Maximizing a Submodular Set Function subject to a Matroid Constraint (Extended Abstract)

Gruia Calinescu[1], Chandra Chekuri[2], Martin Pál[3], and Jan Vondrák[4]

[1] Computer Science Dept., Illinois Institute of Technology, Chicago, IL. calinescu@iit.edu.
[2] Dept. of Computer Science, University of Illinois, Urbana, IL 61801. chekuri@cs.uiuc.edu.
[3] Google Inc., 1440 Broadway, New York, NY 10018. mpal@google.com.
[4] Dept. of Mathematics, Princeton University, Princeton, NJ 08544. jvondrak@math.princeton.edu.

Abstract. Let $f : 2^N \to \mathcal{R}^+$ be a non-decreasing submodular set function, and let $(N, \mathcal{I})$ be a matroid. We consider the problem $\max_{S \in \mathcal{I}} f(S)$. It is known that the greedy algorithm yields a 1/2-approximation [9] for this problem. It is also known, via a reduction from the max-k-cover problem, that there is no $(1 - 1/e + \epsilon)$-approximation for any constant $\epsilon > 0$, unless $P = NP$ [6]. In this paper, we improve the 1/2-approximation to a $(1-1/e)$-approximation, when f is a sum of weighted rank functions of matroids. This class of functions captures a number of interesting problems including set coverage type problems. Our main tools are the pipage rounding technique of Ageev and Sviridenko [1] and a probabilistic lemma on monotone submodular functions that might be of independent interest.

We show that the generalized assignment problem (GAP) is a special case of our problem; although the reduction requires $|N|$ to be exponential in the original problem size, we are able to interpret the recent $(1 - 1/e)$-approximation for GAP by Fleischer *et al.* [10] in our framework. This enables us to obtain a $(1 - 1/e)$-approximation for variants of GAP with more complex constraints.

1 Introduction

This paper is motivated by the following optimization problem. We are given a ground set N of n elements and a non-decreasing submodular set function $f : 2^N \to \mathcal{R}^+$. The function f is *submodular* iff $f(A)+f(B) \geq f(A\cup B)+f(A\cap B)$ for all $A, B \subseteq N$. We restrict attention to non-decreasing (or monotone) submodular set functions, that is $f(A) \geq f(B)$ for all $B \subseteq A$ and $f(\emptyset) = 0$. An independence family $\mathcal{I} \subseteq 2^N$ is a family of subsets that is downward closed, that is, $A \in \mathcal{I}$ and $B \subseteq A$ implies that $B \in \mathcal{I}$. A set A is *independent* iff $A \in \mathcal{I}$. A family $\mathcal{I}$ is a p-independence family for an integer $p \geq 1$ if for all $A \in \mathcal{I}$ and $e \in N$ there exists a set $B \subseteq A$ such that $|B| \leq p$ and $A \setminus B + e$ is independent. For computational purposes we will assume that f and $\mathcal{I}$ are specified as oracles although in many specific settings of interest, an explicit description is often available. The problem (or rather class of problems) of interest in this paper is the following: $\max_{S\in\mathcal{I}} f(S)$. We will be mostly interested in the special case when $\mathcal{I}$ consists of the independent sets of a matroid on N. The problem of maximizing a submodular set function subject to independence constraints has been studied extensively. A number of interesting and useful combinatorial optimization problems, including NP-hard problems, are special cases. Some notable examples are maximum independent set in a matroid, weighted matroid intersection, and maximum coverage. Below we describe some candidates for f and $\mathcal{I}$ that arise frequently in applications.

Modular functions: A function $f : 2^N \rightarrow \mathcal{R}^+$ is modular iff $f(A) + f(B) = f(A\cup B)+f(A\cap B)$. If f is modular then there is a weight function $w : N \rightarrow \mathcal{R}^+$ such that $f(A) = w(A) = \sum_{e\in A} w(e)$.

Set Systems and Coverage: Given a universe U and n subsets $S_1, S_2, \ldots, S_n$ of U we obtain several natural submodular functions on the set $N = \{1, 2, \ldots, n\}$. First, the coverage function f given by $f(A) = |\cup_{i\in A} S_i|$ is submodular. This naturally extends to the weighted coverage function; given a non-negative weight function $w : U \rightarrow \mathcal{R}^+$, $f(A) = w(\cup_{i\in A} S_i)$. We obtain a multi-cover version as follows. For $x \in U$ let $k(x)$ be an integer. For each $x \in U$ and S_i let $c(S_i, x) = 1$ if $x \in S_i$ and 0 if $x \notin S_i$. Given $A \subseteq N$, let $c'(A, x)$, the coverage of x under A, be defined as $c'(A, x) = \min\{k(x), \sum_{i\in A} c(S_i, x)\}$. The function f with $f(A) = \sum_{x\in U} c'(A, x)$ is submodular. A related function defined by $f(A) = \sum_{x\in U} \max_{i\in A} w(S_i, x)$ is also submodular where $w(S_i, x)$ is a non-negative weight for S_i covering x.

Weighted rank functions of matroids and their sums: The rank function of a matroid $\mathcal{M} = (N, \mathcal{I})$, $r_{\mathcal{M}}(A) = \max\{|S| : S \subseteq A, S \in \mathcal{I}\}$, is submodular. Given $w : N \rightarrow \mathcal{R}^+$, the weighted rank function defined by $r_{\mathcal{M},w}(A) = \max\{w(S) : S \subseteq A, S \in \mathcal{I}\}$ is a submodular function. A sum of weighted rank functions is also submodular. Functions arising in this way form a rich class of submodular functions. In particular, all the functions on set systems and coverage mentioned above are captured by this class. However, the class does not include all monotone submodular functions; one notable exception is multi-cover by multisets.

Matroid Constraint: An independence family of particular interest is one induced by a matroid $\mathcal{M} = (N, \mathcal{I})$. A very simple matroid constraint that is of much importance in applications [5, 14, 2, 3, 10] is the partition matroid; N is partitioned into ℓ sets $N_1, N_2, \ldots, N_\ell$ with associated integers $k_1, k_2, \ldots, k_\ell$, and a set $A \subseteq N$ is independent iff $|A \cap N_i| \leq k_i$. In fact even the case of $\ell = 1$ (the uniform matroid) is of interest. Laminar matroids generalize partition matroids. We have a laminar family of sets on N and each set S in the family has an integer value k_S. A set $A \subseteq N$ is independent iff $|A \cap S| \leq k_S$ for each S in the family.

Intersection of Matroids: A natural generalization of the single matroid case is obtained when we consider intersections of different matroids $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_p$ on the same ground set N. That is, $\mathcal{I} = \cap_i \mathcal{I}_i$ where $\mathcal{I}_i$ is the independence family of $\mathcal{M}_i$. A simple example is the family of hypergraph matchings in a p-partite graph ($p = 2$ is simply the family of matchings in a bipartite graph).

Matchings: Given a general graph $G = (V, N)$ the set of matchings forms a 2-independent family. Given a hypergraph $G = (V, N)$ such that each edge $e \in N$ is of cardinality at most p, the set of matchings in G induce a p-independent family. Note that matchings in general graphs are not captured as intersections of matroids.

The Greedy Algorithm: A simple greedy algorithm is quite natural for this problem. The algorithm incrementally builds a solution (without backtracking) starting with the empty set. In each iteration it adds an element that most improves the current solution (according to f) while maintaining independence of the solution. The greedy algorithm yields a $1/p$-approximation for maximizing a modular function subject to a p-independence constraint [12, 13]. For submodular functions, the greedy algorithm yields a ratio of $1/(p+1)$ [9]. [5] These ratios for greedy are tight for all p even when the p-independent system is obtained as an intersection of p matroids. For large but fixed p, the p-dimensional matching problem is NP-hard to approximate to within an $\Omega(\log p/p)$ factor [11].

For the problem of maximizing a submodular function subject to a matroid constraint (special case of $p = 1$), the greedy algorithm achieves a ratio of $1/2$. When the matroid is the simple uniform matroid ($S \subseteq N$ is independent iff $|S| \leq k$) the greedy algorithm yields a $(1-1/e)$-approximation [14]. This special case already captures the maximum coverage problem for which it is shown in [6] that, unless $P = NP$, no $1 - 1/e + \epsilon$ approximation is possible for any constant $\epsilon > 0$. This paper is motivated by the following question. Is there a $(1-1/e)$-approximation algorithm for maximizing a submodular function subject to (any given) matroid constraint? We resolve this question for a subclass of monotone submodular functions, which can be expressed as a sum of weighted rank functions of matroids. The following is our main result.

Theorem 1. *Given a ground set N, let $f(S) = \sum_{i=1}^{m} g_i(S)$ where $g_1, \ldots, g_m : 2^N \to \mathcal{R}^+$ are weighted rank functions, g_i defined by a matroid $\mathcal{M}_i = (N, \mathcal{X}_i)$ and weight function $w_i : N \to \mathcal{R}^+$. Given another matroid $\mathcal{M} = (N, \mathcal{I})$ and membership oracles for $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_m$ and $\mathcal{M}$, there is a polynomial time $(1-1/e)$-approximation for the problem $\max_{S \in \mathcal{I}} f(S)$.*

As immediate corollaries we obtain a $(1-1/e)$-approximation for a number of coverage problems under a matroid constraint. It is known that there exist submodular monotone functions that cannot be expressed as a sum of weighted rank functions of matroids (see [16], 44.6e). For such functions, our framework does not seem to apply at this moment. We leave it as an open question whether a $(1-1/e)$-approximation is possible for all monotone submodular functions.

Our main tools are the the pipage rounding technique of Ageev and Sviridenko [1], and the following useful lemma.

[5] We give a somewhat new proof of this result in the full version of the paper. If only an α-approximate oracle ($\alpha \leq 1$) is available for the function evaluation, the ratio obtained is $\alpha/(p+\alpha)$. Several old and recent applications of greedy can be explained using this observation.

Lemma 1. *Let $f : 2^N \to \mathcal{R}^+$ be a monotone submodular function and let $f^* : [0,1]^N \to \mathcal{R}^+$ be defined as $f^*(y) = \min_S(f(S) + \sum_i y_i(f(S+i) - f(S)))$. For $y \in [0,1]^N$, let $\hat{y}$ denote a random vector in $\{0,1\}^N$ obtained by independently setting $\hat{y}_i = 1$ with probability y_i and 0 otherwise. Then, $\mathbf{E}[f(\hat{y})] \geq (1-1/e)f^*(y)$.*

We give a non-trivial application of Theorem 1 to variants of the generalized assignment problem (GAP). In GAP we are given n bins and m items. Each item i specifies a size s_{ji} and a value (or profit) v_{ji} for each bin j. Each bin has capacity 1 and the goal is to assign a subset of items to bins such that the bin capacities are not violated and the profit of the assignment is maximized. Recently Fleischer *et al.* [10] gave a $(1 - 1/e)$-approximation for this problem, improving upon a $1/2$-approximation [4]. We rederive the same ratio casting the problem as a special case of submodular function maximization. Moreover our techniques allow us to obtain a $(1-1/e)$-approximation for GAP even under any given laminar matroid constraint on the bins. A simple and easy to understand example is GAP with the added constraint that at most k of the n bins be used.

Theorem 2. *Let A be an instance of GAP with n bins and m items and let B be the set of bins. Let $\mathcal{M} = (B, \mathcal{I})$ be a laminar matroid on B. There is a polynomial time $(1 - 1/e)$-approximation to find a maximum profit assignment to bins such that the subset $S \subseteq B$ of bins that are used in the assignment satisfy the constraint $S \in \mathcal{I}$.*

We note that the approximation ratio for GAP has been improved to $1 - 1/e + \delta_1$ for a small $\delta_1 > 0$ in [8] using the same LP as in [10]. However, the algorithm in [10] extends to even more general assignment problems in which the sets of items allowed in a bin are further constrained; for such allocation problems it is shown in [10] that it is NP-hard to obtain an approximation ratio of $1 - 1/e + \epsilon$ for any constant $\epsilon > 0$. Our framework also extends to this wider class of assignment problems and hence $1 - 1/e$ is the best approximation factor one can achieve with this approach.

1.1 Preliminaries

Given a submodular function $f : N \to \mathcal{R}^+$ and $A \subset N$, the function f_A defined by $f_A(S) = f(S \cup A) - f(A)$ is also submodular. Further, if f is monotone, f_A is also monotone. For $i \in N$, we abbreviate $S \cup \{i\}$ by $S + i$. By $f_A(i)$, we denote the "marginal value" $f(A+i) - f(A)$. Submodularity is equivalent to $f_A(i)$ being non-increasing as a function of A for every fixed i.

Given a matroid $\mathcal{M} = (N, \mathcal{I})$, we denote by $r_{\mathcal{M}}$ the rank function of $\mathcal{M}$ where $r_{\mathcal{M}}(A) = \max\{|S| : S \subseteq A, S \in \mathcal{I}\}$. The rank function is monotone and submodular. We denote by $P(\mathcal{M})$ the polytope associated with $\mathcal{M}$; this is the set of all real vectors $y \in [0,1]^N$ that satisfy the constraints: $y(S) \leq r_{\mathcal{M}}(S)\ \forall S \subseteq N$, where $y(S) = \sum_{i \in S} y_i$. Edmonds showed that the vertices of $P(\mathcal{M})$ are precisely the characteristic vectors of the independent sets of $\mathcal{M}$. Further, given a membership oracle for $\mathcal{M}$ (that is given $S \subseteq N$, the oracle answers if $S \in \mathcal{I}$ or not), one can optimize over $P(\mathcal{M})$.

A base of $\mathcal{M}$ is a set $S \in \mathcal{I}$ such that $r_{\mathcal{M}}(S) = r_{\mathcal{M}}(N)$. The base polytope $B(\mathcal{M})$ of $\mathcal{M}$ is given by $\{y \in P(\mathcal{M}) \mid y(N) = r_{\mathcal{M}}(N)\}$. The extreme points of $B(\mathcal{M})$ are the characteristic vectors of the bases of $\mathcal{M}$. Given the problem $\max_{S \in \mathcal{I}} f(S)$, where $\mathcal{M} = (N, \mathcal{I})$ is a matroid, there always exists an optimum solution S^* where S^* is a base of $\mathcal{M}$. Note that this is false if f is not monotone. Thus, for monotone f, it is equivalent to consider the problem $\max_{S \in \mathcal{B}} f(S)$ where $\mathcal{B}$ is the set of bases of $\mathcal{M}$. See [16] for more details on matroids and polyhedral aspects.

2 Pipage Rounding Framework

Ageev and Sviridenko [1] developed an elegant technique for rounding solutions of linear and non-linear programs that they called "pipage rounding". Subsequently, Srinivasan [17] and Gandhi *et al.* [15] interpreted some applications of pipage rounding as a deterministic variant of dependent randomized rounding. In a typical scenario, randomly rounding a fractional solution of a linear program does not preserve the feasibility of constraints, in particular equality constraints. Nevertheless, the techniques of [1, 17, 15] show that randomized rounding can be applied in a certain controlled way to guide a solution that respects certain class of constraints. In particular these techniques were used to round fractional solutions to the assignment problem. In this paper we show that the rounding framework applies quite naturally to our problem. Further, our analysis also reveals the important role of submodularity in this context.

We now describe the pipage rounding framework as adapted to our problem. We follow [1] in spirit although our notation and description is somewhat different and tailored to our application; given a monotone submodular function $f : 2^N \to \mathcal{R}^+$ and a matroid $\mathcal{M} = (N, \mathcal{I})$, we wish to solve $\max_{S \in \mathcal{I}} f(S)$. Let $y_i \in \{0, 1\}$ be a variable that indicates whether i is picked in a solution to the problem. Then $\max_{S \in \mathcal{I}} f(S)$ can be written as the following problem: $\max\{f(y) : y \in P(\mathcal{M}), y \in \{0, 1\}^N\}$. As we observed in Section 1.1, this is equivalent to $\max\{f(y) : y \in B(\mathcal{M}), y \in \{0, 1\}^N\}$ where $B(\mathcal{M})$ is the base polytope of $\mathcal{M}$.

The framework relies on the ability to solve a relaxation of the problem in polynomial time. To obtain a relaxation we let $y \in [0, 1]^N$. This also requires us to find an *extension* of f to a function $\tilde{f} : [0, 1]^N \to \mathcal{R}^+$ such that the problem $\max\{\tilde{f}(y) : y \in P(\mathcal{M})\}$ can be solved in polynomial time. We require two properties of the extension: (i) $\tilde{f}(y) = f(y)$ for all $y \in \{0, 1\}^N$, and (ii) monotonicity, that is $\tilde{f}(y) \geq \tilde{f}(z)$, for all $y \geq z$; $y, z \in [0, 1]^N$. Note that the optimum value of the relaxation is at least the integral optimum solution denoted by OPT. Given an optimum fractional solution y^* to the relaxation, our goal is to round y^* to an integer solution z such that $f(z) \geq \alpha \tilde{f}(y^*) \geq \alpha \mathrm{OPT}$. Clearly the quality of the relaxation depends on the extension function $\tilde{f}$. The rounding framework relies on a potential function $F : [0, 1]^N \to \mathcal{R}^+$, derived from f, that guides the rounding and at the same time allows one to derive bounds on the quality of the approximation. The reason to consider $\tilde{f}$ and F separately will become clear later. Assuming the existence of $\tilde{f}$ and F, we describe the pipage rounding algorithm for our problem.

Given $y \in [0,1]^n$ we say that i is fractional in y if $0 < y_i < 1$. For $y \in P(\mathcal{M})$, a set $A \subseteq N$ is *tight* if $y(A) = r_{\mathcal{M}}(A)$. The following useful proposition follows easily from the submodularity of the rank function $r_{\mathcal{M}}$.

Proposition 1. *If A and B are two tight sets with respect to y then $A \cap B$ and $A \cup B$ are also tight with respect to y.*

The monotonicity of $\tilde{f}$ also implies the following.

Proposition 2. *There exists an optimum solution y^* to $\max\{\tilde{f}(y) : y \in P(\mathcal{M})\}$ such that $y^*(N) = \sum_{i \in N} y_i^* = r_{\mathcal{M}}(N)$.*

Alternatively we can solve the problem $\max\{\tilde{f}(y) : y \in B(\mathcal{M})\}$ which would automatically ensure that $y^*(N) = r_{\mathcal{M}}(N)$. We are interested in tight sets that contain a fractional variable. Observe that a tight set with a fractional variable has at least two fractional variables. Given a tight set A with fractional variables i, j, we let $y_{ij}(\epsilon)$ be the vector obtained by adding ϵ to y_i and subtracting ϵ from y_j and leaving the other values unchanged. Let $\epsilon_{ij}^+(y) = \max\{\epsilon \geq 0 \mid y_{ij}(\epsilon) \in P(\mathcal{M})\}$. Similarly we let $\epsilon_{ij}^-(y) = \min\{\epsilon \leq 0 \mid y_{ij}(\epsilon) \in P(\mathcal{M})\}$. We let $y_{ij}^+ = y_{ij}(\epsilon_{ij}^+)$ and $y_{ij}^- = y_{ij}(\epsilon_{ij}^-)$. For a given y and $i, j \in N$, we define a real-valued function $F_{ij}^y : [\epsilon_{ij}^-(y), \epsilon_{ij}^+(y)] \rightarrow \mathcal{R}^+$ where $F_{ij}^y(\delta) = F(y_{ij}(\delta))$.

Algorithm **PipageRound(y):**
  While (y is not integral) do
    Let A be a minimal tight set containing fractional $i, j \in A$
    If $(F(y_{ij}^+) \geq F(y_{ij}^-))$ $\quad y \leftarrow y_{ij}^+$
    Else $\quad y \leftarrow y_{ij}^-$
  EndWhile
  Output y, $f(y)$.

Lemma 2. *The pipage rounding algorithm outputs an integral feasible y in $O(n^2)$ iterations. Given an oracle access to F and a membership oracle for $\mathcal{M}$, the algorithm can be implemented in polynomial time.*

Proof (sketch). Using Proposition 2, we assume that N is tight with respect to y. Since y_{ij}^+ and y_{ij}^- both belong to $P(\mathcal{M})$, the algorithm maintains the invariant that $y \in P(\mathcal{M})$ and that N is tight. Thus there is always a tight set with two fractional variables as long as y is not integral. We observe that the algorithm does not alter a variable y_i once $y_i \in \{0,1\}$. To simplify the algorithm's analysis we can alter it slightly so that the set A that is picked in each iteration is not only minimal but also of minimum cardinality among such minimal sets. Let $y(h)$ be the vector y at the beginning of iteration h. We claim that $y(h+n-1)$ has at least one more integral variable than $y(h)$. This will give us the desired bound of $O(n^2)$ on the total number of iterations.

To prove the claim, let A_h be the tight set picked by the algorithm, and $i_h, j_h \in A_h$ the two fractional variables modified in iteration h. If one of them becomes integral in $y(h+1)$, we are done. Otherwise we claim that $|A_{h+1}| < |A_h|$, hence after $n-1$ iterations we are guaranteed to have one more integral variable. To see that $|A_{h+1}| < |A_h|$, assume wlog that $y(h+1) = y(h)^+_{i_h j_h}$; since i_h, j_h are still fractional, there is a new tight set B with respect to $y(h+1)$, which prevented us from going further. B contains exactly one of i_h, j_h, otherwise $y(B)$ does not change in iteration h. From Proposition 1, it follows that $B \cap A_h$ is also tight, it contains a fractional variable, and $|B \cap A_h| < |A_h|$. In the next iteration, we can use $A_{h+1} = B \cap A_h$. To implement an iteration, we need to compute y^+_{ij}, y^-_{ij} and the new tight set in polynomial time. These can be done by appealing to known methods [16]. We defer the details to a full version of the paper.

To obtain a guarantee on the quality of the solution, F needs to satisfy some properties, as suggested in [1].

- F is an extension of f and $F(y) \geq \alpha \tilde{f}(y)$ for all $y \in [0,1]^N$.
- F^y_{ij} is *convex* for all y and i, j.

Given the above two conditions, it is shown in [1] that the pipage rounding algorithm yields the following: given an optimum fractional solution y^*, the rounding yields an integral solution z such that $F(z) \geq F(y^*)$. This follows from the convexity requirement on F^y_{ij}; either $F(y^+_{ij}) \geq F(y)$ or $F(y^-_{ij}) \geq F(y)$ and the choice of the algorithm ensures that in each iteration the value of F does not decrease. Therefore we can conclude that $f(z) = F(z) \geq F(y^*) \geq \alpha \tilde{f}(y^*)$. Since $\tilde{f}(y^*) \geq \mathrm{OPT}$, we have $f(z) \geq \alpha \mathrm{OPT}$.

3 Extensions of submodular functions

In this section, we address the issue of extending a monotone submodular function $f : 2^N \rightarrow \mathcal{R}^+$ to continuous functions $\tilde{f}, F : [0,1]^N \rightarrow \mathcal{R}^+$, as required by the framework.

F as the expected value of f: We consider a simple and natural candidate for F that is implicitly generated from f. Define $F(y) = \mathbf{E}[f(\hat{y})]$ where $\hat{y}$ is a random integer vector obtained from y by independently rounding each i to 1 with probability y_i and to 0 with probability $1 - y_i$. In shorthand, we write $F = \mathbf{E}f$. We can evaluate $F = \mathbf{E}f$ to any desired accuracy by taking several independent samples. We defer details that show that a polynomial number of samples suffice to obtain a $(1 - 1/poly(n))$-approximation to $F(y)$. Alternatively we could use a randomized version of the pipage rounding that does not require us to evalute F explicitly.

In [1], F was given as an explicit function for some simple functions and the convexity of F^y_{ij} was explicitly shown. A nice feature of $F = \mathbf{E}f$ is that the convexity requirement is satisfied for all submodular f.

Lemma 3. *For any submodular f, if $F = \mathbf{E}f$, then F^y_{ij} is convex for all $y \in [0,1]^N$ and $i, j \in N$.*

Proof. Let $F = \mathbf{E}f$. For $S \subseteq N \setminus \{i,j\}$ and $y \in [0,1]^N$, let $p_y(S) = \prod_{l \in S} y_l \prod_{l \in N\setminus\{i,j\}\setminus S}(1-y_l)$ be the probability that S is precisely the set obtained by randomized rounding on $N \setminus \{i,j\}$. Then

$$F(y) = \sum_{S \subseteq N\setminus\{i,j\}} p_y(S)\ \big((1-y_i)(1-y_j)f(S) + (1-y_i)y_j f(S+j)\big)$$
$$+ y_i(1-y_j)f(S+i) + y_i y_j f(S+i+j)).$$

We have $F^y_{ij}(\delta) = F(y_{ij}(\delta))$. Let $x = y_{ij}(\delta)$, i.e. $x_i = y_i + \delta$, $x_j = y_j - \delta$ and $x_l = y_l$ for all $l \in N\setminus\{i,j\}$. Hence it follows that $p_x(S) = p_y(S)$ for $S \subseteq N\setminus\{i,j\}$. It can be seen that $F(y_{ij}(\delta)) = F(x) = c_2\delta^2 + c_1\delta + c_0$ where c_2, c_1, c_0 do not depend on δ (they depend only on y and f). Thus to show that $F^y_{ij}(\delta)$ is convex in δ, it is sufficient to prove that $c_2 \geq 0$. It is easy to check that

$$c_2 = \sum_{S \subseteq N\setminus\{i,j\}} p_y(S)(-f(S) + f(S+j) + f(S+i) - f(S+i+j)).$$

By submodularity, $f(S+i)+f(S+j) \geq f((S+i)\cap(S+j)) + f((S+i)\cup(S+j)) = f(S) + f(S+i+j)$ which proves that $c_2 \geq 0$.

Next, we need an extension $\tilde{f}$ such that $\max\{\tilde{f}(y) : y \in P(\mathcal{M})\}$ can be solved in polynomial time. The approximation guarantee is the largest α such that $F(y) \geq \alpha \tilde{f}(y)$.

Extension f^+: Our first candidate for $\tilde{f}$ is an extension similar to the objective function of the "Configuration LP" [10, 7, 8].

– $f^+(y) = \max\left\{\sum_{S\subseteq N} \alpha_S f(S) : \sum_S \alpha_S \leq 1, \alpha_S \geq 0\ \&\ \forall j; \sum_{S:j\in S} \alpha_S \leq y_j\right\}$.

Extension f^*: Another candidate is a function appearing in [14] and subsequently [9, 18, 19], where it is used indirectly in the analysis of the greedy algorithm for submodular function maximization:

– $f^*(y) = \min\left\{f(S) + \sum_{j\in N} f_S(j) y_j : S \subseteq N\right\}$.

Unfortunately, as the theorem below shows, it is NP-hard to evaluate $f^+(y)$ and $f^*(y)$ and also to optimize them over matroid polytopes.

Theorem 3. *It is NP-hard to compute $f^+(y)$ or $f^*(y)$ for a given $y \in [0,1]^n$ and a given monotone submodular function f. Also, there is $\delta > 0$ such that for a given matroid $\mathcal{M}$ it is NP-hard to find any point $z \in P(\mathcal{M})$ such that $f^+(z) \geq (1-\delta)\max\{f^+(y) : y \in P(\mathcal{M})\}$. Similarly, it is NP-hard to find any point $z \in P(\mathcal{M})$ such that $f^*(z) \geq (1-\delta)\max\{f^*(y) : y \in P(\mathcal{M})\}$. These results hold even for coverage-type submodular functions and partition matroids.*

We defer the proof to a full version of the paper; the authors are unaware of prior work that might have addressed this question. Still, both $f^+(y)$ and $f^*(y)$ will be useful in our analysis. We remark that for any class of submodular functions where either $f^+(y)$ or $f^*(y)$ is computable in polynomial time, we obtain a $(1-1/e)$-approximation for our problem.

It is known and easy to see that for $y \in \{0,1\}^N$, both f^+ and f^* functions coincide with f and thus they are indeed extensions of f. For any $y \in [0,1]^N$, we first show the following.

Lemma 4. *For any monotone submodular f, $F(y) \leq f^+(y) \leq f^*(y)$.*

Proof. To see the first inequality, let $\alpha_S = \prod_{i \in S} y_i \prod_{i \notin S}(1-y_i)$ be the probability that we obtain $\hat{y} = \chi_S$ by independent rounding of y. Since $\sum_{S: j \in S} \alpha_S = \Pr[\hat{y}_j = 1] = y_j$, this is a feasible solution for $f^+(y)$ and therefore $f^+(y) \geq \sum_S \alpha_S f(S) = \mathbf{E}[f(\hat{y})] = F(y)$.

For the second inequality, consider any feasible vector α_S and any set $T \subseteq N$:

$$\sum_S \alpha_S f(S) \leq \sum_S \alpha_S \left(f(T) + \sum_{j \in S} f_T(j) \right) \leq f(T) + \sum_{j \in N} y_j f_T(j)$$

using submodularity and the properties of α_S. By taking the maximum on the left and the minimum on the right, we obtain $f^+(y) \leq f^*(y)$.

It is tempting to conjecture that $f^+(y)$ and $f^*(y)$ are in fact equal, due to some duality relationship. However, this is not the case: both inequalities in Lemma 4 can be sharp and both gaps can be close to $1-1/e$. For the first inequality, consider the submodular function $f(S) = \min\{|S|, 1\}$ and $y_j = 1/n$ for all j; then $F(y) = 1-(1-1/n)^n$ and $f^+(y) = 1$. For the second inequality, choose a large but fixed k, $f(S) = 1-(1-|S|/n)^k$ and $y_j = 1/k$ for all j. The reader can verify that $f^+(y) = 1-(1-1/k)^k$, while $f^*(y) \geq 1-k/n \to 1$ as $n \to \infty$. We prove that $1-1/e$ is the worst possible gap for both inequalities. Moreover, even the gap between $F(y)$ and $f^*(y)$ is bounded by $1-1/e$.

Lemma 5. *For any monotone submodular f, $F(y) \geq \left(1-\frac{1}{e}\right) f^*(y)$.*

Proof. For each element $j \in N$, set up an independent Poisson clock $\mathcal{C}_j$ of rate y_j, i.e. a device which sends signals at random times, in any infinitesimal time interval of size dt independently with probability $y_j dt$. We define a random process which starts with an empty set $S(0) = \emptyset$ at time $t = 0$. At any time when the clock $\mathcal{C}_j$ sends a signal, we include element j in S, which increases its value by $f_S(j)$. (If j is already in S, nothing happens; the marginal value $f_S(j)$ is zero in this case.) Denote by $S(t)$ the random set we have at time t. By the definition of a Poisson clock, $S(1)$ contains element j independently with probability $1-e^{-y_j} \leq y_j$. Since such a set can be obtained as a subset of the random set defined by $\hat{y}$, we have $\mathbf{E}[f(S(1))] \leq F(y)$ by monotonicity. We show that $\mathbf{E}[f(S(1))] \geq (1-1/e)f^*(y)$ which will prove the claim.

Let $t \in [0,1]$. Condition on $S(t) = S$ and consider how $f(S(t))$ changes in an infinitesimal interval $[t, t+dt]$. The probability that we include element j is $y_j dt$.

Since dt is very small, the events for different elements j are effectively disjoint. Thus the expected increase of $f(S(t))$ is (up to $O(dt^2)$ terms)

$$\mathbf{E}[f(S(t+dt)) - f(S(t)) \mid S(t) = S] = \sum_{j \in N} f_S(j) y_j dt \geq (f^*(y) - f(S))dt$$

using the definition of $f^*(y)$. We divide by dt and take the expectation over S:

$$\frac{1}{dt}\mathbf{E}[f(S(t+dt)) - f(S(t))] \geq f^*(y) - \mathbf{E}[f(S(t))].$$

We define $\phi(t) = \mathbf{E}[f(S(t))]$, i.e. $\frac{d\phi}{dt} \geq f^*(y) - \phi(t)$. We solve this differential inequality by considering $\psi(t) = e^t\phi(t)$ and $\frac{d\psi}{dt} = e^t(\frac{d\phi}{dt} + \phi(t)) \geq e^t f^*(y)$. Since $\psi(0) = \phi(0) = 0$, this implies

$$\psi(x) = \int_0^x \frac{d\psi}{dt} dt \geq \int_0^x e^t f^*(y) dt = (e^x - 1) f^*(y)$$

for any $x \geq 0$. We conclude that $\mathbf{E}[f(S(t))] = \phi(t) = e^{-t}\psi(t) \geq (1 - e^{-t})f^*(y)$ and $F(y) \geq \mathbf{E}[f(S(1))] \geq (1 - 1/e)f^*(y)$.

We remark that we did not actually use submodularity in the proof of Lemma 5! Formally, it can be stated for all monotone functions f. However, $f^*(y)$ is not a proper extension of f when f is not submodular (e.g., $f^*(y)$ is identically zero if $f(S) = 0$ for $|S| \leq 1$). So the statement of Lemma 5 is not very meaningful in this generality.

To summarize what we have proved so far, we have two relaxations of our problem:

- $\max\{f^+(y) : y \in P(\mathcal{M})\}$
- $\max\{f^*(y) : y \in P(\mathcal{M})\}$

Our framework together with Lemma 4 and Lemma 5 implies that both of these relaxations have integrality gap at most $1 - 1/e$. Theorem 3 shows NP-hardness of solving the relaxations. We show how to use the framework efficiently in a restricted case of interest which is described in the following section.

4 Sums of weighted rank functions

We achieve a $(1 - 1/e)$-approximation, under a matroid constraint $\mathcal{M}$, for any submodular function f that can be expressed as a sum of "weighted rank functions" of matroids. This is the most general subclass of submodular functions for which we are able to use the framework outlined in Section 2 in an efficient way. Here we describe this in detail.

Weighted rank functions of matroids: Given a matroid $(N, \mathcal{X})$ and a weight function $w : N \to \mathcal{R}^+$, we define a *weighted rank function* $g : 2^N \to \mathcal{R}^+$,

$$g(S) = \max\{\sum_{j \in I} w_j : I \subseteq S \ \& \ I \in \mathcal{X}\}.$$

It is well known that such a function is monotone and submodular. A simple special case is when $\mathcal{X} = \{I \mid |I| = 1\}$. Then $g(S)$ returns simply the maximum-weight element of S; this will be useful in our application to GAP.

Sums of weighted rank functions: We consider functions $f : 2^N \to \mathcal{R}^+$ of the form $f(S) = \sum_{i=1}^m g_i(S)$ where each g_i is a weighted rank function for matroid $(N, \mathcal{X}_i)$ with weights w_{ij}. Again, $f(S)$ is monotone and submodular.

The functions that can be generated in this way form a fairly rich subclass of monotone submodular functions. In particular, they generalize submodular functions arising from coverage systems. Coverage-type submodular functions can be obtained by considering a simple uniform matroid $(N, \mathcal{X})$ with $\mathcal{X} = \{I \subseteq N \mid |I| \leq 1\}$. For a collection of sets $\{A_j\}_{j \in N}$ on a ground set $[m]$, we can define m collections of weights on N, where $w_{ij} = 1$ if A_j contains element i, and 0 otherwise. Then the weighted rank function $g_i(S) = \max\{w_{ij} : j \in S\}$ is simply an indicator of whether $\bigcup_{j \in S} A_j$ covers element i. The sum of the rank functions $g_i(S)$ gives exactly the size of this union $f(S) = \sum_{i=1}^m g_i(S) = \left|\bigcup_{j \in S} A_j\right|$. Generalization to the weighted case is straightforward.

LP formulation for sums of weighted rank functions: For a submodular function given as $f(S) = \sum_{i=1}^m g_i(S)$ where $g_i(S) = \max\{w_i(I) : I \subseteq S, I \in \mathcal{X}_i\}$, consider an extension $g_i^+(y)$ for each g_i, as defined in Section 3:

$$g_i^+(y) = \max\{\sum_{S \subseteq N} \alpha_S g_i(S) : \sum_S \alpha_S \leq 1, \alpha_S \geq 0 \ \& \ \forall j; \sum_{S : j \in S} \alpha_S \leq y_j\}.$$

Here, we can assume without loss of generality that α_S is nonzero only for $S \in \mathcal{X}_i$ (otherwise replace each S by a subset $I \subseteq S, I \in \mathcal{X}_i$, such that $g_i(S) = w_i(I)$). Therefore, g_i^+ can be written as

$$g_i^+(y) = \max\{\sum_{I \in \mathcal{X}_i} \alpha_I \sum_{j \in I} w_{ij} : \sum_{I \in \mathcal{X}_i} \alpha_I \leq 1, \alpha_I \geq 0 \ \& \ \forall j; \sum_{I \in \mathcal{X}_i : j \in I} \alpha_I \leq y_j\}.$$

We can set $x_{ij} = \sum_{I \in \mathcal{X}_i : j \in I} \alpha_I$ and observe that a vector $x_i = (x_{ij})_{j \in N}$ can be obtained in this way if and only if it is a convex linear combination of independent sets; i.e., if it is in the matroid polytope $P(\mathcal{X}_i)$. The objective function becomes $\sum_{j \in N} w_{ij} \sum_{I \in \mathcal{X}_i : j \in I} \alpha_I = \sum_{j \in N} w_{ij} x_{ij}$ and so we can write equivalently

$$g_i^+(y) = \max\{\sum_{j \in N} w_{ij} x_{ij} : x_i \in P(\mathcal{X}_i) \ \& \ \forall j; x_{ij} \leq y_j\}.$$

We sum up these functions to obtain an extension $\tilde{f}(y) = \sum_{i=1}^m g_i^+(y)$. This leads to the following LP formulation for the problem $\max\{\tilde{f}(y) : y \in P(\mathcal{M})\}$:

We can solve the LP using the ellipsoid method, since a separation oracle can be efficiently implemented for each matroid polytope, and therefore also for this LP. To obtain a $(1-1/e)$-approximation (Theorem 1) via the above LP using the pipage rounding framework from Section 2, it is sufficient to prove the following lemma.

$$\begin{aligned} &\max \sum_{i=1}^{m} \sum_{j \in N} w_{ij} x_{ij}; \\ &\forall i, j; x_{ij} \leq y_j, \\ &\forall i; x_i \in P(\mathcal{X}_i), \\ &y \in P(\mathcal{M}). \end{aligned}$$

Lemma 6. *For any sum of weighted rank functions f, $F(y) \geq (1-1/e)\tilde{f}(y)$.*

Proof. By Lemma 5, $F(y) \geq (1-1/e)f^*(y)$ and hence it suffices to prove that $f^*(y) \geq \tilde{f}(y)$. By Lemma 4, $g_i^+(y) \leq g_i^*(y)$ where $g_i^*(y) = \min_{S_i}(g_i(S_i) + \sum_j y_j g_{i,S_i}(j))$. (Here, $g_{i,S_i}(j) = g_i(S_i + j) - g_i(S_i)$.) Consequently,

$$\begin{aligned} \tilde{f}(y) &= \sum_{i=1}^{m} g_i^+(y) \leq \sum_{i=1}^{m} \min_{S_i}(g_i(S_i) + \sum_{j \in N} y_j g_{i,S_i}(j)) \\ &\leq \min_S \sum_{i=1}^{m}(g_i(S) + \sum_{j \in N} y_j g_{i,S}(j)) = \min_S(f(S) + \sum_{j \in N} y_j f_S(j)) = f^*(y). \end{aligned}$$

5 The Generalized Assignment Problem

Here we consider an application of our techniques to the Generalized Assignment Problem ("GAP"). An instance of GAP consists of n bins and m. Each item i has two non-negative numbers for each bin j; a value v_{ji} and a size s_{ji}. We seek an assignment of items to bins such that the total size of items in each bin is at most 1, and the total value of all items is maximized.

In [10], a $(1-1/e)$-approximation algorithm for GAP has been presented. The algorithm uses LP_1.

In LP_1, $\mathcal{F}_j$ denotes the collection of all feasible assignments for bin j, i.e. sets satisfying $\sum_{i \in S} s_{ji} \leq 1$. The variable $x_{j,S}$ represents bin j receiving a set of items S. Although this is an LP of exponential size, it is shown in [10] that it can be solved to an arbitrary precision in polynomial time. Then the fractional solution can be rounded to an integral one to obtain a $(1-1/e)$ approximation.

$$\begin{aligned} LP_1: \quad &\max \sum_{j, S \in \mathcal{F}_j} y_{j,S} v_j(S); \\ &\forall j; \sum_{S \in \mathcal{F}_j} y_{j,S} \leq 1, \\ &\forall i; \sum_{j, S \in \mathcal{F}_j : i \in S} y_{j,S} \leq 1, \\ &\forall j, S;\ y_{j,S} \geq 0. \end{aligned}$$

We show in this section that this $(1-1/e)$-approximation algorithm can be interpreted as a special case of submodular maximization subject to a matroid constraint[6], and this framework also allows some generalizations of GAP[7]. For this purpose, we reformulate the problem as follows.

We define $N = \{(j,S) \mid 1 \leq j \leq n,\ S \in \mathcal{F}_j\}$ and a submodular function $f : 2^N \to \mathcal{R}^+$,

$$f(\mathcal{S}) = \sum_{i=1}^{m} \max\{v_{ji} : \exists (j,S) \in \mathcal{S}, i \in S\}.$$

We maximize this function subject to a matroid constraint $\mathcal{M}$, where $\mathcal{S} \in \mathcal{M}$ iff $\mathcal{S}$ contains at most one pair (j,S) for each j. Such a set $\mathcal{S}$ corresponds to an assignment of set S to bin j for each $(j,S) \in \mathcal{S}$. This is equivalent to GAP: although the bins can be assigned overlapping sets in this formulation, we only count the value of the most valuable assignment for each item. We can write $f(\mathcal{S}) = \sum_{i=1}^{m} g_i(\mathcal{S})$ where $g_i(\mathcal{S}) = \max\{v_{ji} : \exists (j,S) \in \mathcal{S}, i \in S\}$ is a weighted rank function of a matroid $\mathcal{X}_i$ on N. In the matroid $\mathcal{X}_i$ an element $(j,S) \in N$ has weight v_{ji} if $i \in S$ and 0 otherwise. A set is independent in $\mathcal{X}_i$ iff its cardinality is at most 1. Therefore the problem falls under the umbrella of our framework.

We now write explicitly the LP arising from interpreting GAP as a submodular function problem. We have variables $y_{j,S}$ for each j and $S \in \mathcal{F}_j$. In addition, for each matroid $\mathcal{X}_i$, we define copies of these variables $x_{i,j,S}$. The resulting linear program is given as LP_2.

$$LP_2 : \quad \max \sum_{j,S \in \mathcal{F}_j, i \in S} v_{ji} x_{i,j,S}; \quad \forall i,j,S;\ x_{i,j,S} \leq y_{j,S}, \quad \forall i;\ x_i \in P(\mathcal{X}_i), \quad y \in P(\mathcal{M}).$$

LP_2 has exponentially many variables and exponentially many constraints. However, observe that a feasible solution $y_{j,S}$ for LP_1 is also feasible for LP_2, when we set $x_{i,j,S} = y_{j,S}$ for $i \in S$ and 0 otherwise. This is because the constraint $\sum_{j,S:i \in S} y_{j,S} \leq 1$ in LP_1 implies $x_i \in P(\mathcal{X}_i)$, and the constraint $\sum_S y_{j,S} \leq 1$ implies $y \in P(\mathcal{M})$. Therefore, we can solve LP_1 using the techniques of [10] and then convert the result into a feasible solution of LP_2. Finally, we can apply the pipage rounding technique to obtain a $(1-1/e)$-approximation.

This is simply a reformulation of the algorithm from [10]. However, the flexibility of our framework allows a more complicated matroid constraint $\mathcal{M}$ than each bin choosing at most one set. We briefly discuss this below.

Laminar matroid constraints on the bins: Let B be the set of bins in a GAP instance. Consider a *laminar* matroid $\mathcal{M}$ on B. We consider the problem of assigning items to a subset of bins $B' \subseteq B$ such that B' is independent in $\mathcal{M}$. An example is when $\mathcal{M}$ is the simple uniform matroid; that is B' is independent iff $|B'| \leq k$. This gives rise to a variant of GAP in which at most k of the n bins

[6] This formulation of GAP is also described in [10] as personal communication from an author of this paper.

[7] In [10] more general allocation problems are considered that allow constraints on the sets of items packable within a bin. Our approach also works for such problems but in this extended abstract we limit our discussion to GAP.

can be used. One can modify LP_1 by adding a new constraint: $\sum_{j,S\in\mathcal{F}_j} y_{j,S} \leq k$, to obtain a relaxation LP_3 for this new problem.

$$\begin{aligned} LP_3: \quad \max \sum_{j,S\in\mathcal{F}_j} y_{j,S} v_j(S); \\ \forall j; \sum_{S\in\mathcal{F}_j} y_{j,S} \leq 1, \\ \forall i; \sum_{j,S\in\mathcal{F}_j : i\in S} y_{j,S} \leq 1, \\ \sum_{j,S\in\mathcal{F}_j} y_{j,S} \leq k, \\ \forall j,S;\ y_{j,S} \geq 0. \end{aligned}$$

Using the same ideas as those in [10], one can solve LP_3 to an arbitrary precision in polynomial time. The simple rounding scheme of [10] for LP_1 does not apply to LP_3. However, as before, we can see that a solution to LP_3 is feasible for LP_2 where the matroid $\mathcal{M}$ now also enforces the additional constraint that at most k elements from N are chosen. Thus pipage rounding can be used to obtain a $(1-1/e)$-approximation. A similar reasoning allows us to obtain a $(1-1/e)$-approximation for any laminar matroid constraint on the bins B. We defer the details to a full version of the paper.

6 Conclusions

We obtained a $(1-1/e)$-approximation for an interesting and useful class of submodular functions. We note that the methods in the paper apply to some interesting submodular functions that are not in the class. An example is the maximum multiset multicover problem which generalizes the multicover problem defined in Section 1. The difference between multicover and multiset multicover is that a set can cover an element multiple times (at most the requirement of the element). We can obtain a $(1-1/e)$ approximation for this problem even though this function cannot be expressed as a weighted sum of matroid rank functions. We defer the details. It would be of much interest to prove or disprove the existence of a $(1-1/e)$-approximation for all monotone submodular functions. Note that our hardness results (Theorem 3) hold even when f can be expressed as a sum of weighted rank functions of matroids, yet we can obtain a $(1-1/e)$-approximation in this case.

The unconstrained problem $\max_{S\subseteq N} f(S)$ is NP-hard and hard to approximate if f is a *non-monotone* submodular set function; the Max-Cut problem is a special case. However, the pipage rounding framework is still applicable to non-monotone functions (as already shown in [1]). For non-monotone functions, the problem we need to consider is $\max_{S\in\mathcal{B}} f(S)$ where $\mathcal{B}$ is the set of bases of $\mathcal{M}$. It is easy to see that Lemma 2 and Lemma 3 still apply. Thus, the approximation ratio that can be guaranteed depends on the extension $\tilde{f}$.

Pipage rounding [1] and dependent randomized rounding [17, 15] are based on rounding fractional solutions to the assignment problem into integer solutions while maintaining the quality of a solution that is a function of the variables on the edges of the underlying bipartite graph. A number of applications are given in [1, 17, 15]. This paper shows that submodularity and uncrossing properties of solutions to matroids and other related structures are the basic ingredients in the applicability of the pipage rounding technique. We hope this insight will lead to more applications in the future.

References

1. A. Ageev and M. Sviridenko. Pipage rounding: a new method of constructing algorithms with proven performance guarantee. *J. of Combinatorial Optimization*, 8:307–328, 2004.
2. C. Chekuri and A. Kumar. Maximum coverage problem with group budget constraints and applications. *Proc. of APPROX*, Springer LNCS, 72–83, 2004.
3. C. Chekuri and M. Pál. A recursive greedy algorithm for walks in directed graphs. *Proc. of IEEE FOCS*, 2005.
4. C. Chekuri and S. Khanna. A PTAS for the multiple knapsack problem. *SIAM J. on Computing*, 35(3):713–728, 2004.
5. G. Cornuejols, M. Fisher and G. Nemhauser. Location of bank accounts to optimize float: an analytic study of exact and approximate algorithms. *Management Science*, 23: 789–810, 1977.
6. U. Feige. A threshold of $\ln n$ for approximating set cover. *JACM*, 45(4):634–652, 1998
7. U. Feige. On maximizing welfare when utility functions are subadditive. *Proc. of ACM STOC*, 41–50, 2006.
8. U. Feige and J. Vondrák. Approximation algorithms for allocation problems: Improving the Factor of $1 - 1/e$. *Proc. of IEEE FOCS*, 667–676, 2006.
9. M. L. Fisher, G. L. Nemhauser and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions - II. *Math. Prog. Study*, 8:73–87, 1978.
10. L. Fleischer, M.X. Goemans, V.S. Mirrokni and M. Sviridenko. Tight approximation algorithms for maximum general assignment problems. *Proc. of ACM-SIAM SODA*, 611–620, 2006.
11. E. Hazan, S. Safra and O. Schwartz. On the complexity of approximating k-set packing. *Proc. of APPROX*, 2003.
12. T. A. Jenkyns. The efficiency of the "greedy" algorithm. *Proc. of 7th South Eastern Conference on Combinatorics, Graph Theory and Computing*, 341–350, 1976.
13. B. Korte and D. Hausmann. An analysis of the greedy heuristic for independence systems. *Annals of Discrete Math.*, 2:65–74, 1978.
14. 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.
15. R. Gandhi, S. Khuller, S. Parthasarathy and A. Srinivasan. Dependent rounding and its applications to approximation algorithms. *JACM*, 53(3):324–360, 2006.
16. A. Schrijver. Combinatorial optimization - polyhedra and efficiency. Springer, 2003.
17. A. Srinivasan. Distributions on level-sets with applications to approximation algorithms. *Proc. of IEEE FOCS*, 588–597, 2001.
18. L. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2:385–393, 1982.
19. L. Wolsey. Maximizing real-valued submodular functions: Primal and dual heuristics for location Problems. *Math. of Operations Research*, 7:410–425, 1982.

A Analysis of Greedy for p-independent families

We consider the problem $\max_{S \in \mathcal{I}} f(S)$ where $f : 2^N \rightarrow \mathcal{R}^+$ is a monotone submodular function and $\mathcal{I} \subseteq N$ is a p-independent family. $\mathcal{I}$ is a p-independent family if for each $S \in \mathcal{I}$ and $e \in N \setminus S$, there exist at most $k \leq p$ elements $e_1, e_2, \ldots, e_k \in S$ such that $S \cup \{e\} - \{e_1, \ldots, e_k\} \in \mathcal{I}$. Note that the intersection of p matroids on N yields a p-independence family. The class is more general as can be seen by the example of matchings in hypergraphs of maximum degree p. Fisher, Nemhauser and Wolsey [9] showed that the natural greedy algorithm has a tight approximation ratio of $1/(p+1)$. In [9], a proof is given for a special case when the p-system $\mathcal{I}$ is induced by the intersection of p matroids on N; it is mentioned that the proof extends to an arbitrary p-system. The proof in [9] extends the analysis of Jenkyns [12], and Korte and Hausmann [13], who showed a bound of $1/p$ if f is modular. Here we give a proof of the result in [9] using a somewhat more combinatorial approach. We draw on the proof in [2] for a (very) special case of the problem.

For $A \subset N$, we let f_A denote the submodular function defined by $f_A(X) = f(A \cup X) - f(A)$. We first define the greedy algorithm formally.

Algorithm **Greedy:**
  $S \leftarrow \emptyset$, $A \leftarrow \emptyset$
  repeat
    $A \leftarrow \{e \mid S \cup \{e\} \in \mathcal{I}\}$
    If $(A \neq \emptyset)$ then
      $e \leftarrow \text{argmax}_{e' \in A} f_S(e')$
      $S \leftarrow S \cup \{e\}$
    Endif
  until $(A = \emptyset)$
  Output S

It is easy to see the greedy algorithm can be implemented using valuation oracles for f and $\mathcal{I}$. We let $e_1, e_2, \ldots, e_k$ be the elements added to S by Greedy and for $i \geq 0$, we let S_i denote the set $\{e_1, e_2, \ldots, e_i\}$. Let $\delta_i = f(S_i) - f(S_{i-1})$ be the improvement in the solution value when e_i is added. Note that $f(S_k) = \sum_i \delta_i$. We now prove that Greedy yields a $1/(p+1)$ approximation. Fix some optimum solution O. We show the existence of mutually disjoint sets $O_1, O_2, \ldots, O_k$, whose union is $O \setminus S_k$, with the following property: for $1 \leq i \leq k$, $\delta_i \geq \frac{1}{p} f_{S_k}(O_i)$. Assume that we have a partition of $O \setminus S_k$ with this property. Then,

$$
\begin{aligned}
f(S_k) = \sum_{i=1}^{k} \delta_i \; &\geq \; \sum_{i=1}^{k} \frac{1}{p} f_{S_k}(O_i) \\
&\geq \frac{1}{p} f_{S_k}(\cup_i O_i) \; \geq \; \frac{1}{p} f_{S_k}(O) \\
&\geq \frac{1}{p}(f(O) - f(S_k)) \\
&\geq \frac{1}{p+1} f(O).
\end{aligned}
$$

We use submodularity in the second step above. We now prove the desired property by iteratively constructing sets $O = T_0, T_1, \ldots, T_k = S_k$ such that for $1 \leq i \leq k$, (i) T_i is independent, (ii) $S_i \subseteq T_i$ and (iii) $O_i = T_{i-1} \setminus T_i$. In other words we start with O and transform it into S_k by adding the elements $e_1, e_2, \ldots, e_k$ and maintaining independence. We obtain T_i from T_{i-1} as follows. Consider the set $X_i = T_{i-1} \cup \{e_i\}$. If X_i is independent we let $T_i = X_i$ in which case $O_i \subseteq \{e_i\}$ and $f_{S_k}(O_i) = 0$. Otherwise, X_i is not independent. Since $\mathcal{I}$ is a p-independent family, there exist $p_i \leq p$ elements $e_i^1, e_i^2, \ldots, e_i^{p_i} \in T_{i-1}$ such that $T_{i-1} \cup \{e_i\} - \{e_i^1, e_i^2, \ldots, e_i^{p_i}\}$ is independent. We let $O_i = \{e_i^1, e_i^2, \ldots, e_i^{p_i}\}$ and $T_i = X_i \setminus O_i$. Note that $|O_i| \leq p$ and that T_i is independent. By construction $S_i \subseteq T_i$. Since each e_i^j was a candidate choice for Greedy in iteration i, it follows that for $1 \leq j \leq p_i$,

$$\delta_i \geq f_{S_{i-1}}(e_i^j) \geq f_{S_k}(e_i^j).$$

and hence

$$\delta_i \geq \frac{1}{p_i} \sum_j f_{S_k}(e_i^j) \geq \frac{1}{p_i} f_{S_k}(O_i) \geq \frac{1}{p} f_{S_k}(O_i).$$

This finishes the proof.

Approximate Greedy: We observe that in many settings the oracle for f is approximate. It is easy to adapt the above analysis to show the following. If we have an α-approximate oracle for f (here $\alpha \leq 1$) then the Greedy algorithm would yield an approximation of $\alpha/(p+\alpha)$ approximation for the problem.

B A sampling lemma for submodular functions

Here, we show a corollary of Lemma 5 which is useful in some applications.

Lemma 7. *Let $f : 2^N \to \mathcal{R}^+$ be a monotone submodular function, $A_1, A_2, \ldots, A_k \subseteq N$ and $p_1, \ldots, p_k \geq 0$ such that $\sum_{i=1}^k p_i = 1$. Let $S = A_1(p_1) \cup A_2(p_2) \cup \ldots \cup A_k(p_k)$ where $A_i(p_i)$ are independently sampled random sets, and each item of A_i appears in $A_i(p_i)$ independently with probability p_i. Then*

$$\mathbf{E}[f(S)] \geq \left(1 - \frac{1}{e}\right) \sum_{i=1}^{k} p_i f(A_i).$$

Proof. For each $j \in N$, let $y_j = \sum_{i:j\in A_i} p_i$. By the definition of $f^+(y)$ (see Section 3), we have $\sum_{i=1}^{k} p_i f(A_i) \le f^+(y)$.

The probability that j appears in $S = A_1(p_1) \cup A_2(p_2) \cup \ldots \cup A_k(p_k)$ is $1 - \prod_{i:j\in A_i}(1-p_i) \ge 1 - e^{-y_j}$. Recall the random set $S(1)$ defined in the proof of Lemma 5 which contains element j independently with probability $1 - e^{-y_j}$. By monotonicity, we get $\mathbf{E}[f(S)] \ge \mathbf{E}[f(S(1))]$. The proof of Lemma 5 shows that $\mathbf{E}[f(S(1))] \ge (1-1/e)f^*(y)$. By Lemma 4, this also implies $\mathbf{E}[f(S(1))] \ge (1-1/e)f^+(y)$ which proves the claim.

We note that Lemma 7 can be proved directly, even without the assumption of independence among the elements of each $A_i(p_i)$. The important assumption is independence between different sets $A_i(p_i)$. We omit the proof from this extended abstract.

Applications.

- This lemma provides an alternative proof of the $1-1/e$ approximation factor for sums of weighted rank functions (Lemma 6). This follows directly by considering $\tilde{f}(y) = \sum_i g_i^+(y)$ and applying Lemma 7 to each $g_i^+(y)$ separately.
- The lemma provides a very simple way to achieve a $(1-1/e)$-approximation from the configuration LP (used in [7, 8]; see also Section 5) under submodular utility functions. It can be shown that allocating each item i independently with probabilities $p_{ji} = \sum_{S:i\in S} y_{j,S}$ to bin (or "player") j provides a $(1-1/e)$-approximation. Note that this technique cannot be used for GAP, since capacity constraints could be violated.
- The lemma can be used to prove other interesting statements which are not immediately obvious, such as the following: In a graph G with edge weights w_e, consider an assignment of edge variables x_e such that x is in the forest polytope of G, i.e. for any vertex set W, $\sum_{e\in E(W)} x_e \le |W| - 1$. Let R be a random set of edges where e appears independently with probability x_e and let F be the maximum-weight forest in R. Then $\mathbf{E}[w(F)] \ge (1 - 1/e)\sum_e w_e x_e$.

C Hardness of $f^+(y)$

C.1 Computing $f^+(y)$

Here we show that it is NP-hard to compute $f^+(y)$. Our reduction is indirect. We show that the separation oracle for the dual of the LP for a slight generalization of $f^+(y)$ is NP-hard. We obtain the desired result using strong duality and the equivalence of optimization and separation.

Recall that $f^+(y) = \max\left\{\sum_{S\subseteq N} \alpha_S f(S) : \sum_S \alpha_S \le 1, \alpha_S \ge 0 \;\&\; \forall j; \sum_{S:j\in S} \alpha_S \le y_j\right\}$.

We consider a new problem $f^+(y_0, y) = \max\left\{\sum_{S\subseteq N} \alpha_S f(S) : \sum_S \alpha_S \le y_0, \alpha_S \ge 0 \;\&\; \forall j; \sum_{S:j\in S} \alpha_S \le y_j\right\}$ which slightly generalizes $f^+(y)$. Note that $f^+(y) = f^+(1, y)$. Our first observation is that $f^+(y_0, y)$ can be reduced to computing $f^+(y)$ as follows. Given

(y_0, y), we can assume without loss of generality that $y_i \leq y_0$ for $i \in N$. We observe that for any scalar $\rho > 0$, $f^+(\rho y_0, \rho y) = \rho f^+(y_0, y)$ and hence $f^+(y_0, y) = y_0 f^+(1, y/y_0) = y_0 f^+(y/y_0)$.

The dual of the LP for $f^+(y_0, y)$ is given by

$$\begin{aligned} \min \quad & y_0 z + \sum_{j \in N} y_j \gamma_j \\ & z + \sum_{j \in S} \gamma_j \geq f(S) \\ & z, \gamma \geq 0. \end{aligned}$$

Thus the separation oracle for the dual corresponds to solving the following problem: given a submodular function f on N and non-negative weights γ on N and a non-negative z, is there a set S such that $z < f(S) - \gamma(S)$. We now show that this decision problem is NP-Complete via a reduction from the vertex cover problem; given a graph $G = (N, E)$ and an integer k, is there a vertex cover of size k in G? We define a submodular function f on N as follows: for $S \subseteq N$, $f(S)$ is the number of edges with an end point in S. For $j \in N$ let $\gamma_j = 1/2$ and let $z = |E| - (k+1)/2$. We claim that there is a set S with $z < f(S) - \gamma(S)$ iff G has a vertex cover of size at most k. To see this let S^* be a set that achieves $\max_{S \subseteq N}(f(S) - \gamma(S))$. We observe that $f(S^*) = |E|$ for otherwise we can add a vertex v to S^* that covers one more edge and therefore

$$f(S^* + v) - \gamma(S^* + v) \geq f(S^*) + 1 - (|S^*| + 1)/2 > f(S^*) - |S^*|/2,$$

contradicting the choice of S^*. Thus $\max_{S \subseteq N}(f(S) - \gamma(S)) = |E| - k^*/2$ where k^* is the size of the minimum vertex cover in G. It follows that there is a set S such that $z < f(S) - \gamma(S)$ iff $k^* \leq k$. This proves that the separation oracle for the dual is NP-Complete.

By the equivalence of separation and optimization, optimizing the dual is NP-hard; note that (y_0, y) the coefficients of the objective function are arbitrary. By strong duality, this implies that computing $f^+(y_0, y)$ is NP-hard and we have shown a polynomial-time reduction from $f^+(y_0, y)$ to $f^+(y)$ and hence this proves the hardness of $f^+(y)$.

C.2 Optimizing $f^+(y)$

We have shown that it is NP-hard to compute $f^+(y)$ for a given monotone submodular function f and vector $y \in [0,1]^n$. However, it is still not obvious that we cannot optimize $f^+(y)$ over a matroid polytope $P(\mathcal{M})$, in the sense of finding a point $y \in P(\mathcal{M})$ where $f^+(y)$ is (approximately) optimal, possibly without computing the optimal value itself. Here we show that this problem is also NP-hard, even for coverage-type functions f.

Proposition 3. *There is $\delta > 0$ such that for a given monotone submodular function f and matroid $\mathcal{M}$, it is NP-hard to find any point $y^* \in P(\mathcal{M})$ such that*

$$f^+(y^*) \geq (1 - \delta) \max\{f^+(y) : y \in P(\mathcal{M})\}.$$

Proof. We use the following hardness result for the max-k-cover problem [6, **?**]:

For any $\epsilon > 0$, given a collection of sets $\{S_i \subseteq U \mid i \in N\}$ where $|S_i| = s$ and $|U| = ks$, it is NP-hard to distinguish the case where U can be covered by k disjoint sets S_i and the case where any collection of k sets covers at most $(1 - 1/e + \epsilon)|U|$ elements.

Given such a collection of sets, we define a coverage function $f(A) = |\bigcup_{i \in A} S_i|$. A natural matroid here is the uniform matroid $\mathcal{M} = \{I \subseteq N \mid |I| \leq k\}$, i.e. $P(\mathcal{M}) = \{y \in [0,1]^N \mid \sum y_i \leq k\}$.

Note that if there are k sets indexed by $A, |A| = k$, such that $\bigcup_{i \in A} S_i = U$, then $\mathbf{1}_A \in P(\mathcal{M})$ and $f^+(\mathbf{1}_A) = f(A) = |U|$. On the other hand, if there are no such k sets, then $\max\{f^+(y) : y \in P(\mathcal{M})\} < |U|$, because $f^+(y)$ for any $y \in P(\mathcal{M})$ is a convex linear combination $\sum \alpha_A f(A)$ where $\sum \alpha_A |A| \leq k$, and therefore all sets in the linear combination cannot satisfy $f(A) = |U|$. This proves that it is NP-hard to compute $\max\{f^+(y) : y \in P(\mathcal{M})\}$. However, we want to prove NP-hardness even for the problem of finding a (near-)optimal point y, without computing its value.

We extend the set system by k additional sets $T_1, \ldots, T_k$, each containing $(1 - \frac{1}{4e})s$ elements disjoint from everything else. The index set becomes $N' = N \cup K$, $|K| = k$. Our goal is to show that in case of a YES instance (k sets covering U), the optimum is attained for a vector y supported by N. In case of a NO instance, it is more profitable to use the sets $T_1, \ldots, T_k$ and therefore the optimal point is $y = \mathbf{1}_K$. By careful analysis, we get hardness even for finding an approximately optimal point.

First, consider a YES instance, where $f(A) = |U| = ks$ for $A \subseteq N, |A| = k$. Then $f^+(\mathbf{1}_A) = ks$ is clearly an optimum over $P(\mathcal{M})$, because $f(B) \leq |B|s$ for any $B \subseteq N'$ and hence $f^+(y) \leq \sum_{i \in N'} y_i s \leq ks$ for $y \in P(\mathcal{M})$. Moreover, any vector y such that $\sum_{i \in K} y_i > 4e\delta k$ has $f^+(y) < (1 - \delta)ks$, because each element $i \in K$ can contribute only $(1 - \frac{1}{4e})s$ rather than s. Therefore, any $(1 - \delta)$-approximate optimum satisfies $\sum_{i \in K} y_i \leq 4e\delta k$.

Next, consider a NO instance, i.e. for any $A \subseteq N, |A| \leq k$, we have $f(A) \leq (1 - 1/e + \epsilon)ks$. By submodularity, we get $f(A) \leq (1 - 1/e + \epsilon)|A|s$ for $|A| > k$, since otherwise we could find a subset $A' \subseteq A$ of size k and value $f(A') > (1 - 1/e + \epsilon)ks$. We also have $f(A) \leq |A|s$ for any $|A| < (1 - 1/e + \epsilon)k$. By combining these 3 bounds in the ranges of $|A|$ between $[(1 - 1/e + \epsilon)k, k]$, above k and below $(1 - 1/e + \epsilon)k$, we obtain

$$f(A) \leq (1 - 1/e + \epsilon)(|A| + (1/e - \epsilon)k)s$$

for all $A \subseteq N$. Since this bound is linear in $|A|$, it implies a similar bound for $f^+(y), y \in [0,1]^N$: Assume $\sum_{i \in N} y_i = a$. Then $f^+(y)$ is a convex linear combination $\sum_{A \subseteq N} \alpha_A f(A)$ where $\sum_{A \subseteq N} \alpha_A |A| = a$ and $\sum_{A \subseteq N} \alpha_A = 1$. Consequently,

$$f^+(y) \leq (1 - 1/e + \epsilon)(a + (1/e - \epsilon)k)s.$$

Now assume $y \in [0,1]^{N \cup K}$ where $\sum_{i \in N} y_i = a$ and $\sum_{i \in K} y_i = k - a$. Since each element of K contributes $(1 - \frac{1}{4e})s$, independently of other elements, we have

$$f^+(y) \leq (1 - \frac{1}{e} + \epsilon)(a + (\frac{1}{e} - \epsilon)k)s + (k - a)(1 - \frac{1}{4e})s \leq (1 + \frac{3}{4e} - \frac{1}{e^2})ks - (\frac{3}{4e} - \epsilon)as.$$

If $a \geq (1-4e\delta)k$, we get $f^+(y) \leq (1-\frac{1}{e^2}+O(\epsilon+\delta))ks$, while $f^+(\mathbf{1}_K) = (1-\frac{1}{4e})ks$ is a substantially larger value. Therefore, any near-optimal point $y \in P(\mathcal{M})$ must satisfy $\sum_{i \in K} y_i = k - a > 4e\delta k$.

If we could find any $(1-\delta)$-approximate optimum $y \in P(\mathcal{M})$, we could test whether $\sum_{i \in K} y_i$ is above or below $4e\delta k$ and hence distinguish YES and NO instances.

D Hardness of $f^*(y)$

D.1 Computing $f^*(y)$

In this section, we show that it is NP-hard to compute $f^*(y)$.

Proposition 4. *Let $G = (V, E)$ be a graph. For $S \subseteq V$, let $c(S)$ denote the number of edges incident with S, and let $f(S) = |S| + c(S)$. Then there is $\delta > 0$ such that it is NP-hard to $(1-\delta)$-approximate $f^*(y)$ for a given graph G and $y = (3/4, \ldots, 3/4)$.*

Note that $c(S)$ is a coverage-type submodular function and hence $f(S)$ also, by adding one distinct element to each set in the system.

Proof. Let's analyze the extension $f^*(y) = \min_S f(S) + \sum_j y_j f_S(j)$ for this particular submodular function. Here, $f_S(j) = 1 + c_S(j)$ where $c_S(j)$ is the number of edges incident with j but not with S. For $y = (3/4, \ldots, 3/4)$, we get

$$f^*(y) = \min_S |S|+c(S)+\frac{3}{4}\sum_{j \in V \setminus S}(1+c_S(j)) = \min_S |S|+c(S)+\frac{3}{4}|V \setminus S|+\frac{3}{2}|E(\bar{S})|.$$

Here, $E(\bar{S})$ denotes the edges disjoint from S, which get a contribution of $3/4$ from each endpoint. Using $c(S) + |E(\bar{S})| = |E|$ and joining the terms, we get

$$f^*(y) = \min_S \frac{3}{4}|V| + \frac{3}{2}|E| + \frac{1}{4}|S| - \frac{1}{2}c(S).$$

Note that if S doesn't cover all edges, we can decrease this expression by including another vertex in S, covering a new edge. The minimum can be attained only when all edges are covered by S and $c(S) = |E|$. Therefore, $f^*(y) = \frac{3}{4}|V| + |E| + \frac{1}{4}k^*$ where k^* is the minimum vertex cover of G. Since this number is NP-hard to compute, so is $f^*(y)$. In fact, vertex cover is APX-hard, even for instances where $|E| = O(|V|)$, and hence $f^*(y)$ is also APX-hard.

D.2 Optimizing $f^*(y)$

It remains to show that it is also NP-hard to find any point $y \in P(\mathcal{M})$ approximately optimizing $f^*(y)$ over $P(\mathcal{M})$ (see the discussion in Section C.2).

Proposition 5. *There is $\delta > 0$ such that for a given monotone submodular function f and matroid $\mathcal{M}$, it is NP-hard to find any point $y^* \in P(\mathcal{M})$ such that*

$$f^*(y^*) \geq (1-\delta)\max\{f^*(y) : y \in P(\mathcal{M})\}.$$

Proof. We use the following hardness result on Vertex Cover in r-uniform hypergraphs [**?**]:

For any $\epsilon > 0$ and $r > 2/\epsilon$ constant, it is NP-hard for a given r-uniform hypergraph H on n vertices to distinguish whether the minimum vertex cover in H has size at most ϵn or at least $(1-\epsilon)n$.

Here, the reduction is somewhat less intuitive. We denote the vertices of H by V, edges by E, and we define our ground set to be $N = V' \cup V''$, a union of two disjoint copies of V. In other words, we produce two clones j', j'' for each vertex $j \in V$. Our submodular function is the following:

$$f(S) = |S \cap V'| + 2|S \cap V''| + c(S)$$

where $c(S)$ is the number of edges $e \in E$ such that S contains some clone of a vertex in e. In other words, $c(S)$ is the vertex-cover function for a derived hypergraph where each vertex is replaced by two clones.

We prove that it is NP-hard to optimize $f^*(y)$ over $P(\mathcal{M})$ for the following matroid:

$$\mathcal{M} = \{I \subseteq V' \cup V'' \mid \forall j \in V; |I \cap \{j', j''\}| \leq 1\},$$

i.e. we allow at most one clone of each vertex to be chosen. The matroid polytope $P(\mathcal{M})$ is defined by the constraints $y_{j'} + y_{j''} \leq 1$ for each vertex j. Since the maximum of a monotone function over $P(\mathcal{M})$ must be attained on the basis face of $P(\mathcal{M})$, we can assume that $y_{j'} + y_{j''} = 1$ for any candidate for an optimum.

Observe that maximizing $f(S)$ over independent sets $S \in \mathcal{M}$ yields $f(V'') = 2n + |E|$, which does not depend on the structure of H in any way. However, $f^*(y)$ for fractional vectors $y \in P(\mathcal{M})$ will depend on the minimum vertex cover of H in a way similar to the previous section. Then the value can be possibly larger than $2n + |E|$. The intuition here is that if the minimum vertex cover is very small, the gain for a fractional vector is not significant and it is more profitable to take $y = \mathbf{1}_{V''}$ where $f^*(y) = 2n + |E|$. On the other hand, if the minimum vertex cover is very large, the gain for a fractional solution outweighs the loss incurred by the fact that elements of V' contribute 1 rather than 2 to $f(S)$. Then we can achieve a value close to $\frac{7}{3}n + |E|$, for $y_{j'} = 1/3, y_{j''} = 2/3$. (The reader can verify that this is the exact optimum when the minimum vertex cover is $|V| = n$.) Note that the optimal point does not necessarily indicate what the minimum vertex cover is, but testing how close the optimal point is to being integral can help us decide about the size of the minimum vertex cover.

Let's write $f^*(y)$ in the following form:

$$f^*(y) = \min_S f(S) + \sum_{i \in N} y_i f_S(i) = \min_S |S \cap V'| + 2|S \cap V''| + c(S) + y(V' \setminus S) + 2y(V'' \setminus S) + \sum_{e: e \cap S = \emptyset} y(e).$$

Again, we want to argue that the minimum is attained for a vertex cover. We assume that $y_{j'} + y_{j''} = 1$ for each pair of clones, i.e. we get $y(e) = r > 4$ for each edge. Therefore, covering a new edge by including a new vertex in S can only decrease the subject of minimization and

$$f^*(y) = \min_{\text{vertex cover } S} |S \cap V'| + 2|S \cap V''| + y(V' \setminus S) + 2y(V'' \setminus S) + |E|.$$

Consider a YES instance, where H has a vertex cover of size ϵn. We can set $y_{j'} = 0, y_{j''} = 1$ for each $j \in V$ and we get $f^*(y) = f(V'') = 2n + |E|$. We do not claim that this is exactly the optimum but we claim that any approximately optimal point must be close to $\mathbf{1}_{V''}$. Suppose that for at least $n/2$ vertices, we have $y_{j'} \geq 1/6$ and $y_{j''} \leq 5/6$. Then choosing $S \subseteq V'$ to be a vertex cover of size at most ϵn shows that

$$\begin{aligned} f^*(y) &= |S \cap V'| + 2|S \cap V''| + y(V' \setminus S) + 2y(V'' \setminus S) + |E| \leq \epsilon n + y(V') + 2y(V'') + |E| \\ &\leq \epsilon n + \frac{n}{2}\left(\frac{1}{6} + 2 \cdot \frac{5}{6}\right) + \frac{n}{2}(0 + 2 \cdot 1) + |E| \leq \left(2 - \frac{1}{12} + \epsilon\right) n + |E|. \end{aligned}$$

For sufficiently small ϵ, this is smaller than the optimum. Thus for any near-optimal point $y \in P(\mathcal{M})$, more than $n/2$ vertices must satisfy $y_{j'} < 1/6, y_{j''} > 5/6$.

On the other hand, consider a NO instance, where any vertex cover has size at least $(1-\epsilon)n$. Here, let's choose $y^*_{j'} = 1/3, y^*_{j''} = 2/3$ for each $j \in V$. This gives

$$f^*(y^*) = |S \cap V'| + 2|S \cap V''| + \frac{1}{3}|V' \setminus S| + \frac{4}{3}|V'' \setminus S| + |E| = \frac{2}{3}|S| + \frac{1}{3}|V'| + \frac{4}{3}|V''| + |E|$$

for some vertex cover S. Since $|S| \geq (1-\epsilon)n$, we have

$$f^*(y^*) \geq \frac{2}{3}(1-\epsilon)n + \frac{1}{3}n + \frac{4}{3}n + |E| = \left(\frac{7}{3} - \frac{2}{3}\epsilon\right) n + |E|.$$

We claim that any near-optimal point y must be close y^*. Suppose on the contrary that there are at least $n/2$ vertices for which $y_{j'} \leq 1/6$ and $y_{j''} \geq 5/6$. Let

$$S = \{j' \in V' \mid y_{j'} \geq 1/3\} \cup \{j'' \in V'' \mid y_{j'} < 1/3\}.$$

This is a vertex cover, containing exactly one clone of each vertex. Note that we have $y_{j'} < 1/3$ for $j' \in V' \setminus S$, and at least $n/2$ of these coordinates in fact satisfy $y_{j'} \leq 1/6$. For $j'' \in V'' \setminus S$, we have $y_{j''} \leq 2/3$. Therefore

$$\begin{aligned} f^*(y) &\leq |S \cap V'| + 2|S \cap V''| + y(V' \setminus S) + 2y(V'' \setminus S) + |E| \\ &\leq |S \cap V'| + 2|S \cap V''| + \left(\frac{1}{3}|V' \setminus S| - \frac{n}{2} \cdot \frac{1}{6}\right) + \frac{4}{3}|V'' \setminus S| + |E| \\ &= |S| + |V''| + \frac{1}{3}|(V' \cup V'') \setminus S| - \frac{1}{12}n + |E| = \frac{7}{3}n - \frac{1}{12}n + |E| \end{aligned}$$

which is smaller than $f^*(y^*)$. Therefore, for any near-optimal vector, more than $n/2$ vertices must satisfy $y_{j'} > 1/6, y_{j''} < 5/6$. If we could find such a near-optimal vector y, we could distinguish YES and NO instances by testing how many vertices satisfy this condition. We note that the hard instances of H in [**?**] have constant (albeit large) degrees, therefore $|E| = O(n)$ and the hardness gap is a constant factor.

E Submodular welfare maximization

Here we describe another application of our framework, to the *social welfare maximization* problem. First, we review the problem and known results.

Social welfare maximization. *Given a set X of m items, and n players, each of which has a monotone utility function $w_i : 2^X \to \mathcal{R}^+$. The goal is to partition X into disjoint subsets $S_1, \ldots, S_n$ in order to maximize the social welfare $\sum_{i=1}^{n} w_i(S_i)$.*

This problem arises in combinatorial auctions and has been studied intensively in recent years [**?**,**?**,**?**,7, 8]. Before studying its complexity status, one needs to specify how the algorithm accesses the input of the problem. Usually, an algorithm is allowed to ask certain queries about the players' utility functions. This leads to different *oracle models* of computation. The two types of oracle most commonly considered are:

- *Value oracle:* returns the value of $w_i(S)$ for a given player i and $S \subseteq X$.
- *Demand oracle:* returns the set S maximizing $w_i(S) - \sum_{j \in S} p_j$ for a given player i and an assignment of prices p_j.

Previous work. In general, the problem is NP-hard to approximate within a factor of $m^{1/2-\epsilon}$ for any $\epsilon > 0$, even in the demand oracle model. Positive results have been achieved only under strong assumptions on the utility functions. A particular case of interest is when the utility functions are submodular.

In the value oracle model, the greedy algorithm achieves a $1/2$-approximation for all submodular functions [**?**]. On the hardness side, it has been shown that a $(1 - 1/e + \epsilon)$-approximation for any fixed $\epsilon > 0$ would imply $P = NP$ [**?**]. This hardness result holds in particular for submodular functions induced by explicitly given set coverage systems. It is not known whether a $(1 - 1/e)$-approximation can be achieved for all submodular functions.

In the demand oracle model, on the other hand, a $(1 - 1/e)$-approximation was presented in [**?**]. This algorithm has been subsequently generalized to all fractionally subadditive functions [7] and improved to $1 - 1/e + \delta, \delta > 0$, for submodular functions [8].

Our result. Our framework gives almost immediately a $(1-1/e)$-approximation algorithm in the case of utility functions in the form of a sum of weighted rank functions of matroids. We need a value oracle for each of the weighted rank functions, a demand oracle is not required. As a special case, this includes the coverage-type utility functions for which this approximation factor is optimal [**?**]. We do not provide a $(1-1/e)$-approximation for all submodular functions; this question seems to be intimately tied to the question of $(1-1/e)$-approximate maximization of submodular functions subject to a matroid constraint.

The reduction: We describe here how the submodular welfare maximization problem fits in our framework. For a given set of items X and number of players n, we define a ground set $N = [n] \times X$. The elements of N can be viewed as clones of items, one clone of each item for each player. For each player i, we define a mapping $\pi_i : 2^N \to 2^X$,

$$\pi_i(S) = \{j \in X : (i,j) \in S\},$$

which simply takes all the clones in S corresponding to player i. Given utility functions $w_1, \ldots, w_n : 2^X \to \mathcal{R}^+$, we define a function $f : 2^N \to \mathcal{R}^+$,

$$f(S) = \sum_{i=1}^{n} w_i(\pi_i(S)).$$

It can be seen that if w_i is submodular, then $w_i \circ \pi_i$ is submodular, and hence f is submodular as well.

The problem of partitioning $X = S_1 \cup S_2 \cup \ldots \cup S_n$ so that $\sum_{i=1}^n w_i(S_i)$ is maximized is equivalent to finding $S = \bigcup_{i=1}^n (\{i\} \times S_i) \subseteq N$, containing at most 1 clone of each item, so that $f(S)$ is maximized. Sets of this type form a partition matroid:

$$\mathcal{I} = \{S \subseteq N \mid \forall j; |\{i : (i,j) \in S\}| \leq 1\}.$$

Therefore, the welfare maximization problem is equivalent to maximizing $f(S)$ subject to $S \in \mathcal{I}$.

If we assume that $w_i : 2^X \to \mathcal{R}^+$ is a sum of weighted rank functions of matroids $g_{i1}, \ldots, g_{ik}$, the functions $g_{ij} \circ \pi_i$ are again weighted rank functions of matroids on N. Therefore, $f = \sum_{i=1}^n w_i \circ \pi_i = \sum_{i=1}^n \sum_{j=1}^k g_{ij} \circ \pi_i$ is a sum of weighted rank functions as well. Our framework implies that if a value oracle is available for each of the weighted rank functions g_{ij}, then we can find a $(1-1/e)$-approximation to the problem $\max_{S \in \mathcal{I}} f(S)$ in polynomial time. This $(1-1/e)$-approximates the submodular welfare problem.

We conclude by observing that by this reduction, an α-approximation for maximizing any given class of monotone functions (closed under taking sums) subject to a matroid constraint, implies the same α-approximation for the maximum social welfare problem and the same class of utility functions.