A Unified Continuous Greedy Algorithm for Submodular Maximization

Moran Feldman
Technion

Joseph (Seffi) Naor
Technion

Roy Schwartz
Technion

***Abstract*—The study of combinatorial problems with a submodular objective function has attracted much attention in recent years, and is partly motivated by the importance of such problems to economics, algorithmic game theory and combinatorial optimization. Classical works on these problems are mostly combinatorial in nature. Recently, however, many results based on continuous algorithmic tools have emerged. The main bottleneck of such continuous techniques is how to approximately solve a non-convex relaxation for the submodular problem at hand. Thus, the efficient computation of better fractional solutions immediately implies improved approximations for numerous applications. A simple and elegant method, called "continuous greedy", successfully tackles this issue for monotone submodular objective functions, however, only much more complex tools are known to work for general non-monotone submodular objectives.**

In this work we present a new unified continuous greedy algorithm which finds approximate fractional solutions for both the non-monotone and monotone cases, and improves on the approximation ratio for many applications. For general non-monotone submodular objective functions, our algorithm achieves an improved approximation ratio of about $1/e$. For monotone submodular objective functions, our algorithm achieves an approximation ratio that depends on the density of the polytope defined by the problem at hand, which is always at least as good as the previously known best approximation ratio of $1 - 1/e$. Some notable immediate implications are an improved $1/e$-approximation for maximizing a non-monotone submodular function subject to a matroid or $O(1)$-knapsack constraints, and information-theoretic tight approximations for Submodular Max-SAT and Submodular Welfare with k players, for *any* number of players k.

A framework for submodular optimization problems, called the *contention resolution framework*, was introduced recently by Chekuri et al. [11]. The improved approximation ratio of the unified continuous greedy algorithm implies improved approximation ratios for many problems through this framework. Moreover, via a parameter called *stopping time*, our algorithm merges the relaxation solving and re-normalization steps of the framework, and achieves, for some applications, further improvements. We also describe new monotone balanced contention resolution schemes for various matching, scheduling and packing problems, thus, improving the approximations achieved for these problems via the framework.

I. Introduction

The study of combinatorial problems with submodular objective functions has attracted much attention recently, and is motivated by the principle of economy of scale, prevalent in real world applications. Moreover, submodular functions are commonly used as utility functions in economics and algorithmic game theory. From a theoretical perspective, submodular maximization plays a major role in combinatorial optimization since many optimization problems can be represented as constrained variants of submodular maximization. Two such well studied problems are Max-Cut and Max-k-Cover [22], [23], [26], [28], [29], [31], [40], [41], [46]. In this paper, we consider the basic problem of maximizing a non-negative submodular function[1] $f : 2^{\mathcal{N}} \to \mathbb{R}^+$ over a ground set $\mathcal{N}$ under the constraint that the solution must belong to a down-monotone family[2] of subsets $\mathcal{I} \subseteq 2^{\mathcal{N}}$. This basic (constrained) submodular maximization problem generalizes the above two mentioned classic combinatorial optimization problems.

The techniques used to compute approximate solutions to various (constrained) submodular maximization problems can be partitioned into two main approaches. The first approach is combinatorial in nature, and is mostly based on local search techniques and greedy rules. This approach has been used as early as the late 70's for maximizing monotone submodular functions under the constraint that the solution should be an independent set of one of several specific matroids [12], [20], [24], [25], [27], [32], [42], [43]. Lately, this approach has been extended to include both non-monotone submodular objective functions [16], [19], [21], [48] and additional constraint sets $\mathcal{I}$ [37] (*e.g.*, independent sets of matroids intersection). Though for some problems this approach yields the current state of the art solutions [37], or even tight results [45], these solutions are usually tailored for the specific structure of the problem at hand, making extensions quite difficult.

The second approach for approximating (constrained) submodular maximization problems overcomes the above obstacle. This approach resembles a common paradigm for designing approximation algorithms and is composed of two steps. In the first step, a fractional solution is found for a relaxation of the problem. In

[1]A function $f : 2^{\mathcal{N}} \to \mathbb{R}$ is *submodular* if for every $A \subseteq B \subseteq \mathcal{N}$ and $e \in \mathcal{N}$: $f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B)$. An equivalent definition is that for every $A, B \subseteq \mathcal{N}$: $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$.

[2]A family of subsets $\mathcal{I} \subseteq 2^{\mathcal{N}}$ is *down-monotone* if $B \in \mathcal{I}$ and $A \subseteq B$ imply $A \in \mathcal{I}$. Note that many natural families of subsets $\mathcal{I}$ are down-monotone, *e.g.*, families induced by matroid and knapsack constraints.

the second step, the fractional solution is rounded to obtain an integral one while incurring only a small loss in the objective. This approach has been used to obtain improved approximations to various problems [7], [9]–[11], [34], [36]. Most notable of these results is an *asymptotically tight* approximation for maximizing a monotone submodular function given a single matroid constraint [7], [42], [43]. Two issues arise when using this approach. First, since the objective function is not linear, it is not clear how to formulate a relaxation which can be solved or even approximated efficiently. Second, given a fractional solution, one needs a rounding procedure which outputs an integral solution without losing too much in the objective function.

Let us elaborate on the first issue, namely how to find good fractional solutions to (constrained) submodular maximization problems. The standard relaxation for such a problem has a variable for every element of the groundset $\mathcal{N}$ taking values from the range $[0, 1]$. As with linear programming relaxations, the family $\mathcal{I}$ is replaced by a set of linear inequality constraints on the variables which define a down-monotone polytope[3] $\mathcal{P}$. Unlike the linear case, the formulation of an objective function for the relaxation is not obvious. A good objective function is a continuous extension of the given integral objective f which allows for efficient computation of a good fractional solution. The extension commonly used to overcome this difficulty, in the context of (constrained) submodular maximization problems, is the *multilinear extension* of f, denoted by F. The multilinear extension $F(x)$ for any $x \in [0, 1]^{\mathcal{N}}$ is the expected value of f over a random subset $R(x) \subseteq \mathcal{N}$. Each element $e \in \mathcal{N}$ is chosen independently to be in $R(x)$ with probability x_e. Formally, for every $x \in [0, 1]^{\mathcal{N}}$, $F(x) \triangleq \mathbb{E}[R(x)] = \sum_{S \subseteq \mathcal{N}} f(S) \prod_{e \in S} x_e \prod_{e \notin S}(1 - x_e)$. Such relaxations are very common since first introduced by [6] (see [7], [36], [47], [48] for several additional examples).

Even though the objective function defined by the multilinear extension is neither convex nor concave, it is still possible to efficiently compute an approximate feasible fractional solution for the relaxation, assuming its feasibility polytope $\mathcal{P}$ is down monotone and solvable[4]. The first method proposed for computing such a solution is the *continuous greedy* method [6]. It is simple and quick, and its analysis is rather short and intuitive. However, it is only known to work for the multilinear extensions of *monotone*[5] submodular functions f. For non-monotone functions f and specific polytopes, other methods are known for solving the multilinear extension, *e.g.*, for a constant number of knapsack constraints [36] and for a single matroid [21], [48]. These methods use extensions of the local search approach, as opposed to the simple continuous greedy method, making the analysis quite involved. Recently, three algorithms for the non-monotone case and general down-monotone solvable polytopes were suggested by [11]. Similarly to [21], [36], these three algorithms are also based on extensions of the local search approach. The best of the three (with respect to its approximation guarantee) uses a simulated annealing technique [21]. Therefore, these algorithms, and especially the best of the three, have quite a complex analysis.

[3]A polytope $\mathcal{P} \subseteq [0, 1]^{\mathcal{N}}$ is *down-monotone* if $x \in \mathcal{P}$ and $0 \leq y \leq x$ imply $y \in \mathcal{P}$.

[4]A polytope $\mathcal{P}$ is *solvable* if linear functions can be maximized over it in polynomial time. Using the ellipsoid algorithm, one can prove $\mathcal{P}$ is solvable by giving a polynomial-time algorithm that given x determines whether $x \in \mathcal{P}$.

[5]A function $f : 2^{\mathcal{N}} \to \mathbb{R}$ is *monotone* if $A \subseteq B \subseteq \mathcal{N}$ implies $f(A) \leq f(B)$.

A. *Our Results*

We present a new unified continuous greedy algorithm which finds approximate fractional solutions for both the non-monotone and monotone cases, and improves on the approximation ratio for many applications. For general non-monotone submodular objective functions, our algorithm achieves an improved approximation ratio of about $1/e$. For monotone submodular objective functions, our algorithm achieves an approximation ratio that depends on the density of the polytope defined by the problem at hand, which is always at least as good as the previously known best approximation ratio of $1 - 1/e$. Some notable applications are an improved $1/e$-approximation for maximizing a non-monotone submodular function subject to a matroid or $O(1)$-knapsack constraints, and tight approximations for Submodular Max-SAT and Submodular Welfare with k players, for *any* number of players k.

It turns out that the unified continuous greedy algorithm works very well with a framework presented by [11] for solving submodular optimization problems via a relaxation. Naïvely plugging our algorithm into the framework, immediately produces improved results due to its approximation ratio. Moreover, we prove that a careful use of our algorithm can further improve the framework's performance. We also show how to extend the framework to various scheduling, matching and packing problems, thus, improving upon the current best known results for these problems.

1) Measured Continuous Greedy: Though our algorithm is quite intuitive, it is based on a simple but crucially useful insight on which we now elaborate. The continuous greedy algorithm of [7] starts with an empty solution and at each step moves by a small δ in the direction of a feasible point $x \in \mathcal{P}$. Let y be the current position of the algorithm. Then x is chosen greedily (hence the name "continuous greedy") by solving $x = \text{argmax}\{w(y) \cdot x \mid x \in \mathcal{P}\}$ where the weight vector $w(y) \in \mathbb{R}^{\mathcal{N}}$ is $w(y)_e = F(y \vee \mathbf{1}_e) - F(y)$, for every $e \in \mathcal{N}$. Thus, x is chosen according to the *residual increase* of each element e, *i.e.*, $F(y \vee \mathbf{1}_e) - F(y)$. However, one would intuitively expect that the step should be chosen according to the *gradient* of $F(y)$. Our unified algorithm compensates for the difference between the residual increase of elements at point y, and $\nabla F(y)$, by *distorting* the direction x as to mimic the value of $\nabla F(y)$. This is done by decreasing x_e, for every $e \in \mathcal{N}$, by a multiplicative factor of $1 - y_e$. Therefore, our unified continuous greedy algorithm is called *measured continuous greedy*.

The measured continuous greedy algorithm, unlike local search based algorithms [11], has a parameter called *stopping time*. The stopping time controls a trade-off between two important properties of the fractional solution found by the algorithm. The first property is the value of the solution: a larger stopping time implies a better fractional solution. The second property is how much slack does the fractional solution has: a smaller stopping time implies more slack (refer to Section I-A3 for uses of the second property).

For monotone submodular objectives, the dependance of the approximation ratio on the stopping time T is identical for both our algorithm and the continuous greedy algorithm of [7]. This is somewhat counter intuitive, since our algorithm makes a somewhat "smaller" step in each iteration (recall that the movement in direction e is reduced by a multiplicative factor of $1-y_e$). This seems to suggest that the known continuous greedy algorithm is a bit wasteful. The smaller steps of our algorithm prevent this waste, keep its fractional solution within the polyope for a longer period of time, and thus, allow the use of larger stopping times.

The following two theorems quantify the guaranteed performance of the measured continuous greedy algorithm for non-monotone and monotone submodular functions. We denote by OPT the optimal integral solution. Note that the first claim of Theorem I.2, $x/T \in \mathcal{P}$, repeats, in fact, the guarantee of the continuous greedy algorithm of [7]. However, the second claim of the same theorem enables us to obtain improved approximation guarantees for several well studied problems. This property states that one can use stopping times larger than 1. The maximal stopping time that can be used depends on the density of the underlaying polytope[6] (notice that the density resembles the width parameter used by [3]).

Theorem I.1. *For any non-negative submodular function $f : 2^{\mathcal{N}} \to \mathbb{R}^+$, down-monotone solvable polytope $\mathcal{P} \subseteq [0,1]^{\mathcal{N}}$ and stopping time $T \in [0,1]$, the* measured continuous greedy *algorithm finds a point $x \in [0,1]^{\mathcal{N}}$ such that $F(x) \geq [Te^{-T} - o(1)] \cdot f(OPT)$ and $x/T \in \mathcal{P}$.*

[6]Let $\sum_{e\in\mathcal{N}} a_{i,e}x_e \leq b_i$ denote the i^{th} inequality constraint of the polytope. The *density* of $\mathcal{P}$ is defined by: $d(\mathcal{P}) = \min_{1\leq i\leq m} \frac{b_i}{\sum_{e\in\mathcal{N}} a_{i,e}}$.

Theorem I.2. *For any normalized monotone submodular function $f : 2^{\mathcal{N}} \to \mathbb{R}^+$, down-monotone solvable polytope $\mathcal{P} \subseteq [0,1]^{\mathcal{N}}$ and stopping time $T \geq 0$, the* measured continuous greedy *algorithm finds a point $x \in [0,1]^{\mathcal{N}}$ such that $F(x) \geq [1-e^{-T}-o(1)] \cdot f(OPT)$. Additionally,*

1) $x/T \in \mathcal{P}$.
2) Let $T_{\mathcal{P}} = -\ln(1 - d(\mathcal{P}) + n\delta)/d(\mathcal{P})$. Then, $T \leq T_{\mathcal{P}}$ implies $x \in \mathcal{P}$.

Theorem I.2 gives an approximation ratio of $1 - e^{-T_{\mathcal{P}}} \approx 1 - (1 - d(\mathcal{P}))^{1/d(\mathcal{P})}$. In some cases one can get a cleaner approximation ratio of exactly $1 - (1 - d(\mathcal{P}))^{1/d(\mathcal{P})}$ by guessing the most valuable single element of OPT (the technique of guessing the most valuable single element of OPT is not new, and can be found, *e.g.*, in [7]). The following theorem examplifies that. A *binary* polytope $\mathcal{P}$ is a polytope defined by constraints with only $\{0,1\}$ coefficients.

Theorem I.3. *Given a binary down-monotone solvable polytope $\mathcal{P}$ with a bounded $T_{\mathcal{P}}$ and a normalized monotone submodular function $f : 2^{\mathcal{N}} \to \mathbb{R}^+$, there is a polynomial time algorithm outputting a point $x \in \mathcal{P}$ with $F(x) \geq [1 - (1 - d(\mathcal{P}))^{1/d(\mathcal{P})}] \cdot f(OPT)$.*

2) Main Applications: Theorems I.1, I.2 and I.3 immediately provide improved approximations for various problems. We elaborate now on a few of these, starting with the non-monotone case. Theorem I.1, gives an improved $(1/e - o(1))$-approximation for finding a fractional solution for any down-monotone and solvable polytope $\mathcal{P}$. Examples of some well-studied problems for which this provides improved approximation are maximization of a non-monotone submodular function f over a single matroid [11], [21], [48] and over $O(1)$ knapsack constraints [11], [33], [36]. For both we provide an improved approximation of about $1/e$. Note that both problems are known to have an approximation of roughly ≈ 0.325 [11] via the technique of [21].

For the monotone case, Theorems I.2 and I.3 are used to immediately obtain improved approximations for various problems. Most notable is the well studied Submodular Welfare problem (refer to [7], [8], [13]–[15], [17], [18], [30], [38], [44], [47], [48] for previous results on Submodular Welfare and additional closely related variants of the problem). The above theorems provide *tight* approximations for *any* number of players k, which exactly matches the $(1-(1-1/k)^k)$-hardness result [47], [48]. This improvement is most significant for small values of k. Another problem we consider is Submodular Max-SAT. Submodular Max-SAT is a

Table I
MAIN APPLICATIONS.

Problem \ Constraint	This Paper	Previous Result		Hardness*	
Matroid (non-monotone)	$1/e - o(1)$	≈ 0.325	[11]	≈ 0.478	[21]
$O(1)$-Knapsacks (non-monotone)	$1/e - \varepsilon$	≈ 0.325	[11]	$1/2$**	
Submodular Welfare (k players)	$1-(1-1/k)^k$	$\max\left\{1-\frac{1}{e}, \frac{k}{(2k-1)}\right\}$	[7], [14]	$1-(1-1/k)^k$	[48]
Submodular Max-SAT	$3/4$	$2/3$***	[2]	$3/4$	[48]

* All hardness results are for the value oracle model, and are information theory based.
** Can be derived from the method of [48].
*** The results of [2] were achieved independently of ours.

generalization of both Max-SAT and Submodular Welfare with two players, in which a monotone submodular function f is defined over the clauses of a CNF formula, and the goal is to find an assignment maximizing the value of f over the set of satisfied clauses. For Submodular Max-SAT we get a $3/4$ approximation. The above is summarized in Table I.

3) Framework Extension: As previously mentioned, though the rounding approach to (constrained) submodular maximization problems is flexible, there are two issues that need addressing. The first one is to approximately solve a relaxation for the problem, and the other is to round the fractional solution. Building upon [3], [11] proposes a general *contention resolution framework* for rounding fractional solutions. Intuitively, the scheme works as follows. First, an approximate fractional solution x is found for the multilinear extension relaxation. Second, x is re-normalized, and a random subset of elements is sampled according to probabilities determined by x. Third and last, some of the sampled elements are discarded to ensure feasibility.

The first step can be performed by any algorithm for finding approximate fractional solutions for the multilinear relaxation. Let α be its approximation guarantee. The re-normalization factor and the decision which elements to discard are determined by a constraint specific *contention resolution scheme*. Formally, a (b, c)-balanced contention resolution scheme for a constraint family $\mathcal{I}$ is an algorithm that gets a vector $x \in b\mathcal{P}(\mathcal{I})$ (where $\mathcal{P}(\mathcal{I})$ is the convex hull of $\mathcal{I}$), picks a random set $R(x)$ according to probabilities determined by x, and then outputs a set $S \in \mathcal{I}$ obeying $\Pr[e \in S | e \in R(x)] \geq c$ for every $e \in \mathcal{N}$. If the contention resolution scheme is monotonic, *i.e.*, the probability of e to be in S only increases when elements are removed from $R(x)$, then the framework guarantees αbc approximation for maximizing a submodular function subject to the constraint family $\mathcal{I}$. One advantage of this framework is the ease by which it deals with intersections of constraints of different types (*e.g.*, matroids, knapsack constraints and matchoids).

We extend the framework of [11] by showing that finding a fractional solution for the relaxation and the re-normalization step, can both be done simultaneously using the measured continuous greedy algorithm. Equipped with this observation, we can replace the expression αbc for the approximation ratio with an improved one for both the non-monotone and the monotone cases. The improvement achieved by the new expression is most significant for small values of b, as can be seen by applications such as *k-sparse packing*.

The idea behind our method is to use b as the stopping time of Theorems I.1 and I.2, hence, directly getting a re-normalized fractional solution (as both theorems ensure). The following theorem presents the improved expressions for the approximation ratio and its proof is deferred to a full version of this paper.

Theorem I.4. *If there is a monotone (b, c)-balanced contention resolution scheme for $\mathcal{I}$, then there is an approximation of $\left(e^{-b}bc - o(1)\right)$ for $\max_{S\in\mathcal{I}}\{f(S)\}$ assuming f is non-negative and submodular, and an approximation of $\left((1-e^{-b})c - o(1)\right)$ assuming f is monotone.*

Note that the results of Theorem I.4 are better then the (αbc)-approximation of [11]. This is true, since for the non-monotone case $e^{-b} > 0.325$ for every $b \in (0, 1]$, and for the monotone case $1 - e^{-b} \geq (1 - 1/e)b$ for every $b \in (0, 1]$.

For example, consider the *k-sparse packing* problem presented by [3], who provided an approximation of $(e-1)/(e^2 \cdot k) - o(1)$ in case f is monotone. Using Theorem I.4 we can improve this approximation factor and obtain a guarantee of $1/(e \cdot k) - o(1)$ (details are deferred to a full version of this paper). In fact, we are able to improve the approximation guarantee for any constraint family $\mathcal{I}$ which contains, for example, constraints for the intersection of k matroids or k-matchoids. Additional details regarding such problems are deferred to a full version of this paper.

We also note that a simple proof of the framework which does not use the FKG inequality, but rather relays on a coupling argument of random subsets, can be presented. We defer details of this proof to a full version of this paper.

4) Balanced Contention Resolution Schemes: We provide monotone balanced contention resolution schemes for various matching, scheduling and packing problems. Using these schemes and Theorem I.4, we are able to improve the known approximation ratios for these problems. A comprehensive list of our schemes and the problems for which they provide improvement is deferred to a full version of this paper.

Two notable examples for such problems are job interval selection with k-identical machines and a *linear* objective function, and broadcast scheduling with a monotone submodular objective function. For job interval scheduling with k-identical machines, and a *linear* objective function, we get an approximation ratio approaching 1 for large values of k. The previously best approximation ratio for this problems approaches $1 - e^{-1}$ for large k's [5]. For broadcast scheduling with a monotone submodular objective function, we get an approximation ratio of $1/4$. This matches the best known approximation for the linear variant [4].

II. Preliminaries

In addition to the multilinear extension, we make use of the Lovász extension (introduced in [39]). Let $T_\lambda(z)$ be the set of elements whose coordinate in z is at least λ. The Lovász extension of a submodular function $f : 2^{\mathcal{N}} \to \mathbb{R}$ is defined as $\hat{f}(x) = \int_0^1 f(T_\lambda(x))d\lambda$. This definition can also be interpreted in probabilistic terms as the expected value of f over the set $T_\lambda(x)$, where $\lambda \sim Unif[0,1]$. Beside its use in relaxations for minimization problems, the Lovász extension can also be used to lower bound the multilinear extension via the following theorem.

Theorem II.1 (Lemma A.4 in [48]). *Let $F(x)$ and $\hat{f}(x)$ be the multilinear and Lovász extensions, respectively, of a submodular function $f : 2^{\mathcal{N}} \to \mathbb{R}$. Then, $F(x) \geq \hat{f}(x)$ for every $x \in [0,1]^{\mathcal{N}}$.*

For two vectors $x, y \in [0,1]^{\mathcal{N}}$, we use $x \vee y$ and $x \wedge y$ to denote the coordinate-wise maximum and minimum, respectively, of x and y (formally, $(x \vee y)_e = \max\{x_e, y_e\}$ and $(x \wedge y)_e = \min\{x_e, y_e\}$). We also make use of the notation $\partial_e F(x) = F(x \vee \mathbf{1}_e) - F(x \wedge \mathbf{1}_{\bar{e}})$, where $\mathbf{1}_e$ and $\mathbf{1}_{\bar{e}}$ are the characteristic vectors of the sets $\{e\}$ and $\mathcal{N} - \{e\}$, respectively. The multilinear nature of F yields the following useful observation, relating these terms to each other.

Observation II.2. *Let $F(x)$ be the multilinear extension of a submodular function $f : 2^{\mathcal{N}} \to \mathbb{R}$. Then, for every $e \in \mathcal{N}$,*

$$\partial_e F(x) = \frac{F(x \vee \mathbf{1}_e) - F(x)}{1 - x_e} = \frac{F(x) - F(x \wedge \mathbf{1}_{\bar{e}})}{x_e} \quad .$$

Consider a down-monotone polytope $\mathcal{P} \subseteq [0,1]^{\mathcal{N}}$ defined by positive sign constraints ($x \geq 0$) and additional m inequality constraints. Let $\sum_{e \in \mathcal{N}} a_{i,e} x_e \leq b_i$ denote the i^{th} inequality constraint. The *density* of $\mathcal{P}$ is defined by: $d(\mathcal{P}) = \min_{1 \leq i \leq m} \frac{b_i}{\sum_{e \in \mathcal{N}} a_{i,e}}$. Since $\mathcal{P}$ is a down monotone polytope within the hypercube $[0,1]^{\mathcal{N}}$, one can assume all coefficients $a_{i,e}$ and b_i are non-negative, and $0 < d(\mathcal{P}) \leq 1$. Further details on the properties of the density are deferred to a full version of this paper.

Given a matroid $M = (\mathcal{N}, \mathcal{I})$, its matroid polytope $\mathcal{P}(M)$ is the convex-hall of all its independent sets. The polytope $\mathcal{P}(M)$ is down-monotone since the family of independent sets $\mathcal{I}$ is down-monotone. Also, $\mathcal{P}(M)$ is solvable because the greedy algorithm can be used to maximize a linear function over $\mathcal{P}(M)$. The following theorem shows that it is possible to round a fractional point in $\mathcal{P}(M)$ without any loss, even when the objective function is a general non-negative submodular function. This theorem is based on the method of [1].

Theorem II.3 (Lemma A.8 in [48]). *Given a matroid $M = (\mathcal{N}, \mathcal{I})$, a point $x \in \mathcal{P}(M)$ and a submodular function $f : 2^{\mathcal{N}} \to \mathbb{R}^+$ with its multilinear extension $F : [0,1]^{\mathcal{N}} \to \mathbb{R}^+$, there is a polynomial time algorithm, called* pipage rounding*, outputting a random independent set $S \in \mathcal{I}$ such that $\mathbb{E}[f(S)] \geq F(x)$.*

The explicit representation of both submodular functions and matroids might be exponential in the size of their ground set. The standard way to bypass this difficulty is to assume access to these objects via oracles. For a submodular function $f : 2^{\mathcal{N}} \to \mathbb{R}^+$, given a set $S \subseteq \mathcal{N}$, the oracle returns the value of $f(S)$.[7] For a matroid $M = (\mathcal{N}, \mathcal{I})$, given a set $S \subseteq \mathcal{N}$, the oracle answers whether $S \in \mathcal{I}$.

In some proofs we use the value W defined as $W \triangleq n \cdot \max_{e \in \mathcal{N}} f(e)$. We assume throughout the paper that $\{e\} \in \mathcal{I}$ for every element $e \in \mathcal{N}$. Any element which violates this assumption belongs to a non-feasible solution, and can be removed. It is clear that this assumption implies $f(S) \leq W \leq n \cdot f(OPT)$, for every set $S \subseteq \mathcal{N}$.

III. Measured Continuous Greedy

In this section we describe the unified measured continuous greedy algorithm that works for both non-monotone and monotone cases. We analyze it for general non-monotone submodular functions and then refine the analysis to get improved results for monotone submodular functions. The parameter T of the algorithm is the *stopping time* mentioned in Theorems I.1 and I.2.

[7] Such an oracle is called *value oracle*. Other, stronger, oracle types for submodular functions are also considered in the literature, but the value oracle is probably the most widely used.

Algorithm 1: MeasuredContGreedy($f, \mathcal{P}, T$)

```
   // Initialization
1  Set: n ← |N|, δ ← T(⌈n^5 T⌉)^{-1}.
2  Initialize: t ← 0, y(0) ← 1_∅.
   // Main loop
3  while t < T do
4      foreach e ∈ N do
5          w_e(t) ← F(y(t) ∨ 1_e) − F(y(t)).
6      I(t) ← argmax{x · w(t) | x ∈ P}.
7      foreach e ∈ N do
8          y_e(t + δ) ← y_e(t) + δI_e(t) · (1 − y_e(t)).
9      t ← t + δ.
10 Return y(T).
```

It is important to note the differences of Algorithm 1 with respect to the known continuous greedy algorithm of [6]. As mentioned before, we distort the direction y which the algorithm goes to at each step. This can bee seen in line 8 of Algorithm 1 as we multiply $I_e(t)$ with $1 - y_e(t)$. There are a few technical issues to consider:

1) The way δ is defined implies that δ^{-1} has two properties: it is at least n^5, and it is dividable by T^{-1}. The last property guarantees that after $T\delta^{-1}$ iterations, t will be exactly T.
2) In some applications, the calculation of $w_e(t)$ can be done efficiently. In cases where it is not true, $w_e(t)$ can be estimated by averaging its value for enough independent random samples of $R(y(t))$. This is a standard practice (see, *e.g.*, [7]), and we omit details from this extended abstract.

Due to space limitations, many proofs of this section are deferred to the full version of this paper.

A. Analysis for Non-Monotone f

In this subsection we analyze the measured continuous greedy algorithm for general non-negative submodular functions, and prove Theorem I.1.

Lemma III.1. *For every $T \geq 0$, Algorithm 1 produces a solution x such that $x/T \in \mathcal{P}$.*

The following two lemmata give together a lower bound on the improvement achieved by the algorithm in each iteration. This lower bound is stated explicitly in Corollary III.4.

Lemma III.2. *For every time $0 \leq t < T$, $\sum_{e \in \mathcal{N}}(1 - y_e(t)) \cdot I_e(t) \cdot \partial_e F(y(t)) \geq F(y(t) \vee \mathbf{1}_{OPT}) - F(y(t))$.*

Lemma III.3. *Consider two vectors $x, x' \in [0,1]^{\mathcal{N}}$ such that for every $e \in \mathcal{N}$, $|x_e - x'_e| \leq \delta$. Then, $F(x') - F(x) \geq \sum_{e \in \mathcal{N}}(x'_e - x_e) \cdot \partial_e F(x) - O(n^3\delta^2) \cdot f(OPT)$.*

Corollary III.4. *For every time $0 \leq t < T$,*

$$F(y(t+\delta)) - F(y(t)) \geq$$
$$\delta \cdot [F(y(t) \vee \mathbf{1}_{OPT}) - F(y(t))] - O(n^3\delta^2) \cdot f(OPT).$$

Proof: Follows from Lemmata III.2 and III.3. ■

The last corollary gives a lower bound in terms of $F(y(t) \vee \mathbf{1}_{OPT})$. To make this lower bound useful, we need to lower bound the term $F(y(t) \vee \mathbf{1}_{OPT})$. This is done by the following two lemmata.

Lemma III.5. *Consider a vector $x \in [0,1]^{\mathcal{N}}$. Assuming $x_e \leq a$ for every $e \in \mathcal{N}$, then for every set $S \subseteq \mathcal{N}$, $F(x \vee \mathbf{1}_S) \geq (1-a)f(S)$.*

Lemma III.6. *For every time $0 \leq t \leq T$ and element $e \in \mathcal{N}$, $y_e(t) \leq 1 - (1-\delta)^{t/\delta} \leq 1 - e^{-t} + O(\delta)$.*

Proof: We prove the first inequality by induction on t. For $t = 0$, the inequality holds because $y_e(0) = 0 = 1 - (1-\delta)^{0/\delta}$. Assume the inequality holds for some t, let us prove it for $t + \delta$.

$$\begin{aligned} y_e(t+\delta) &= y_e(t)(1 - \delta I_e(t)) + \delta I_e(t) \\ &\leq 1 - (1-\delta)^{t/\delta} + \delta I_e(t)(1-\delta)^{t/\delta} \\ &\leq 1 - (1-\delta)^{(t+\delta)/\delta}. \end{aligned}$$

We now derive the second inequality:

$$\begin{aligned} 1 - (1-\delta)^{t/\delta} &\leq 1 - e^{-t}(1-\delta)^t \\ &\leq 1 - e^{-t} + O(\delta), \end{aligned}$$

where the last inequality holds since $t \in [0, T]$. ■

Corollary III.7. *For every time $0 \leq t < T$,*

$$F(y(t+\delta)) - F(y(t)) \geq$$
$$\delta \cdot [e^{-t} \cdot f(OPT) - F(y(t))] - O(n^3\delta^2)f(OPT).$$

Proof: By Lemma III.6, every coordinate in $y(t)$ is at most $1 - e^{-t} + O(\delta)$. Therefore, by Lemma III.5, $F(y(t) \vee \mathbf{1}_{OPT})] \geq [e^{-t} - O(\delta)] \cdot f(OPT)$. Plugging this into Corollary III.4 completes the proof. ■

At this point we have a lower bound on the improvement achieved in each iteration in terms of t, $f(OPT)$ and $F(y(t))$. In order to complete the analysis of the algorithm, we need to a lower bound on the value of $F(y(t))$ for every time t. Let $g(t)$ be defined as follows: $g(0) = 0$ and $g(t+\delta) = g(t) + \delta[e^{-t}f(OPT) - g(t)]$. The next lemma shows that a lower bound on $g(t)$ also gives a lower bound on $F(y(t))$.

Lemma III.8. *For every time $0 \leq t \leq T$,*

$$g(t) \leq F(y(t)) + O(n^3\delta) \cdot tf(OPT).$$

The function g is given by a recursive formula, thus, evaluating it is not immediate. Instead, we show that the function $h(t) = te^{-t} \cdot f(OPT)$ lower bounds g within the range $[0, 1]$.

Lemma III.9. *For every* $0 \le t \le T \le 1$*,* $g(t) \ge h(t)$*.*

Corollary III.10. *For* $T \in [0, 1]$*,*

$$F(y(T)) \ge [Te^{-T} - o(1)] \cdot f(OPT).$$

Proof: Recall that $\delta \le n^{-5}$, hence, $O(n^3\delta) = o(1)$. Apply Lemmata III.8 and III.9 to complete the proof. ∎

Theorem I.1 now follows immediately from Lemma III.1 and Corollary III.10.

B. Analysis for Monotone f

In this subsection we analyze the measured continuous greedy algorithm for normalized monotone submodular functions, and prove Theorem I.2. The proof of Theorem I.3 builds on the proofs in this subsection and is deferred to a full version of this paper. Observe that we can use all claims of Subsection III-A because a normalized monotone submodular function is a specific case of a non-negative submodular function.

Our proof has two parts. In the first part we modify the proof from Subsection III-A to show that $F(y(T)) \ge [(1-e^T) - o(1)] \cdot f(OPT)$. This part essentially reproduces the result Calinescu et al. [7] achieve using their continuous greedy algorithm. However, we are able to show that Algorithm 1 achieves the same bound when making a somewhat "smaller" step in each iteration. The novel part of the proof is the second part showing that if $\mathcal{P}$ is a packing polytope and $T \le T_{\mathcal{P}}$, then $y(T) \in \mathcal{P}$. Let us begin with the first part of the proof. The following observation logically replace Lemma III.5.

Observation III.11. *Consider a vector* $x \in [0, 1]^{\mathcal{N}}$*, then for every set* S*,* $\mathbb{E}[f(R(x) \cup S)] \ge f(S)$*.*

Proof: Follows from the monotonicity of f. ∎

Using Observation III.11 instead of Lemma III.5 in the proof of Corollary III.7, we get the following improved corollary.

Corollary III.12. *For every time* $0 \le t < T$*,*

$$F(y(t+\delta)) - F(y(t)) \ge$$
$$\delta \cdot [f(OPT) - F(y(t))] - O(n^3\delta^2) \cdot f(OPT).$$

We now define $\tilde{g}(t)$, the counterpart of g from Subsection III-A, as follows: $\tilde{g}(0) = 0$ and $\tilde{g}(t+\delta) = g(t) + \delta[f(OPT) - g(t)]$. Using $\tilde{g}$, we get the following counterpart of Lemma III.8.

Lemma III.13. *For every time* $0 \le t \le T$*,*

$$\tilde{g}(t) \le F(y(t)) + O(n^3\delta) t f(OPT).$$

Let $\tilde{h}(t)$ be the function $\tilde{h}(t) = (1 - e^{-t}) \cdot f(OPT)$. $\tilde{h}$ is the counterpart of h, and using it we can write the following counterparts of Lemma III.9 and Corollary III.10.

Lemma III.14. *For every* $0 \le t \le T$*,* $\tilde{g}(t) \ge \tilde{h}(t)$*.*

Corollary III.15. $F(y(T)) \ge [1 - e^{-T} - o(1)] \cdot f(OPT)$*.*

Proof: Recall that $\delta \le n^{-5}$, hence, $O(n^3\delta) \cdot T = O(n^{-2}) \cdot T = o(1)$. Apply Lemmata III.13 and III.14 to complete the proof. ∎

The first part of the proof is now complete. We are left to prove that if $T \le T_{\mathcal{P}}$, then $y(T) \in \mathcal{P}$. Consider some general constraint $\sum_{e \in \mathcal{N}} a_e x_e \le b$ of $\mathcal{P}$. We assume $a_e > 0$ for some $e \in \mathcal{N}$, otherwise, the constraint holds always and can be ignored. Let $I_e^t = \delta \cdot \sum_{i=0}^{t/\delta - 1} I_e(\delta \cdot i)$, *i.e.*, I_e^t is the scaled sum of I_e over all times up to t.

Lemma III.16. $\sum_{e \in \mathcal{N}} a_e \cdot I_e^T \le Tb$*.*

Lemma III.17. *For every time* $0 \le t \le T$*,*

$$y_e(t) \le 1 - e^{-I_e^t} + O(\delta) \cdot t.$$

The following lemma is a mathematical observation needed to combine the last two lemmata.

Lemma III.18. *Let* $c_1, c_2 > 0$*, and let* z_1, z_2 *be two variables whose values obey* $c_1 z_1 + c_2 z_2 = s$ *for some constant* s*. Then,* $c_1(1 - e^{-z_1}) + c_2(1 - e^{-z_2})$ *is maximized when* $z_1 = z_2$*.*

The following lemma upper bounds the left hand side of our general constraint $\sum_{e \in \mathcal{N}} a_e \cdot x_e \le b$ at time T.

Lemma III.19. *Let* $\mathcal{N}' \subseteq \mathcal{N}$ *be the set of elements with a strictly positive* a_e*. Then,*

$$\sum_{e \in \mathcal{N}'} a_e \cdot y_e(T) \le \frac{b}{d(\mathcal{P})} \cdot (1 - e^{-Td(\mathcal{P})} + O(\delta) \cdot T).$$

Proof: By Lemma III.17:

$$\sum_{e \in \mathcal{N}'} a_e \cdot y_e(T) \le \sum_{e \in \mathcal{N}'} a_e \cdot (1 - e^{-I_e^T} + O(\delta) \cdot T)$$
$$\le \sum_{e \in \mathcal{N}'} a_e (1 - e^{-I_e^T}) + O(\delta) Tb/d(\mathcal{P}).$$

The second term of the right hand side is independent of the values taken by the I_e^T's, therefore, we can upper bound the entire right hand side by assigning to the I_e^T's values maximizing the first term. Let us determine these values.

Since the summand is an increasing function of I_e^T, the sum $\sum_{e \in \mathcal{N}'} a_e \cdot I_e^T$ should have its maximal value, which is Tb by Lemma III.16. By Lemma III.18, the maximum is attained when I_e^T is identical for all elements $e \in \mathcal{N}'$.

It can be easily seen that the sole solution satisfying these conditions is $I_e^T = Tb / \sum_{e \in \mathcal{N}'} a_e$. Plugging this into the previous bound on $\sum_{e\in\mathcal{N}'} a_e \cdot y_e(T)$, we get:

$$\sum_{e\in\mathcal{N}'} a_e \cdot y_e(T) \leq$$
$$\sum_{e\in\mathcal{N}'} a_e \cdot (1 - e^{-Tb/\sum_{e\in\mathcal{N}'} a_e}) + O(\delta) \cdot Tb/d(\mathcal{P}) =$$
$$(1 - e^{-Tb/\sum_{e\in\mathcal{N}'} a_e}) \cdot \sum_{e\in\mathcal{N}'} a_e + O(\delta) \cdot Tb/d(\mathcal{P}).$$

Let us denote by $\Sigma \triangleq \sum_{e\in\mathcal{N}'} a_e$. The first term of the last expression can now be rewritten as $\Sigma(1 - e^{-Tb/\Sigma})$, and its derivative by Σ is:

$$\frac{d[S(1 - e^{-Tb/\Sigma})]}{d\Sigma} = (1 - e^{-Tb/\Sigma}) - S \cdot \frac{Tb}{S^2} e^{-Tb/\Sigma}$$
$$\geq 1 - e^{Tb/\Sigma} \cdot e^{-Tb/\Sigma} = 0.$$

Hence, increasing the value of Σ only worsens the bound we have on $\sum_{e\in\mathcal{N}'} a_e \cdot y_e(T)$. Plugging $\Sigma = b/d(\mathcal{P})$, which is an upper bound on Σ, we get:

$$\sum_{e\in\mathcal{N}'} a_e \cdot y_e(T) \leq$$
$$(1 - e^{-Tb/(b/d(\mathcal{P}))}) \cdot b/d(\mathcal{P}) + O(\delta) \cdot Tb/d(\mathcal{P}) =$$
$$\frac{b}{d(\mathcal{P})} \cdot (1 - e^{-Td(\mathcal{P})} + O(\delta) \cdot T).$$

The lemma now follows. ■

As long as the upper bound proved in the last lemma is at most b, the constraint $\sum_{e\in\mathcal{N}} a_e x_e \leq b$ is not violated. The next corollary shows that if $T \leq T_\mathcal{P}$, then this is the case.

Corollary III.20. *Given that $T \leq T_\mathcal{P}$, $y(T) \in \mathcal{P}$.*

Proof: Consider an arbitrary constraint $\sum_{e\in\mathcal{N}} a_e x_e \leq b$ of $\mathcal{P}$. By Lemma III.19,

$$\sum_{e\in\mathcal{N}} a_e \cdot y_e(T) \leq$$
$$\sum_{e\in\mathcal{N}'} a_e \cdot y_e(T_\mathcal{P}) \leq$$
$$\frac{b}{d(\mathcal{P})} \cdot (1 - e^{-T_\mathcal{P} d(\mathcal{P})} + O(\delta) \cdot T_\mathcal{P}) =$$
$$\frac{b}{d(\mathcal{P})} \cdot (1 - e^{\ln(1-d(\mathcal{P})+n\delta)} + O(\delta) \cdot T_\mathcal{P}) =$$
$$\frac{b}{d(\mathcal{P})} \cdot (d(\mathcal{P}) - n\delta + O(\delta) \cdot T_\mathcal{P}) \leq b.$$

Therefore, $y(T) \in \mathcal{P}$. ■

Theorem I.2 now follows from Lemma III.1 and Corollaries III.15 and III.20.

IV. Main Applications

This section describes the immediate applications of the measured continuous greedy algorithm. Additional applications involving our algorithm and the framework of [11] are deferred to a full version of this paper.

A. Non-Monotone Applications

Consider the problem of maximizing a non-monotone submodular function subject to a matroid constraint. Formally, given a matroid $M = (\mathcal{N}, \mathcal{I})$ and a non-negative submodular function $f : 2^\mathcal{N} \to \mathbb{R}^+$, the goal is to find an independent set $S \in \mathcal{I}$ maximizing $f(S)$.

Theorem IV.1. *There is a polynomial time $(1/e - o(1))$-approximation algorithm for maximizing a general non-negative submodular function subject to a matroid constraint.*

Proof: Apply Theorem I.1 to $\mathcal{P}(M)$ with stopping time $T = 1$ to obtain $x \in \mathcal{P}(M)$ such that $F(x) \geq [1/e - o(1)] \cdot f(OPT)$. Theorem II.3 states that x can be rounded to produce a random independent set S in M where: $\mathbb{E}[f(S)] \geq [e^{-1} - o(1)] \cdot f(OPT)$. ■

Consider the problem of maximizing a non-monotone submodular function subject to a constant number of knapsack constraints. Formally, we are given a ground set $\mathcal{N}$, a set of d knapsack constraints over this ground set (d is a constant) and a non-negative submodular function $f : 2^\mathcal{N} \to \mathbb{R}^+$. The objective is to find a set $S \subseteq \mathcal{N}$ satisfying all knapsack constraints and maximizing $f(S)$.

Let $\mathcal{P}$ be the polytope defined by the d knapsack constraints and the cube $[0,1]^\mathcal{N}$. Observe that $\mathcal{P}$ is a down monotone solvable polytope. The following theorem shows that it is possible to round fractional points in $\mathcal{P}$.

Theorem IV.2 (Theorem 2.6 in [35])**.** *Suppose there is a polynomial time α-approximation algorithm for finding a point $x \in \mathcal{P}$ maximizing $F(x)$. Then, for every constant $\varepsilon > 0$, there is a polynomial time randomized $(\alpha - \varepsilon)$-approximation algorithm for maximizing a general non-monotone submodular function subject to d knapsack constraints.*

Corollary IV.3. *For any constant $\varepsilon > 0$, there is a polynomial time $(1/e - \varepsilon)$-approximation algorithm for maximizing a general non-negative submodular function subject to knapsack constraints.*

Proof: Apply Theorem I.1 to $\mathcal{P}$ with stopping time $T = 1$ to obtain $x \in \mathcal{P}$ such that $F(x) \geq [1/e - o(1)] \cdot f(OPT)$. The corollary now follows by Theorem IV.2. ■

B. Monotone Applications

Consider the Submodular Welfare and Submodular Max-SAT problems. Both problems are generalized by the (d, r)-Submodular Partition Problem. A (d, r)-partition matroid (considered by [2]) is a matroid defined over a groundset $\mathcal{N} = \mathcal{N}_1 \cup \ldots \cup \mathcal{N}_m$, where $|\mathcal{N}_i| = r$ for every $1 \leq i \leq m$. A set $S \subseteq \mathcal{I}$ is independent if it contains up to d elements of each subset $\mathcal{N}_i$. In the (d, r)-Submodular Partition problem, given a (d, r)-partition matroid M and a normalized monotone submodular function f, the goal is to find an independent set S maximizing $f(S)$.

Observation IV.4. *Let M be a (d, r)-partition matroid, then $\mathcal{P}(M)$ is a binary down-monotone solvable polytope with density $d(\mathcal{P}(\mathcal{M})) = d/r$.*

Lemma IV.5. *There is a polynomial time $(1 - (1 - d/r)^{r/d})$-approximation algorithm for the (d, r)-Submodular Partition problem.*

Proof: Apply Theorem I.3 and Observation IV.4 to $\mathcal{P}(M)$ to obtain $x \in \mathcal{P}(M)$ such that $F(x) \geq [1-(1-d/r)^{r/d}] \cdot f(OPT)$. Theorem II.3 states that x can be rounded to produce a random independent set S in M where: $\mathbb{E}[f(S)] \geq [1 - (1 - d/r)^{r/d}] \cdot f(OPT)$. ∎

In the Submodular Welfare problem there are k players and n elements $\mathcal{E}$. Each player is associated with a normalized monotone submodular utility function f_i. The objective is to partition the elements among the players, and maximize the total utility of the players for the items that were assigned to them. Formally, we need to partition $\mathcal{E}$ into: $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_k$ (where $\mathcal{E}_i$ is the set of elements assigned to player i), and maximize $\sum_{i=1}^{k} f_i(\mathcal{E}_i)$.

Observation IV.6. *The Submodular Welfare problem with k players is a special case of the $(1, k)$-Submodular Partition problem.*

Corollary IV.7. *There is a polynomial time $1 - (1 - 1/k)^k$-approximation algorithm for the Submodular Welfare problem with k players.*

In the Submodular Max-SAT problem we are given a CNF formula and a normalized monotone submodular function over the set of clauses in the formula. The goal is to find an assignment ϕ to the variables maximizing the value of f over the set of clauses satisfied by ϕ.

Observation IV.8. *There is an α-approximation for Submodular Max-SAT if and only if $(1, 2)$-Submodular Partition has.*

Corollary IV.9. *There is a polynomial time $3/4$-approximation algorithm for the Submodular Max-SAT problem.*

References

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

[2] Y. Azar, I. Gamzu, and R. Roth, "Submodular max-sat," in *Proceedings of the ninteenth Annual European Symposium on Algorithms*, ser. ESA '2011, 2011.

[3] N. Bansal, N. Korula, V. Nagarajan, and A. Srinivasan, "On k-column sparse packing programs," in *Integer Programming and Combinatorial Optimization*, ser. Lecture Notes in Computer Science, 2010, pp. 369–382.

[4] A. Bar-Noy, S. Guha, Y. Katz, J. S. Naor, B. Schieber, and H. Shachnai, "Throughput maximization of real-time scheduling with batching," *ACM Trans. Algorithms*, vol. 5, no. 2, pp. 1–17, 2009.

[5] A. Bar-Noy, S. Guha, J. S. Naor, and B. Schieber, "Approximating the throughput of multiple machines under real-time scheduling," in *STOC '99*, 1999, pp. 622–631.

[6] G. Calinescu, C. Chekuri, M. Pal, and J. Vondrák, "Maximizing a submodular set function subject to a matroid constraint," in *IPCO '07*, 2007, pp. 182–196.

[7] ——, "Maximizing a submodular set function subject to a matroid constraint," *to appear in SIAM Journal on Computing, Special Issue for STOC 2008*, 2008.

[8] D. Chakrabarty and G. Goel, "On the approximability of budgeted allocations and improved lower bounds for submodular welfare maximization and gap," in *FOCS '08*, 2008, pp. 687–696.

[9] C. Chekuri, J. Vondrák, and R. Zenklusen, "Multi-budgeted matchings and matroid intersection via dependent rounding," in *SODA 2011*.

[10] ——, "Dependent randomized rounding via exchange properties of combinatorial structures," in *FOCS '10*, 2010, pp. 575–584.

[11] ——, "Submodular function maximization via the multilinear relaxation and contention resolution schemes," in *STOC '11*, 2011.

[12] M. Conforti and G. Cornuèjols, "Submodular set functions, matroids and the greedy algorithm: Tight worst-case bounds and some generalizations of the rado-edmonds theorem," *Disc. Appl. Math.*, vol. 7, no. 3, pp. 251–274, 1984.

[13] S. Dobzinski, N. Nisan, and M. Schapira, "Approximation algorithms for combinatorial auctions with complement-free bidders," in *STOC '05*, 2005, pp. 610–618.

[14] S. Dobzinski and M. Schapira, "An improved approximation algorithm for combinatorial auctions with submodular bidders," in *SODA '06*, 2006, pp. 1064–1073.

[15] U. Feige, "On maximizing welfare when utility functions are subadditive," in *STOC '06*, 2006, pp. 41–50.

[16] U. Feige, V. S. Mirrokni, and J. Vondrák, "Maximizing non-monotone submodular functions," in *FOCS '07*, 2007, pp. 461–471.

[17] U. Feige and J. Vondrák, "Approximation algorithms for allocation problems: Improving the factor of 1 - 1/e," in *FOCS '06*, 2006, pp. 667–676.

[18] U. Feige and J. Vondrák, "The submodular welfare problem with demand queries," *Theory of Computing*, vol. 6, no. 1, pp. 247–290, 2010.

[19] M. Feldman, J. S. Naor, and R. Schwartz, "Nonmonotone submodular maximization via a structural continuous greedy algorithm," *ICALP 2011*, 2011.

[20] M. Fisher, G. Nemhauser, and L. Wolsey, "An analysis of approximations for maximizing submodular set functions - ii," *Math. Prog. Study*, vol. 8, pp. 73–87, 1978.

[21] S. O. Gharan and J. Vondrák, "Submodular maximization by simulated annealing," in *SODA 2011*, 2011.

[22] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, pp. 1115–1145, November 1995.

[23] J. Håstad, "Some optimal inapproximability results," *J. ACM*, vol. 48, pp. 798–859, July 2001.

[24] D. Hausmann and B. Korte, "K-greedy algorithms for independence systems," *Oper. Res. Ser. A-B*, vol. 22, no. 1, pp. 219–228, 1978.

[25] D. Hausmann, B. Korte, and T. Jenkyns, "Worst case analysis of greedy type algorithms for independence systems," *Math. Prog. Study*, vol. 12, pp. 120–131, 1980.

[26] D. S. Hochbaum, *Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems*. Boston, MA, USA: PWS Publishing Co., 1997, pp. 94–143.

[27] T. Jenkyns, "The efficacy of the greedy algorithm," *Cong. Num.*, vol. 17, pp. 341–350, 1976.

[28] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. Plenum Press, 1972, pp. 85–103.

[29] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell, "Optimal inapproximability results for max-cut and other 2-variable csps?" *SIAM J. Comput.*, vol. 37, pp. 319–357, April 2007.

[30] S. Khot, R. J. Lipton, E. Markakis, and A. Mehta, "Inapproximability results for combinatorial auctions with submodular utility functions," in *in Proceedings of WINE 2005*, 2005, pp. 92–101.

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

[32] B. Korte and D. Hausmann, "An analysis of the greedy heuristic for independence systems," *Annals of Discrete Math.*, vol. 2, pp. 65–74, 1978.

[33] A. Kulik and H. Shachnai, "Improved approximations for non-monotone submodular maximization with knapsack constraints," ser. Manuscript, 2010.

[34] A. Kulik, H. Shachnai, and T. Tamir, "Maximizing submodular set functions subject to multiple linear constraints," in *SODA '09*, 2009, pp. 545–554.

[35] ——, "Approximations for monotone and non-monotone submodular maximization with knapsack constraints," 2011, manuscript.

[36] J. Lee, V. S. Mirrokni, V. Nagarajan, and M. Sviridenko, "Non-monotone submodular maximization under matroid and knapsack constraints," in *STOC '09*, 2009, pp. 323–332.

[37] J. Lee, M. Sviridenko, and J. Vondrák, "Submodular maximization over multiple matroids via generalized exchange properties," *Mathematics of Operations Research*, vol. 35, no. 4, pp. 795–806, 2010.

[38] B. Lehmann, D. Lehmann, and N. Nisan, "Combinatorial auctions with decreasing marginal utilities," in *EC '01*, 2001, pp. 18–28.

[39] L. Lovász, "Submodular functions and convexity," in *Mathematical Programming: the State of the Art*, A. Bachem, M. Grötschel, and B. Korte, Eds. Springer, 1983, pp. 235–257.

[40] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. New York, NY, USA: Cambridge University Press, 2005.

[41] R. Motwani and P. Raghavan, *Randomized algorithms*. New York, NY, USA: Cambridge University Press, 1995.

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

[43] G. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of approximations for maximizing submodular set functions - i," *Math. Prog.*, vol. 14, no. 1, pp. 265–294, 1978.

[44] N. Nisan and I. Segal, "The communication requirements of efficient allocations and supporting prices," *Journal of Economic Theory*, vol. 129, pp. 192–224, 2006.

[45] M. Sviridenko, "A note on maximizing a submodular set function subject to knapsack constraint," *Operations Research Letters*, vol. 32, pp. 41–43, 2004.

[46] L. Trevisan, G. B. Sorkin, M. Sudan, and D. P. Williamson, "Gadgets, approximation, and linear programming," *SIAM J. Comput.*, vol. 29, pp. 2074–2097, April 2000.

[47] J. Vondrák, "Optimal approximation for the submodular welfare problem in the value oracle model," in *STOC '08*. ACM, 2008, pp. 67–74.

[48] ——, "Symmetry and approximability of submodular maximization problems," in *FOCS '09*, 2009, pp. 651–670.