2012 IEEE 53rd Annual Symposium on Foundations of Computer Science

A Tight Combinatorial Algorithm for Submodular Maximization Subject to a Matroid Constraint

Yuval Filmus
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada
Email: yuvalf@cs.toronto.edu

Justin Ward
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada
Email: jward@cs.toronto.edu

***Abstract*—We present an optimal, combinatorial 1-1/e approximation algorithm for monotone submodular optimization over a matroid constraint. Compared to the continuous greedy algorithm (Calinescu, Chekuri, Pal and Vondrak, 2008), our algorithm is extremely simple and requires no rounding. It consists of the greedy algorithm followed by local search. Both phases are run not on the actual objective function, but on a related non-oblivious potential function, which is also monotone submodular.**

In our previous work on maximum coverage (Filmus and Ward, 2011), the potential function gives more weight to elements covered multiple times. We generalize this approach from coverage functions to arbitrary monotone submodular functions. When the objective function is a coverage function, both definitions of the potential function coincide. The parameters used to define the potential function are closely related to Pade approximants of exp(x) evaluated at x = 1. We use this connection to determine the approximation ratio of the algorithm.

***Keywords*-approximation algorithms; submodular functions; matroids; local search**

I. Introduction

We consider the problem of maximizing a monotone submodular function f, subject to a single matroid constraint. Formally, let $\mathcal{U}$ be a set of elements and let $f\colon 2^{\mathcal{U}} \to \mathbb{R}_{\geq 0}$ be a function assigning a value to each subset of $\mathcal{U}$. We say that f is *submodular* if

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$$

for all $A, B \subseteq \mathcal{U}$. If additionally, $f(A) \leq f(B)$ whenever $A \subseteq B$, we say that f is *monotone submodular*. Submodular functions exhibit (and are, in fact, alternately characterized by) the property of diminishing returns—if f is submodular then $f(A+x)-f(A) \leq f(B+x)-f(B)$ whenever $B \subseteq A$ and $x \notin A$.[1] Hence, they are useful for modeling various economic and game-theoretic scenarios, as well as various combinatorial problems. In a general monotone submodular maximization problem, we are given a value oracle for f and a membership oracle for some distinguished collection $\mathcal{I} \subseteq 2^{\mathcal{U}}$ of *feasible sets*, and our goal is to find a member of $\mathcal{I}$ that maximizes the value of f.

We consider the restricted setting of monotone submodular matroid maximization, in which $\mathcal{I}$ is a matroid. Matroids are intimately connected to combinatorial optimization: the problem of optimizing a linear function over a hereditary set system (a set system closed under taking subsets) is solved optimally for all possible functions by the standard greedy algorithm if and only if the set system is a matroid [9], [27].

In the case of a *monotone submodular* objective function, the standard greedy algorithm, which chooses at each step the element yielding the largest increase in f while maintaining independence, is (only) a 1/2-approximation [17]. Recently, Calinescu et al. [6], [7], [29] have developed a $(1-1/e)$-approximation for this problem via the *continuous greedy algorithm*, which is essentially a steepest descent algorithm running in continuous time (in practice, a suitably discretized version is used), producing a fractional solution. This solution is rounded using pipage rounding [1] or swap rounding [8].

Feige [10] shows that improving the bound $(1 - 1/e)$ is NP-hard. Nemhauser and Wolsey [24] show that any improvement over $(1-1/e)$ requires an exponential number of queries in the value oracle setting. Vondrák [30] further shows that even when the submodular function f has restricted curvature, the approximation ratio attained by the continuous greedy algorithm is the best possible.

A. Our contribution

In this paper, we propose a conceptually simple randomized polynomial time local search algorithm for the problem of monotone submodular matroid maximization. Like the continuous greedy algorithm, our algorithm delivers the optimal $(1 - 1/e)$-approximation. However, unlike the continuous greedy algorithm, our algorithm is entirely combinatorial, in the sense that it deals only with integral solutions and hence involves no rounding procedure. As such, we believe that the algorithm may serve as a gateway

[1] For an element x and a set A, we use the shorthand $A + x$ for the set $A \cup \{x\}$ and $A - x$ for the set $A \setminus \{x\}$.

0272-5428/12 $26.00 © 2012 IEEE
DOI 10.1109/FOCS.2012.55

to further improved algorithms in contexts where pipage rounding and swap rounding break down, such as submodular maximization subject to multiple matroid constraints.

Our main results are a combinatorial $1 - 1/e - \epsilon$ approximation algorithm running in randomized time $\tilde{O}(\epsilon^{-3}n^4u)$, and a combinatorial $1 - 1/e$ approximation algorithm running in randomized time $\tilde{O}(n^7u^2)$, where n is the rank of the given matroid and u is the size of its ground set. In the full version of the paper, we estimate the runtime of the continuous greedy algorithm, which turns out to be a factor of $\tilde{O}(n)$ faster than our algorithm.

Our approach is based on *non-oblivious local search*, a technique first proposed by Alimonti [2] and by Khanna, Motwani, Sudan and Vazirani [20]. In classical (or, *oblivious*) local search, the algorithm starts at an arbitrary solution, and proceeds by iteratively making small changes that improve the objective function, until no such improvement can be made. The *locality ratio* of a local search algorithm is $\min f(S)/f(O)$, where S is a solution that is locally-optimal with respect to the small changes considered by the algorithm, O is a global optimum, and f is the objective function. The locality ratio provides a natural, worst-case guarantee on the approximation performance of the local search algorithm.

In many cases, oblivious local search may have a very poor locality ratio, implying that a locally-optimal solution may be of significantly lower quality than the global optimum. For example, for submodular matroid maximization, the locality ratio of an algorithm changing a single element at each step is $1/2$ [17]. *Non-oblivious* local search attempts to avoid this problem by making use of a secondary potential function to guide the search. By carefully choosing this auxiliary function, we ensure that poor local optima with respect to the original objective function are no longer local optima.

In previous work [15], we designed an optimal non-oblivious local search algorithm for maximum coverage over a matroid. The non-oblivious potential function used in [15] gives more weight to elements appearing multiple times. In the present work, we extend this approach to general monotone submodular functions. This presents two challenges: defining a non-oblivious potential function without reference to elements, and analyzing the resulting algorithm. In order to define the potential function in general, we use a definition of the potential function from [15] which doesn't refer to elements. Instead, the potential function aggregates the information by applying the objective function to all subsets of the input, weighted according to their size. Intuitively, the resulting potential function gives extra weight to solutions that contain a large number of good subsolutions, or alternatively, remain good solutions on average even when elements are randomly removed. An appropriate setting of the weights yields a function which coincides with the previous definition for coverage functions, but still makes sense for arbitrary monotone submodular functions.

The analysis of the algorithm in [15] is relatively straightforward. For each type of element in the universe of the coverage problem, we must prove a certain inequality among the coefficients defining the potential function. In the general setting, however, we need to construct a proof using only the inequalities given by monotonicity and submodularity. The resulting proof is non-obvious and delicate. For the proof to work, a certain sequence defined by a recurrence relation needs to be non-decreasing. The locality ratio can then be read off the sequence. We describe a way to construct the sequence using the recurrence relation so that it is non-decreasing. In order to show that the resulting performance ratio is at least $1 - 1/e$, we use an explicit formula for the sequence in terms of Padé approximants of e^x.

B. Related work

Fisher, Nemhauser and Wolsey [17], [25] analyze greedy and local search algorithms for submodular maximization subject to various constraints, including single and multiple matroid constraints, obtaining some of the earliest results in the area including a $1/(k+1)$-approximation for monotone submodular maximization subject to k matroid constraints. A recent survey by Goundan and Schulz [19] reviews many results pertaining to the greedy algorithm for submodular maximization.

More recently, Lee, Sviridenko and Vondrák [23] consider the problem of both monotone and non-monotone submodular maximization subject to multiple matroid constraints, attaining a $1/(k+\epsilon)$-approximation for monotone submodular maximization subject to $k \geq 2$ constraints using local search. Feldman et al. [14] show that a local search algorithm attains the same bound for the related class of k-exchange systems, which includes the intersection of k strongly base orderable matroids, as well as the independent set problem in $(k+1)$-claw free graphs. Further work by Ward [31] shows that a non-oblivious local search routine attains an improved $2/(k+3) - \epsilon$ approximation for this class of problems.

In the case of unconstrained non-monotone maximization, Feige, Mirrokni and Vondrák [11] give a $2/5$ approximation via a randomized local search algorithm, and give an upper bound of $1/2$ in the value oracle model. Gharan and Vondrák [18] improved the algorithmic result to 0.41 by enhancing the local search algorithm with ideas borrowed from simulated annealing. Feldman, Naor and Schwarz [13] later improved this to 0.42 by using a variant of the continuous greedy algorithm. Buchbinder, Feldman, Naor and Schwarz have recently obtained an optimal $1/2$ algorithm [5].

In the setting of constrained non-monotone submodular maximization, Lee et al. [22] give a $1/(k + 2 + 1/k + \epsilon)$ approximation subject to k matroid constraints and a $1/(5-\epsilon)$ approximation for k knapsack constraints. Further work by Lee, Sviridenko and Vondrák [23] improves the approximation ratio in the case of k matroid constraints to

$1/(k+1+1/(k-1)+\epsilon)$. Feldman et al. [14] attain this ratio for non-monotone maximization in k-exchange systems. In the case of a single matroid constraint, Feldman, Naor and Schwarz [12] obtain a $1/e$ approximation by using a version of the continuous greedy algorithm. They additionally unify various applications of the continuous greedy and obtain improved approximations for non-monotone submodular maximization subject to a matroid constraint or $O(1)$ knapsack constraints.

C. Follow-up work

A set function f has *curvature* c if for all A and $x \notin A$,

$$f(A+x) \geq f(A) + (1-c)f(\{x\}).$$

The parameter c ranges in $[0,1]$. If a function has curvature c then it also has curvature c' for any $c' \geq c$. Any monotone function has curvature 1, and if a normalized function has curvature 0 then it is linear.

Vondrák [30] showed that the continuous greedy algorithm achieves an approximation ratio of $(1-e^{-c})/c$, and proved that this is optimal for the value oracle model. In [16] we generalize our framework to general curvature, achieving the same approximation ratio. In contrast to Vondrák's algorithm, ours requires knowledge of c.

II. The algorithm

Our non-oblivious local search algorithm is Algorithm 1. The input to the algorithm is a matroid $\mathcal{M}$, a monotone submodular function f, an error parameter ϵ_0, and a sampling parameter N. The matroid $\mathcal{M}$ is given as a universe $\mathcal{U}$ and a collection of independent sets $\mathcal{I} \subseteq 2^{\mathcal{U}}$, itself given as an independence oracle (an oracle deciding whether $S \in \mathcal{I}$ for an arbitrary subset $S \subseteq \mathcal{U}$). Throughout the paper, we let n denote the rank of $\mathcal{M}$ and $u = |\mathcal{U}|$.

Theorem V.5 will show how to choose ϵ_0 and N in order to obtain (with high probability) an approximation ratio as close to $1-1/e$ as desired. Theorem V.6 shows how to use the algorithm as a black-box in order to remove the small extra factor ϵ from the approximation ratio to yield a clean $(1-1/e)$-approximation algorithm.

The algorithm makes repeated calls to the function $\tilde{g}$, defined in Lemma V.2. The function $\tilde{g}$ is an approximation to the function g, which is defined in Section III using a sequence of coefficients $\beta_k^{(m)}$. These, in turn, are defined by coefficients $\gamma_k^{(m)}$, whose construction is detailed in Section III. The function $\tilde{g}$ is calculated by taking N independent samples from a particular random process defined in Lemma V.2 and so depends implicitly on the parameter N.

The purpose of the greedy phase is to produce a set S_{init} with non-negligible $g(S_{\text{init}})$. This will allow us to bound the number of iterations in the main loop. Instead of running the greedy phase with the function g, we could also run it with f, obtaining marginally inferior results. An even simpler approach is to "guess" a set S_1 such that $g(\{S_1\}) \geq \max g(S)/n$, at a multiplicative $O(n \log n)$ cost in the runtime.

Input: $\mathcal{M} = (\mathcal{U}, \mathcal{I})$, f, ϵ_0, N
Let S_{init} be the result of running the standard greedy algorithm on $(\mathcal{M}, \tilde{g})$;
$S \leftarrow S_{\text{init}}$;
repeat
  foreach *element* $a \in S$ *and* $b \in \mathcal{U} \setminus S$ **do**
    $S' \leftarrow S - a + b$;
    if $S' \in \mathcal{I}$ *and* $\tilde{g}(S') > (1+\epsilon_0)\tilde{g}(S)$ **then**
      $S \leftarrow S'$;
      break;
until *No exchange is made*;
return S;

Algorithm 1: The non-oblivious local search algorithm

The rest of the paper is organized as follows. Section III defines the function g and gives several of its properties that will be used in the following proofs. Section IV determines the locality ratio of the algorithm under the assumption that g is computed exactly (i.e. that $\tilde{g} = g$). Section V shows how this assumption can be removed by using a polynomial-time sampling procedure to estimate g and gives a complete analysis of the runtime and approximation ratio of Algorithm 1 when this procedure is used. Finally, section VI discusses future work.

In the following sections, we shall repeatedly make use of the following general property of submodular functions, given in [23].

Lemma II.1 (Lemma 1.1 in [23])**.** *Let f be a submodular function on $\mathcal{U}$. Let $C, S \subseteq \mathcal{U}$, and $\{T_i\}_{i=1}^{l}$ be a collection of subsets of $C \setminus S$ such that each element of $C \setminus S$ appears in exactly k of the subsets T_i. Then,*

$$\sum_{i=1}^{l} [f(S \cup T_i) - f(S)] \geq k\,[f(S \cup C) - f(S)]\,.$$

III. The non-oblivious potential function

We now define our non-oblivious potential function g. Borrowing intuition from the coverage case [15], we seek a function that gives extra weight to solutions that will have more flexibility in future iterations of the local search procedure. One way to do this is to incorporate the value of all subsets of a solution S into the value of $g(S)$. Then, we can give some extra value to solutions that contain a large number of good sub-solutions.

With this in mind, we define $g(S)$ to be a weighted combination of the values $f(T)$ for all $T \subseteq S$. The extra weight that a subset T contributes will depend on both the size of T and the size of the solution S on which we are

evaluating g. Our function g has the general form

$$g(S) = \sum_{k=1}^{|S|} \sum_{T \in \binom{S}{k}} \frac{\beta_k^{(|S|)}}{\binom{|S|}{k}} f(T) = \sum_{k=1}^{|S|} \beta_k^{(|S|)} \mathop{\mathbb{E}}_{T \in \binom{S}{k}} f(T). \tag{1}$$

Here, $\frac{\beta_k^{(|S|)}}{\binom{|S|}{k}}$ is the weight given to the value $f(T)$ for any subset T of size k. Alternatively, we can think of g as computing the expected value of f on a uniformly random subset of S of size k, for each $0 \leq k \leq |S|$, and then weighting the expectations by $\beta_k^{(|S|)}$, for $0 \leq k \leq |S|$. Note that the coefficients β depend on the size of $|S|$; that is, we have a separate sequence of coefficients $\beta_0^{(m)}, \ldots, \beta_m^{(m)}$ for each value of m (where m corresponds here to the size of the set S on which g is being evaluated).

In order to complete the definition of our non-oblivious local search algorithm, we must "only" specify appropriate values for these coefficients. We now turn to this task.

In later proofs, it will be more convenient to work with expressions of the form $\gamma_\ell^{(m)} = \ell \beta_\ell^{(m)}$. Furthermore, our analysis will make use of the additional coefficients $\gamma_0^{(m)}$ and $\gamma_{m+1}^{(m)}$ for each value of $m \geq 0$, although they are not used by the function g. We now define the coefficients.

The values for $\gamma_\ell^{(m)}$ are given by the initial conditions

$$\gamma_0^{(m)} = 1, \quad \gamma_{m+1}^{(m)} = e \tag{γ-BASE}$$

for all $m \geq 0$, and the upward recurrence

$$\gamma_\ell^{(m)} = m\left(\gamma_\ell^{(m-1)} - \gamma_{\ell-1}^{(m-1)}\right) \tag{γ-UP}$$

for $m \geq 1$, and $1 \leq \ell \leq m$.

In the remainder of this section, we focus on proving several properties of the resulting coefficient sequences. These properties will be used in our analysis of Algorithm 1. In Lemmas III.1 and III.2, we derive a recurrence and a closed formula for each sequence $\gamma^{(m)} = \gamma_0^{(m)}, \ldots, \gamma_{m+1}^{(m)}$. In Lemma III.3, we show that each sequence $\gamma^{(m)}$ is non-decreasing. Finally, in Lemma III.4 we derive an upper bound for the sum of each sequence $\gamma^{(m)}$.

Lemma III.1. *For each $m \geq 1$, the sequence $\gamma^{(m)}$ satisfies the recurrence:*

$$\ell\gamma_{\ell+1}^{(m)} = (2\ell - m)\gamma_\ell^{(m)} + (m - \ell + 1)\gamma_{\ell-1}^{(m)} \tag{γ-REC}$$

for $1 \leq \ell \leq m$.

Proof: We proceed by induction on m. In the case that $m = 1$, we must only show that (γ-REC) is valid for $\ell = 1$. In this case, (γ-REC) becomes

$$\gamma_2^{(1)} = \gamma_1^{(1)} + \gamma_0^{(m)}.$$

This equation follows directly from (γ-BASE) and (γ-UP), which give $\gamma_0^{(1)} = 1$, $\gamma_2^{(1)} = e$, and $\gamma_1^{(1)} = (e - 1)$.

In the general case that $m > 1$ we consider 3 subcases based on the value of ℓ. When $\ell = 1$, we have $\ell\gamma_{\ell+1}^{(m)} = \gamma_2^{(m)}$, which is equal to

$$\begin{aligned}
&m(\gamma_2^{(m-1)} - \gamma_1^{(m-1)}) && \text{by } \gamma\text{-UP} \\
&= m\Big[(2 - m + 1)\gamma_1^{(m-1)} && \text{ind. hyp.} \\
&\qquad + (m-1)\gamma_0^{(m-1)} - \gamma_1^{(m-1)}\Big] \\
&= m\Big[(1 - m + 1)\gamma_1^{(m-1)} && \text{algebra} \\
&\qquad - (1 - m + 1)\gamma_0^{(m-1)} + \gamma_0^{(m-1)}\Big] \\
&= (1 - m + 1)\gamma_1^{(m)} + m\gamma_0^{(m)} && \text{by } \gamma\text{-UP and } \gamma\text{-BASE} \\
&= (2\ell - m)\gamma_\ell^{(m)} + (m - \ell + 1)\gamma_0^{(m)} && \text{since } \ell = 1.
\end{aligned}$$

When $\ell = m$, we have $\ell\gamma_{\ell+1}^{(m)} = m\gamma_{m+1}^{(m)}$, which is equal to

$$\begin{aligned}
&m\gamma_m^{(m-1)} && \text{by } \gamma\text{-BASE} \\
&= m\left[m\gamma_m^{(m-1)} - (m-1)\gamma_m^{(m-1)}\right] && \text{algebra} \\
&= m\left[m\gamma_m^{(m-1)} - (m-1)\gamma_{m-1}^{(m-1)} - \gamma_{m-2}^{(m-1)}\right] && \text{ind. hyp.} \\
&= m\left[m(\gamma_m^{(m-1)} - \gamma_{m-1}^{(m-1)}) + \gamma_{m-1}^{(m-1)} - \gamma_{m-2}^{(m-1)}\right] \\
& && \text{algebra} \\
&= m\gamma_m^{(m)} + \gamma_{m-1}^{(m)} && \text{by } \gamma\text{-UP} \\
&= (2\ell - m)\gamma_{m-1}^{(m)} + (m - \ell + 1)\gamma_{m-1}^{(m)} && \text{since } \ell = m.
\end{aligned}$$

Finally, when $2 \leq \ell \leq m - 1$, we have $\ell\gamma_{\ell+2}^{(m)}$ equal to

$$\begin{aligned}
&m(\ell\gamma_{\ell+1}^{(m-1)} - \ell\gamma_\ell^{(m-1)}) && \text{by } \gamma\text{-UP} \\
&= m\Big[(2\ell - m + 1)\gamma_\ell^{(m-1)} && \text{ind. hyp.} \\
&\qquad + (m - \ell)\gamma_{\ell-1}^{(m-1)} - \ell\gamma_\ell^{(m-1)}\Big] \\
&= m\Big[(2\ell - m)\gamma_\ell^{(m-1)} && \text{algebra} \\
&\qquad + (m - \ell)\gamma_{\ell-1}^{(m-1)} - (\ell - 1)\gamma_\ell^{(m-1)}\Big] \\
&= m\Big[(2\ell - m)\gamma_\ell^{(m-1)} + (m - \ell)\gamma_{\ell-1}^{(m-1)} && \text{ind. hyp.} \\
&\qquad - (2\ell - m - 1)\gamma_{\ell-1}^{(m-1)} - (m - \ell + 1)\gamma_{\ell-2}^{(m-1)}\Big] \\
&= m\Big[(2\ell - m)\gamma_\ell^{(m-1)} + (m - \ell + 1)\gamma_{\ell-1}^{(m-1)} && \text{algebra} \\
&\qquad - (2\ell - m)\gamma_{\ell-1}^{(m-1)} - (m - \ell + 1)\gamma_{\ell-2}^{(m-1)}\Big] \\
&= (2\ell - m)\gamma_\ell^{(m)} + (m - \ell + 1)\gamma_{\ell-1}^{(m)}. && \text{by } \gamma\text{-UP}
\end{aligned}$$

■

Additionally, it can be shown that the coefficients are given by the following closed formula.

Lemma III.2. *For all $m \geq 1$, and $1 \leq \ell \leq m$ we have*

$$\gamma_\ell^{(m)} = (-1)^\ell \sum_{k=0}^{m-1} \frac{m!}{k!}\left[(-1)^k \tbinom{m-1-k}{\ell-1-k} e - \tbinom{m-1-k}{\ell-1}\right].$$

Proof: By induction on m, using (γ-BASE) and (γ-UP). ∎

Now, we show that the coefficient sequence $\gamma^{(m)} = \gamma_0^{(m)}, \ldots, \gamma_{m+1}^{(m)}$ is non-decreasing for each value of m. Our proof makes use of a surprising relationship between the coefficients γ and the Padé approximants to the exponential function e^x. A comprehensive overview of the theory of Padé approximants is given by Baker and Graves-Morris [3]. We shall need only need a few fundamental results from the area, which we now state.

We are primarily concerned with the Padé approximants of the function e^x. These were among the first rigorously studied Padé approximants: a full treatment of the function e^x appears in Padé's thesis [26, Part 2, Section 4].[2]

The $[\mu/\nu]$ Padé approximant to the function e^x is given by the rational function $R_{[\mu/\nu]}(x) = P_{[\mu/\nu]}(x) \,/\, Q_{[\mu/\nu]}(x)$, whose numerator and denominator are given by the polynomials:

$$P_{[\mu/\nu]}(x) = \sum_{k=0}^{\mu} \frac{x^k(\mu+\nu-k)!\mu!}{k!(\mu+\nu)!(\mu-k)!},$$
$$Q_{[\mu/\nu]}(x) = \sum_{k=0}^{\nu} \frac{(-x)^k(\mu+\nu-k)!\nu!}{k!(\mu+\nu)!(\nu-k)!}.$$

The following formula gives the error in the approximant $R_{[\mu/\nu]}$.

$$e^x Q_{[\mu/\nu]}(x) - P_{[\mu/\nu]}(x) = (-1)^\nu \frac{x^{\mu+\nu+1}}{(\mu+\nu)!} \int_0^1 e^{xt} t^\nu (1-t)^\mu \, dt. \quad (2)$$

We can restate the explicit formula from Lemma III.2 in terms of Padé numerators and denominators to obtain the following expression for $\gamma_\ell^{(m)}$:

$$(-1)^{\ell-1} m! \tbinom{m-1}{\ell-1} \left[eQ_{[m-\ell/\ell-1]}(1) - P_{[m-\ell/\ell-1]}(1) \right]. \quad (3)$$

Using (3), we can now prove our second main result regarding the γ sequences.

Lemma III.3. *For any $m \geq 1$, the sequence $\gamma^{(m)} = \gamma_0^{(m)}, \gamma_1^{(m)}, \ldots, \gamma_{m+1}^{(m)}$ is non-decreasing.*

Proof: Consider $\gamma_\ell^{(m)}$ and $\gamma_{\ell-1}^{(m)}$ for some $m \geq 1$ and $1 \leq \ell \leq m+1$. By definition, we then have

$$\gamma_\ell^{(m+1)} = (m+1)\left(\gamma_\ell^{(m)} - \gamma_{\ell-1}^{(m)}\right),$$

and so $\gamma_\ell^{(m)} \geq \gamma_{\ell-1}^{(m)}$ if and only if $\gamma_\ell^{(m+1)} \geq 0$. We now show that this must be the case for all $m \geq 1$ and $1 \leq \ell \leq m+1$. From (3) we have:

$$\begin{aligned}
&\gamma_\ell^{(m+1)} \\
&= (-1)^{\ell-1}(m+1)!\tbinom{m}{\ell-1} \\
&\qquad \cdot \left[Q_{[m+1-\ell/\ell-1]}(1)e - P_{[m+1-\ell/\ell-1]}(1)\right] \\
&= (-1)^{\ell-1}(m+1)!\tbinom{m}{\ell-1} \\
&\qquad \cdot \left[\frac{(-1)^{\ell-1}}{(m+1)!} \int_0^1 e^t t^{\ell-1}(1-t)^{m+1-\ell}\, dt\right] \\
&= \tbinom{m}{\ell-1} \int_0^1 e^t t^{\ell-1}(1-t)^{m+1-\ell}\, dt,
\end{aligned}$$

where we have used the formula (2) in the second line. Now, we note that the integral above must be non-negative, since its integrand is non-negative on the entire interval $[0,1]$. Thus, $\gamma_\ell^{(m+1)} \geq 0$. ∎

Finally, we bound the sum of the sequence $\gamma^{(m)}$. This bound will be useful for several of our results in later sections.

Lemma III.4. *Define $\alpha_m = \sum_{k=1}^m \beta_k^{(m)}$. For all $m \geq 1$,*

$$\alpha_m = O(\log m).$$

Proof: From Lemma III.3, we have $\gamma_k^{(m)} \leq \gamma_{m+1}^{(m)} = e$ for all $k \leq m+1$. Therefore,

$$\alpha_m = \sum_{k=1}^m \frac{\gamma_k^{(m)}}{k} \leq e \sum_{k=1}^m \frac{1}{k} = O(\log m).$$

∎

We have now completed our definition of g, and given the necessary properties of the coefficient sequences $\gamma^{(m)}$. We additionally note that if f is monotone submodular then so is g. Moreover if f is a coverage function, then g agrees with the non-oblivious potential function defined in [15] (a full proof of these facts can be found in [16]).

IV. Locality Ratio

In this section, we derive a bound on the locality ratio of Algorithm 1, under the assumption that g is computed *exactly* (i.e. that $\tilde{g} = g$). In the next section, we show how to remove this assumption, by using a sampling procedure. Consider an instance $(\mathcal{U}, \mathcal{I}, f)$ with optimal solution O (without loss of generality, a base), and let S be the solution produced by the algorithm. Then, note that $|S| = |O| = n$. Moreover, for every $a \in S$ and for every $b \in \mathcal{U} \setminus S$ such that $S - a + b \in \mathcal{I}$, we have

$$(1+\epsilon_0) g(S) \geq g(S - a + b).$$

Since both O and S are bases, a theorem of Brualdi [4] shows that there exists a bijection $\pi \colon S \to O$ such that $S - x + \pi(x)$ is a base of $\mathcal{M}$ for all $x \in S$. We index the elements of $S = \{s_1, \ldots, s_n\}$ arbitrarily, and then for each element $s_i \in S$, we define $o_i = \pi(s_i)$. This bijection, used here to index the sets of O and S, is essentially the only

[2] A more recent derivation of the approximants to e^x and the stated error formula can be found in Baker and Graves-Morris [3, Sections 1.2 and 10.3]. Concise statements of all the properties we use may also be found in Underhill and Wragg [28].

property of matroids that we require for the remainder of our analysis.

For a set of indices $I \subseteq [n]$, we use the notation S_I (respectively, O_I) to denote the set $\{s_i : i \in I\}$ (respectively, $\{o_i : i \in I\}$). The notation $[n]$ itself is shorthand for $\{1, \ldots, n\}$.

It will be convenient to work with the following symmetric notation. Let l, b, g be non-negative integers satisfying $l + b \leq n$ and $g + b \leq n$. Then, we define $X_{l,b,g}$ to be the multiset of sets $(S_L \cup O_G)$ for all distinct L, G, satisfying $|L| = l + b$, $|G| = g + b$, $|L \cap G| = b$. That is, $X_{l,b,g}$ is the collection of all sets containing $l + b$ elements from S, and $g + b$ elements from O, where b of the elements have the same index in both S and O.[3] We have $|X_{l,b,g}| = \binom{n}{l}\binom{n-l}{g}\binom{n-l-g}{b}$. We define $F_{l,b,g}$ to be the expected value of a uniformly random set in $X_{l,b,g}$:

$$F_{l,b,g} = \frac{1}{|X_{l,b,g}|} \sum_{a \in X_{l,b,g}} f(a).$$

We adopt the convention that $F_{l,b,g} = 0$ if one of l, b, g is negative. The proof of Theorem IV.3 makes use two inequalities following from local optimality, the definition of g and the submodularity of f. Our first ancillary inequality simply re-expresses the (approximate) local optimality of S in our symmetric notation:

Lemma IV.1.

$$\epsilon_0 n \sum_{k=1}^{n} \beta_k^{(n)} F_{k,0,0} + \sum_{i=1}^{n} \gamma_k^{(n)} (F_{k,0,0} - F_{k-1,0,1}) \geq 0.$$

Proof: Note that $S - s_i + o_i$ is a base for all $1 \leq i \leq n$. Since S is an ϵ_0-approximate local optimum, we must have $(1 + \epsilon_0) g(S) \geq g(S - s_i + o_i)$ for all such i. Summing over $1 \leq i \leq n$ gives:

$$\epsilon_0 n g(S) + n g(S) - \sum_{i=1}^{n} g(S - s_i + o_i) \geq 0. \tag{4}$$

From the definition of g and F, it follows that $g(S) = \sum_{k=1}^{n} \beta_k^{(n)} F_{k,0,0}$. We now focus on the final summation in inequality (4), and consider the coefficients of each term in the summation. First, we consider an arbitrary set in $X_{k,0,0}$. This set has the form S_I where $|I| = k$, and appears as a subset of $S - s_i + o_i$ for each value of $i \notin I$. Thus, it appears in the sum with total weight $(n-k)\frac{\beta_k^{(n)}}{\binom{n}{k}} = \frac{(n-k)\beta_k^{(n)}}{|X_{k,0,0}|}$. Next, we consider an arbitrary set in $X_{k-1,0,1}$. This set has the form $S_I + o_i$ for some I with $|I| = k - 1$ and $i \notin I$. Each such set appears as a subset of $S - s_i + o_i$ for exactly one value of i and so appears in the sum with total weight $\frac{\beta_k^{(n)}}{\binom{n}{k}} = \frac{k\beta_k^{(n)}}{n\binom{n-1}{k-1}} = \frac{k\beta_k^{(n)}}{|X_{k-1,0,1}|}$. It follows that the final summation is equivalent to

$$\sum_{k=1}^{n} (n-k)\beta_k^{(n)} F_{k,0,0} + k\beta_k^{(n)} F_{k-1,0,1}.$$

The claim follows from the definition $\gamma_k^{(n)} = k\beta_k^{(n)}$. ∎

In order to reduce the inequality from Lemma IV.1 to a statement relating $f(S)$ and $f(O)$, we shall need to use the following general inequality, which follows from the submodularity and monotonicity of f.

Lemma IV.2. *For ℓ satisfying $0 \leq \ell \leq n$,*

$$(n-\ell)F_{\ell,0,1} + \ell F_{\ell-1,0,1} \geq \ell F_{\ell-1,0,0} + (n-\ell-1)F_{\ell,0,0} + f(O).$$

Proof: In order to prove Lemma IV.2, we first prove two smaller inequalities. Consider a set S_L, where $|L| = \ell$. From Lemma II.1, we have

$$\sum_{i \in [n] \setminus L} [f(S_L + o_i) - f(S_L)] \geq f(S_L \cup O_{[n]\setminus L}) - f(S_L). \tag{5}$$

Additionally, we have

$$\begin{aligned} \sum_{i \in L} [f(S_L - s_i + o_i) - f(S_L - s_i)] &\geq \sum_{i \in L} [f(S_L + o_i) - f(S_L)] \\ &\geq f(S_L \cup O_L) - f(S_L), \end{aligned} \tag{6}$$

where the first inequality follows from the decreasing marginals characterization of submodularity and the second from Lemma II.1.[4]

Adding (5) and (6), we obtain:

$$\begin{aligned} &\sum_{i \notin L} [f(S_L + o_i) - f(S_L)] + \sum_{i \in L} [f(S_L - s_i + o_i) - f(S_L - s_i)] \\ &\quad \geq f(S_L \cup O_{[n]\setminus L}) + f(S_L \cup O_L) - 2f(S_L) \\ &\quad \geq f(S_L \cup O) - f(S_L) \\ &\quad \geq f(O) - f(S_L), \end{aligned} \tag{7}$$

where the second inequality follows from submodularity of f and the last from monotonicity of f. Inequality (7) is valid for any particular assignment of values from $[n]$ to the indices of S and O. Averaging over all possible such assignments, we obtain the inequality

$$\begin{aligned} &(n-\ell)(F_{\ell,0,1} - F_{\ell,0,0}) + \ell(F_{\ell-1,0,1} - F_{\ell-1,0,0}) \\ &\quad \geq f(O) - F_{\ell,0,0}, \end{aligned}$$

which is equivalent to the inequality stated in the Lemma. ∎

[3] We adopt that convention that if we have some element x in $S \cap O$, then sets containing x will appear multiple times in $X_{l,b,g}$, once when x is treated as an element of S and once when it is treated as an element of O. This will obviate the need to consider specially the intersection $S \cap O$ in our analysis.

[4] Note that we must have $s_i = o_i$ for each element $s_i \in S \cap O$. For all such elements, we have $f(S_L - s_i + o_i) - f(S_L - s_i) = 0$, and so we can disregard all such elements in the summation in (6).

We are now ready to prove our main result for this section.

Theorem IV.3. *Suppose O is a global optimum of f and that S is an ϵ_0-approximate local optimum of g. Then,*

$$(1 + O(\epsilon_0 n \log n)) \cdot e \cdot f(S) \geq (e-1) \cdot f(O).$$

Proof: Consider the inequality given in Lemma IV.1. From the monotonicity of f, we have $F_{k,0,0} \leq F_{n,0,0}$ for all $k \leq n$. Lemma III.4 implies that:

$$\begin{aligned}\epsilon_0 n g(S) &= \epsilon_0 n \sum_{k=1}^{n} \beta_k^{(n)} F_{k,0,0} \\ &\leq \epsilon_0 n \sum_{k=1}^{n} \beta_k^{(n)} F_{n,0,0} = O(\epsilon_0 n \log n)\, F_{n,0,0}.\end{aligned}$$

Furthermore, since $F_{0,0,0} = f(\emptyset) \geq 0$ and $F_{-1,0,1} = 0$, we have $\gamma_0^{(n)}(F_{0,0,0} - F_{-1,0,0}) \geq 0$. These inequalities, together with Lemma (IV.1), imply

$$O(\epsilon_0 n \log n)\, F_{n,0,0} + \sum_{k=0}^{n} \gamma_k^{(n)} (F_{k,0,0} - F_{k-1,0,1}) \geq 0. \quad (8)$$

Since $\gamma^{(n)}$ is non-decreasing, we have $(\gamma_{\ell+1}^{(n)} - \gamma_\ell^{(n)}) \geq 0$ for all $0 \leq \ell \leq n$. Multiplying the inequality from Lemma IV.2 by $(\gamma_{\ell+1}^{(n)} - \gamma_\ell^{(n)})$ gives

$$\begin{aligned}&(\gamma_{\ell+1}^{(n)} - \gamma_\ell^{(n)}) \\ &\cdot [(n-\ell)F_{\ell,0,1} + \ell F_{\ell-1,0,1} - \ell F_{\ell-1,0,0} - (n-\ell-1)F_{\ell,0,0}] \\ &\geq (\gamma_{\ell+1}^{(n)} - \gamma_\ell^{(n)}) \cdot f(O), \end{aligned} \quad (9)$$

for each $\ell \in \{0, \ldots, n\}$. We claim that the inequality that results from adding the $n+1$ inequalities given by (9) to inequality (8) is the desired inequality.

We first consider all terms of the form $F_{k,0,0}$. For $0 \leq k \leq n-1$, the coefficient of $F_{k,0,0}$ is

$$\begin{aligned}&\gamma_k^{(n)} - (n-k-1)(\gamma_{k+1}^{(n)} - \gamma_k^{(n)}) - (k+1)(\gamma_{k+2}^{(n)} - \gamma_{k+1}^{(n)}) \\ &= (n-k)\gamma_k^{(n)} + (2k-n+2)\gamma_{k+1}^{(n)} - (k+1)\gamma_{k+2}^{(n)} = 0,\end{aligned}$$

where the final equality follows from (γ-REC). The coefficient of $F_{n,0,0} = f(S)$ is

$$\begin{aligned}&O(\epsilon_0 n \log n) + \gamma_n^{(n)} + \gamma_{n+1}^{(n)} - \gamma_n^{(n)} \\ &= O(\epsilon_0 n \log n) + \gamma_{n+1}^{(n)} = (1 + O(\epsilon_0 n \log n)) \cdot e.\end{aligned}$$

Now, we consider all terms of the form $F_{k-1,0,1}$. For $1 \leq k \leq n$, the coefficient of $F_{k-1,0,1}$ is

$$\begin{aligned}&-\gamma_k^{(n)} + (n-k+1)(\gamma_k^{(n)} - \gamma_{k-1}^{(n)}) + k(\gamma_{k+1}^{(n)} - \gamma_k^{(n)}) \\ &= k\gamma_{k+1}^{(n)} - (2k-n)\gamma_k^{(n)} - (n-k+1)\gamma_{k-1}^{(n)} \;=\; 0,\end{aligned}$$

where the final equality follows from (γ-REC). Additionally, $F_{-1,0,1} = 0$ by definition. Finally, the coefficient of $f(O)$ is

$$\sum_{k=0}^{n} (\gamma_{k+1}^{(n)} - \gamma_k^{(n)}) = \gamma_{n+1}^{(n)} - \gamma_0^{(n)} = e - 1.$$

■

V. Estimating g

Each evaluation of $g(S)$ requires evaluating f on all subsets of S, and so we cannot compute g directly without using an exponential number of calls to the value oracle f. Fortunately, we can estimate $g(S)$ by sampling.

Suppose that $|S| = m$, and recall that $\alpha_m = \sum_{k=1}^{m} \beta_k^{(m)}$. We define the random set X using the following two step experiment. First, let L be a random variable taking value k with probability $\beta_k^{(m)}/\alpha_m$. Then, choose X as a uniformly random subset of S of size L. Then, from the linearity of expectations we have $\alpha_m \mathbb{E}\, f(X) = g(S)$. We now estimate the error incurred when g is estimated by taking N samples.

We will need the following result, which follows directly from Lemma II.1.

Lemma V.1. *Let f be a non-negative submodular function, and let S be a set of size m. For k in the range $1 \leq k \leq m$, define $F(k) = \mathbb{E}\, f(X_k)$, where X_k is a uniformly random subset of S of size k. Then $F(k) \geq (k/m) f(S)$.*

Proof: Each element $x \in S$ appears in exactly $\binom{m-1}{k-1} = \frac{k}{m}\binom{m}{k}$ of the sets in $\binom{S}{k}$. From Lemma II.1, we then have:

$$\begin{aligned}F(X_k) &= \frac{1}{\binom{m}{k}} \sum_{T \in \binom{S}{k}} [f(T) - f(\emptyset)] \\ &\geq \frac{1}{\binom{m}{k}} \frac{k}{m} \binom{m}{k} [f(S) - f(\emptyset)] \geq \frac{k}{m} f(S).\end{aligned}$$

■

We can now estimate the error arising from the sampling process.

Lemma V.2. *Let S be a set of size m. Let N be a positive integer, and $X_1, \ldots, X_N$ be N i.i.d. random samples drawn from the distribution for X. Define $\tilde{g} = \frac{1}{N}\sum_{i=1}^{N} \alpha_m f(X_i)$. For every $\epsilon > 0$,*

$$\Pr[|\tilde{g}(S) - g(S)| > \epsilon g(S)] \leq 2 \exp\left(-\Omega\left(\frac{\epsilon^2 N}{\log^2 m}\right)\right).$$

Proof: From Lemma III.3 we have $\gamma_k^{(m)} \geq \gamma_0^{(m)} = 1$ for all k. Thus, Lemma V.1 gives

$$\begin{aligned}g(S) &= \sum_{k=1}^{m} \beta_k^{(m)} \mathbb{E}\, f(X_k) \geq \sum_{k=1}^{m} \frac{k}{m} \beta_k^{(m)} f(S) \\ &= \sum_{k=1}^{m} \frac{\gamma_k^{(m)}}{m} f(S) \geq \sum_{k=1}^{m} \frac{1}{m} f(S) = f(S). \quad (10)\end{aligned}$$

Since $f(X_i) \leq f(S)$ for all i by monotonicity, Hoeffding's bound gives that $\Pr[|\tilde{g}(S) - g(S)| > \epsilon g(S)]$ is at most:

$$2 \exp\left(-\frac{2\epsilon^2 g(S)^2 N}{\alpha_m^2 f(S)^2}\right) \leq 2 \exp\left(-\frac{2\epsilon^2 N}{\alpha_m^2}\right)$$

where the inequality follows from (10). The claim then follows Lemma III.4, which bounds α_m. ■

We now suppose that Algorithm 1 uses the function $\tilde{g}$ from the Lemma V.2, where the number of samples N is a parameter of the algorithm. We shall translate the bound from Theorem IV.3, which supposed that g was computed *exactly*, into a bound for the resulting algorithm. First, we prove the following elementary result, relates local optima of g to those of the sampled function $\tilde{g}$.

Lemma V.3. *Let $\delta \leq 1/2$. Suppose that $|\tilde{g}(A) - g(A)| \leq \delta g(A)$, $|\tilde{g}(B) - g(B)| \leq \delta g(B)$ and $(1+\delta)\tilde{g}(A) \geq \tilde{g}(B)$. Then $(1+7\delta)g(A) \geq g(B)$.*

Proof: The premises imply

$$(1+\delta)^2 g(A) \geq (1+\delta)\tilde{g}(A) \geq \tilde{g}(B) \geq (1-\delta)g(B).$$

Therefore

$$\frac{(1+\delta)^2}{1-\delta} g(A) \geq g(B).$$

The expression on the left is bounded by

$$\frac{(1+\delta)^2}{1-\delta} = 1 + \frac{\delta(3+\delta)}{1-\delta} \leq 1 + 7\delta,$$

since the function $(3+\delta)/(1-\delta)$ is increasing and $\delta \leq 1/2$. ∎

We also need to know the approximation ratio of the greedy algorithm when $\tilde{g}$ is used instead of g. Similar results appear in Goundan and Schulz [19] and Calinescu et al. [7], though with a different kind of approximate oracle.

Lemma V.4. *Let $S_{\text{init}} = \{s_1, \ldots, s_n\}$ satisfy*

$$(1+\eta)g(S_{[k]}) \geq g(S_{[k-1]} + x)$$

for all $k \in [n]$ and all $x \in \mathcal{U}$ such that $S_{[k-1]} + x \in \mathcal{I}$. Let G be the maximum value taken by g on $\mathcal{I}$. Then

$$g(S_{\text{init}}) \geq \frac{G}{2 + n\eta}.$$

Proof: Our proof will use the fact (proved formally in [16]) that g is monotone and submodular. Suppose $G = g(O)$. As in Section IV, we index the elements of O so that $o_i = \pi(s_i)$ for all $i \in [n]$. Then, $S_{[k-1]} + o_k \in \mathcal{I}$ for all $k \in [n]$ and

$$(1+\eta)\sum_{k=1}^{n} g(S_{[k]}) \geq \sum_{k=1}^{n} g(S_{[k-1]} + o_k).$$

Since $g(S_{[k]}) \leq g(S_{\text{init}})$ by monotonicity, this implies

$$\begin{aligned}
(1+n\eta)g(S_{\text{init}}) &\geq n\eta g(S_{\text{init}}) + \sum_{k=1}^{n}[g(S_{[k]}) - g(S_{[k-1]})] \\
&\geq \sum_{k=1}^{n}[g(S_{[k-1]} + o_k) - g(S_{[k-1]})] \\
&\geq \sum_{k=1}^{n}[g(S_{\text{init}} + o_k) - g(S_{\text{init}})] \\
&\geq g(S_{\text{init}} \cup O) - g(S_{\text{init}}) \\
&\geq g(O) - g(S_{\text{init}}),
\end{aligned}$$

where we have used decreasing marginals in the third inequality, Lemma II.1 in the fourth, and monotonicity in the last. Rearranging,

$$(2+n\eta)g(S_{\text{init}}) \geq g(O).$$

∎

Now, we are ready to prove our main theorems regarding the runtime and approximation performance of Algorithm 1.

Theorem V.5. *There is a global constant $C_1 > 0$ such that the following is true. Given $\epsilon > 0$, set the parameters of Algorithm 1 as follows:*

$$\epsilon_0 = \frac{\epsilon}{n \log n}, \quad N = C_1 \epsilon_0^{-2} \log^2 n \log(\epsilon^{-1} n^2 u \log n).$$

With probability $1 - o(1)$, Algorithm 1 is a $(1 - \frac{1}{e} - O(\epsilon))$-approximation algorithm, running in time $\tilde{O}(\epsilon^{-3} n^4 u)$.

Proof: We analyze the algorithm under the assumption that whenever we evaluate $\tilde{g}(S)$, the value we obtain satisfies

$$(1-\epsilon_0)g(X) \leq \tilde{g}(X) \leq (1+\epsilon_0)g(X).$$

Later we will show that this happens with probability $1 - o(1)$. We can assume that $\epsilon = O(1)$.

Let O be the optimal solution for the instance we are considering and let S be the solution produced by the algorithm. Lemma V.3 shows that S is an $O(\epsilon_0)$-approximate local optimum of g. Theorem IV.3 then shows that

$$\begin{aligned}
f(S) &\geq \left(1 - \frac{1}{e}\right) f(O) - O(\epsilon_0 n \log n) f(S) \\
&\geq \left(1 - \frac{1}{e} - O(\epsilon_0 n \log n)\right) f(O) \\
&= \left(1 - \frac{1}{e} - O(\epsilon)\right) f(O).
\end{aligned}$$

We now bound the number of improvements our algorithm can make. Let G be the maximum value taken by g on $\mathcal{I}$. Applying Lemma V.4 with $1 + \eta = (1+\epsilon_0)/(1-\epsilon_0) = 1 + O(\epsilon_0)$, we deduce

$$\tilde{g}(S_{\text{init}}) \geq (1-\epsilon_0)g(S_{\text{init}}) \geq \frac{G}{2 + O(n\epsilon_0)}.$$

Every time the algorithm applies an improvement, it must improve $\tilde{g}(S)$ by at least a factor of $(1+\epsilon_0)$. Furthermore, $\tilde{g}(S) \leq (1+\epsilon_0)G$ for all S we consider. Thus, the number of improvements Algorithm 1 can make is at most:

$$\begin{aligned}
\log_{1+\epsilon_0} \frac{(1+\epsilon_0)G}{\tilde{g}(S_{\text{init}})} &\leq (1+\epsilon_0) \log_{1+\epsilon_0}(2 + O(n\epsilon_0)) \\
&= O(\epsilon_0^{-1}) = O(\epsilon^{-1} n \log n).
\end{aligned}$$

Finally, we derive a bound on the number of samples needed to ensure that with high probability $|\tilde{g}(X) - g(X)| \leq \epsilon_0 g(X)$ for all sets considered by the algorithm. The initial greedy step requires at most nu total evaluations of g, and

each improvement step requires at most nu evaluations. Thus, the algorithm requires $C_1\epsilon^{-1}n^2u\log n$ total evaluations of g for some constant C_1. Define

$$N = \Theta(\epsilon_0^{-2}\log^2 n\log(\epsilon^{-1}n^2u\log n)) = \Theta(\epsilon^{-2}n^2\log^4 n\log(\epsilon^{-1}n^2u\log n)).$$

Lemma V.2 shows that the probability that for a given set X, $|\tilde{g}(X)-g(X)| > \epsilon_0 g(X)$ is $o((\epsilon^{-1}n^2u\log n)^{-1})$. Hence this never happens for any set considered by the algorithm with probability $1-o(1)$.

The final algorithm requires a total of $\tilde{O}(\epsilon^{-3}n^4u)$ calls to the value oracle for f and $O(\epsilon^{-1}n^2u\log n)$ calls to the independence oracle for $\mathcal{M}$. Its runtime is proportional to the total number of oracle calls to f. ■

We can remove the ϵ from our approximation ratio by using a partial enumeration technique described by Khuller et al. [21] and employed by Calinescu et al. [6]. Effectively, we try to "guess" a single set in the optimal solution, and then run Algorithm 1 on an instance in which all solutions contain this set. We then iterate over all possible guesses.

Formally, for a matroid $\mathcal{M} = (\mathcal{U}, \mathcal{I})$ and an element $x \in \mathcal{U}$, the contracted matroid $\mathcal{M}/x$ is a matroid on $\mathcal{U}-x$ in which a set A is independent if and only if $A+x \in \mathcal{I}$. Similarly, for $x \in \mathcal{U}$ we define the contracted function $f_x(A)$ on $\mathcal{U}-x$ by $f_x(A) = f(A+x) - f(\{x\})$ and note that if f is monotone submodular, then so is f_x.

Algorithm 2 simply runs Algorithm 1 with suitable parameters on the instance $\mathcal{M}/x, f_x$ for each $x \in \mathcal{F}$, and returns the best resulting solution. The following theorem

Input: $\mathcal{M} = (\mathcal{U}, \mathcal{I})$, f
for $x \in \mathcal{U}$ **do**
  Let $\epsilon = C_2 n^{-1}$;
  Set ϵ_0 and N as in Theorem V.5;
  Let S_x be the result of running Algorithm 1 on $(\mathcal{M}/x, f_x, \epsilon_0, N)$;
Let $y = \operatorname{argmax}_{x\in\mathcal{U}} f(S_x + x)$;
return $S_y + y$

Algorithm 2: Clean $1-1/e$ approximation algorithm

follows easily from the analysis of Calinescu et al. [7]. We present a proof here for the sake of completeness.

Theorem V.6. *With probability $1-o(1)$, Algorithm 2 is a $(1-\frac{1}{e})$-approximation algorithm running in time $\tilde{O}(n^7u^2)$.*

Proof: Consider an instance $(\mathcal{M}, \mathcal{I})$, where n is the rank of $\mathcal{M}$. Let O be some optimal solution for this instance, and $y = \arg\max_{x\in O} f(\{x\})$. Submodularity implies that $f(O) \le \sum_{x\in O} f(\{x\}) \le nf(\{y\})$. Furthermore, Theorem V.5 shows that each call to Algorithm 1 is a $(1-1/e-O(\epsilon))$-approximation algorithm with probability $1-o(1)$. Thus, with probability $1-o(1)$ we have

$$f_y(S_y) \ge \left(1 - \frac{1}{e} - O(\epsilon)\right) f_y(O-y).$$

Let S be the solution produced by Algorithm 2 on $(\mathcal{M}, \mathcal{I})$. Then, $f(S) \ge f(S_y + y)$, and

$$\begin{aligned} f(S_y+y) &= f(\{y\}) + f_y(S_y) \\ &\ge f(\{y\}) + \left(1 - \frac{1}{e} - O(\epsilon)\right) f_y(O-y) \\ &= f(\{y\}) + \left(1 - \frac{1}{e} - O(\epsilon)\right)(f(O) - f(\{y\})) \\ &\ge \left(\frac{1}{en} + 1 - \frac{1}{e} - \frac{C_2C_3}{n}\right) f(O), \end{aligned}$$

for some constant C_3, with probability $1-o(1)$. By choosing an appropriate constant C_2, we can ensure that $C_2C_3 \le e^{-1}$. Then, $f(S) \ge (1-1/e)f(O)$ as desired. Algorithm 2 makes u calls to Algorithm 1, each taking time $\tilde{O}(\epsilon^{-3}n^4u) = \tilde{O}(n^7u)$, so its runtime is $\tilde{O}(n^7u^2)$. ■

VI. Future work

An immediate open question is whether our algorithm can be made deterministic, even if only for particular classes of functions. If f is a coverage function, we have already shown [15] that g can be computed explicitly. However, this result required access to the representation of f and so is not possible in the general value oracle model. Even reducing the amount of sampling needed to compute g would be useful, as it would improve the runtime of the algorithm.

A more general question is whether this approach can be extended to other submodular maximization problems, including non-monotone maximization or maximization over multiple matroid constraints. A major difficulty in extending the continuous greedy algorithm to this latter case is that the rounding phase does not generalize to multiple matroids. As our technique does not require any rounding, it is a natural candidate for improvement in this area.

Finally, we ask whether it is possible to match the improved performance of the continuous greedy algorithm for other problems, or restricted settings by combinatorial algorithms similar to our own. For example, in [16] we show that a variant of our algorithm matches the performance of the continuous greedy algorithm in the case that the curvature of the submodular function is constrained. It is unclear, however, whether our algorithm may match or improve results for other applications of the continuous greedy algorithm, say those presented in [12].

Acknowledgement

Y. F. thanks Anupam Gupta for introducing him the problem, and Allan Borodin for introducing him non-oblivious local search. J. W. thanks Maxim Sviridenko for several helpful discussions about the problem.

REFERENCES

[1] A. A. Ageev and M. I. Sviridenko, "Pipage rounding: A new method of constructing algorithms with proven performance guarantee," *J. of Combinatorial Optimization*, vol. 8, no. 3, pp. 307–328, Sep. 2004.

[2] P. Alimonti, "New local search approximation techniques for maximum generalized satisfiability problems," in *CIAC*, 1994, pp. 40–53.

[3] G. A. Baker and P. Graves-Morris, *Padé Approximants*, 2nd ed., ser. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1996, no. 59.

[4] R. A. Brualdi, "Comments on bases in dependence structures," *Bull. of the Austral. Math. Soc.*, vol. 1, no. 02, pp. 161–167, 1969.

[5] N. Buchbinder, M. Feldman, J. S. Naor, and R. Schwartz, "A tight linear time $(1/2)$-approximation for unconstrained submodular maximization," in *FOCS*, 2012.

[6] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák, "Maximizing a submodular set function subject to a matroid constraint (Extended abstract)," in *IPCO*, 2007, pp. 182–196.

[7] ——, "Maximizing a monotone submodular function subject to a matroid constraint," *SIAM J. Comput.*, vol. 40, no. 6, pp. 1740–1766, 2011.

[8] C. Chekuri, J. Vondrak, and R. Zenklusen, "Dependent randomized rounding via exchange properties of combinatorial structures," *FOCS*, pp. 575–584, 2010.

[9] J. Edmonds, "Matroids and the greedy algorithm," *Math. Programming*, vol. 1, no. 1, pp. 127–136, 1971.

[10] U. Feige, "A threshold of ln n for approximating set cover," *J. ACM*, vol. 45, pp. 634–652, Jul. 1998.

[11] U. Feige, V. S. Mirrokni, and J. Vondrak, "Maximizing non-monotone submodular functions," in *FOCS*, 2007, pp. 461–471.

[12] M. Feldman, J. Naor, and R. Schwartz, "A unified continuous greedy algorithm for submodular maximization," in *FOCS*, 2011, pp. 570–579.

[13] M. Feldman, J. S. Naor, and R. Schwartz, "Nonmonotone submodular maximization via a structural continuous greedy algorithm," in *ICALP*, 2011, pp. 342–353.

[14] M. Feldman, J. S. Naor, R. Schwartz, and J. Ward, "Improved approximations for k-exchange systems," in *ESA*, 2011, pp. 784–798.

[15] Y. Filmus and J. Ward, "The power of local search: Maximum coverage over a matroid," in *STACS*, 2012, pp. 601–612.

[16] ——, "A tight combinatorial algorithm for submodular maximization subject to a matroid constraint," *Preprint*, 2012, arXiv:1204.4526.

[17] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, "An analysis of approximations for maximizing submodular set functions—II," in *Polyhedral Combinatorics*. Springer Berlin Heidelberg, 1978, pp. 73–87.

[18] S. O. Gharan and J. Vondrák, "Submodular maximization by simulated annealing," in *SODA*, 2011, pp. 1098–1116.

[19] P. R. Goundan and A. S. Schulz, "Revisiting the greedy approach to submodular set function maximization," 2007, (manuscript).

[20] S. Khanna, R. Motwani, M. Sudan, and U. Vazirani, "On syntactic versus computational views of approximability," *SIAM J. Comput.*, vol. 28, no. 1, pp. 164–191, 1999.

[21] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Inf. Process. Lett.*, vol. 70, pp. 39–45, April 1999.

[22] J. Lee, V. S. Mirrokni, V. Nagarajan, and M. Sviridenko, "Non-monotone submodular maximization under matroid and knapsack constraints," in *STOC*, 2009, pp. 323–332.

[23] J. Lee, M. Sviridenko, and J. Vondrák, "Submodular maximization over multiple matroids via generalized exchange properties," *Math. of Oper. Res.*, vol. 35, no. 4, pp. 795–806, Nov. 2010.

[24] G. L. Nemhauser and L. A. Wolsey, "Best algorithms for approximating the maximum of a submodular set function," *Math. of Operations Res.*, vol. 3, no. 3, pp. 177–188, 1978.

[25] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—I," *Math. Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[26] H. Padé, "Sur la représentation approchée d'une fonction par des fractions rationnelles," *Annales scientifiques de l'École Normale Supérieure, Ser. 3*, vol. 9, pp. 3–93 (supplement), 1892.

[27] R. Rado, "Note on independence functions," *Proc. of the London Math. Soc.*, vol. s3-7, no. 1, pp. 300–320, 1957.

[28] C. Underhill and A. Wragg, "Convergence properties of padé approximants to $\exp(z)$ and their derivatives," *IMA J. of Applied Math.*, vol. 11, no. 3, pp. 361–367, 1973.

[29] J. Vondrák, "Optimal approximation for the submodular welfare problem in the value oracle model," in *STOC*, 2008, pp. 67–74.

[30] ——, "Submodularity and curvature: the optimal algorithm," in *RIMS Kokyuroku Bessatsu*, S. Iwata, Ed., vol. B23, Kyoto, 2010.

[31] J. Ward, "A $(k+3)/2$-approximation algorithm for monotone submodular k-set packing and general k-exchange systems," in *STACS*, 2012, pp. 42–53.