The Complexity of the Matroid-Greedoid Partition Problem

Vera Asodi* and Christopher Umans†

Abstract

We show that the maximum matroid-greedoid partition problem is NP-hard to approximate to within $1/2 + \varepsilon$ for any $\varepsilon > 0$, which matches the trivial factor $1/2$ approximation algorithm. The main tool in our hardness of approximation result is an *extractor code* with polynomial rate, alphabet size and list size, together with an efficient algorithm for list-decoding. We show that the recent extractor construction of Guruswami, Umans and Vadhan [5] can be used to obtain a code with these properties.

We also show that the parameterized matroid-greedoid partition problem is fixed-parameter tractable.

1 Introduction

Matroid theory is a general framework that captures a number of classical combinatorial optimization problems. Many natural problems can be formulated as matroid problems, including minimum weight spanning tree, maximum matching and various connectivity problems, and can be solved by general algorithms for matroids. For example, the problem of finding a maximum matching in a bipartite graph is a special case of the maximum two matroid intersection problem. Edmonds [4] gave a polynomial time algorithm for this problem.

A greedoid is a generalization of a matroid that captures even more optimization problems. Recall that a matroid is a set system $M = (E, \mathcal{I})$, where the independent sets $\mathcal{I} \subseteq 2^E$ have the following properties.

(1) $\emptyset \in \mathcal{I}$.

(2) If $X \in \mathcal{I}$ and $Y \subseteq X$ then $Y \in \mathcal{I}$.

(3) If $X, Y \in \mathcal{I}$ and $|X| > |Y|$ then there is an element $x \in X \setminus Y$ such that $Y \cup \{x\} \in \mathcal{I}$.

[*] California Institute of Technology, 1200 E. California Blvd., Pasadena, CA 91125. Supported by a CMI postdoctoral fellowship. Email: vasodi@ist.caltech.edu

[†] California Institute of Technology, 1200 E. California Blvd., Pasadena, CA 91125. Supported by NSF CCF-0346991, BSF 2004329, a Sloan Research Fellowship, and an Okawa Foundation research grant. Email: umans@cs.caltech.edu

A greedoid is a set system $G = (E, \mathcal{F})$, where the feasible sets $\mathcal{F}$ have properties (1) and (3). Some well known examples of greedoids are rooted trees in a directed or undirected graph and ideals in a partially ordered set. For further examples see, e.g., [8, 7]. The maximum feasible set problem for greedoids is solvable by the greedy algorithm.

When studying algorithmic problems for matroids and greedoids, the matroid or the greedoid is given by a polynomial time oracle, that is, a procedure that, given a subset $A \subseteq E$, checks in time polynomial in $|E|$ whether A is independent in the case of a matroid or feasible in the case of a greedoid.

Whereas some of the matroid algorithms extend to greedoids, there are matroid problems that have polynomial time algorithms, but their generalization to greedoids is NP-hard. One example is the generalization of the two matroid intersection problem to the intersection of a matroid and a greedoid. Mielikäinen and Ukkonen [9] proved that the maximum matroid-greedoid intersection problem is NP-hard, and is even NP-hard to approximate within a factor of $|E|^{1-\varepsilon}$ for any fixed $\varepsilon > 0$. A closely related problem is the maximum matroid-greedoid partition problem.

Definition 1.1 *Let $M = (E, \mathcal{I})$ be a matroid and $G = (E, \mathcal{F})$ a greedoid. A partition is a set $Z \subseteq E$ for which there is a partition $Z = X \cup Y$, $X \cap Y = \emptyset$, such that $X \in \mathcal{I}$ and $Y \in \mathcal{F}$. The maximum matroid-greedoid partition problem is to find the maximum cardinality partition Z.*

The corresponding problem for two matroids – even k matroids for any k – is reducible to the two matroid intersection problem, and therefore is in P. The matroid-greedoid versions of the two problems are related as well, and, as Mielikäinen and Ukkonen mentioned in [9], their NP-hardness result for the matroid-greedoid intersection problem also proves that the matroid-greedoid partition problem is NP-hard. However, the hardness of approximation does not carry over to the partition problem.

In this paper we study the maximum matroid-greedoid partition problem and prove a tight inapproximability result for it. First, observe that the problem can be easily approximated within a factor of $\frac{1}{2}$, by finding a maximum independent set $X \in \mathcal{I}$, a maximum feasible set $Y \in \mathcal{F}$, and setting Z to be the larger set among X and Y. Our main result in this paper is that this trivial algorithm is essentially the best one can do. Namely, we prove that it is NP-hard to approximate the maximum matroid-greedoid partition within a factor of $\frac{1}{2} + \varepsilon$ for any constant $\varepsilon > 0$.

Besides the hardness of approximation result, we also study the parameterized version of the matroid-greedoid partition problem, that is, when the objective is to find a partition of size k for a given parameter k, if one exists. Note that if the maximum partition is of size greater than k, then a partition of size k exists. In [9], Mielikäinen and Ukkonen showed that the parameterized matroid-greedoid intersection problem is W[P]-hard, and raised the question of the fixed-parameter tractability of the partition problem. We show that, unlike the intersection problem, the parameterized matroid-greedoid partition problem is fixed-parameter tractable, i.e. it can be solved in time $O(f(k)n^c)$ where f is an arbitrary function and c is a constant independent of k.

1.1 Motivation

Matroids, greedoids, and their associated optimization problems constitute an approach to combinatorial problems whose goal is generality and uniformity. This is a good approach from the algorithmic perspective because it is better to have a single generic algorithm than many specific algorithms for individual problems. But in seeking greater generality one may discard combinatorial structure that turns out to be algorithmically beneficial. The results in this paper identify limits on the benefits of generalization, with respect to matroid-greedoid partition problems.

If the maximum matroid-greedoid partition problem could be approximated arbitrarily well, then it would be a powerful algorithmic tool, despite being NP-hard. Our results show that this is definitely *not* the case, and indeed by taking advantage of the generality afforded by formulating problems as matroid-greedoid partition problems, one gives up any hope of obtaining non-trivial approximation algorithms. Thus this type of generalization is not useful from the perspective of approximation.

On the other hand, our results show that this type of generalization *is* useful from the perspective of fixed-parameter algorithms, since we show that the matroid-greedoid partition problem is fixed-parameter tractable.

1.2 Techniques

Our main tool is an error-correcting code with strong list-decodability properties. Using just this object, we are able to produce tight inapproximability results "from scratch." In particular, we do not rely on Probabilistically Checkable Proof (PCP) machinery anywhere in the proof. This stands in contrast to the vast majority of non-trivial hardness of approximation results, which use PCPs (either directly, or indirectly by giving a gap-preserving reduction from problem whose inapproximability is proven using PCPs).

Our result is one of relatively few known applications of the strong list-decodability properties of extractor codes (see the discussion following Theorem 3.1 for why we really *need* precisely this type of strong list-decodability). We suspect that the codes that we construct in Section 4 (which are now possible using the constructions in [5]) may find additional applications in hardness of approximation and even complexity as a whole, as they achieve a very natural and useful parameter setting: polynomially large alphabet, blocklength, and list-size, coupled with polynomial-time list-decoding in the "extractor code" regime.

Outline The rest of this paper is organized as follows. For completeness, we present in Section 2 the reduction of Mielikäinen and Ukkonen [9] that proves that the maximum matroid-greedoid partition problem is NP-hard. In Section 3 we build significantly upon this reduction in order to prove our hardness of approximation result. In Section 4 we describe the error-correcting codes which are the key ingredient in our reduction. Finally, in Section 5 we study the fixed parameter version of the problem.

2 NP-hardness

Mielikäinen and Ukkonen [9] proved that the matroid-greedoid intersection problem is NP-hard, and mentioned that their proof applies to the matroid-greedoid partition problem as well. We give their proof here, to illustrate some of the ideas that appear in the reduction of our main result.

Theorem 2.1 ([9]) *The maximum matroid-greedoid partition problem is NP-hard.*

Proof: We prove the theorem by a reduction from SAT. Let Φ be a Boolean formula in variables $x_1, x_2, \ldots, x_n$. We construct a matroid $M = (E, \mathcal{I})$ and a greedoid $G = (E, \mathcal{F})$ as follows. Let $E = \{0,1\} \times [n]$ be the set of the elements of the matroid and the greedoid. The elements correspond to the truth values that can be assigned to each variable. The independent sets of the matroid M are all the sets that contain at most one of the elements $(0,i),(1,i)$ for each i, that is,

$$\mathcal{I} = \{I \subseteq E : \forall 1 \leq i \leq n, |I \cap (\{0,1\} \times \{i\})| \leq 1\}.$$

Clearly, M is a matroid, and has a polynomial time oracle.

G has two types of feasible sets. The first type, $\mathcal{A}$, consists of all sets of cardinality at most n that do not contain $(0,n)$ and $(1,n)$, that is,

$$\mathcal{A} = \{F \subseteq \{0,1\} \times [n-1] : |F| \leq n\}.$$

The second type, $\mathcal{B}$, consists of all the sets that contain exactly one of the elements $(0,i),(1,i)$ for each i, and correspond to satisfying assignments. Again, it is easy to see that G has a polynomial time oracle. We now show that G is a greedoid. Clearly $\emptyset \in \mathcal{F}$. Suppose $X, Y \in \mathcal{F}$, $|X| > |Y|$. Note that since $|Y| < |X| \leq n$, $Y \in \mathcal{A}$. If $X \in \mathcal{A}$ then for any $x \in X \setminus Y$, $Y \cup \{x\} \in \mathcal{A}$. Suppose now that $X \in \mathcal{B}$. If there is an element $(\sigma, i) \in X \setminus Y$ with $i \neq n$ then $Y \cup \{(\sigma, i)\} \in \mathcal{A}$. Otherwise, $Y = X \cap (\{0,1\} \times [n-1])$, and $X \setminus Y$ consists of a single element (σ, n) where $\sigma \in \{0,1\}$. Thus $Y \cup \{(\sigma, n)\} = X \in \mathcal{B}$.

Now, if Φ has a satisfying assignment let $Y \subseteq E$ be a set corresponding to a satisfying assignment and let $X = E \setminus Y$. Since $X \in \mathcal{I}$ and $Y \in \mathcal{F}$ we get a solution of size $2n$ to the partition problem.

If Φ has no satisfying assignment then $\mathcal{F} = \mathcal{A}$. Thus, for any solution $Z = X \cup Y$ to the partition problem, Z contains at most one of the elements $(0,n)$ and $(1,n)$, and hence $|Z| \leq 2n - 1$. □

3 Hardness of Approximation

Theorem 3.1 *It is NP-hard to approximate the maximum matroid-greedoid partition within a factor of $\frac{1}{2}+\varepsilon$ for any constant $\varepsilon > 0$. Specifically, it is NP hard to distinguish between the case the maximum partition contains all the elements of the matroid and the greedoid, and the case it contains at most $\frac{1}{2} + \varepsilon$ fraction of them.*

We prove the theorem by a reduction from SAT that is based on the NP-hardness proof from Section 2. Before getting into the details of the proof, let us sketch the general idea of it. First, note that the reduction must produce a matroid-greedoid pair for which the maximal independent sets of the matroid and the maximal feasible sets of the greedoid have the same cardinality, otherwise, the trivial approximation algorithm would give a factor greater than $\frac{1}{2}$. The main idea in the reduction in Section 2 is that if there is no satisfying assignment, then all the feasible sets of the greedoid are contained in the first $n-1$ blocks (in our discussion, the i-th "block" is the the subset $\{0,1\} \times \{i\} \subseteq E$), and since the independent sets of the matroid contain at most one element from each block, no partition contains both elements of the n-th block. Therefore in that reduction, in the positive case (when there is a satisfying assignment), the maximum partition is of size $2n$, and in the negative case it is of size $2n-1$. In order to prove hardness of approximation we need a larger gap between the two cases.

A natural approach is to modify the reduction so that in the negative case, the feasible sets of the greedoid are contained in the first r blocks, where r is as small as possible. As we will see below, we will eventually need the blocks to be larger than 2; and, just as the reduction in Section 2 used the sets in $\mathcal{B}$ to "encode" satisfying assignments, our modified reduction will view some of feasible sets of the greedoid as encodings of satisfying assignments. The general setup can thus be described in terms of an error correcting code $C : \{0,1\}^n \to \Sigma^N$ (where n is the number of variables in the original instance of SAT). The universe for both the matroid and greedoid will be $\Sigma \times [N]$, and as above, we will refer to the subset $\Sigma \times \{i\}$ as the i-th "block". In our reduction the feasible sets of the greedoid are of two types, $\mathcal{A}$ and $\mathcal{B}$ (just as in the reduction in Section 2): the feasible sets in $\mathcal{A}$ will be subsets of the first r blocks, where r is as small as possible. The feasible sets in $\mathcal{B}$ will have the following properties. Each $F \in \mathcal{B}$ contains elements from every one of the first k blocks, for some $r < k \leq N$, and F corresponds to an encoding of a satisfying assignment. By this we mean that F's intersection with each block is interpreted as giving information about the codeword's value in the corresponding coordinate, and there is a codeword c consistent with this information for which $C^{-1}(c)$ is a satisfying assignment. (Note that this scheme actually allows us to reduce from *any* NP problem, by having the sets in $\mathcal{B}$ correspond to encodings of NP witnesses for that problem).

What properties are needed from C? We first argue that the alphabet Σ must have cardinality larger than 2. For suppose that the code C is binary, i.e. $\Sigma = \{0,1\}$, and that the matroid is defined as in Section 2, that is, an independent set contains at most one element from each block. Then, for a set $F \in \mathcal{B}$, the blocks from which F contains exactly one element should be thought of as "known" coordinates of the codeword and the rest as erasures, and we therefore need a code with efficient list decoding for erasures (and no errors). However, since the maximal feasible sets are of size N, we must have $r \geq \frac{N}{2}$, regardless of the code we use (since in $\mathcal{A}$ we must fit N elements into r blocks of size 2 each). But then, in the negative case, the size of the maximum partition will be at least $\frac{3}{2}N$ (since we can occupy one element from each block with a independent set of the matroid, and at least r additional elements with a feasible set of the greedoid contained entirely in the first r blocks) and hence the hardness result we will get is for a factor of at least $\frac{3}{4}$.

To get the hardness result of $\frac{1}{2}+\varepsilon$ under this general reduction scheme, we need an error correcting code with a larger alphabet Σ. This enables us to define the matroid in a more general way: instead of containing one element from each block, an independent set may contain *almost all* the elements from each of the first $r \approx \frac{N}{2}$ blocks, and only a small fraction from each of the rest. As before, the greedoid will include as feasible sets all sets up to a certain size that are contained entirely within the first r blocks. Now in the negative case, a partition might again include all the elements from the first r blocks, *but* it can only contain a small fraction from each of the rest (coming just from an independent set of the matroid), thus getting the desired gap – between a partition of size about $N/2$ times the size of the blocks, and a partition of size N times the size of the blocks.

However, if the alphabet is larger, we need to allow the feasible sets of the greedoid to contain many elements in each block – and when trying to interpret this as information about a codeword we can no longer assume that in each coordinate of the codeword we either know the exact symbol or have an erasure. Instead, a potential feasible set will be interpreted as giving in each coordinate a *set* of possible symbols, from which we need to recover a list of codewords (and we declare such a set feasible iff the list of codewords includes one that encodes a satisfying assignment). Moreover, since the greedoid should have a polynomial time oracle, we need to be able to recover this list of codewords efficiently; i.e. we need C to have an efficient list-decoding procedure, in the regime where the information about coordinate is a set of possible values that may be nearly the size of the entire alphabet. This is a very strong demand, but it can be achieved by the extractor codes that are described in Section 4.

Given a code with these properties, we can define the greedoid, as before, in terms of two classes of feasible sets: $\mathcal{A}$, which are all subsets up to a certain size of the first r blocks, where r is roughly $\frac{N}{2}$, and $\mathcal{B}$, which are sets that contain elements from each one of the first k blocks, for $r < k \leq N$, and that encode satisfying assignments. The exact choice of r will guarantee that the sets in $\mathcal{B}$ correspond to decoding problems that have polynomial sized lists and can be solved efficiently. The exact definitions should also ensure that F is a greedoid. Recall that the crucial point in the proof that F is a greedoid in Theorem 2.1 is that, if $X \in \mathcal{B}$, $Y \in \mathcal{A}$, and $|X| > |Y|$, then there is either an element $x \in X \setminus Y$ that belongs to one of the first $n-1$ blocks, and thus $Y \cup \{x\} \in \mathcal{A}$, or X and Y agree on the first $n-1$ blocks, and hence adding the element from the n-th block in X to Y will make it a satisfying assignment (and equal to X), and therefore in $\mathcal{B}$. We will make a similar argument in our proof (in case (3) below).

In the rest of this section we give the details of the proof of Theorem 3.1 assuming that we have error correcting codes with the required properties, and in Section 4 we prove the existence of such codes.

Proof: Fix a constant $\varepsilon > 0$. Let Φ be a Boolean formula in variables $x_1, x_2, \ldots, x_n$. We construct a matroid $M = (E, \mathcal{I})$ and a greedoid $G = (E, \mathcal{F})$ as follows. Let $C : \{0,1\}^n \to \Sigma^N$ be a code with the following properties:

- $|\Sigma| = q$ is polynomial in n.
- N is polynomial in n.
- There exist constants $\alpha, \beta > 0$ with $\alpha + \beta - \alpha\beta < \varepsilon$, such that for any sets $S_1, S_2, \ldots, S_N \subseteq \Sigma$ such that at least αN of them are of size at most $(1-\beta)q$, $C^{-1}(S_1 \times S_2 \times \ldots \times S_N)$ is of size polynomial in N, and can be computed in polynomial time.

In Section 4, we construct a code C with exactly these properties (specifically, we apply Corollary 4.5 with $m = \Theta(\log n)$).

Let $E = \Sigma \times [N]$ be the set of the elements of M and G. Define the independent sets of M by

$$\begin{aligned}\mathcal{I} = \{I \subseteq E : \ \forall \ \ 1 \leq i \leq r, |I \cap (\Sigma \times \{i\})| \leq (1-\delta)q, \\ \forall \ \ r < i \leq N, |I \cap (\Sigma \times \{i\})| \leq \beta q\},\end{aligned}$$

where we define $r = \left(\frac{1}{2} + \gamma\right) N$, $\gamma = \alpha + \frac{\beta}{2(1-\beta)}$ and $\delta = \frac{\beta + 2\gamma - 2\beta\gamma}{1+2\gamma}$. By this choice of δ, every independent set is of cardinality at most $(1-\delta)qr + \beta q(N-r) = \frac{qN}{2} = \frac{|E|}{2}$. The choice of γ is needed for the greedoid. Clearly, M is a matroid and has a polynomial time oracle.

We construct the feasible sets of G as follows. For a set $F \subseteq E$ define the following:

- $T(F) = \{1 \leq i \leq N : F \cap (\Sigma \times \{i\}) \neq \emptyset\}$
- $C^{-1}(F) = C^{-1}(S_1 \times S_2 \times \ldots \times S_N)$, where $S_i = \{\sigma \in \Sigma : (\sigma, i) \in F\}$ for all $i \in T(F)$, and $S_i = \Sigma$ for all $i \in [N] \setminus T(F)$. We say that F satisfies Φ if $C^{-1}(F)$ contains a satisfying assignment.

Let $\mathcal{F} = \mathcal{A} \cup \mathcal{B}$, where $\mathcal{A}$ and $\mathcal{B}$ are defined as follows. Let

$$\mathcal{A} = \left\{F \subseteq \Sigma \times [r] : |F| \leq \frac{qN}{2}\right\}.$$

For $r < k \leq N$ let

$$\mathcal{B}_k = \left\{F \subseteq E : |F| \leq \frac{qN}{2}, T(F) = [k], F \text{ satisfies } \Phi\right\}$$

and let

$$\mathcal{B} = \bigcup_{k=r+1}^{N} \mathcal{B}_k.$$

By the choice of γ, for every set $F \subseteq E$ such that $|F| \leq \frac{qN}{2}$ and $T(F) = [k]$ with $k > r$, there are at least αN indices $i \in [k]$ for which $|F \cap (\Sigma \times \{i\})| \leq (1-\beta)q$. Hence, $C^{-1}(F)$ can be computed in polynomial time, and thus G has a polynomial time oracle.

We now show that G is a greedoid. Clearly $\emptyset \in \mathcal{F}$. Suppose $X, Y \in \mathcal{F}$, $|X| > |Y|$. We consider four cases:

1. $X, Y \in \mathcal{A}$:
   For any $(\sigma, i) \in X \setminus Y$, $Y \cup \{(\sigma, i)\} \in \mathcal{A}$.
2. $X \in \mathcal{A}$ and $Y \in \mathcal{B}$:
   For any $(\sigma, i) \in X \setminus Y$, $Y \cup \{(\sigma, i)\} \in \mathcal{B}$.
3. $X \in \mathcal{B}$ and $Y \in \mathcal{A}$:
   If there exists an element $(\sigma, i) \in X \setminus Y$ with $i \leq r$ then $Y \cup \{(\sigma, i)\} \in \mathcal{A}$. Otherwise,

$$X \cap (\Sigma \times [r]) \subseteq Y, \tag{1}$$

and in particular, $T(Y) = [r]$. Since $X \in \mathcal{B}$, $T(X) = [k]$ for some $k > r$ and there exists a satisfying assignment w such that $(C(w)_i, i) \in X$ for all $1 \leq i \leq k$. By (1), $(C(w)_i, i) \in Y$ for all $1 \leq i \leq r$. Thus, $(C(w)_{r+1}, r+1) \in X \setminus Y$ and $Y \cup \{(C(w)_{r+1}, r+1)\}$ satisfies Φ. Therefore $Y \cup \{(C(w)_{r+1}, r+1)\} \in \mathcal{B}_{r+1}$.

4. $X, Y \in \mathcal{B}$:
Suppose $X \in \mathcal{B}_k$ and $Y \in \mathcal{B}_\ell$. If there exists an element $(\sigma, i) \in X \setminus Y$ with $i \leq \ell$ then $Y \cup \{(\sigma, i)\} \in \mathcal{B}_\ell$. Otherwise,
$$X \cap (\Sigma \times [\ell]) \subseteq Y, \tag{2}$$
and in particular $k > \ell$. Since $X \in \mathcal{B}$ there exists a satisfying assignment w such that $(C(w)_i, i) \in X$ for all $1 \leq i \leq k$. By (2), $(C(w)_i, i) \in Y$ for all $1 \leq i \leq \ell$. Thus, $(C(w)_{\ell+1}, \ell+1) \in X \setminus Y$ and $Y \cup \{(C(w)_{\ell+1}, \ell+1)\}$ satisfies Φ. Therefore $Y \cup \{(C(w)_{\ell+1}, \ell+1)\} \in \mathcal{B}_{\ell+1}$.

Now, if Φ has a satisfying assignment w, let $Y \subseteq E$ be any set such that:

(i) $|Y \cap (\Sigma \times \{i\})| = \delta q$ for all $1 \leq i \leq r$

(ii) $|Y \cap (\Sigma \times \{i\})| = (1 - \beta)q$ for all $r < i \leq N$

(iii) $(C(w)_i, i) \in Y$ for all $1 \leq i \leq N$

and let $X = E \setminus Y$. Since $X \in \mathcal{I}$ and $Y \in \mathcal{F}$ we get a solution of size qN to the partition problem.

If Φ has no satisfying assignment then $\mathcal{F} = \mathcal{A}$. Thus, for any feasible set $Y \in \mathcal{F}$, $Y \subseteq \Sigma \times [r]$. Therefore, for any solution $Z = X \cup Y$ to the partition problem,
$$|Z| \leq qr + \beta q(N - r) = \left(\frac{1}{2} + \alpha + \beta - \alpha\beta\right) qN < \left(\frac{1}{2} + \varepsilon\right) qN$$
Therefore, it is NP-hard to approximate the matroid-greedoid partition problem within a factor of $\frac{1}{2} + \varepsilon$. □

4 Error-correcting codes for the reduction

We need error-correcting codes over a polynomially large alphabet Σ with polynomial blocklength, and with the following very strong list-decoding property: given subsets of Σ of size $(1-\beta)|\Sigma|$ for an α fraction of the coordinates (and viewing the other coordinates as erasures), there should be only polynomially many codewords whose coordinates fall into the associated subsets. We need this to hold for arbitrarily small α and β.

It is shown in [13] that “ordinary” codes (i.e. Reed-Solomon codes) cannot satisfy this requirement, but that these strong demands can be met by viewing *randomness extractors* as codes.

Recall that the statistical difference between two distributitions D, D' over a domain A is defined by $d(D, D') = \frac{1}{2}\sum_{x \in A} |D(x) - D'(x)|$. We say a distribution D over $\{0,1\}^t$ is ε-close to uniform if $d(D, U_t) < \varepsilon$, where U_t is the uniform distribution over $\{0,1\}^t$.

Definition 4.1 *A* (k,ε)-extractor *is a function* $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ *with the following property: for every random variable* $\mathbf{X}$ *distributed over* $\{0,1\}^n$ *with min-entropy at least* k*, the distribution* $E(\mathbf{X}, \mathbf{U_t})$ *is* ε*-close to uniform.*

If $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ is a (k,ε)-extractor, then the associated *extractor code* over $\Sigma = \{0,1\}^m$ encodes $x \in \{0,1\}^n$ as $(E(x,y))_{y\in\{0,1\}^t}$, and has list-size 2^k, provided $\alpha + \beta - \alpha\beta < \varepsilon$. Ta-Shma and Zuckerman [13] also showed that certain extractor constructions (namely [14] and [11]) have *efficient decoding*, which means that the list can be recovered from the sets describing the "received word" in polynomial time in the size of the list.

Since we require a polynomial-size alphabet, blocklength and list-size, we need extractors with $k, t, m = O(\log n)$. This is at the extreme low end of the typical values for the k and m parameters, and while a number of constructions achieve these parameters, we are not aware of any which admit efficient decoding.

In this section we show that by combining the recent construction of Guruswami, Umans and Vadhan [5] with a family of pairwise-independent hash functions, we can obtain the desired parameters and also have efficient decoding.

In this following description, we will work directly with objects defined in terms of their list-decoding properties (as advocated by Vadhan [15]), to avoid having to define a variety of other pseudorandom objects. However, we note that the list-decoding properties we describe imply that the object in Theorem 4.2 is a *condenser*, and the ones in Theorems 4.3 and 4.4 are *dispersers* (see, e.g., [10] for definitions and known constructions). The reason our final object is a disperser rather than an extractor is that we only need our codes to handle erasures, as opposed to errors[1].

Our final object, which is described in Theorem 4.4, is the composition of two intermediate ones, described in Theorems 4.2 and 4.3. The first is a variant of the main construction in [5] (we repeat the short proof from [5] in order to show that the set $LIST_C(T,\varepsilon)$ can actually be efficiently computed):

Theorem 4.2 *For all positive integers* $\ell \le n$*, and* $\varepsilon > 0$*, there is an explicit function* $C : \mathbb{F}_q^n \times \mathbb{F}_q \to \mathbb{F}_q^\ell$*, where* q *is the smallest power of* 4 *larger than* $(n^2/\varepsilon)^2$*, with the following property: for every* $T \subseteq \mathbb{F}_q^\ell$ *of cardinality less than* $q^{\ell/2}$*, the set*

$$LIST_C(T,\varepsilon) = \left\{ x : \Pr_{y\in F_q}[C(x,y) \in T] \ge \varepsilon \right\}$$

has cardinality at most $q^{\ell/2}$ *and can be computed in time polynomial in* $q^{\ell/2}$ *for such sets* T*.*

Proof: Set $h = q^{1/2}$. Pick a degree n polynomial $E(Y)$ that is irreducible over $\mathbb{F}_q$ (the field $\mathbb{F}_q$ and $E(Y)$ can be constructed deterministically in time polynomial in n and $\log q$ since the characteristic is fixed [12]). For an arbitrary polynomial $f(Y)$ of Y, denote by $f_i(Y)$ the polynomial $f(Y)^{h^i}$ mod $E(Y)$. Identify $\mathbb{F}_q^n$ with univariate polynomials of degree at most $n-1$, and define

$$C(f,y) = (f_0(y), \cdots, f_{\ell-1}(y)).$$

[1] Note that although list-decoding from erasures is trivial for linear codes, it is highly non-trivial in the setting in which the non-erased symbols are only known to lie in a fairly large set.

Now we give the algorithm for computing $LIST_C(T, \varepsilon)$. Fix a subset $T \subseteq F_q^\ell$ of cardinality at most $h^\ell - 1$. Find a polynomial $Q(Y_0, \cdots, Y_{\ell-1})$ over $\mathbb{F}_q$ with individual degrees at most $h-1$ that vanishes on T. This can be done in time polynomial in $\log q$ and h^ℓ by simply solving a system of homogeneous linear equations.

Now, every $f \in LIST_C(T, \varepsilon)$ satisfies

$$\Pr_{y \in F_q}[Q(f_0(y), f_1(y), \ldots, f_{\ell-1}(y)) = 0] \geq \varepsilon,$$

and because the degree of the univariate polynomial $Q(f_0(Y), \ldots, f_{\ell-1}(Y))$ is at most $h\ell n$, and $\varepsilon q > h\ell n$, this polynomial must be the zero polynomial. Therefore $f(Y)$ viewed as an element of the extension field $\mathbb{F}_q[Y]/E(Y)$ is a root of the polynomial

$$Q^*(Z) \stackrel{\text{def}}{=} Q(Z, Z^h, Z^{h^2}, \ldots, Z^{h^{\ell-1}}).$$

There are at most $\deg(Q^*) \leq (1 + h + h^2 + \cdots + h^{\ell-1})(h-1) = h^\ell - 1 < q^{\ell/2}$ such roots, and they can be found in time polynomial in $n \log q$ (the log of the size of the extension field) and the degree h^ℓ. Every element of $LIST_C(T, \varepsilon)$ is a root, and given the set of roots, it is easy to check for each one whether or not it is in $LIST_C(T, \varepsilon)$. □

Our second object is a pairwise independent hash family which by [6] is an extractor with large seed length, although the theorem statement below only implies that it is a *disperser*:

Theorem 4.3 *For all positive integers $m \leq n$, and $\varepsilon > 0$, there is an explicit function $D : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$, with $d = O(n)$, with the following property: for every $T \subseteq \{0,1\}^d \times \{0,1\}^m$ of cardinality at most $(1-\varepsilon)2^{d+m}$, the set*

$$LIST_D(T) = \{x : \forall y \ \ (y, D(x,y)) \in T\}$$

has cardinality at most $2^m/\varepsilon^2$ and can be computed in time polynomial in 2^n and 2^d for such sets T.

Proof: We use a pairwise independent family $\mathcal{H}$ of hash functions from n bits to m bits. Standard constructions have $\log |\mathcal{H}| = O(n)$; the function D is given by $D(x, h \in \mathcal{H}) = h(x)$. The bound on the cardinality of $LIST_D(T)$ follows directly from the Leftover Hash Lemma [6]. Efficient computation of $LIST_D(T)$ is trivial: simply compute $D(x,y)$ for each $x \in \{0,1\}^n$ and $y \in \{0,1\}^d$, each time checking membership in T. □

Our final object comes from composing the previous two:

Theorem 4.4 *For all positive integers $m \leq n$, and $\varepsilon > 0$, there is an explicit function $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$, with $t = O(m + \log n + \log(1/\varepsilon))$, with the following property: for every $T \subseteq \{0,1\}^t \times \{0,1\}^m$ of cardinality at most $(1-\varepsilon)2^{t+m}$, the set*

$$LIST_E(T) = \{x : \forall v \ \ (v, E(x,v)) \in T\}$$

has cardinality at most $O(2^m n^8/\varepsilon^6)$ and can be computed in time polynomial in 2^m and $\frac{n}{\varepsilon}$ for such sets T.

Proof: Set $\varepsilon' = \varepsilon/2$. Set q to be the smallest power of 4 larger than $(n^2/\varepsilon')^2$ as in Theorem 4.2, and set ℓ to be the smallest integer such that $q^{\ell/2-1} > 2^{m+2}/\varepsilon'^2$. Let $C : \mathbb{F}_q^n \times \mathbb{F}_q \to \mathbb{F}_q^\ell$ be the function from Theorem 4.2 (with its parameter ε set to ε').

Set $n' = 2\log(q2^m/\varepsilon'^2)$ and let $D : \{0,1\}^{n'} \times \{0,1\}^d \to \{0,1\}^m$ be the function from Theorem 4.3.

View $\{0,1\}^n$ as sitting inside $\mathbb{F}_q^n$, and define $E : \{0,1\}^n \times (\mathbb{F}_q \times \{0,1\}^d) \to \{0,1\}^m$ as follows: $E(x;y,z) \stackrel{\text{def}}{=} D(C(x,y),z)$. We have $t = \log q + d = O(m + \log n + \log(1/\varepsilon))$ as promised.

Now we give the algorithm for computing $LIST_E(T)$. Fix a subset $T \subseteq (F_q \times \{0,1\}^d) \times \{0,1\}^m$ of cardinality at most $(1-\varepsilon)q2^{d+m}$. Define $T_y = \{(z,w) : (y,z,w) \in T\}$. By an averaging argument, at least an ε' fraction of $y \in F_q$ have $|T_y| \le (1-\varepsilon')2^{d+m}$. Let $S \subseteq \mathbb{F}_q$ be the set of such y. The key observation is that

$$LIST_E(T) \subseteq LIST_C\left(\bigcup_{y\in S} LIST_D(T_y), \varepsilon'\right).$$

To see why, consider an x for which $\forall y,z\ \ (y,z,D(C(x,y),z)) \in T$. Then for each $y \in F_q$, $C(x,y) \in LIST_D(T_y)$ by definition. Therefore for any $S \subseteq \mathbb{F}_q$,

$$\Pr_{y\in\mathbb{F}_q}\left[C(x,y) \in \bigcup_{y\in S} LIST_D(T_y)\right] \ge \frac{|S|}{q},$$

and then the claim follows from $|S| \ge \varepsilon' q$ and the definition of $LIST_C$.

We can find $LIST_D(T_y, \varepsilon')$ for each $y \in S$ in time polynomial in $2^{n'}$ and 2^d, and note that Theorem 4.3 guarantees that each such set has cardinality at most $2^m/\varepsilon'^2$. Therefore, the argument to $LIST_C$ is a set of cardinality at most $q2^m/\varepsilon'^2 < q^{\ell/2}$. Therefore we can compute $LIST_C$ in time polynomial in $q^{\ell/2}$ by Theorem 4.2, and the overall size of $LIST_E(T)$ is then at most $q^{\ell/2} < q^2 2^m/\varepsilon'^2 < O(2^m n^8/\varepsilon^6)$ as promised. □

Corollary 4.5 *Let m, n, ε and $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ be as in Theorem 4.4. Fix sets $S_y \subseteq \{0,1\}^m$, indexed by $y \in \{0,1\}^t$. If at least an α fraction of the S_y have $|S_y| \le (1-\beta)2^m$, for $\alpha + \beta - \alpha\beta < \varepsilon$, then the set $\{x : \forall y\ \ E(x,y) \in S_y\}$ has cardinality at most poly$(2^m, n, 1/\varepsilon)$ and it can be computed from the S_y in time poly$(2^m, n, 1/\varepsilon)$.*

5 Fixed Parameter Tractability

Parameterized complexity studies the complexity of problems that have a distinguished parameter k as part of their input. One can express the complexity of exact algorithms for such a problem in terms of the usual n (the instance size) and k. Problems possessing an algorithm with running time $O(f(k)n^c)$ where f is an arbitrary function and c is a constant independent of k are said to be *fixed-parameter tractable.* This more refined notion of complexity is informative for problems where one might expect to be solving problems with small values of the parameter k, and a complexity theory around this notion has been developed [2, 3].

In the parameterized matroid-greedoid partition problem we are looking for a partition of size k if one exists. In [9], Mielikäinen and Ukkonen proved that the parameterized matroid-greedoid intersection problem is W[P]-hard (which means that it is likely to be fixed-parameter intractable, under accepted complexity assumptions), and left the corresponding question for the partition problem open. In this section we show that, as in the approximation case, there is a difference between the complexity of the two problems, and that the parameterized matroid-greedoid partition problem is fixed-parameter tractable.

We prove the above in two steps. We first present a randomized algorithm that solves it, and then derandomize it using "almost k-wise independent" random variables. Our algorithm finds a partition of size k if one exists, that is, if the size of the maximum partition is at least k. If the size of the maximum partition is less than k it finds a maximum partition.

Definition 5.1 *A sample space $S \subseteq \{0,1\}^n$ is (ε, k)-independent if for any $1 \leq i_1 < i_2 < \ldots < i_k \leq n$ and any $\alpha \in \{0,1\}^k$*

$$\left| \Pr\left[x_{i_1} x_{i_2} \ldots x_{i_k} = \alpha\right] - 2^{-k} \right| \leq \varepsilon$$

where $x = x_1 x_2 \ldots x_n$ is chosen uniformly from S.

In [1] the authors give explicit constructions of (ε, k)-independent sample spaces of size $|S| = O(\frac{k^2 \log^2 n}{\varepsilon^2})$ that can be sampled using $O(\log |S|)$ random bits. In our proof, we only need to make sure that for any $1 \leq i_1 < i_2 < \ldots < i_k \leq n$ and any $\alpha \in \{0,1\}^k$ the sample space contains some x for which $x_{i_1} x_{i_2} \ldots x_{i_k} = \alpha$. Thus, taking ε to be strictly smaller than 2^{-k} will suffice, and the space can then be sampled using $O(k + \log \log n)$ random bits.

Theorem 5.2 *The parameterized matroid-greedoid partition problem with parameter k can be solved in time $f(k)n^c$, where n is the number of elements of the matroid and the greedoid, f is some function of k and c is a constant independent of k.*

Proof: We prove the theorem in two steps. We first present a randomized polynomial time algorithm that solves the problem, and then show how it can be derandomized efficiently. Let $M = (E, \mathcal{I})$ be a matroid and $G = (E, \mathcal{F})$ a greedoid on a set E of n elements, and let k be a positive integer. The following randomized algorithm finds, with probability at least 2^{-k}, a partition $Z = X \cup Y$ such that $X \in \mathcal{I}$, $Y \in \mathcal{F}$, $X \cap Y = \emptyset$ and $|Z| \leq k$. Partition E at random into two sets A and B. Use the greedy algorithm to find a subset $X \subseteq A$ such that $X \in \mathcal{I}$ that is either maximal, or of size k if the cardinality of a maximal independent set in M is larger than k. Then use the greedy algorithm to find a subset $Y \subseteq B$ such that $Y \in \mathcal{F}$ that is either maximal, or of size $k - |X|$ if the cardinality of a maximal feasible set in G is larger than $|X| - k$. Return $Z = X \cup Y$.

We now show that the algorithm succeeds with probability at least 2^{-k}. Suppose $Z^* = X^* \cup Y^*$, with $X^* \in \mathcal{I}$ and $Y^* \in \mathcal{F}$, is an optimal solution to the parameterized problem, i.e. Z^* is a partition of size k if a maximal partition has size at least k, otherwise Z^* is a maximal partition. Then the probability that the algorithm succeeds is

$$\Pr\left(|Z| = |Z^*|\right) \geq \Pr\left(X^* \subseteq A, Y^* \subseteq B\right) = 2^{-|Z^*|} \geq 2^{-k}.$$

We can derandomize the above algorithm by using a (k, ε)-independent sample space S for any $\varepsilon < 2^{-k}$, and running the randomized algorithm with partitions according to all $x \in S$. Since $\varepsilon < 2^{-k}$, for all $1 \leq i_1 < i_2 < \ldots < i_k \leq n$ and $\alpha \in \{0,1\}^k$ the sample space contains some x for which $x_{i_1} x_{i_2} \ldots x_{i_k} = \alpha$. Thus, for any $X, Y \subseteq E$ with $X \cap Y = \emptyset$ and $|X| + |Y| \leq k$, there is an $x \in S$ such that when the partition is done according to x we have $X \subseteq A$ and $Y \subseteq B$. Therefore, the algorithm finds an optimal solution.

By [1] there exists an (ε, k)-independent sample space of size $O(\frac{k^2 \log^2 n}{\varepsilon^2}) = O(2^{2k} k^2 \log^2 n)$ if we choose, say, $\varepsilon = 2^{-(k+1)}$, and we can enumerate all elements in its support in time poly$(n, 2^k)$. We run the greedy algorithm for the matroid and the greedoid for each string in the sample space, for an overall running time of $f(k)n^c$, for some function f of k, and a constant c that is independent of k. □

Remark 1 *This algorithm gives the same result for the parameterized greedoid-greedoid partition problem, and in fact, can easily be generalized to the parameterized m-greedoid partition problem for any fixed m, by using (ε, k)-independent sample spaces over an alphabet of size m. For the latter problem, the randomized algorithm works by finding a random partition of the universe into m parts, and using the greedy algorithm on the i-th greedoid in the i-th part to obtain the parts of the final partition. The derandomization proceeds analogously: an element x of the (ε, k)-independent sample space over an alphabet of size m is interpreted as specifying a partition of the universe, and we run over all such x in the sample space.*

References

[1] N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures and Algorithms*, 3(3):289–304, 1992.

[2] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness i: Basic results. *SIAM J. Comput.*, 24(4):873–921, 1995.

[3] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer, 1999.

[4] J. Edmonds. Minimum partition of a matroid into independent subsets. *Journal of Research of the National Bureau of Standards*, 69B:67–72, 1965.

[5] V. Guruswami, C. Umans, and S. P. Vadhan. Unbalanced expanders and randomness extractors from parvaresh-vardy codes. In *IEEE Conference on Computational Complexity*, pages 96–108. IEEE Computer Society, 2007.

[6] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random generation from one-way functions (extended abstract). In *STOC*, pages 12–24. ACM, 1989.

[7] B. Korte and L. Lovász. Mathematical structures underlying greedy algorithms. In *FCT '81: Proceedings of the 1981 International FCT-Conference on Fundamentals of Computation Theory*, pages 205–209, London, UK, 1981. Springer-Verlag.

[8] B. H. Korte, L. Lovász, and R. Schrader. *Greedoids*. Springer, 1991.

[9] T. Mielikäinen and E. Ukkonen. The complexity of maximum matroid-greedoid intersection and weighted greedoid maximization. *Discrete Appl. Math.*, 154(4):684–691, 2006.

[10] R. Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the European Association for Theoretical Computer Science*, 77:67–, June 2002. Columns: Computational Complexity.

[11] R. Shaltiel and C. Umans. Simple extractors for all min-entropies and a new pseudorandom generator. *Journal of the ACM*, 52(2):172–216, 2005. Conference version appeared in FOCS 2001.

[12] V. Shoup. New algorithms for finding irreducible polynomials over finite fields. *Mathematics of Computation*, 54(189):435–447, 1990.

[13] A. Ta-Shma and D. Zuckerman. Extractor codes. *IEEE Transactions on Information Theory*, 50(12):3015–3025, 2004.

[14] L. Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, 48(4):860–879, 2001.

[15] S. Vadhan. The unified theory of psuedorandomness. In *SIGACT News*, volume 38, September 2007.