Submodular Maximization over Multiple Matroids via Generalized Exchange Properties

Jon Lee
IBM T.J. Watson Research Center, Yorktown Heights, NY
email: jonlee@us.ibm.com

Maxim Sviridenko
IBM T.J. Watson Research Center, Yorktown Heights, NY
email: sviri@us.ibm.com

Jan Vondrák
IBM Almaden Research Center, San Jose, CA
email: jvondrak@us.ibm.com

Submodular-function maximization is a central problem in combinatorial optimization, generalizing many important NP-hard problems including Max Cut in digraphs, graphs and hypergraphs, certain constraint satisfaction problems, maximum-entropy sampling, and maximum facility-location problems. Our main result is that for any $k \geq 2$ and any $\varepsilon > 0$, there is a natural local-search algorithm that has approximation guarantee of $1/(k+\varepsilon)$ for the problem of maximizing a monotone submodular function subject to k matroid constraints. This improves upon the $1/(k+1)$-approximation of Fisher, Nemhauser and Wolsey, obtained in 1978. Also, our analysis can be applied to the problem of maximizing a linear objective function and even a general non-monotone submodular function subject to k matroid constraints. We show that in these cases the approximation guarantees of our algorithms are $1/(k-1+\varepsilon)$ and $1/(k+1+1/(k-1)+\varepsilon)$, respectively.

Our analyses are based on two new exchange properties for matroids. One is a generalization of the classical Rota Exchange Property for matroid bases, and another is an exchange property for two matroids, based on the structure of matroid intersection.

Key words: matroid ; submodular function ; approximation algorithm

MSC2000 Subject Classification: Primary: 68W25 , 68R05 ; Secondary: 05B35 , 90C27

OR/MS subject classification: Primary: Analysis of Algorithms , Mathematics/Combinatorics

---

Introduction.[1] In this paper, we consider the problem of maximizing a non-negative submodular function f, defined on a (finite) ground set N, subject to matroid constraints. A function $f : 2^N \to \mathbb{R}$ is *submodular* if for all $S, T \subseteq N$, $f(S \cup T) + f(S \cap T) \leq f(S) + f(T)$. Furthermore, all submodular functions that we deal with are assumed to be non-negative. Throughout, we assume that our submodular function f is given by a *value oracle*; i.e., for a given set $S \subseteq N$, an algorithm can query an oracle to find the value $f(S)$. Without loss of generality, we take the ground set N to be $[n] = \{1, 2, \ldots, n\}$.

We assume some familiarity with matroids (see [29]) and associated algorithmics (see [31]). Briefly, to set our notation, we denote a matroid $\mathcal{M}$ by an ordered pair $(N, \mathcal{I})$, where N is the (finite) ground set of $\mathcal{M}$, and $\mathcal{I}$ is the set of independent sets of $\mathcal{M}$. We recall that the defining properties for the set $\mathcal{I}$ of independent sets of a matroid are as follows: (i) $\emptyset \in \mathcal{I}$; (ii) $X \subset Y \in \mathcal{I} \Rightarrow X \in \mathcal{I}$; (iii) $X, Y \in \mathcal{I}$, $|X| > |Y|$ $\Rightarrow \exists\ e \in X \setminus Y$ such that $Y \cup \{e\} \in \mathcal{I}$. For a given matroid $\mathcal{M}$, the associated *matroid constraint* is: $S \in \mathcal{I}(\mathcal{M})$. In our usage, we deal with k matroids $\mathcal{M}_i = (N, \mathcal{I}_i)$, $i = 1, \ldots, k$, on the common ground set N. We assume that each matroid is given by an *independence oracle*, answering whether or not $S \in \mathcal{I}_i$. It is no coincidence that we use N for the ground set of our submodular function f as well as for the ground set of our matroids $\mathcal{M}_i = (N, \mathcal{I}_i)$, $i = 1, \ldots, k$. Indeed, our optimization problem is

$$\max \left\{ f(S) \ : \ S \in \cap_{i=1}^{k} \mathcal{I}_i \right\}.$$

Where necessary, we make some use of other standard matroid operations and notation. For a matroid $\mathcal{M} = (N, \mathcal{I})$, its *rank function* $r_{\mathcal{M}} : 2^N \to \mathbb{Z}_+$ is defined as $r_{\mathcal{M}}(X) := \max\{|Y| \ : \ Y \subset X,\ Y \in \mathcal{I}\}$, $\forall X \subset N$. A *base* of $\mathcal{M}$ is an independent set of maximum cardinality $r_{\mathcal{M}}(N)$. For a set $S \subset N$, we let $\mathcal{M} \backslash S$, $\mathcal{M}/S$, and $\mathcal{M}|S$ denote deletion of S, contraction of S, and restriction to S, respectively. The *deletion* $\mathcal{M} \backslash S$ is the matroid with ground set $N \setminus S$ having as its set of independent sets $\{X \subset N \setminus S \ : \ X \in \mathcal{I}\}$. The *restriction* $\mathcal{M}|S$ is simply $\mathcal{M} \backslash (N \setminus S)$. Contraction is a bit more complicated;

[1]A preliminary version of this work appeared in the proceedings of APPROX 2009

let T be an independent subset of S such that $|T| = r_{\mathcal{M}}(S)$. Then the *contraction* $\mathcal{M}/S$ is the matroid with ground set $N \setminus S$ having as its set of independent sets $\{X \subset N \setminus S \ : \ X \cup T \in \mathcal{I}\}$.

Previous Results Optimization of submodular functions is a central topic in combinatorial optimization (see, e.g. [25, 31]). While submodular minimization is polynomially solvable (see [20, 32]), maximization variants are usually NP-hard because they include either Max Cut, variants of facility location, and set coverage problems.

A classical technique for submodular maximization is the greedy algorithm. The greedy algorithm was first applied to a wide range of submodular maximization problems in the late-70's and early-80's (see [8, 9, 10, 15, 16, 21, 26, 13]). The most relevant result for our purposes is that the greedy algorithm gives a $1/(k+1)$-approximation for the problem of maximizing a monotone submodular function subject to k matroid constraints (see [13]). Due to a simple reduction, this problem also encapsulates the problem of maximizing a linear function subject to $k+1$ matroid constraints.[2] Thus we get a $1/k$-approximation for the problem of maximizing a linear function subject to k matroid constraints, $k \geq 3$. This result appeared first in [15, 16, 21] even in more general setting of k-independence systems. Until recently, the greedy algorithm had the best established performance guarantee for these problems under general matroid constraints.

Recently, improved results have been achieved using the *multilinear extension* of a submodular function and *pipage rounding* (see [1, 5, 33, 6]). In particular, Vondrák [33] designed the continuous greedy algorithm which achieves a $(1-1/e)$-approximation for our problem with $k = 1$, i.e. monotone submodular maximization subject to a single matroid constraint (see also [6]). This result is optimal in the oracle model even for the case of a uniform matroid constraint (see [27]), and also optimal unless $P = NP$ for the special case of maximum coverage (see [11]).

Another algorithmic technique that has been used for submodular maximization is local search. Cornuéjols et al. [9] show that a local-search algorithm achieves a constant-factor approximation guarantee for the maximum uncapacitated facility-location problem which is a special case of submodular maximization. Analogously, Fisher, Nemhauser, Wolsey [26] show a similar result for the problem of maximizing a monotone submodular function subject to a single cardinality constraint (i.e. a uniform matroid constraint). We remark that local search in this case is known to yield only a 1/2-approximation, i.e. it performs worse than the greedy algorithm (see [26]).

The maximum k-dimensional matching problem is a problem of maximizing a linear function subject to k special partition matroid constraints. Improved algorithms for maximum k-dimensional matching have been designed using local search. The best known approximation factors are $2/(k+\varepsilon)$ in the unweighted case (i.e., 0/1 weights), and $2/(k+1+\varepsilon)$ for a general linear function, even in the more general cases of weighted set packing (see [18]) and independent set problems in $(k+1)$-claw free graphs (see [3]). The latter result was obtained after a series of improvements over the basic local-search algorithm (see [2, 7, 3]).

However, general matroid constraints seem to complicate the matter. Of course for $k = 1$, we can efficiently find a maximum-weight set with the greedy algorithm, and for $k = 2$ we can efficiently find a maximum-weight set with augmenting-path type algorithms based on exchange properties (see [31], for example). Prior to this paper, the best approximation for the problem of maximum independent set in the intersection of $k \geq 3$ matroids was $1/k$ (for a recent discussion, see [30]). On the hardness side, it is known that unless $P = NP$, there is no approximation better than $O(\log k/k)$ for k-dimensional matching (see [17]), and hence neither for the intersection of k general matroids. The $1/(k+1)$-approximation for submodular maximization subject to k matroids (see [13]) can be improved in the case when all k constraints correspond to partition matroids. For any fixed $k \geq 2$ and $\varepsilon > 0$, a simple local-search algorithm gives a $1/(k+\varepsilon)$-approximation for this variant of the problem (see [23]). The analysis strongly uses the properties of partition matroids. It is based on relatively simple exchange properties of partition matroids that do not hold in general.

Local-search algorithms were also designed for non-monotone submodular maximization. The best

[2] Given a problem $\max\{w(S) : S \in \bigcap_{i=0}^{k} \mathcal{I}_i\}$ where $w(S)$ is linear, we can equivalently consider the problem $\max\{f(S) : S \in \bigcap_{i=1}^{k} \mathcal{I}_i\}$, where $f(S) = \max\{w(I) : I \subseteq S, I \in \mathcal{I}_0\}$, the *weighted rank function* of $\mathcal{M}_0$, is known to be monotone submodular.

approximation guarantee known for unconstrained submodular maximization is $2/5 - \varepsilon$ (see [12]). For the problem of non-monotone submodular maximization subject to k matroid constraints, the best known approximation is $1/(k + 2 + 1/k + \varepsilon)$, for any constant $k \geq 1$ and $\varepsilon > 0$ (see [23]). For $k = 1$, this gives a $1/(4 + \epsilon)$-approximation; this result has been recently improved to a factor of roughly 0.309, using multilinear relaxation (see [34]).

Our Results and Techniques In this paper we analyze a natural local-search algorithm: Given a feasible solution, i.e. a set S that is independent in each of the k matroids, our local-search algorithm tries to add at most $2p$ elements and delete at most $2kp$ elements from S. If there is a local move that generates a feasible solution and improves the objective value, our algorithm repeats the local-search procedure with that new solution, until no improvement is possible. Our main result is that for $k \geq 2$, every locally-optimal feasible solution S satisfies the inequality

$$(k + 1/p) \cdot f(S) \;\geq\; f(S \cup C) + (k - 1 + 1/p) \cdot f(S \cap C),$$

for every feasible solution C. We also provide an approximate variant of the local-search procedure that finds an approximate locally-optimal solution in polynomial time, while losing a factor of $1 + \varepsilon$ on the left-hand side of the above inequality (Lemma 3.2). Therefore, for any fixed $k \geq 2$ and $\varepsilon > 0$, we obtain a polynomial-time algorithm with approximation guarantee $1/(k + \varepsilon)$ for the problem of maximizing a monotone non-decreasing submodular function subject to k matroid constraints. This algorithm gives a $1/(k - 1 + \varepsilon)$-approximation in the case when the objective function is linear because this problem can be viewed as a weighted rank-function maximization (which is submodular) subject to $k - 1$ matroid constraints; see Corollary 3.1. These results are tight for our local-search algorithm, which follows from [2].

We also obtain an approximation algorithm for non-monotone submodular functions. In this case, one round of local search is not enough, but applying the local search iteratively, as in [23], one can obtain an approximation algorithm with performance guarantee of $1/(k + 1 + \frac{1}{k-1} + \varepsilon)$.

The main technical contributions of this paper are two new exchange properties for matroids. One is a generalization of the classical Rota Exchange Property (Lemma 2.6) and another is an exchange property for the intersection of two matroids (Lemma 2.3), which generalizes an exchange property based on augmenting paths which was used in [23] for partition matroids. We believe that both properties and their proofs are interesting in their own right.

In §1.1, we establish some useful properties of submodular functions. In §2, we establish our exchange properties for matroids. In §3, we describe and analyze our local-search algorithm. In §4, combining our techniques with the iterative local search from [23], we establish improved approximation results for maximizing a general (non-monotone) submodular function subject to $k \geq 2$ matroid constraints. In §5, we show that our analysis is tight for monotone submodular functions.

1. Some Useful Properties of Submodular Functions.

Lemma 1.1 *Let f be a submodular function on N. Let $S, C \subseteq N$ and let $\{T_l\}_{l=1}^t$ be a collection of subsets of $C \setminus S$ such that each element of $C \setminus S$ appears in exactly k of these subsets. Then*

$$\textstyle\sum_{l=1}^t \left(f(S \cup T_l) - f(S)\right) \;\geq\; k\; \left(f(S \cup C) - f(S)\right).$$

Proof. Let $s = |S|$ and $c = |C \cup S|$. We will use the notation $[n]$ to denote the set $\{1, \ldots, n\}$ (by convention $[0] = \emptyset$). Without loss of generality, we can assume that $S = \{1, 2, \ldots, s\}$ and that $C \setminus S = \{s+1, s+2, \ldots, c\}$. Then for any $T \subseteq C \setminus S$, by submodularity: $f(S \cup T) - f(S) \;\geq\; \sum_{p \in T} \left(f([p]) - f([p-1])\right)$. Summing up over all sets T_l, we get

$$\begin{aligned} &\textstyle\sum_{l=1}^t \left(f(S \cup T_l) - f(S)\right) \geq \sum_{l=1}^t \sum_{p \in T_l} \left(f([p]) - f([p-1])\right) \\ &= k \textstyle\sum_{p=s+1}^c \left(f([p]) - f([p-1])\right) = k\left(f(S \cup C) - f(S)\right). \end{aligned}$$

The first equality follows from the fact that each element in $\{s + 1, \ldots, c\}$ appears in exactly k sets T_l, and the second equality follows from a telescoping summation. □

Lemma 1.2 *Let f be a submodular function on N. Let $S' \subseteq S \subseteq N$, and let $\{T_l\}_{l=1}^t$ be a collection of subsets of $S \setminus S'$ such that each element of $S \setminus S'$ appears in exactly k of these subsets. Then*

$$\textstyle\sum_{l=1}^t \left(f(S) - f(S \setminus T_l)\right) \;\leq\; k\; \left(f(S) - f(S')\right).$$

Proof. Let $s = |S|$ and $c = |S'|$. Without loss of generality, we can assume that $S' = \{1, 2, \dots, c\} = [c] \subseteq \{1, 2, \dots, s\} = [s] = S$. For any $T \subseteq S$, $f(S) - f(S \setminus T) \leq \sum_{p \in T} (f([p]) - f([p-1]))$ by submodularity. Using this we obtain

$$\begin{aligned}\sum_{l=1}^{t} (f(S) - f(S \setminus T_l)) &\leq \sum_{l=1}^{t} \sum_{p \in T_l} (f([p]) - f([p-1])) \\ &= k \sum_{p=c+1}^{s} (f([p]) - f([p-1])) = k \ (f(S) - f(S')).\end{aligned}$$

The first equality follows from $S \setminus C = \{c+1, \dots, s\}$ and the fact that each element of $S \setminus C$ appears in exactly k of the sets $\{T_l\}_{l=1}^{t}$. The last equality is due to a telescoping summation. □

2. New Exchange Properties of Matroids.

2.1 Intersection of two matroids. An exchange digraph is a well-known construct for devising efficient algorithms for exact maximization of linear functions over the intersection of two matroids (for example, see [31]). We are interested in submodular maximization, k matroids and approximation algorithms; nevertheless, we are able to make use of such exchange digraphs, once we establish some new properties of them.

Let $\mathcal{M}_l = (N, \mathcal{I}_l)$, $l = 1, 2$, be two matroids on ground set N. For $I \in \mathcal{I}_1 \cap \mathcal{I}_2$, we define two digraphs $D_{\mathcal{M}_1}(I)$ and $D_{\mathcal{M}_2}(I)$ on node set N as follows:

- For each $i \in I, j \in N \setminus I$ with $I \cup \{j\} \setminus \{i\} \in \mathcal{I}_1$, we have an arc (i, j) of $D_{\mathcal{M}_1}(I)$;
- For each $i \in I, j \in N \setminus I$ with $I \cup \{j\} \setminus \{i\} \in \mathcal{I}_2$, we have an arc (j, i) of $D_{\mathcal{M}_2}(I)$.

The arcs in $D_{\mathcal{M}_l}(I)$, $l = 1, 2$, encode valid swaps in $\mathcal{M}_l$.

When we refer to a matching (or perfect matching) in $D_{\mathcal{M}_l}(I)$ for $l = 1, 2$ we mean a matching in an undirected graph where the arcs of the graph $D_{\mathcal{M}_l}(I)$ are treated as undirected edges. We use two known lemmas from matroid theory.

Lemma 2.1 ([31], Corollary 39.12*a*) *If $|I| = |J|$ and $I, J \in \mathcal{I}_l$ for $l = 1, 2$, then $D_{\mathcal{M}_l}(I)$ contains a perfect matching between $I \setminus J$ and $J \setminus I$.*

Lemma 2.2 ([31], Theorem 39.13) *Let $|I| = |J|$, $I \in \mathcal{I}_l$ for $l = 1, 2$, and assume that $D_{\mathcal{M}_l}(I)$ has a unique perfect matching between $I \setminus J$ and $J \setminus I$. Then $J \in \mathcal{I}_l$.*

Next, we define a digraph $D_{\mathcal{M}_1,\mathcal{M}_2}(I)$ on node set N as the union of $D_{\mathcal{M}_1}(I)$ and $D_{\mathcal{M}_2}(I)$. A dicycle in $D_{\mathcal{M}_1,\mathcal{M}_2}(I)$ corresponds to a chain of feasible swaps. Observe that it is not necessarily the case that the entire cycle gives a valid exchange in both matroids. However, Lemma 2.2 implies that if the cycle decomposes into two matchings that are unique in $D_{\mathcal{M}_1}(I)$ and $D_{\mathcal{M}_2}(I)$, respectively, then we can legally perform the exchange corresponding to this cycle. This motivates the following definition.

Definition 2.1 *We call a dicycle C in $D_{\mathcal{M}_1,\mathcal{M}_2}(I)$* irreducible *if $C \cap D_{\mathcal{M}_1}(I)$ is the unique perfect matching in $D_{\mathcal{M}_1}(I)$ and $C \cap D_{\mathcal{M}_2}(I)$ is the unique perfect matching in $D_{\mathcal{M}_2}(I)$ on their vertex set $V(C)$. Otherwise, we call C* reducible.

The following, which is our first technical lemma, allows us to consider only irreducible cycles. The proof follows the ideas of matroid intersection (see Lemma 41.5α in [31]). This lemma holds trivially for partition matroids with $s = 0$.

Lemma 2.3 *Let $\mathcal{M}_l = (N, \mathcal{I}_l)$, $l = 1, 2$, be matroids on ground set N. Suppose that $I, J \in \mathcal{I}_1 \cap \mathcal{I}_2$ and $|I| = |J|$. Then there is $s \geq 0$ and a collection of irreducible dicycles $\{C_1, \dots, C_m\}$ (allowing repetition of arcs and dicycles themselves) in $D_{\mathcal{M}_1,\mathcal{M}_2}(I)$, using only elements of $I \Delta J$, so that each element of $I \Delta J$ appears in exactly 2^s of the dicycles.*

Proof. Consider $D_{\mathcal{M}_1,\mathcal{M}_2}(I) = D_{\mathcal{M}_1}(I) \cup D_{\mathcal{M}_2}(I)$. By Lemma 2.1, there is a perfect matching between $I \setminus J$ and $J \setminus I$, both in $D_{\mathcal{M}_1}(I)$ and $D_{\mathcal{M}_2}(I)$. We denote these two perfect matchings by M_1, M_2. The union $M_1 \cup M_2$ forms a subgraph of out-degree 1 and in-degree 1 on $I \Delta J$. Therefore, it decomposes into a collection of dicycles $C_1, \dots, C_m$. If they are all irreducible, we are done and $s = 0$.

If C_i is not irreducible, it means that either $M'_1 = C_i \cap D_{\mathcal{M}_1}(I)$ or $M'_2 = C_i \cap D_{\mathcal{M}_2}(I)$ is not a unique perfect matching on $V(C_i)$. Let us assume, without loss of generality, that there is another perfect matching $M''_1 \neq M'_1$ in $D_{\mathcal{M}_1}(I)$. We consider the disjoint union $M'_1 + M''_1 + M'_2 + M'_2$, duplicating arcs where necessary. This is a subgraph of out-degree 2 and in-degree 2 on $V(C_i)$, which decomposes into dicycles $C_{i1}, \ldots, C_{it}$, covering each vertex of C_i exactly twice:

$$V(C_{i1}) + V(C_{i2}) + \ldots + V(C_{it}) = 2V(C_i).$$

Because $M'_1 \neq M''_1$, we have a chord of C_i in M''_1, and we can choose the first dicycle so that it does not cover all of $V(C_i)$. So we can assume that we have $t \geq 3$ dicycles, and at most one of them covers all of $V(C_i)$. If there is such a dicycle among $C_{i1}, \ldots, C_{it}$, we remove it and duplicate the remaining dicycles. Either way, we get a collection of dicycles $C_{i1}, \ldots, C_{it'}$ such that each of them is shorter than C_i and together they cover each vertex of C_i exactly twice.

We repeat this procedure for each reducible dicycle C_i. For irreducible dicycles C_i, we just duplicate C_i to obtain $C_{i1} = C_{i2} = C_i$. This completes one stage of our procedure. After the completion of the first stage, we have a collection of dicycles $\{C_{ij}\}$ covering each vertex in $I \Delta J$ exactly twice.

As long as there exists a reducible dicycle in our current collection of dicycles, we perform another stage of our procedure. This means decomposing all reducible dicycles and duplicating all irreducible dicycles. In each stage, we double the number of dicycles covering each element of $I \Delta J$. To see that this cannot be repeated indefinitely, observe that every stage decreases the size of the longest reducible dicycle. All dicycles of length 2 are irreducible, and therefore the procedure terminates after a finite number of stages s. Then, all cycles are irreducible and together they cover each element of $I \Delta J$ exactly 2^s times. □

We remark that of course the procedure in the proof of Lemma 2.3 is very inefficient, but it is not part of our algorithm — it is only used for the proof.

Next, we extend this Lemma 2.3 to sets I, J of different size, which forces us to deal with dipaths as well as dicycles.

Definition 2.2 *We call a dipath or dicycle A* feasible *in $D_{\mathcal{M}_1,\mathcal{M}_2}(I)$, if*

- *$I \Delta V(A) \in \mathcal{I}_1 \cap \mathcal{I}_2$, and*
- *For any sub-dipath $A' \subset A$ such that each endpoint of A' is either an endpoint of A or an element of I, we also have $I \Delta V(A') \in \mathcal{I}_1 \cap \mathcal{I}_2$.*

First, we establish that irreducible dicycles are feasible.

Lemma 2.4 *Any irreducible dicycle in $D_{\mathcal{M}_1,\mathcal{M}_2}(I)$ is also feasible in $D_{\mathcal{M}_1,\mathcal{M}_2}(I)$.*

Proof. An irreducible dicycle C consists of two matchings $M_1 \cup M_2$, which are the unique perfect matchings on $V(C)$, in $D_{\mathcal{M}_1}(I)$ and $D_{\mathcal{M}_2}(I)$ respectively. Therefore, we have $I \Delta V(C) \in \mathcal{I}_1 \cap \mathcal{I}_2$ by Lemma 2.2.

Consider any sub-dipath $A' \subset C$ whose endpoints are in I. (C has no endpoints, so the other case in Definition 2.2 does not apply.) This means that A' has even length. Suppose that $a_1 \in V(A')$ is the endpoint incident to an edge in $M_1 \cap A'$ and $a_2 \in V(A')$ is the other endpoint, incident to an edge in $M_2 \cap A'$. Note that any subset of M_1 or M_2 is again a unique perfect matching on its respective vertex set, because otherwise we could produce a different perfect matching on $V(C)$. We can view $I \Delta V(A')$ in two possible ways:

- $I \Delta V(A') = (I - a_1) \Delta (V(A') - a_1)$; because $V(A') - a_1$ has a unique perfect matching $M_2 \cap A'$ in $D_{\mathcal{M}_2}(I)$, this shows that $I \Delta V(A') \in \mathcal{I}_2$.
- $I \Delta V(A') = (I - a_2) \Delta (V(A') - a_2)$; because $V(A') - a_2$ has a unique perfect matching $M_1 \cap A'$ in $D_{\mathcal{M}_1}(I)$, this shows that $I \Delta V(A') \in \mathcal{I}_1$.

□

Finally, we establish the following property of possible exchanges between arbitrary solutions I, J (not necessarily of the same size).

Lemma 2.5 *Let $\mathcal{M}_1 = (N, \mathcal{I}_1)$ and $\mathcal{M}_2 = (N, \mathcal{I}_2)$ be two matroids and let $I, J \in \mathcal{I}_1 \cap \mathcal{I}_2$. Then there is $s \geq 0$ and a collection of dipaths/dicycles $\{A_1, \ldots, A_m\}$ (possibly with repetition), feasible in $D_{\mathcal{M}_1,\mathcal{M}_2}(I)$, using only elements of $I\Delta J$, so that each element of $I\Delta J$ appears in exactly 2^s dipaths/dicycles A_i.*

Proof. If $|I| = |J|$, we are done by Lemmas 2.3 and 2.4. If $|I| \neq |J|$, we extend the matroids by new "dummy elements" E, independent of everything else (in both matroids), and add them to I or J, to obtain sets of equal size $|\tilde{I}| = |\tilde{J}|$. We denote the extended matroids by $\tilde{\mathcal{M}}_1 = (N \cup E, \tilde{\mathcal{I}}_1), \tilde{\mathcal{M}}_2 = (N \cup E, \tilde{\mathcal{I}}_2)$. We consider the graph $D_{\tilde{\mathcal{M}}_1,\tilde{\mathcal{M}}_2}(\tilde{I})$. Observe that the dummy elements do not affect independence among other elements, so the graphs $D_{\mathcal{M}_1,\mathcal{M}_2}(I)$ and $D_{\tilde{\mathcal{M}}_1,\tilde{\mathcal{M}}_2}(\tilde{I})$ are identical on $I \cup J$.

Applying Lemma 2.3 to $\tilde{I}, \tilde{J}$, we obtain a collection of irreducible dicycles $\{C_1, \ldots, C_m\}$ on $\tilde{I}\Delta\tilde{J}$ such that each element appears in exactly 2^s dicycles. Let $A_i = C_i \setminus E$. Obviously, the sets $V(A_i)$ cover $I\Delta J$ exactly 2^s times. We claim that each A_i is either a feasible dicycle, a feasible dipath, or a collection of feasible dipaths (in the original digraph $D_{\mathcal{M}_1,\mathcal{M}_2}(I)$).

First, assume that $C_i \cap E = \emptyset$. Then $A_i = C_i$ is an irreducible cycle in $D_{\mathcal{M}_1,\mathcal{M}_2}(I)$ (the dummy elements are irrelevant). By Lemma 2.4, we know that $A_i = C_i$ is a feasible dicycle.

Next, assume that $C_i \cap E \neq \emptyset$. C_i is still a feasible dicycle, but in the extended digraph $D_{\tilde{\mathcal{M}}_1,\tilde{\mathcal{M}}_2}(\tilde{I})$. We remove the dummy elements from C_i to obtain $A_i = C_i \setminus E$, a dipath or a collection of dipaths. Consider any sub-dipath A' of A_i, possibly $A' = A_i$, satisfying the assumptions of Definition 2.2. A_i does not contain any dummy elements. If both endpoints of A' are in I, it follows from the feasibility of C_i that $\tilde{I}\Delta V(A') \in \tilde{\mathcal{I}}_1 \cap \tilde{\mathcal{I}}_2$, and hence $I\Delta V(A') = (\tilde{I}\Delta V(A')) \setminus E \in \mathcal{I}_1 \cap \mathcal{I}_2$.

If an endpoint of A' is outside of I, then it must be an endpoint of A_i. This means that it has a dummy neighbor in $\tilde{I} \cap C_i$ that we deleted. (Note that this case can occur only if we added dummy elements to I, i.e. $|I| < |J|$.) In that case, extend the path to A'', by adding the dummy neighbor(s) at either end. We obtain a dipath from $\tilde{I}$ to $\tilde{I}$. By the feasibility of C_i, we have $\tilde{I}\Delta V(A'') \in \tilde{\mathcal{I}}_1 \cap \tilde{\mathcal{I}}_2$, and therefore $I\Delta V(A') = (\tilde{I}\Delta V(A'')) \setminus E \in \mathcal{I}_1 \cap \mathcal{I}_2$. □

2.2 A generalized Rota-exchange property. Next, we establish a very useful property for a pair of bases of one matroid.

Lemma 2.6 *Let $\mathcal{M} = (N, \mathcal{I})$ be a matroid and A, B bases in $\mathcal{M}$. Let $A_1, \ldots, A_m$ be subsets of A such that each element of A appears in exactly q of them. Then there are sets $B_1, \ldots, B_m \subseteq B$ such that each element of B appears in exactly q of them, and for each i, $A_i \cup (B \setminus B_i) \in \mathcal{I}$.*

Remark 2.1 *A very special case of Lemma 2.6, namely when $m = 2$ and $q = 1$, attracted significant interest when it was conjectured by G.-C. Rota and proved in [4, 14, 35]; see* (39.58) *and* (42.13) *in [31]. The case of general m and $q = 1$ can be derived by repeated applications of that property [19]; see* (42.15) *in [31].*

Our proof uses the standard concept of matroid union (see [31]). Briefly, for matroids $\mathcal{M}_i = (N, \mathcal{I}_i)$, $i = 1, \ldots, k$, the *union* $\vee_{i=1}^k \mathcal{M}_i$ has ground set N, and a subset S of N is independent if it can be partitioned into sets $S_i \in \mathcal{I}_i$ $i = 1, \ldots, k$. It is well known that such a union is itself a matroid. Moreover, there is an efficient algorithm for checking whether a subset S of N is independent in the union $\vee_{i=1}^k \mathcal{M}_i$, assuming that oracles for checking independence in each of the $\mathcal{M}_i$ are available.

Our proof here differs slightly from the one we gave in the conference version [24]. We do not use the notion of a dual matroid here, and the proof is simpler.

Proof. We can assume for convenience that A and B are disjoint (otherwise we can make each B_i equal to A_i on the intersection $A \cap B$ and continue with a matroid where $A \cap B$ is contracted).

For each i, we define a matroid $\mathcal{N}_i = (\mathcal{M}/A_i)|B$, where we contract A_i and restrict to B. In other words, $S \subseteq B$ is independent in $\mathcal{N}_i$ exactly when $A_i \cup S \in \mathcal{I}$. The rank function of $\mathcal{N}_i$ is

$$r_{\mathcal{N}_i}(S) \;=\; r_{\mathcal{M}/A_i}(S) \;=\; r_{\mathcal{M}}(A_i \cup S) - |A_i|.$$

Observe that $C_i \subseteq B$ is independent in $\mathcal{N}_i$ exactly if $B_i = B \setminus C_i$ is a feasible swap for A_i, i.e. $(B \setminus B_i) \cup A_i \in \mathcal{I}$. What we want is a collection of feasible swaps $B_1, \ldots, B_m \subseteq B$ such that each element of B appears in exactly q of such swaps. Equivalently, by taking complements in B, we can ask for a collection of sets $C_1, \ldots, C_m \subseteq B$ such that each C_i is a base of $\mathcal{N}_i$ and each element of B appears in exactly $m-q$ of such sets. To find such a collection of bases, we use the *matroid union* theorem (see Corollary 42.1a in [31]).

Consider a new ground set $\hat{B} = B \times [m-q]$. We view the elements $\{(i,j) : j \in [m-q]\}$ as parallel copies of i. For $T \subseteq \hat{B}$, we define its *projection* to B as

$$\pi(T) \;=\; \{i \in B \mid \exists j \in [m-q] \text{ with } (i,j) \in T\}.$$

A natural extension of $\mathcal{N}_i$ to $\hat{B}$ is a matroid $\hat{\mathcal{N}_i}$ where a set T is independent iff it contains at most one copy (i,j) of each element $i \in B$ and $\pi(T)$ is independent in $\mathcal{N}_i$. The rank function of $\hat{\mathcal{N}_i}$ is

$$r_{\hat{\mathcal{N}_i}}(T) \;=\; r_{\mathcal{N}_i}(\pi(T)).$$

The question now is whether $\hat{B}$ can be partitioned into $C'_1, \ldots, C'_m$ so that C'_i is a base in $\hat{\mathcal{N}_i}$. If this is true, then we are done, because each $C_i = \pi(C'_i)$ would be a base of $\mathcal{N}_i$ and each element of B would appear in $m-q$ sets C_i. To prove this, consider the union of our matroids, $\hat{\mathcal{N}} := \hat{\mathcal{N}_1} \vee \hat{\mathcal{N}_2} \vee \ldots \vee \hat{\mathcal{N}}_m$. By the matroid union theorem (see, e.g., Corollary 42.1a in [31]), this matroid has rank

$$r_{\hat{\mathcal{N}}}(\hat{B}) \;=\; \min\nolimits_{T \subseteq \hat{B}} \left(|\hat{B} \setminus T| + \textstyle\sum_{i=1}^m r_{\hat{\mathcal{N}_i}}(T) \right). \tag{1}$$

For any $T \subseteq \hat{B}$, using the facts that $\pi(T) \subseteq B$ is independent in $\mathcal{M}$ and $\sum_{i=1}^m |A_i| = q|A|$, we have

$$\begin{aligned}
\textstyle\sum_{i=1}^m r_{\hat{\mathcal{N}_i}}(T) \quad &= \quad \sum_{i=1}^m r_{\mathcal{N}_i}(\pi(T)) \\
&= \quad \sum_{i=1}^m (r_{\mathcal{M}}(A_i \cup \pi(T)) - |A_i|) \\
&= \quad \sum_{i=1}^m (r_{\mathcal{M}}(A_i \cup \pi(T)) - r_{\mathcal{M}}(\pi(T))) + \sum_{i=1}^m (r_{\mathcal{M}}(\pi(T)) - |A_i|) \\
&= \quad \sum_{i=1}^m (r_{\mathcal{M}}(A_i \cup \pi(T)) - r_{\mathcal{M}}(\pi(T))) + m|\pi(T)| - q|A| \\
&\geq \quad q\,(r_{\mathcal{M}}(A \cup \pi(T)) - r_{\mathcal{M}}(\pi(T))) + m|\pi(T)| - q|A| \\
&= \quad (m-q)|\pi(T)|
\end{aligned}$$

where the inequality follows from the submodularity of the matroid rank functions and Lemma 1.1, and the last equality follows from the facts $r_{\mathcal{M}}(A \cup \pi(T)) = r_{\mathcal{M}}(A) = |A|$ and $r_{\mathcal{M}}(\pi(T)) = |\pi(T)|$.

Because T is in $\hat{B} = B \times [m-q]$, the projection π can decrease the size of a set at most by a factor of $m-q$. Consequently, the matroid union theorem (1) implies that

$$r_{\hat{\mathcal{N}}}(\hat{B}) \;\geq\; \min\nolimits_{T \subseteq \hat{B}} \left(|\hat{B} \setminus T| + (m-q)|\pi(T)| \right) \;\geq\; \min\nolimits_{T \subseteq \hat{B}} (|\hat{B} \setminus T| + |T|) \;=\; |\hat{B}|.$$

In fact, equality holds because the rank cannot be more than $|\hat{B}|$. This means that $\hat{B}$ can be partitioned into $C'_1 \cup \ldots \cup C'_m$ where each C'_i is independent in $\hat{\mathcal{N}_i}$. However, the ranks of $\hat{B}$ in the $\hat{\mathcal{N}_i}$ sum up to $\sum_{i=1}^m r_{\hat{\mathcal{N}_i}}(\hat{B}) = \sum_{i=1}^m |B \setminus B_i| = |\hat{B}|$, so this implies that each C'_i is a base of $\hat{\mathcal{N}_i}$. Then, each $C_i = \pi(C'_i)$ is a base of $\mathcal{N}_i$ and $B_i = B \setminus C_i$ are the sets demanded by the lemma. □

Finally, we give a version of Lemma 2.6 where the two sets need not be bases.

Lemma 2.7 *Let $\mathcal{M} = (N, \mathcal{I})$ be a matroid and $I, J \in \mathcal{I}$. Let $I_1, \ldots, I_m$ be subsets of I such that each element of I appears in at most q of them. Then there are sets $J_1, \ldots, J_m \subseteq J$ such that each element of J appears in at most q of them, and for each i, $I_i \cup (J \setminus J_i) \;\in\; \mathcal{I}$.*

Proof. We reduce this statement to Lemma 2.6. Let A, B be bases such that $I \subseteq A$ and $J \subseteq B$. Let q_e be the number of appearances of an element $e \in I$ in the subsets $I_1, \ldots, I_m$ and let $q' = \max_{e \in I} q_e$. Obviously, $q' \leq q$. We extend I_i arbitrarily to A_i, $I_i \subseteq A_i \subseteq A$, so that each element of A appears in exactly q' of them. By Lemma 2.6, there are sets $B_i \subseteq B$ such that each element of B appears in exactly

q' of them, and $A_i \cup (B \setminus B_i) \in \mathcal{I}$ for each i. We define $J_i = J \cap B_i$. Then, each element of J appears in at most $q' \le q$ sets J_i, and

$$I_i \cup (J \setminus J_i) \subseteq A_i \cup (B \setminus B_i) \;\in\; \mathcal{I}.$$

□

3. Local-Search Algorithm. At each iteration of our local-search algorithm, given a current feasible solution $S \in \cap_{j=1}^k \mathcal{I}_j$, our algorithm seeks an improved solution by looking at a polynomial number of options to change S. If the algorithm finds a better solution, it moves to the next iteration, otherwise the algorithm stops. Specifically, given a current solution $S \in \cap_{j=1}^k \mathcal{I}_j$, the local move that we consider is

p**-exchange operation**: If there is $S' \in \cap_{j=1}^k \mathcal{I}_j$ such that **(i)** $|S' \setminus S| \le 2p$, $|S \setminus S'| \le 2kp$, and **(ii)** $f(S') > f(S)$, then $S \leftarrow S'$.

The p-exchange operation for $S' \subseteq S$ is called a *delete operation*. Our main result is the following lower bound on the value of the locally-optimal solution.

Lemma 3.1 *For every $k \ge 2$ and every $C \in \cap_{j=1}^k \mathcal{I}_j$, a locally-optimal solution S under p-exchanges, satisfies*

$$(k + 1/p) \cdot f(S) \;\ge\; f(S \cup C) + (k - 1 + 1/p) \cdot f(S \cap C).$$

Proof. Our proof is based on the new exchange properties of matroids: Lemmas 2.5 and 2.7. By applying Lemma 2.5 to the independent sets C and S in matroids $\mathcal{M}_1$ and $\mathcal{M}_2$, we obtain a collection of dipaths/dicycles $\{A_1, \ldots, A_m\}$ (possibly with repetition), feasible in $D_{\mathcal{M}_1,\mathcal{M}_2}(S)$, using only elements of $C \Delta S$, so that each element of $C \Delta S$ appears in exactly 2^s paths/cycles A_i .

We would like to define the sets of vertices corresponding to the exchanges in our local-search algorithm, based on the sets of vertices in paths/cycles $\{A_1, \ldots, A_m\}$. The problem is that these paths/cycles can be much longer than the maximal cardinality of a set allowable in a p-exchange operation. To handle this, we index vertices of the set of $C \setminus S$ in each path/cycle A_i for $i = 1, \ldots, m$, in such a way that vertices along any path or cycle are numbered consecutively. The vertices of $S \setminus C$ remain unlabeled. Because one vertex appears in 2^s paths/cycles, it might get different labels corresponding to different appearances of that vertex.

We also define $p+1$ copies of the collection $\{A_1, \ldots, A_m\}$. For each copy $q = 0, \ldots, p$ of $\{A_1, \ldots, A_m\}$ with labels as described above, we throw away appearances of vertices from $C \setminus S$ that were labeled by q modulo $p+1$ from each A_i. By throwing away some appearances of the vertices, we are changing our set of paths in each copy of the original sets $\{A_1, \ldots, A_m\}$. Let $\{A_{q1}, \ldots, A_{qm_q}\}$ be the resulting collection of paths for $q = 0, \ldots, p$. Now each path A_{qi} contains at most $2p$ vertices from $C \setminus S$ and at most $2p+1$ vertices from $S \setminus C$.

Because our original collection of paths/cycles was feasible in $D_{\mathcal{M}_1,\mathcal{M}_2}(S)$ (see definition 2.2), each of the paths in the new collections correspond to feasible exchanges for matroids $\mathcal{M}_1$ and $\mathcal{M}_2$, i.e. $S \Delta V(A_{qi}) \in \mathcal{I}_1 \cap \mathcal{I}_2$. Consider now the collection of paths $\{A_{qi} | q = 0, \ldots, p, i = 1, \ldots, m_q\}$. By construction, each element of the set $S \setminus C$ appears in exactly $(p+1)2^s$ paths, and each element of $C \setminus S$ appears in exactly $p2^s$ paths, because each vertex has $2^s(p+1)$ appearances in total, and each appearance is thrown away in exactly one out of $p+1$ copies of the original sets $\{A_1, \ldots, A_m\}$. Let $L_{qi} = S \cap V(A_{qi})$ denote the set of vertices in the path A_{qi} belonging to the locally-optimal solution S, and let $W_{qi} = C \cap V(A_{qi})$ denote the set of vertices in the path A_{qi} belonging to the set C.

For each matroid $\mathcal{M}_i$ for $i = 3, \ldots, k$, independent sets $S \in \mathcal{I}_i$ and $C \in \mathcal{I}_i$, and collection of sets $\{W_{qi} \mid q = 0, \ldots, p; i = 1, \ldots, m_q\}$ (note that some of these sets might be empty), we apply Lemma 2.7. For convenience, we re-index the collection of sets $\{W_{qi} \mid q = 0, \ldots, p, i = 1, \ldots, m_q\}$. Let $W_1, \ldots, W_t$ be that collection, after re-indexing, for $t = \sum_{q=0}^p m_q$. By Lemma 2.7, for each $i = 3, \ldots, k$ there exists a collection of sets $X'_{1i}, \ldots, X'_{ti}$ such that $W_j \cup (S \setminus X'_{ji}) \in \mathcal{I}_i$. Moreover, each element of S appears in at most $p2^s$ of the sets from collection $X'_{1i}, \ldots, X'_{ti}$.

We consider the set of p-exchanges that correspond to adding the elements of the set W_j to the set S and removing the set of elements $\Lambda_j = L_j \cup (\cup_{i=3}^k X'_{ji})$ for $j = 1, \ldots, t$. Note that $|\Lambda_j| \;\le\; (2p+1) + (k -$

$2)2p = (k-1)2p+1 \le 2kp$. By Lemmas 2.5 and 2.7, the sets $W_j \cup (S \setminus \Lambda_j)$ are independent in each of the matroids $\mathcal{M}_1, \ldots, \mathcal{M}_k$. By the fact that S is a locally-optimal solution, we have

$$f(S) \ \ge\ f\left((S \setminus \Lambda_j) \cup W_j\right), \qquad \forall j = 1, \ldots, t. \tag{2}$$

Using inequalities (2) together with submodularity for $j = 1, \ldots, t$, we have

$$f(S \cup W_j) - f(S) \ \le\ f\left((S \setminus \Lambda_j) \cup W_j\right) - f\left(S \setminus \Lambda_j\right) \ \le\ f(S) - f\left(S \setminus \Lambda_j\right). \tag{3}$$

Moreover, we know that each element of the set $C \setminus S$ appears in exactly $p2^s$ sets W_j , and each element $e \in S \setminus C$ appears in $n_e \le (p+1)2^s + (k-2)p2^s$ sets Λ_j .

Consider the sum of t inequalities (3), and add $(p+1)2^s + (k-2)p2^s - n_e$ inequalities

$$f(S) \ \ge\ f(S \setminus \{e\}) \tag{4}$$

for each element $e \in S \setminus C$. These inequalities correspond to the delete operations. We obtain

$$\begin{aligned} \sum_{j=1}^{t} [f(S \cup W_j) - f(S)] \quad &\le \quad \sum_{j=1}^{t} [f(S) - f(S \setminus \Lambda_j)] \\ &+ \quad \sum_{e \in S \setminus C} \left((p+1)2^s + (k-2)p2^s - n_e\right) [f(S) - f(S \setminus \{e\})]. \end{aligned} \tag{5}$$

Applying Lemma 1.2 to the right-hand side of the inequality (5) and Lemma 1.1 to the left-hand side of the inequality (5), we have

$$p2^s\,[f(S \cup C) - f(S)] \ \le\ \left((p+1)2^s + (k-2)p2^s\right)[f(S) - f(S \cap C)],$$

which is equivalent to

$$(k + 1/p) \cdot f(S) \ \ge\ f(S \cup C) + (k - 1 + 1/p) \cdot f(S \cap C).$$

The result follows. □

Simple consequences of Lemma 3.1 are bounds on the value of a locally-optimal solution when the submodular function f has additional structure.

Corollary 3.1 *For $k \ge 2$, a locally-optimal solution S, and any $C \in \cap_{j=1}^{k} \mathcal{I}_j$, the following inequalities hold:*

(*i*) $f(S) \ \ge\ f(C)/\,(k + 1/p)$ *if function f is monotone submodular,*

(*ii*) $f(S) \ \ge\ f(C)/\,(k - 1 + 1/p)$ *if function f is linear.*

Proof. For a monotone submodular function f, a locally-optimal solution S, and any $C \in \cap_{j=1}^{k} \mathcal{I}_j$ by the Lemma 3.1 we obtain

$$(k + 1/p) \cdot f(S) \ \ge\ f(S \cup C) + (k - 1 + 1/p) \cdot f(S \cap C) \ \ge\ f(S \cup C) \ \ge\ f(C).$$

For a linear function f, we obtain

$$(k + 1/p) \cdot f(S) \ \ge\ f(S \cup C) + (k - 1 + 1/p) \cdot f(S \cap C) \ \ge\ f(S \cup C) + f(S \cap C) \ =\ f(S) + f(C).$$

Another way to derive this bound is to notice that this problem can be viewed as a weighted rank-function maximization subject to $k - 1$ matroid constraints. Because the weighted rank function is monotone submodular, we can derive the second claim of the corollary from the first one. □

The local-search algorithm defined at the beginning of this section could run for an exponential amount of time before reaching a locally-optimal solution. To ensure polynomial runtime, we follow the standard approach of approximate local search under a suitable (small) parameter $\varepsilon > 0$ as described in Figure 1. The following is a simple extension of Lemma 3.1.

Lemma 3.2 *For an approximate locally-optimal solution S and any $C \in \cap_{j=1}^{k} \mathcal{I}_j$,*

$$(1 + \varepsilon)\,(k + 1/p) \cdot f(S) \ \ge\ f(S \cup C) + (k - 1 + 1/p) \cdot f(S \cap C),$$

where $\varepsilon > 0$ is the parameter used in the procedure of Figure 1.

Input: Finite ground set $N := [n]$, value-oracle access to submodular function $f : 2^N \to \mathbb{R}$, and matroids $\mathcal{M} = (N, \mathcal{I}_i)$, for $i \in [k]$.

1. Set $v \leftarrow \arg\max\{f(u) \mid u \in N\}$ and $S \leftarrow \{v\}$.
2. While the following local operation is possible, update S accordingly:
   p**-exchange operation.** If there is a feasible S' such that
   (i) $|S' \setminus S| \leq 2p$, $|S \setminus S'| \leq 2kp$, and
   (ii) $f(S') \geq (1 + \frac{\varepsilon}{n(k+1)})f(S)$,
   then $S \leftarrow S'$.

Output: S.

Figure 1: The approximate local-search procedure

Proof. The proof of this lemma is almost identical to the proof of the Lemma 3.1 — the only difference is that left-hand sides of inequalities (2) and inequalities (4) are multiplied by $1 + \frac{\varepsilon}{n(k+1)}$. After following the steps in the proof of Lemma 3.1, we obtain the following inequality:

$$\left(k + 1/p + \frac{\varepsilon\lambda}{n(k+1)p2^s}\right) \cdot f(S) \geq f(S \cup C) + (k - 1 + 1/p) \cdot f(S \cap C),$$

where $\lambda = t + \sum_{e \in S \setminus C} [(p+1)2^s + (k-2)p2^s - n_e] \leq t + |S|kp2^s$ is the total number of inequalities (2) and (4). Because $t \leq |C|p2^s$, we obtain that $\lambda \leq |C|p2^s + |S|kp2^s \leq n(k+1)p2^s$. □

Lemma 3.2 implies the following:

Theorem 3.1 *For any constant $k \geq 2$, $\delta > 0$, there exists a polynomial $1/(k+\delta)$-approximation algorithm for maximizing a non-negative non-decreasing submodular function subject to k matroid constraints. This bound improves to $1/(k-1+\delta)$ for linear functions.*

4. Local search for non-monotone submodular functions. Combining our techniques with the iterative local search from [23], we also improve the known approximation results for maximizing a general (non-monotone) submodular function subject to $k \geq 2$ matroid constraints. The previously known approximation factor in this case was $1/(k + 2 + \frac{1}{k} + \delta)$ (see [23]). We essentially replace k by $k - 1$ in this formula.

Our algorithm is exactly the same as in [23] (see Figure 2). As a subroutine, we use the local-search algorithm from the previous section (see Figure 1). Our algorithm has k iterations. At each iteration, the algorithm finds a locally-optimal solution S_k by applying the algorithm from Figure 1 to the set of remaining elements V_k (initially $V_1 = N$). Then the algorithm deletes the locally-optimal solution from the current ground set, i.e. $V_{k+1} = V_k \setminus S_k$, and goes to the next iteration. Intuitively, the algorithm finds k good solutions that do not intersect each other.

Theorem 4.1 *For any fixed $k \geq 2$ and constant $\delta > 0$, there exists a polynomial $1/(k + 1 + \frac{1}{k-1} + \delta)$-approximation algorithm for maximizing a non-negative submodular function subject to k matroid constraints.*

For example, for maximizing a non-monotone submodular function subject to two matroid constraints, we obtain a factor arbitrarily close to $1/4$, improving the previously known $2/9$ (see [23]). We remark that in the case of one matroid constraint, the best known approximation factor is $\simeq 0.325$ (see [28]).

Our theorem is proven by an easy adaptation of the proof from [23]. For the sake of completeness, we present the full proof here.

Proof. Let C denote an optimal solution and let $C_i = C \cap V_i$, where $V_i = V \setminus (S_1 \cup \ldots \cup S_{i-1})$ are the sets generated by the algorithm. Note that $C = C_1 \supseteq C_2 \supseteq C_3 \ldots$. The crucial inequality in the analysis is given by Lemma 3.2: for each $i = 1, \ldots, k$,

$$(1+\varepsilon)\,(k + 1/p) \cdot f(S_i) \;\geq\; f(S_i \cup C_i) + (k - 1 + 1/p) \cdot f(S_i \cap C_i).$$

Input: Finite ground set $N := [n]$, value-oracle access to submodular function $f : 2^N \to \mathbb{R}$, and matroids $\mathcal{M} = (N, \mathcal{I}_i)$, for $i \in [k]$.

1. Set $V_1 = N$.
2. For $i = 1, \ldots, k$, do:
   (i) Apply the local-search algorithm on ground set V_i to find a solution S_i to the problem
   $$\max\{f(S) : S \subseteq V_i, S \in \cap_{i=1}^k \mathcal{I}_i\}$$
   (ii) Set $V_{i+1} = V_i \setminus S_i$.

Output: The maximum of $f(S_1), \ldots, f(S_k)$.

Figure 2: Iterative local search for non-monotone submodular functions

We set $(1+\varepsilon)(k+1/p) = (k+\delta)$ and note that $S_i \cap C_i = S_i \cap C$. In fact we use the following (weaker) inequalities:

$$\begin{aligned}
(k+\delta) \cdot f(S_1) &\geq f(S_1 \cup C_1) + (k-1) \cdot f(S_1 \cap C) \\
(k+\delta) \cdot f(S_2) &\geq f(S_2 \cup C_2) + (k-2) \cdot f(S_2 \cap C) \\
&\cdots \\
(k+\delta) \cdot f(S_k) &\geq f(S_k \cup C_k)
\end{aligned}$$

First, we add up the first two inequalities and we use submodularity: $f(S_1 \cup C_1) + f(S_2 \cup C_2) \geq f(S_1 \cup S_2 \cup C) + f(C_2)$. We get:

$$(k+\delta) \sum_{i=1}^{2} f(S_i) \ \geq\ f(S_1 \cup S_2 \cup C) + f(C_2) + (k-1) f(S_1 \cap C) + (k-2) f(S_2 \cap C).$$

We use submodularity again: $f(C_2) + f(S_1 \cap C) \geq f(C) = OPT$. Therefore, the first two inequalities yield

$$(k+\delta) \sum_{i=1}^{2} f(S_i) \ \geq\ OPT + f(S_1 \cup S_2 \cup C) + (k-2) \sum_{i=1}^{2} f(S_i \cap C).$$

We claim that by induction, the first j inequalities yield

$$(k+\delta) \sum_{i=1}^{j} f(S_i) \ \geq\ (j-1) OPT + f(S_1 \cup \ldots \cup S_j \cup C) + (k-j) \sum_{i=1}^{j} f(S_i \cap C).$$

To see the inductive step, add the $(j+1)$-th inequality:

$$(k+\delta) \cdot f(S_{j+1}) \ \geq\ f(S_{j+1} \cup C_{j+1}) + (k-j-1) \cdot f(S_{j+1} \cap C).$$

Again, submodularity gives $f(S_1 \cup \ldots \cup S_j \cup C) + f(S_{j+1} \cup C_{j+1}) \geq f(S_1 \cup \ldots \cup S_{j+1} \cup C) + f(C_{j+1})$, and $f(C_{j+1}) + \sum_{i=1}^{j} f(S_i \cap C) \geq f(C) = OPT$. Hence we get the desired inequality after $j+1$ steps:

$$(k+\delta) \sum_{i=1}^{j+1} f(S_i) \ \geq\ j \cdot OPT + f(S_1 \cup \ldots \cup S_{j+1} \cup C) + (k-j-1) \sum_{i=1}^{j+1} f(S_i \cap C).$$

After k steps, we get $(k+\delta) \sum_{i=1}^{k} f(S_i) \geq (k-1) \cdot OPT$ which means

$$\max f(S_i) \geq \frac{1}{k} \sum_{i=1}^{k} f(S_i) \geq \frac{k-1}{k(k+\delta)} OPT \geq \frac{1}{k+1+\frac{1}{k-1}+2\delta} OPT.$$

□

5. Tightness of analysis. Next, we demonstrate that our analysis of local search for maximizing linear and monotone submodular functions is almost tight. By local search, we mean for fixed $p > 0$, adding $\leq p$ elements and removing $\leq kp$ elements at a time which is a $p/2$-exchange operation. It was known (see [2]) that such an algorithm cannot give better than $1/(k-1+1/p)$-approximation for the weighted k-set packing problem ($k \geq 3$). From the example of [2], the same bound follows also for weighted k-dimensional matching and hence also for the more general problems of maximizing a linear function subject to k matroid constraints. We repeat the proof from [2] for the sake of completeness.

Proposition 5.1 *For any $k, p \geq 2$, there are instances of maximizing a linear function subject to k partition matroids, where a local optimum with respect to $p/2$-exchanges has value $OPT/(k-1+1/p)$.*

Proof. Let $G = (V, E)$ be a k-regular bipartite graph of girth at least $2p+2$ (see [22] for a much stronger result), with bipartition $V = A \cup B$. We define vertex weights $w_i = 1$ for $i \in A$ and $w_j = k - 1 + 1/p$ for $j \in B$. Being a k-regular bipartite graph, G can be decomposed into k matchings, $E = M_1 \cup M_2 \cup \ldots \cup M_k$. For each M_i, we define a partition matroid $\mathcal{M}_i = (V, \mathcal{I}_i)$ where $S \in \mathcal{I}_i$ iff S contains at most one vertex from each edge in M_i. We maximize $w(S)$ over $S \in \bigcap_{i=1}^k \mathcal{I}_i$. Equivalently, we seek a maximum-weight independent set in G.

Clearly, A and B are both feasible solutions. Because $|A| = |B|$, we have $w(A)/w(B) = 1/(k-1+1/p)$. We claim that A is a local optimum. Consider any set obtained by a local move, $A' = (A \setminus K) \cup L$ where $K \subseteq A$, $L \subseteq B$ and $|L| \leq p$. For A' to be independent, $K \cup L$ must contain all edges incident with L. (Otherwise, there is an edge contained in A'.) Also, $K \cup L$ cannot contain any cycle, because every cycle in G has at least $p+1$ vertices on each side. Therefore, $K \cup L$ induces a forest with $k|L|$ edges. Hence $|K \cup L| \geq k|L| + 1$, i.e. $|K| \geq (k-1)|L| + 1$. The value of A' is

$$w(A') = w(A) - |K| + (k-1+1/p)|L| \leq w(A) - |K| + (k-1)|L| + 1 \leq w(A).$$

$\square$

We obtain the same result for maximizing monotone submodular functions subject to $k-1$ matroid constraints, by incorporating one of the matroid constraints into the objective function. That is, in the example above, we would remove one matroid constraint, for example $\mathcal{M}_k$, and replace the objective function by the *weighted rank function* $f(S) = \max\{w(I) : I \subseteq S, I \in \mathcal{I}_k\}$. This is known to be a monotone submodular function (see [31]), and it can be seen that again, A is a local optimum and it has value $OPT/(k-1+1/p)$. Hence we get the following.

Proposition 5.2 *For any $k, p \geq 2$, there are instances of maximizing a monotone submodular function subject to k partition matroids, where a local optimum with respect to $p/2$-exchanges has value $OPT/(k+1/p)$.*

We note that the iterative local search (see Figure 2) does not help for linear or monotone submodular functions; we could have many copies of the local optimum A that are equivalent in all the matroids (using parallel copies). The iterative local-search algorithm would repeatedly find local optima having the same value.

However, we do not know whether our analysis for non-monotone submodular functions is tight (Theorem 4.1). It is unlikely that our algorithm for non-monotone functions achieves the $1/(k+1/p)$ bound, but we do not have a counterexample.

References

[1] A. Ageev, M. Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee, *J. Comb. Opt.* 8(3) (2004), 307–328.

[2] E. Arkin, R. Hassin. On local search for weighted k-set packing, *Math. of Oper. Research* 23(3) (1998), 640–648.

[3] P. Berman. A $d/2$ approximation for maximum weight independent set in d-claw free graphs, *Nordic J. Comput.* 7(3) (2000), 178–184.

[4] T. Brylawski. Some properties of basic families of subsets, *Disc. Math.* 6 (1973), 333–341.

[5] G. Calinescu, C. Chekuri, M. Pál, J. Vondrák. Maximizing a submodular set function subject to a matroid constraint, *IPCO 2007*, 182–196.

[6] G. Calinescu, C. Chekuri, M. Pál, J. Vondrák. Maximizing a submodular set function subject to a matroid constraint, to appear in the *SIAM J. on Comp.*

[7] B. Chandra, M. Halldórsson. Greedy local improvement and weighted set packing approximation, *J. Algorithms* 39(2) (2001), 223–240.

[8] M. Conforti, G. Cornuéjols. Submodular set functions, matroids and the greedy algorithm: Tight worst-case bounds and some generalizations of the Rado-Edmonds theorem, *Disc. Appl. Math.* 7(3) (1984), 251–274.

[9] G. Cornuéjols, M. Fischer, G. Nemhauser. Location of bank accounts to optimize float: An analytic study of exact and approximation algorithms, *Management Sci.*, 23 (1977), 789–810.

[10] G. Cornuéjols, M. Fischer, G. Nemhauser. On the uncapacitated location problem, *Annals of Disc. Math.* 1 (1977), 163–178.

[11] U. Feige. A threshold of $\ln n$ for approximating set cover. *J. of ACM* 45 (1998), 634–652.

[12] U. Feige, V. Mirrokni, J. Vondrák. Maximizing non-monotone submodular functions, *FOCS 2007*, 461–471.

[13] M. Fisher, G. Nemhauser, L. Wolsey. An analysis of approximations for maximizing submodular set functions II. *Math. Prog. Study* 8 (1978), 73–87.

[14] C. Greene. A multiple exchange property for bases, *Proc. Amer. Math. Soc.* 39 (1973), 45–50.

[15] D. Hausmann, B. Korte. K-greedy algorithms for independence systems, *Z. Oper. Res. Ser. A-B* 22(5) (1978).

[16] D. Hausmann, B. Korte, T. Jenkyns. Worst case analysis of greedy type algorithms for independence systems, *Math. Prog. Study* 12 (1980), 120–131.

[17] E. Hazan, S. Safra, O. Schwartz. On the complexity of approximating k-set packing, *Computational Complexity* 15(1) (2006) 20–39.

[18] C. Hurkens, A. Schrijver. On the size of systems of sets every t of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems, *SIAM J. Disc. Math.* 2(1) (1989), 68–72.

[19] C. Greene and T. Magnanti, Some abstract pivot algorithms, *SIAM J. Appl. Math.* 29 (1975), 530–539.

[20] S. Iwata, L. Fleischer, S. Fujishige. A combinatorial, strongly polynomial-time algorithm for minimizing submodular functions, *J. of ACM* 48 (2001), 761–777.

[21] T. Jenkyns. The efficacy of the greedy algorithm, *Cong. Num.* 17 (1976), 341–350.

[22] F. Lazebnik, V. Ustimenko, A. Woldar. A new series of dense graphs of high girth, *Bulletin of the AMS* 32(1) (1995), 73–79.

[23] J. Lee, V. Mirrokni, V. Nagarajan, M. Sviridenko. Maximizing non-monotone submodular functions under matroid and knapsack constraints, to appear in SIDMA, preliminary version appeared in *STOC 2009*.

[24] J. Lee, M. Sviridenko, J. Vondrák. Submodular maximization over multiple matroids via generalized exchange properties, *Proc. of APPROX*, Springer LNCS, 244–257, 2009.

[25] L. Lovász. Submodular functions and convexity. In: A. Bachem et al., eds, *Mathematical Programmming: The State of the Art*, 235–257.

[26] G. Nemhauser, L. Wolsey, M. Fisher. An analysis of approximations for maximizing submodular set functions I. *Math. Prog.* 14 (1978), 265–294.

[27] G. Nemhauser, L. Wolsey. Best algorithms for approximating the maximum of a submodular set function, *Math. Oper. Res.*, 3(3) (1978), 177–188.

[28] S. Oveis-Gharan, J. Vondrák. Submodular maximization by simulated annealing, *manuscript*, 2010.

[29] J. Oxley. Matroid theory. Oxford University Press, New York, 1992.

[30] J. Reichel, M. Skutella. Evolutionary algorithms and matroid optimization problems, Proc. of GECCO '07, 947–954, 2007.

[31] A. Schrijver. Combinatorial Optimization: Polyhedra and Efficiency. Springer, 2003.

[32] A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time, *J. Comb. Theory, Ser. B* 80 (2000), 346–355.

[33] J. Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model, *Proc. of ACM STOC 2008*, 67–74.

[34] J. Vondrák. Symmetry and approximability of submodular maximization problems, *Proc. of IEEE FOCS 2009*, 651–670.

[35] D. Woodall. An exchange theorem for bases of matroids, *J. Comb. Theory, Ser. B* 16 (1974), 227–228.