger, Vincenzo Bonifaci, Fabrizio Grandoni, and Guido Schäfer. Budgeted matching and budgeted matroid intersection via the gasoline puzzle. *Mathematical Programming*, 128(1):355–372, 2011.
3. Karl Bringmann and Vasileios Nakos. A fine-grained perspective on approximating subset sum and partition. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1797–1815. SIAM, 2021.
4. Gruia Calinescu, Chandra Chekuri, Martin Pal, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.
5. Paolo M. Camerini, Giulia Galbiati, and Francesco Maffioli. Random pseudo-polynomial algorithms for exact matroid problems. *Journal of Algorithms*, 13(2):258–273, 1992.
6. Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *Journal of the ACM (JACM)*, 51(4):557–594, 2004.
7. Alberto Caprara, Hans Kellerer, Ulrich Pferschy, and David Pisinger. Approximation algorithms for knapsack problems with cardinality constraints. *European Journal of Operational Research*, 123(2):333–345, 2000.
8. Venkatesan T Chakaravarthy, Anamitra Roy Choudhury, Sivaramakrishnan R Natarajan, and Sambuddha Roy. Knapsack cover subject to a matroid constraint. In *Proc. FSTTCS*, 2013.
9. Richard Chang, Benny Chor, Oded Goldreich, Juris Hartmanis, Johan Håstad, Desh Ranjan, and Pankaj Rohatgi. The random oracle hypothesis is false. *Journal of Computer and System Sciences*, 49(1):24–39, 1994.
10. Chandra Chekuri and Sanjeev Khanna. A polynomial time approximation scheme for the multiple knapsack problem. *SIAM Journal on Computing*, 35(3):713–728, 2005.
11. Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 575–584. IEEE, 2010.
12. Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Multi-budgeted matchings and matroid intersection via dependent rounding. In *Proc. SODA*, pages 1080–1097, 2011.
13. Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.
14. Marek Cygan, Marcin Mucha, Karol Wegrzycki, and MichaI WIodarczyk. On problems equivalent to (min,+)-convolution. *ACM Transactions on Algorithms (TALG)*, 15(1):1–25, 2019.
15. Ilan Doron-Arad, Ariel Kulik, and Hadas Shachnai. An AFPTAS for bin packing with partition matroid via a new method for LP rounding. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, 2023.
16. Ilan Doron-Arad, Ariel Kulik, and Hadas Shachnai. Budgeted matroid maximization: A parameterized viewpoint. *IPEC*, 2023.
17. Ilan Doron-Arad, Ariel Kulik, and Hadas Shachnai. An EPTAS for budgeted matching and budgeted matroid intersection via representative sets. In *Proc. ICALP*, pages 49:1–49:16, 2023.
18. Ilan Doron-Arad, Ariel Kulik, and Hadas Shachnai. An EPTAS for budgeted matroid independent set. In *Proc. SOSA*, pages 69–83, 2023.

19 Ilan Doron-Arad, Ariel Kulik, and Hadas Shachnai. An FPTAS for budgeted laminar matroid independent set. *Operations Research Letters*, 51:632–637, 2023.

20 Ilan Doron-Arad, Ariel Kulik, and Hadas Shachnai. Lower bounds for matroid optimization problems with a linear constraint. *arXiv preprint*, 2023. `arXiv:2307.07773`.

21 Jack Edmonds. Minimum partition of a matroid into independent subsets. *J. Res. Nat. Bur. Standards Sect. B*, 69:67–72, 1965.

22 Fedor V Fomin, Petr A Golovach, Tuukka Korhonen, Kirill Simonov, and Giannos Stamoulis. Fixed-parameter tractability of maximum colored path and beyond. In *Proc. SODA*, pages 3700–3712, 2023.

23 András Frank. *Connections in combinatorial optimization*, volume 38. Oxford University Press Oxford, 2011.

24 Kilian Grage, Klaus Jansen, and Kim-Manuel Klein. An EPTAS for machine scheduling with bag-constraints. In *The 31st ACM Symposium on Parallelism in Algorithms and Architectures*, pages 135–144, 2019.

25 Fabrizio Grandoni and Rico Zenklusen. Approximation schemes for multi-budgeted independence systems. In *Proc. ESA*, pages 536–548, 2010.

26 Refael Hassin and Asaf Levin. An efficient polynomial time approximation scheme for the constrained minimum spanning tree problem using matroid intersection. *SIAM Journal on Computing*, 33(2):261–268, 2004.

27 Sung-Pil Hong, Sung-Jin Chung, and Bum Hwan Park. A fully polynomial bicriteria approximation scheme for the constrained spanning tree problem. *Operations Research Letters*, 32(3):233–239, 2004.

28 Per M Jensen and Bernhard Korte. Complexity of matroid property algorithms. *SIAM Journal on Computing*, 11(1):184–190, 1982.

29 Richard M Karp, Eli Upfal, and Avi Wigderson. The complexity of parallel computation on matroids. In *26th Annual Symposium on Foundations of Computer Science (sfcs 1985)*, pages 541–550. IEEE, 1985.

30 Ariel Kulik and Hadas Shachnai. There is no EPTAS for two-dimensional knapsack. *Information Processing Letters*, 110(16):707–710, 2010.

31 Eugene L Lawler. Matroid intersection algorithms. *Mathematical programming*, 9(1):31–56, 1975.

32 Eugene L. Lawler. Fast approximation algorithms for knapsack problems. *Math. Oper. Res.*, 4(4):339–356, 1979.

33 László Lovász. The matroid matching problem. *Algebraic methods in graph theory*, 2:495–517, 1978.

34 James G Oxley. *Matroid theory*, volume 3. Oxford University Press, USA, 2006.

35 Rudi Pendavingh and Jorn Van der Pol. On the number of matroids compared to the number of sparse paving matroids. *arXiv preprint*, 2014. `arXiv:1411.0935`.

36 Ram Ravi and Michel X Goemans. The constrained minimum spanning tree problem. In *Algorithm Theory – SWAT'96: 5th Scandinavian Workshop on Algorithm Theory Reykjavík, Iceland, July 3–5, 1996 Proceedings 5*, pages 66–75. Springer, 1996.

37 Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer, 2003.

38 Prabhakant Sinha and Andris A Zoltners. The multiple-choice knapsack problem. *Operations Research*, 27(3):503–515, 1979.

39 José A Soto. A simple PTAS for weighted matroid matching on strongly base orderable matroids. *Discrete Applied Mathematics*, 164:406–412, 2014.

40 Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004.