Disjoint Bases in a Polymatroid

Gruia Călinescu* Chandra Chekuri† Jan Vondrák‡

May 26, 2008

Abstract

Let $f : 2^N \to \mathcal{Z}^+$ be a polymatroid (an integer-valued non-decreasing submodular set function with $f(\emptyset) = 0$). We call $S \subseteq N$ a *base* if $f(S) = f(N)$. We consider the problem of finding a maximum number of disjoint bases; we denote by m^* be this base packing number. A simple upper bound on m^* is given by $k^* = \max\{k : \sum_{i \in N} f_A(i) \geq k f_A(N), \forall A \subseteq N\}$ where $f_A(S) = f(A \cup S) - f(A)$. This upper bound is a natural generalization of the bound for matroids where it is known that $m^* = k^*$. For polymatroids, we prove that $m^* \geq (1 - o(1))k^*/\ln f(N)$ and give a randomized polynomial time algorithm to find $(1 - o(1))k^*/\ln f(N)$ disjoint bases, assuming an oracle for f. We also derandomize the algorithm using minwise independent permutations and give a deterministic algorithm that finds $(1 - \epsilon)k^*/\ln f(N)$ disjoint bases. The bound we obtain is almost tight since it is known there are polymatroids for which $m^* \leq (1 + o(1))k^*/\ln f(N)$. Moreover it is known that unless $NP \subseteq DTIME(n^{\log\log n})$, for any $\epsilon > 0$, there is no polynomial time algorithm to obtain a $(1 + \epsilon)/\ln f(N)$-approximation to m^*. Our result generalizes and unifies two results in the literature.

1 Introduction

A polymatroid $f : 2^N \to \mathcal{Z}^+$ on a ground set N is a non-decreasing (monotone) integer-valued submodular function. A function f is monotone iff $f(A) \leq f(B)$ for all $A \subseteq B$. A function f is submodular iff $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ for all $A, B \subseteq N$.[1] Polymatroids generalize matroids which have the additional condition that $f(\{i\}) \leq 1$ for all $i \in N$. We call a subset $S \subseteq N$ a *base* if $f(S) = f(N)$.[2]

We consider the problem of finding a maximum number of disjoint bases $S_1, S_2, \ldots, S_m$. Let m^* be the maximum achievable for f. For a subset $A \subseteq N$, the function $f_A : 2^N \to \mathcal{Z}^+$ is defined by $f_A(S) = f(S \cup A) - f(S)$. It is easy to check that f_A is submodular if f is submodular. An upper bound on m^* is

*Computer Science Dept., Illinois Institute of Technology, Chicago, IL. calinescu@iit.edu. Supported in part by NSF grant CCF-0515088.

†Dept. of Computer Science, University of Illinois, Urbana, IL 61801. chekuri@cs.uiuc.edu. Supported in part by NSF grant CCF-0728782.

‡Dept. of Mathematics, Princeton University, Princeton, NJ 08544. jvondrak@math.princeton.edu.

[1] Equivalently, a monotone function f is submodular if for any $S \subseteq T \subseteq N$, we have $\sum_{i \in T \setminus S}(f(S \cup \{i\}) - f(S)) \geq f(T) - f(S)$. [16, Proposition 7]

[2] A base in a matroid is also required to be minimal and hence independent. Our notion of a base corresponds to that of a spanning set for a matroid, or that of a set cover for a set system $\{A_j\}_{j \in N}$, when $f(S) = |\bigcup_{j \in S} A_j|$. Since we are concerned with the problem of packing disjoint bases, minimality plays no role here.

given by the following:

$$k^* = \max\{k : \sum_{i \in N} f_A(i) \geq k f_A(N), \forall A \subseteq N\}.$$

Indeed, $m^* \leq k^*$, since if S is a base, then for all $A \subseteq N$, $f_A(N) = f(N) - f(A) = f(S) - f(A) = f_A(S) \leq \sum_{i \in S} f_A(i)$. The last inequality follows immediately from the submodularity of f_A. By summing over k disjoint bases, we get $\sum_{i \in N} f_A(i) \geq k f_A(N)$.

A similar formula as the above for k^* is used by Wolsey [18]. Section 3 gives some possibly illuminating examples of k^*. An equivalent alternative definition for k^* is:

$$k^* = \min_{A : f(A) < f(N)} \frac{\sum_{i \in N} f_A(i)}{f_A(N)}.$$

For matroids f, Edmonds [9] showed that $m^* = k^*$ and gave a polynomial time algorithm to obtain m^* disjoint bases. For polymatroids, the situation is different: there are polymatroids f where $m^* \leq (1 + o(1))k^*/\ln f(N)$ [11]. Here, as in the rest of the paper, the little-oh notation is with respect to $f(N)$. Our main result is the following.

Theorem 1.1 *For every polymatroid $f : 2^N \rightarrow \mathcal{Z}^+$, $m^* \geq (1 - o(1))k^*/\ln f(N)$. Moreover, given an oracle access to f, there is a randomized polynomial time algorithm that finds $(1 - o(1))k^*/\ln f(N)$ disjoint bases with high probability. There is a deterministic algorithm that, for every fixed $\epsilon > 0$, finds $(1 - \epsilon)k^*/\ln f(N)$ bases.* [3]

The theorem is proved in Section 2. Section 3 gives two applications of this theorem. Below we briefly discuss the proof techniques and these two applications.

We prove Theorem 1.1 using a simple argument based on a random permutation. We derandomize the algorithm using ϵ-approximate minwise independent permutations. One could also obtain a proof using a random coloring algorithm used in [11] for the Domatic Partition problem (defined in Subsection 3.1) and a probabilistic lemma on submodular functions from a recent work of the authors [3]. The random permutation based algorithm is easier to analyze. The derandomization via ϵ-approximate minwise independent permutations is quite natural although we rely on involved results on their existence. We also sketch a straightforward generalization of the theorem to the capacitated case.

As an example of using Theorem 1.1, consider a polymatroid that arises from a set coverage system. We are given $|N|$ sets $X_1, X_2, \ldots, X_{|N|}$, each a subset of a universe $\mathcal{U}$. For $S \subseteq N$, let $f(S) = |\bigcup_{j \in S} X_j|$; this is a monotone submodular function. Then, k^* can be interpreted as the minimum coverage of an element; $k^* = \delta = \min_{a \in \mathcal{U}} |\{j : a \in X_j\}|$. In this case, our theorem states that there are at least $(1 - o(1))\delta/\ln n$ disjoint set covers, where $n = |\mathcal{U}|$ and δ the minimum coverage. This result is implicit in [11].

One of the motivations for proving Theorem 1.1 is the result obtained in [5] on a maximum number of disjoint connected spanning hypergraphs. The random coloring algorithm in [5] is very similar to that in [11] for the Domatic Partition problem. Both problems can be cast as finding disjoint bases in a polymatroid and Theorem 1.1 can be used to rederive the packing results in [11, 5] in a unified way.

[3] Technically, we should take the floor of all our upper-bounds. E.g., if $k^*/\ln f(N) = 1.9$, we guarantee only one base. The floor is necessary, as hypergraph 2-coloring can be cast in our framework, similar to the Domatic Partition problem as explained in Subsection 3.1, and Erdős [8] probabilistically constructs examples with $k^* = q$, $f(N) = (1 + o(1))\frac{e \ln 2}{4} q^2 2^q$, and where two disjoint bases do not exist.

Since the Domatic Partition problem is a special case of our problem, from the results in [11], it follows that unless $NP \subseteq DTIME(n^{\log\log n})$, m^* is hard to approximate to within a factor of $(1-\epsilon)\ln f(N)$ for any fixed $\epsilon > 0$. In [11], the hardness for Domatic Partition is shown via involved techniques that were used to obtain similar hardness for the corresponding optimization problem, namely the Minimum Dominating Set problem [10]. In Section 5 we discuss a connection between fractional packings and optimization via linear programming duality. An equivalence theorem claimed in [15] could be used to prove hardness of approximation for integer packing problems via the hardness of approximation of a corresponding optimization problem; however, the proof of this theorem in [15] is yet incomplete. Nevertheless, we give a detailed discussion on this topic for the sake of completeness since the approach would yield simpler proofs of the hardness of the Domatic Partition problem and other packing problems.

The parameter k^* has a natural interpretation in special cases, nevertheless it turns out in general to be NP-hard to compute. In fact we show that there is a constant $c > 0$ such that it is NP-hard to approximate the parameter k^* to within a $(c \log f(N))$-factor: this is essentially the same hardness as computing m^*! The proof is shown in Section 4.

2 Proof of Theorem 1.1

We assume that the value of k^* is known. We cannot compute k^* efficiently, however our algorithm is guaranteed to obtain $(1-o(1))k/\ln f(N)$ bases for any value of $k \leq k^*$ and hence we can try all the values for k in the range $1, 2, \ldots, |N|$ to obtain the desired result. We could also do a binary search which is useful in the capacitated case.

Let $k = \lfloor k^*/(\ln f(N) + \ln\ln f(N))\rfloor$. If $k \leq 1$ we simply output N as a single base. Otherwise, pick a random permutation of the elements of N, and partition N them into at least k parts by grouping together consecutive elements of $\lfloor |N|/k \rfloor$ each (except perhaps for the last group).

Fix one such part of $\lfloor |N|/k \rfloor$ elements, let e_i be its i^{th} element in the given permutation, and let $A_i = \{e_1, \ldots, e_i\}$, with $A_0 = \emptyset$. A_i and e_i are random choices. It is immediate that for $i = 1, \ldots, \lfloor |N|/k \rfloor$,

$$f_{A_i}(N) = f_{A_{i-1}}(N) - f_{A_{i-1}}(e_i)$$

and therefore

$$E[f_{A_i}(N)] = E[f_{A_{i-1}}(N)] - E[f_{A_{i-1}}(e_i)]. \qquad (1)$$

¿From the definition of k^*, and noting that $f_{A_{i-1}}(e) = 0$ for $e \in A_{i-1}$, we have

$$\sum_{e \in N} f_{A_{i-1}}(e) = \sum_{e \in N \setminus A_{i-1}} f_{A_{i-1}}(e) \geq k^* f_{A_{i-1}}(N),$$

which together with the fact that the choice of e_i is uniform from $N \setminus A_{i-1}$ implies:

$$E[f_{A_{i-1}}(e_i)] \geq \frac{1}{|N|+1-i} k^* E[f_{A_{i-1}}(N)] \geq \frac{k^*}{|N|} E[f_{A_{i-1}}(N)]. \qquad (2)$$

¿From this and (1) we deduce:

$$E[f_{A_i}(N)] \leq E[f_{A_{i-1}}(N)](1 - \frac{k^*}{|N|}) \leq E[f_{A_{i-1}}(N)]e^{-k^*/|N|}.$$

Applying induction and the fact that $k^*/|N| \cdot \lfloor |N|/k \rfloor \geq \ln f(N) + \ln\ln f(N) - 1$ (here we used $k^* \leq |N|$), we get:

$$E[f_{A_{\lfloor |N|/k \rfloor}}(N)] \leq e^{-\frac{k^*}{|N|}\cdot\lfloor |N|/k \rfloor} f(N) \leq \frac{e}{\ln f(N)}.$$

Since f is integer-valued, Markov's inequality gives:

$$Pr[f_{A_{\lfloor |N|/k \rfloor}}(N) > 0] \leq \frac{e}{\ln f(N)}$$

This holds for a fixed part $A_{\lfloor |N|/k \rfloor}$, and this part is a base when $f_{A_{\lfloor |N|/k \rfloor}}(N) = 0$. Let q be the random variables giving the number of parts which are bases. By linearity of expectations, we obtain that

$$E[q] \geq \left(1 - \frac{e}{\ln f(N)}\right) k \geq \frac{(1 - o(1))k^*}{\ln f(N)}.$$

To obtain the result with high probability, one has to notice that q cannot exceed k^*, and repeat the experiment a polynomial number of times.

A Deterministic Algorithm: We obtain a deterministic algorithm by derandomizing the random permutation based algorithm using approximately minwise independent family of permutations. A *minwise* independent family of permutations on $\{1, 2, \ldots, n\}$ is a subset $\mathcal{F} \subseteq S_n$ (S_n is the family of all permutations on $\{1, 2, \ldots, n\}$) with the following property: for a permutation picked at random from $\mathcal{F}$, for every $X \subseteq \{1, 2, \ldots, n\}$ and $x \in X$, the probability that x is the first element in X in the chosen permutation is exactly equal to $1/|X|$. These permutations were defined in [1] for their applications in obtaining sketches for checking similarity in documents. In [1], exponential lower bounds are shown on the size of any minwise independent family. For our purposes, it suffices to have a family that is ϵ-approximate minwise independent; $\mathcal{F}$ is such a family if the probability that $x \in X$ is the first element from X in a randomly chosen permutation from $\mathcal{F}$ is at most $(1+\epsilon)/|X|$ and at least $(1-\epsilon)/|X|$ for every subset X and $x \in X$. For each fixed $\epsilon > 0$, there exists a polynomial sized family of ϵ-approximate minwise indepenent permutations as shown in [14] (see also [17]). Moreover, given ϵ, the family can be explicitly constructed in polynomial time.

Let $\mathcal{F}$ be an ϵ-approximate minwise independent family of permutations. Suppose we chose a random permutation from $\mathcal{F}$ in the algorithm instead of choosing a random permutation from S_n the set of all permutations. We rework the analysis highlighting only the main points. We observe that (1) remains valid. However, (2) is no longer valid since e_i is not uniform in $N \setminus A_{i-1}$. Still, conditioned on A_{i-1}, e_i is approximately uniform in $N \setminus A_{i-1}$. More precisely, from the ϵ-approximate minwise independent property of $\mathcal{F}$, the probability that $e \in N \setminus A_{i-1}$ is e_i is in the interval $[\frac{1-\epsilon}{|N|-i+1}, \frac{1+\epsilon}{|N|-i+1}]$. Thus we obtain the following in place of (2).

$$E[f_{A_{i-1}}(e_i)] \geq \frac{1-\epsilon}{|N|+1-i} k^* E[f_{A_{i-1}}(N)] \geq \frac{(1-\epsilon)k^*}{|N|} E[f_{A_{i-1}}(N)]. \tag{3}$$

The rest of the analysis is essentially the same. To ensure that $Pr[f_{A_{\lfloor N/k \rfloor}}(N) > 0] \leq \frac{e}{\ln f(N)}$ we choose $k = \frac{(1-\epsilon)k^*}{\ln f(N) + \ln\ln f(N)}$. Thus, the the number of bases we obtain in expectation is, as before,

$$E[q] \geq \left(1 - \frac{e}{\ln f(N)}\right) k \geq \frac{(1-2\epsilon)k^*}{\ln f(N)}$$

for sufficiently large $f(N)$.

The above proves that there exists a permutation in $\mathcal{F}$ that yields $(1-2\epsilon)k^*/\ln f(N)$ bases. Since $\mathcal{F}$ is of polynomial size (in $|N|$) and is explicilty available, we try all permutations in $\mathcal{F}$ and obtain the desired number of bases.

This finishes the proof of Theorem 1.1.

The capacitated problem: We briefly discuss the capacitated problem. We are given non-negative integer capacities $u : N \rightarrow \mathcal{Z}^+$ and the goal is to output a maximum number of bases such that no element i is contained in more than u_i bases. Let $m^*(u)$ denote the base packing number for a given capacity vector u. The upper bound k^* generalizes in the natural way to $k^*(u)$ in the capacitated setting as follows:

$$k^*(u) = \max\{k : \sum_{i \in N} u_i f_A(i) \geq k f_A(N), \forall A \subseteq N\}.$$

One can reduce the capacitated problem to the uncapcatitated problem by creating u_i copies of i for each $i \in N$ and hence it follows that $m^*(u) \geq (1-o(1))k^*(u)/\ln f(N)$. The algorithmic problem of obtaining the bases requires some more care since the reduction to the uncapacitated case is not polynomial. However, one can use standard and simple scaling ideas that we sketch below. The output is specified in a compact form as a set of bases with corresponding integer multiplicities.

Let $u_{\min} = \min_i u_i$ and $u_{\max} = \max_i u_i$ and let $n = |N|$. First, we observe that $u_{\min} \leq k^* \leq n u_{\max}$. As in the uncapacitated case, the algorithm guesses k^* using binary search (in the interval $[u_{\min}, n u_{\max}]$). The rest of the discussion is for a given guess k^*. If $k^* \leq n^3$ then we can use the reduction to the uncapacitated setting. Otherwise let k be the largest integer in $[0..k^*]$ such that k is divisible by n^2. Note that $k > k^* - n^2$ and by our assumption that $k^* > n^3$, we have $k \geq (1-1/n)k^*$. We work with k in place of k^*. This will result in the loss of at most a $(1-1/n)$ factor in the number of bases. We assume without loss of generality that $u_i \leq k$ for all i; if $u_i > k$, we can set $u_i = k$. We obtain a modified capacity u' by setting

$$u'_i = \lfloor \frac{n^2 u_i}{k} \rfloor \quad i \in N.$$

The modified capacities satisfy the property that $0 \leq u'_i \leq n^2$ for all i. We can now apply the reduction to the uncapacitated case. Let m be the number of bases obtained in the modified instance. We obtain mk/n^2 bases in the original instance by creating k/n^2 copies (note that k/n^2 is an integer by the choice of k) of each base in the modified instance. It can be easily checked that scaling loses only a multiplicative factor of $(1-2/n)$ in the number of bases.

3 Applications

We show that two results in the literature [11, 5] can be cast as special cases of Theorem 1.1.

3.1 Domatic Partition

Given a universe $\mathcal{U}$ and m subsets $X_1, X_2, \ldots, X_m$ of $\mathcal{U}$, define f on the set $N = \{1, 2, \ldots, m\}$ by $f(S) = |\cup_{i \in S} X_i|$. It is easy to check that f is a polymatroid. A special case is when $\mathcal{U}$ is the set of vertices of an undirected graph $G = (V, E)$, and for $v \in V$, we have $X_v = \{v\} \cup \{u \in V \mid (u, v) \in E\}$; thus both $\mathcal{U}$ and N are equal to V, and $f(N) = |V| = n$. Then a polymatroid base is a dominating set in G and the problem of partitioning V into bases is the well-studied Domatic Partition problem (sometimes called Domatic Number). For this polymatroid k^* can be computed and is equal to $\delta + 1$, where δ is the minimum degree of the graph. Feige et al. [11] use a random coloring to obtain $(1 - o(1))\delta/\ln n$ disjoint dominating sets, and show how their method can be derandomized using conditional probabilities.

For $f(S) = |\cup_{i \in S} X_i|$ derived from an arbitrary set system, we obtain $k^* = \delta$ where $\delta = \min_{a \in U} |\{i \in N : a \in X_i\}|$ is the minimum coverage of an element. Let $n = |\mathcal{U}|$ denote the size of the universe. It can be proved by a random coloring along the lines of [11] that there exist $(1 - o(1))\delta/\ln n$ disjoint collections of sets X_i, each of which covers the entire universe. This is also a special case of our main theorem.

3.2 Disjoint Connected Spanning Hypergraphs

Let $H = (V, E)$ be a hypergraph; that is its edges $e \in E$ are arbitrary subsets of V. Given a set of edges $A \subseteq E$, let $c(A)$ be the number of connected components of (V, A), and define $f(A) = |V| - c(A)$. It is easy to check that f is a polymatroid on E, and a base of f corresponds to a connected subgraph of H. Partitioning the above polymatroid into bases was considered in [5], where a $\Theta(1/\ln(|V|))$-approximation is given. We improve the constant in the Θ and provide, in our opinion, a cleaner and simpler proof. We remark that the main result in [5] is to pack element disjoint Steiner trees in a graph. They reduce the problem, using a graph reduction step, to the case when G is a bipartite graph with terminals on one side of the bipartition and non-terminals (Steiner nodes) on the other side. Finding element disjoint Steiner trees in the above bipartite graph is the same as finding edge-disjoint connected hypergraphs; the hypergraph induced by the bipartite graph has a node for each terminal and a (hyper)edge for each Steiner node.

Assume for simplicity that H is connected. Let q be the size of the minimum global cut in H. Cheriyan and Salavatipour [5] use q in their proof. We do not know how to compute k^*, but we show below that $\frac{1}{2}q \le k^* \le q$.

To show $k^* \le q$, consider a set $Q \subseteq E$ with $|Q| = q$, and $c(E \setminus Q) > 1$; Q is a minimum global cut in H. Let $R = E \setminus Q$; then $f_R(E) = c(R) - 1$ and for all $e \in Q$, we have $f_R(e) \le c(R) - 1$. Then $\sum_{e \in E} f_R(e) = \sum_{e \in Q} f_R(e) \le q(c(R) - 1) = q f_R(E)$; this shows that $k^* \le q$.

To show $k^* \ge \frac{1}{2}q$, consider an arbitrary $A \subseteq E$. Let $V_1, V_2, \ldots, V_j$ be the connected components of the hypergraph induced by A. For any edge $e \in E \setminus A$, let j_e be the number of these connected components that e intersects. For $i = 1, 2, \ldots, j$, let Q_i be the set of edges e with $e \not\subseteq V_i$ and $e \cap V_i \neq \emptyset$. Note that for $e \in E \setminus A$, $j_e \ge 2$ if $e \in \cup_i Q_i$ and otherwise $j_e = 1$. Each Q_i is a cut and therefore $|Q_i| \ge q$. We conclude with:

$$
\begin{aligned}
\sum_{e \in E} f_A(e) = \sum_{e \in E \setminus A} f_A(e) = \sum_{e \in E \setminus A} (j_e - 1) \quad &= \quad \sum_{e \in \cup_i Q_i} (j_e - 1) \\
&\ge \quad \sum_{e \in \cup_i Q_i} \frac{1}{2} j_e = \frac{1}{2} \sum_{i=1}^{j} |Q_i| \\
&\ge \quad \frac{1}{2} q j \ge \frac{1}{2} q (j - 1) = \frac{1}{2} q f_A(E).
\end{aligned}
$$

4 Hardness of computing k^*

Theorem 4.1 *There is $c > 0$ and a family of polymatroids $\mathcal{F}$ such that given $f \in \mathcal{F}$, it is NP-hard to distinguish between $k^* = 1$ and $k^* \geq c \log f(N)$. Moreover, for each $f \in \mathcal{F}$, f can be explicitly evaluated in polynomial time.*

We prove the NP-hardness of computing k^* via a reduction from Vertex Cover in Hypergraphs. It is known that it is NP-hard to approximate the minimum vertex cover in a r-uniform-hypergraph to within a factor of $(r - 1 - \epsilon)$ [7]. More specifically, it is NP-hard to distinguish between the case where there is a vertex cover of size $n/(r-1-\epsilon)$, and the case where any vertex cover has size at least $(1-\epsilon)n$. It is enough for our purposes to use the following weaker version:

For a given r-uniform hypergraph H on n vertices, $r \geq 4$, it is NP-hard to distinguish between the case where there is a vertex cover smaller than $\frac{3}{5}n$ and the case where any vertex cover has size at least $\frac{4}{5}n$.

This hardness result holds even for superconstant r, up to $r = c \log m$, where m is the number of edges in H and $c > 0$ some constant. This follows relatively easily from known results [13, 6, 7]; we show the details in the appendix. Given a r-uniform hypergraph H, $r = c \log m$, we define a polymatroid f such that $k^* = 1$ in case H has a vertex cover smaller than $\frac{3}{5}n$, and $k^* \geq r/2$ in case every vertex cover has size at least $\frac{4}{5}n$. This will prove that distinguishing between $k^* = 1$ and $k^* \geq r/2$ is NP-hard. In addition, f will satisfy the property that $f(N) = (1 + o(1))m$, i.e. $r = c \log m = (1 - o(1))c \log f(N)$ which are the desired properties for Theorem 4.1.

Let $H = (V, E)$ be the given r-uniform-hypergraph with n denoting $|V|$. The polymatroid $f : 2^V \to \mathcal{Z}^+$ is defined as follows

$$f(S) = g(|S|) + h(S)$$

where $g(s) = \min\{s, \frac{4}{5}n\}$ and $h(S)$ denotes the number of edges in E covered (i.e. hit) by S in H. (We assume for simplicity that n is divisible by 5. This is actually the case for the hard instances of H that we present in Appendix A.) It is easy to see that g and h are submodular and hence f is submodular. The role of $g(s)$ is to make the value of k^* reflect the threshold of $\frac{4}{5}n$ for vertex cover. Recall the definition of $k^* = \max\{k : \forall S; \sum_{i \in N} f_S(i) \geq k f_S(N)\}$. Let us define

$$f_k(S) := f(S) + \frac{1}{k} \sum_{i \in N} f_S(i)$$

I.e., $k^* = \max\{k : \forall S; f_k(S) \geq f(N)\}$. Note that $f_1(S) \geq f(N)$ for any $S \subseteq N$, therefore k^* is at least 1. In general, we have

$$f_k(S) = g(s) + h(S) + \frac{1}{k}(n-s)(g(s+1) - g(s)) + \frac{1}{k} \sum_{i \in N} h_S(i)$$

where $s = |S|$; the $(n - s)$ term is due to $f_S(i)$, $g_S(i)$ and $h_S(i)$ being all 0 for $i \in S$. Note that each edge uncovered by S contributes to r marginal values $h_S(i)$, since there are r vertices potentially covering it. Thus

$$f_k(S) \quad = \quad g(s) + h(S) + \frac{1}{k}(n-s)(g(s+1) - g(s)) + \frac{r}{k}(|E| - h(S)) = g_k(s) + \frac{r}{k}|E| - \frac{r-k}{k} h(S)$$

where we set $g_k(s) = g(s) + \frac{1}{k}(n-s)(g(s+1) - g(s))$. This function is equal to $g_k(s) = s + \frac{1}{k}(n-s)$ for $s < \frac{4}{5}n$, and it is a constant $g_k(s) = \frac{4}{5}n$ for $s \geq \frac{4}{5}n$.

Suppose that $k \leq r/2$, i.e. $\frac{r-k}{k} \geq 1$. Then, if there is an edge uncovered by S, we can always include a new vertex in S, covering a new edge, which increases $g_k(s)$ by at most $1 - 1/k$ and increases $h(S)$ by at least 1. Therefore, $f_k(S)$ decreases; since we want to consider the minimum of $f_k(S)$, we can assume that S is a vertex cover. For a vertex cover, $h(S) = |E|$ and

$$f_k(S) = g_k(s) + |E|.$$

If there is a vertex cover S of size $s < \frac{3}{5}n$, we have $f_2(S) = s + \frac{1}{2}(n - s) + |E| < \frac{4}{5}n + |E| = f(N)$. In this case, $k^* < 2$. Otherwise, we can assume that any vertex cover has size at least $\frac{4}{5}n$. Then we can take $k = r/2$ and we know that the minimum of $f_k(S)$ is attained for a vertex cover, so we have $\min_S f_k(S) \geq \frac{4}{5}n + |E| = f(N)$. So in this case, $k^* \geq r/2$.

5 Hardness of Fractional and Integer Packings

We observed that unless $NP \subseteq DTIME(n^{\log \log n})$, for any $\epsilon > 0$, there is no polynomial time algorithm to obtain a $(1 + \epsilon)/ \ln f(N)$-approximation to the number of disjoint bases for f. This follows from the inapproximability of the Domatic Partition [11]. The proof of Feige *et al.* involves new techniques combining multi-prover protocols and zero-knowledge proofs. The underlying ideas are borrowed from the hardness results for the Minimum Dominating Set problem. Note that the Domatic Partion and the Minimum Dominating Set problem are two natural optimization problems related to dominating sets in graphs. A natural question is whether there is a direct connection that would tie the hardness of these two problems. This has been explored in [15] although some of the connections were mentioned explicitly only in the context of the Steiner tree problem. Here we discuss this connection explicitly in a general setting for the sake of completeness.

Let us define the more general problem of finding *integer* packings of combinatorial structures [4]. Let N be a ground set of n elements and $\mathcal{S} \subseteq 2^N$ be set of *feasible* solutions. One obtains different problems by defining $\mathcal{S}$ appropriately. We will assume the existence of a polynomial time oracle for the following problem: given a set $S \subseteq N$, decide if $S \in \mathcal{S}$ or not. Further, we also assume that there is a polynomial time algorithm to decide if $\mathcal{S} = \emptyset$ or not. The maximum integer packing problem corresponds to finding a maximum number of disjoint subsets $S_1, S_2, \ldots, S_m$ of N such that $S_i \in \mathcal{S}$ for $1 \leq i \leq m$. More generally, one obtains a capacitated version of the problem where each $e \in N$ has an integer capacity u_e and the goal is to find a maximum number of feasible solutions from $\mathcal{S}$ (we allow a solution to be included with multiplicity) such that each e is in at most u_e sets. One can write this as a large integer program as follows. For each $I \in \mathcal{S}$ we have a variable x_I. Then the IP is:

$$\begin{aligned} \max \sum_{I \in \mathcal{S}} x_I & & & \\ \sum_{I : e \in I} x_I & \leq & u_e & \qquad \forall e \in N \\ x_I & \in & \mathcal{Z}^+ & \qquad \forall I \in \mathcal{S}. \end{aligned}$$

One obtains a *fractional* packing by letting x_I take on values in $\mathcal{R}^+$, which corresponds to taking the LP relaxation of the above integer program. It is known that the complexity of finding a maximum *fractional*

[4]We remark that the discussion in this section applies equally well to coverings instead of packings.

packing of feasible solutions in $\mathcal{S}$ is related, via LP duality, to the complexity of finding a minimum cost structure in $\mathcal{S}$. This can be seen by writing the dual of the LP relaxation above.

$$\begin{aligned}
\min \sum_e u_e y_e & & & \\
\sum_{e \in I} y_e & \geq & 1 & \quad \forall I \in \mathcal{S} \\
y_e & \geq & 0 & \quad \forall e \in N.
\end{aligned}$$

The separation oracle for the dual is the following: given *costs* y_e on the elements of N, find the minimum cost feasible set in $\mathcal{S}$. By strong duality and the equivalence of separation and optimization [12], an optimum fractional packing can be found in polynomial time for a problem defined by $\mathcal{S}$ iff a minimum cost set in $\mathcal{S}$ can be found in polynomial time. Note that the equivalence requires treating the problem in a general setting with arbitrary capacities and costs. Does this equivalence extend to the approximation setting? The easier direction is that an α-approximation for the minimization problem implies an α-approximation for the fractional packing. This was shown in [4] although similar results had been shown for specific problems in prior work. The more difficult direction is that α-approximation for the fractional packing problem implies an α-approximation for the minimization problem. This result was claimed in [15], however the proof does not appear to be complete.

What does the above say about the complexity of maximum integer packings? There are several examples that show that finding maximum integer packings can be significantly harder than finding maximum fractional packings, even when the maximum fractional packing can be computed in polynomial time. However, it is intuitively clear that finding a maximum integer packing is at least as hard as finding a maximum fractional packing. One needs to be careful to some extent in that there could be capacities u for which finding a maximum fractional packing might be hard while finding an integer packing might be easy for a trivial reason. Therefore, in this discussion, we assume that u is part of the input. In this setting we will prove that an α-approximation for the maximum integer packing problem implies an $\alpha(1-\epsilon)$ approximation for finding a maximum fractional packing where ϵ can be chosen to be as small as $1/\text{poly}(n)$. The implication of this is that the approximability of the integer packing problem is no easier than that of the fractional packing problem. Therefore, if the equivalence of the approximability of fractional packing with that of minimization can be shown, we can conclude the following. If the minimization problem over $\mathcal{S}$ is hard to approximate to within a factor of α then finding a maximum integer packing over $\mathcal{S}$ is hard to approximate within a factor of α.

We now prove the fact stated above: an α-approximation for the maximum integer packing problem implies an $(1-\epsilon)\alpha$ approximation for finding a maximum fractional packing. The idea is simple; if u is scaled by a large enough factor, integer packing is approximately equivalent to fractional packing and hence we can use an integer packing algorithm to obtain a good fractional packing. Let us fix N and $\mathcal{S}$. Given an integer vector u, let $\nu(u)$ and $\gamma(u)$ be the maximum fractional and integer packing numbers for u. It is obvious that $\nu(\lambda u) = \lambda \nu(u)$ for any $\lambda \geq 0$. We will show the following: $\gamma(u) \geq (1-\epsilon)\nu(u)$ if $u_{\min} = \min_e u_e \geq c(\epsilon) \ln n$ for some function c. To approximate $\nu(u)$ to within a $(1-\epsilon)$ factor we approximate $\gamma(\lambda u)$ where $\lambda \geq c(\epsilon) \ln n$ for an appropriately chosen function c. It suffices to prove that $\gamma(\lambda u) \geq (1-\epsilon)\nu(\lambda u)$. This is a straightforward application of the Chernoff-Hoeffding bounds. Let x be an optimum solution for the fractional packing problem with capacities λu. First, by choosing a basic solution the linear program, the support of x has at most n sets. We can assume without loss of generality that $\nu(u) \geq 1$ and hence $\sum_I x_I = \nu(\lambda u) \geq \lambda$. We assume for simplicity that $x_I \in [0,1]$ for each $I \in \mathcal{S}$ (otherwise we can work with the fractional and integer parts separately). We obtain an integer packing $\hat{x}$ by independently setting, for each I, $\hat{x}_I = 1$ with probability $x_I(1-\epsilon/2)$ and $\hat{x} = 0$ otherwise. By standard analysis, since $\lambda u_e \geq \lambda \geq c(\epsilon)\ln n$ for each e and $\sum_i x_I \geq \lambda = c(\epsilon)\ln n$, with high probability $\hat{x}$ is a feasible integer packing such that $\sum_I \hat{x}_I \geq (1-\epsilon)\sum_I x_I = (1-\epsilon)\nu(\lambda u)$. Finally, we can scale this integer packing by $1/\lambda$ to obtain a fractional packing approximating the optimum with capacities u:

$$\nu(u) = \frac{1}{\lambda}\nu(\lambda u) \geq \frac{1}{\lambda}\gamma(\lambda u) \geq \frac{1-\epsilon}{\lambda}\nu(\lambda u) = (1-\epsilon)\nu(u).$$

The function $c(\epsilon)$ can be chosen to be c'/ϵ^2 for a sufficiently large constant c'. Thus, one can choose $\epsilon = 1/\text{poly}(n)$ while keeping λ polynomial in n. We mention that simple deterministic argument also gives the above reduction [15].

The main purpose of the discussion above was to conclude that the Domatic Partition problem is hard to approximate to within a $(1-o(1))\ln n$ factor unless $NP \subseteq DTIME(n^{\ln\ln n})$, (already shown in [11]) by simply invoking a similar hardness of approximation for the Minimum Dominating Set problem [10]. We would indeed be able to do this if the equivalence theorem claimed in [15] is proven.

6 Conclusions

In [11] an $O(\ln\Delta(G))$ approximation is given for the Domatic Partition problem where $\Delta(G)$ is the maximum degree in the given graph G. This ratio improves the $(1-o(1))\ln n$ approximation when $\Delta(G)$ is small compared to n. The corresponding question in the polymatroid setting is the following: is there an $O(\ln\Delta(f))$ approximation for m^* where $\Delta(f) = \max_{i\in N} f(i)$? Wolsey [18] showed that the greedy algorithm gives a $(1+\ln\Delta(f))$ approximation for the minimum weight base in a polymatroid f. From the duality discussed in Section 5, a $(1+\ln\Delta(f))$ approximation for the maximum fractional packing follows. Can we obtain a similar bound for integer packings? The $O(\ln\Delta(G))$ bound mentioned above of [11] is based on a simple randomized coloring algorithm; the analysis uses the Lovász Local Lemma. For general polymatroids a similar randomized algorithm does not work. Nevertheless we do not know of any f for which $m^* = o(k^*/\ln\Delta(f))$.

Acknowledgment. We thank Julia Chuzhoy for her help with the hardness result that appears in Appendix A.

References

[1] A. Broder, M. Charikar, A. Frieze and M. Mitzenmacher. Min-Wise Independent Permutations. *JCSS*, 60(3):630–659, 2000. Preliminary version in *Proc. of ACM STOC*, 1998.

[2] A. Broder, M. Charikar and M. Mitzenmacher. A derandomization using min-wise independent permutations. *J. of Discrete Algorithms*, 1(1): 11-20, 2003.

[3] G. Călinescu, C. Chekuri, M. Pál and J. Vondrák. Maximizing a Submodular Set Function subject to a Matroid Constraint. Manuscript, Nov 2006.

[4] R. Carr and S. Vempala. Randomized Metarounding. *Random Structers and Algorithms*, 20(3): 343–352, 2002. Preliminary version in *Proc. of ACM STOC*, 2000.

[5] J. Cheriyan and M. Salavatipour. Packing Element-Disjoint Steiner Trees. *Proc. of APPROX*, Springer LNCS, 52–61, 2005.

[6] I. Dinur, V. Guruswami and S. Khot. Vertex Cover on k-uniform hypergraphs is hard to approximate within factor $(k - 3 - \epsilon)$. *ECCC Technical Report TR02-027*, 2002.

[7] I. Dinur, V. Guruswami, S. Khot and O. Regev. A New Multilayered PCP and the Hardness of Hypergraph Vertex Cover. *Proc. of ACM STOC*, 595–601, 2003.

[8] P. Erdős. On a combinatorial problem II. *Acta Math. Acad. Sci. Hungar.* 15:445-447, 1964.

[9] J. Edmonds. Minimum Partition of a Matroid into Independent Subsets. *J. Res. Nat. Bur. Standards B*, 69:73–77, 1965.

[10] U. Feige. A threshold of $\ln n$ for approximating set cover. *JACM*, 45(4):634–652, 1998

[11] U. Feige, M. Halldorrson, G. Kortsarz and A. Srinivasan. Approximating the domatic number. *SIAM J. on Computing*, 32:172–195, 2002.

[12] M. Grötschel, L. Lovász and A. Schrijver. Geometric Algorithms and Combinatorial Optimization. Springer-Verlag, 1987.

[13] J. Holmerin. Vertex cover on 4-uniform hypergraph is hard to approximate within $2 - \epsilon$. *Proc. of ACM STOC*, 544–552, 2002.

[14] P. Indyk. A Small Approximately Min-Wise Independent Family of Hash Functions. *J. of Algorithms*, 38(1): 84-90, 2001. Preliminary version in *Proc. of ACM-SIAM SODA*, 1999.

[15] K. Jain, M. Mahdian and M. Salavatipour. Packing Steiner Trees. *Proc. of ACM-SIAM SODA*, 2003.

[16] G. L. Nemhauser, L. A. Wolsey and M. L. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Math. Prog.*, 14:265–294, 1978.

[17] M. Saks, A. Srinivasan, S. Zhou and D. Zuckerman. Low discrepancy sets yield approximate min-wise independent permutation families. *Inf. Process. Lett.*, 73(1-2): 29–32, 2000.

[18] L. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.

A A hardness result for vertex cover in hypergraphs

In Section 4, we used the following hardness result:

For a given r-uniform hypergraph H with n vertices and m edges, $r = c \log m$ for some constant $c > 0$, it is NP-hard to distinguish between the case where there is a vertex cover smaller than $\frac{3}{5}n$ and the case where any vertex cover has size at least $\frac{4}{5}n$.

Here we show how this follows from results stated explicitly in the literature [13, 6, 7]. (We are indebted to Julia Chuzhoy for pointing out this reduction to us.) We borrow the following result from [6]:

Theorem A.1 ([6]) *For any $\delta > 0$, it is NP-hard to approximate Vertex Cover in 4-uniform hypergraphs to within $2 - \delta$. In particular, it is NP-hard to distinguish between the minimum vertex cover being at most $(\frac{1}{2} + \delta)n$ and at least $(1 - \delta)n$, where n is the number of vertices.*

(The first part of the above theorem corresponds to Theorem 3.1 in [6] and the second part of the statement is stated in two lemmas immediately following Theorem 3.1 in [6].)

For a given 4-uniform hypergraph H and a given parameter $d \geq 1$, we perform the following operation to define a new hypergraph H'. We replace each vertex v of H by a set of $10d$ "clones" C_v. For each 4-edge $\{v_1, v_2, v_3, v_4\} \in E(H)$, we define $\binom{10d}{d}^4$ hyperedges of H', where each hyperedge is formed by choosing d clones for each of the 4 vertices v_1, v_2, v_3, v_4: $e = A_1 \cup A_2 \cup A_3 \cup A_4$, $A_i \subset C_{v_i}$, $|A_i| = d$. If H had n vertices and m edges, H' now has $n' = 10dn$ vertices and $\binom{10d}{d}^4 m = 2^{O(d)} m$ hyperedges. We choose $d = \log m$, so that H' has $m' = poly(m)$ hyperedges of size $r' = 4d = \Omega(\log m')$.

We show that it is NP-hard to distinguish whether the minimum vertex cover in H' is smaller than $\frac{3}{5}n'$ or at least $\frac{4}{5}n'$. Consider $\delta < \frac{1}{10}$ and the two cases which are NP-hard to distinguish due to Theorem 3.1 [6]:

- If there is a vertex cover S in H of size $|S| \leq (\frac{1}{2} + \delta) < \frac{3}{5}n$, then we can take all the clones for each vertex in S. This gives a vertex cover $S' = \bigcup_{v \in S} C_v$ of size $10d|S| < 6dn = \frac{3}{5}n'$.
- If any vertex cover in H has size at least $(1 - \delta)n > \frac{9}{10}n$, consider any set S' of $\frac{4}{5}n' = 8dn$ vertices in H'. Let $W \subseteq V(H)$ denote vertices that have at least $9d$ clones in S'. Since $9d|W| \leq |S'| = 8dn$, we have $|W| \leq \frac{8}{9}n$. Therefore, W cannot be a vertex cover in H. Consider an edge $\{v_1, v_2, v_3, v_4\}$ disjoint from W. Since each v_i has at most $9d$ clones in S', there is still a set of d clones $A_i \subset C_{v_i}$ disjoint from S'. We have found a hyperedge $A_1 \cup A_2 \cup A_3 \cup A_4 \in E(H')$ disjoint from S', i.e. S' is not a vertex cover in H'. We have proved that there is no vertex cover of size $\frac{4}{5}n'$.

Remark. Note that this construction does not produce r' larger than $O(\log m')$, even for larger d, because m' grows exponentially with d. Indeed, the hardness result stated here (with a gap between $\frac{3}{5}n$ and $\frac{4}{5}n$) does not hold for $\omega(\log m)$-uniform hypergraphs. This is not surprising, since for an r-uniform hypergraph with m hyperedges, where $r = c \log m$, $c > 2$, taking each vertex independently with probability $2/c$ gives a vertex cover with high probability.