Budget-Smoothed Analysis for Submodular Maximization

Aviad Rubinstein
Computer Science Department, Stanford University, CA, USA

Junyao Zhao
Computer Science Department, Stanford University, CA, USA

Abstract

The greedy algorithm for monotone submodular function maximization subject to cardinality constraint is guaranteed to approximate the optimal solution to within a $1 - 1/e$ factor. Although it is well known that this guarantee is essentially tight in the worst case – for greedy and in fact any efficient algorithm, experiments show that greedy performs better in practice. We observe that for many applications in practice, the empirical distribution of the budgets (i.e., cardinality constraints) is supported on a wide range, and moreover, all the existing hardness results in theory break under a large perturbation of the budget.

To understand the effect of the budget from both algorithmic and hardness perspectives, we introduce a new notion of *budget-smoothed analysis*. We prove that greedy is optimal for *every* budget distribution, and we give a characterization for the worst-case submodular functions. Based on these results, we show that on the algorithmic side, under realistic budget distributions, greedy and related algorithms enjoy provably better approximation guarantees, that hold even for worst-case functions, and on the hardness side, there exist hard functions that are fairly robust to all the budget distributions.

2012 ACM Subject Classification Theory of computation → Submodular optimization and polymatroids

Keywords and phrases Submodular optimization, Beyond worst-case analysis, Greedy algorithms, Hardness of approximation

Digital Object Identifier 10.4230/LIPIcs.ITCS.2022.113

Related Version *Full Version*: `https://arxiv.org/abs/2102.05782`

Funding *Aviad Rubinstein*: Supported by NSF CCF-1954927, and a David and Lucile Packard Fellowship.
Junyao Zhao: Supported by NSF CCF-1954927.

Acknowledgements We thank Eric Balkanski and Matt Weinberg for interesting discussions and comments on earlier drafts.

1 Introduction

Monotone submodular function maximization subject to a cardinality constraint is a fundamental problem in combinatorial optimization with a wide variety of applications including feature selection, sensor placement, influence maximization in social networks, document summarization, etc. (see e.g. [33] and references therein). We will use influence maximization in social networks as a running example: an advertiser has a limited budget of k free product samples that she wishes to distribute to seed consumers, who will then propagate the news about the product to their friends, then their friends' friends, etc. The standard approach to this problem [30] models the expected final reach of the campaign as a monotone submodular function $f : \mathcal{P}([n]) \to \mathbb{R}$ of the set of seed consumers (where $[n]$ is the set of all users in the network). The goal of the optimization problem is to find a set of k seed consumers that (approximately) maximizes f.

Classic work shows that the simple greedy algorithm achieves a $1 - 1/e$-approximaiton to the optimal solution in the worst case [37]. Furthermore, this bound is tight for algorithms that make sub-exponential queries to the function [36, 49]; and even succinctly representable functions (e.g. simple models of influence propagation on a social network graph) do not allow better approximation algorithms unless $\mathsf{P} = \mathsf{NP}$ [22]. In theory, this tight characterization of the optimal approximation factor is very satisfying.

Given the importance of this problem in practical applications, it is also interesting to ask what is the optimal approximation factor that can be obtained on realistic instances. As one can expect, the performance of the greedy algorithm tends to be significantly better in practice (e.g. [48, 8]). When reasoning about real-world instances, there is a natural tradeoff between quality and generality of the guarantees: at one extreme, worst-case analysis only gives a $(1 - 1/e)$-approximation but applies to every instance; at the other extreme we could, in principle[1], empirically evaluate the performance of the greedy algorithm on each instance of interest, but we would have to redo this for every new instance. Ideally, we want to extend the classic worst-case model –while making minimal assumptions– to explain why efficient algorithms like greedy should obtain better-than-$(1 - 1/e)$-approximation in practice.

Coming up with useful and realistic assumptions about submodular functions continues to be an interesting and active topic of research. In Section 1.5 we survey several natural restrictions, including recent success stories that allow for improved approximation algorithms [32, 47, 9, 29, 51, 12, 48, 42]. Deferring details for later, we argue that the bottom line of this discussion is that submodular functions are complex objects, and as such modeling their beyond-worst-case behavior is tricky and application-dependent, and moreover, it is often intractable to verify the model assumptions in practice.

In this work, we consider beyond-worst-case analysis (and hardness) of submodular maxmization from a novel perspective by focusing on modeling the average-case behavior of a much simpler object: the cardinality constraint. As we now explain, our approach of perturbing the cardinality constraint is motivated by both theory and practice.

In practice, many applications of submodular maximization have multiple users with the same or similar objective but various budgets (i.e., cardinality constraints). In the example about influence maximization, multiple advertisers could advertise and propagate on the same social network and hence maximize essentially the same, possibly worst-case, submodular function. However, their budgets can easily vary by an order of magnitude or more because of different sizes of business or different amounts of funds. A concrete example is the distribution of the campaign budgets of the candidates in the 2020 Democratic Party primary elections [1]. Thus even if the social network/submodular function is worst-case, the "average" advertiser uses an "average" budget which is independent of the social network/submodular function. In a different example about feature selection, the engineers wish to make predictions in the testing phase using a small subset from the high-dimensional feature space that is selected during the training phase, and in the training phase, they apply the same standard machine learning model (e.g., linear regression, logistic regression) to the same standard datasets (e.g., ImageNet [40]) and hence optimize the same monotone (approximately) submodular objective [16, 18]. However, the number of features they want to choose can easily range from one hundred to one million depending on the computational power they have or the model complexity they prefer. Therefore, from a practical point of view, it is interesting to understand whether the average-case behavior of the budget makes the problem of submodular maximization easier to some extent.

[1] In general calculating the approximation factor on a real-world instance requires computing the value of the optimal solution. The recent work [8] provides an instance-specific method to estimate the optimal value.

In theory, all the known worst-case instances for cardinality-constrained monotone submodular maximization [37, 49, 22] are sensitive to large budget perturbations: even outputting a random solution achieves $\gtrsim 0.95$ approximation, when we perturb (i.e., multiply) the cardinality constraint by a significant multiplicative factor[2] like 0.1 or 10. Hence, from a theoretical point of view, it is interesting to investigate the effect perturbing the cardinality constraint has on the hardness of approximation results.

With the above motivations from theory and practice, we initiate the study of submodular function maximization in a semi-adversarial setting, where the (empirical) distribution of the cardinality constraints is supported on a wide range (e.g. $[x, 10x]$), from both the algorithmic and the hardness perspectives. Namely, we hope to answer the question of whether a large random perturbation of the cardinality constraint allows efficient algorithms to achieve higher optimal approximation ratio or there is a stronger hardness result that is robust to any such perturbations.

To formalize this question, we propose a simple and elegant framework called *budget-smoothed analysis*. The name is inspired by the celebrated smoothed analysis for linear programming (LP) [45]. Admittedly, the analogy is not perfect: we consider much larger perturbations than smoothed analysis for LP. However, much like smoothed analysis for LP, the generative process of random perturbations for budget-smoothed analysis for submodular maximization is grounded in concrete applications (such as viral marketing and feature selection – see discussion above).

1.1 The budget-smoothed analysis model

We study monotone submodular function maximization subject to a cardinality constraint in the following semi-adversarial setting:

▶ **Definition 1** (Budget-smoothed analysis)**.**

1. *The distribution $\tilde{\mathcal{D}}$ of budgets (e.g. uniform over $[x, 10x]$) is given as input to the adversary.*
2. *The adversary chooses a (monotone submodular) function f.*
3. *The cardinality constraint $k \sim \tilde{\mathcal{D}}$ is drawn at random and given as input to the algorithm.*
4. *The algorithm (approximately) maximizes f over all sets of size at most k.*

For any distribution $\tilde{\mathcal{D}}$, we're interested in the expected ratio $\mathcal{R}_{ALG}(f, \tilde{\mathcal{D}})$ between the value obtained by the algorithm and the optimal solution,

$$\mathcal{R}_{ALG}(f, \tilde{\mathcal{D}}) := \mathbb{E}_{k \sim \tilde{\mathcal{D}}} \left[\frac{f(ALG)}{f(OPT)} \right].$$

For notational convenience, we make the following change to the above model: Rather than sampling the budget from a distribution, each instance will be characterized by a base budget k_0 and a *budget perturbation distribution* $\mathcal{D}$, with the final cardinality constraint being $k := \rho \cdot k_0$ for $\rho \sim \mathcal{D}$. This will allow us to talk about a distribution $\mathcal{D}$ like "uniform over $[x, 10x]$" while studying the asymptotic complexity as the instance size and cardinality constraint go to infinity.

[2] Clearly, tiny perturbations of the constraint cannot escape the hardness of approximation results, because by submodularity, a $(1+\varepsilon)$-multiplicative perturbation in budget cannot affect the value of the solution by more than a $(1+\varepsilon)$-factor.

The budget-smoothed analysis model has the following advantages – First, it is simple and clean in theory, which not only provides a formal setup for studying our aforementioned question but also has the flexibility to be integrated with other models that make beyond-worst-case assumptions about the submodular functions. Second, it is easy-to-apply in practice, since calculating the (empirical) distribution of the budgets is much more tractable than verifying the complex assumptions about the submodular functions.

1.2 Our results

We revisit the classic problem of monotone submodular maximization in our model of budget-smoothed analysis, and in particular, we investigate the following fundamental questions:
Question 1: What is the optimal efficient algorithm?
Question 2: What are the worst-case instances for an arbitrary budget distribution?
Question 3: What are the optimal approximation factors for the budget distributions that are supported on a wide range (e.g. $[x, 10x]$) and what is the best budget distribution?

▶ Remark. All the hardness results below hold both in the black-box oracle model (for any algorithm that makes a subexponential number of queries), or assuming $\mathsf{P} \neq \mathsf{NP}$ in the computational model for coverage functions (on a polynomial-size graph).

Result 1 (main theorem): Optimal approximation algorithms

Our main theorem shows that a large class of algorithms that are (near-)optimal in the classic setting continue to obtain (near-)optimal approximation factors under budget-smoothed analysis for *any* distribution (Theorem 5 and Observation 9). This class includes the classic greedy algorithms, as well as (variants of) recent efficient parallel algorithms, and Map-Reduce algorithms (see the appendix of the full version). In particular, these algorithms are optimal even in comparison to algorithms that know the budget perturbation distribution $\mathcal{D}$. In other words, they intrinsically adjust themselves to the budget distribution optimally. The proof of the main theorem relies on a characterization of the worst-case instances in the model of budget-smoothed analysis, and therefore, we completely answer Questions 1 and 2.

The main theorem and Question 3 set up a win-win scenario for us: either we explain (a fraction of) the success of greedy for some interesting distributions, or we get a stronger hardness result that hold against all the budget distributions. Either way, we would bring new insights to the classic problem of submodular maximization.

Applying the main theorem, we manage to give partial answers (Result 2 and 3) to Question 3 on both positive and negative sides.

Result 2: Optimal approximation factors

For any budget perturbation distribution $\mathcal{D}$, we formulate a simple (but non-convex) mathematical program (Section 5) that computes the optimal possible approximation factor for a given budget distribution. We also include some numerical estimates for natural distributions (Table 1). For the special case of $\mathcal{D}$ supported on two budgets, we also give a closed-form solution (see the appendix of the full version).

These results are interesting on both positive and negative sides – On the positive side, the optimal approximation ratios have modest but non-negligible improvements for many interesting distributions *even for worst-case submodular functions*, which explain a fraction of the success of greedy algorithms. On the negative side, we pin down the worst-case instances for these distributions which remain significantly hard to approximate (and studying these instances might provide new insights about the structure of beyond-worst-case submodular functions).

Table 1 Empirical Results.

Budget perturbation distribution	Worst-case approximation ratio
Baseline (no perturbation)	0.6321
Uniform over $[1, 10]$	0.6675
Log-scale-uniform over $[1, 10]$	0.6674
Log-scale-uniform over $[1, 600]$	0.6808
Top 10 social/political campaigns on Facebook	0.6625
2020 Democratic presidential candidates	0.6727

We calculated tight approximation factors for worst-case monotone submodular functions for several exemplary budget distributions. See Section 5.1 for details.

Result 3: Bounding the best-case budget distribution

We also prove that for every budget distribution and any efficient algorithm, the optimal budget-smoothed analysis approximation factor is bounded away from 1, and in particular, it is at most 0.9087 (Theorem 10).

It is worth mentioning that because the program in Result 2 is non-convex, we are only able to compute the optimal approximation factors after discretizing the budget distributions with a limited number of budgets, and thus, the positive results may still have a lot of room to improve[3]. This leaves an interesting open problem: close the gap and give a complete answer to Question 3.

1.3 Broader discussion

Our model of budget-smoothed analysis introduces a new (and more tractable) angle for studying beyond-worst-case analysis and average-case hardness. We believe that there are countless future directions and applications to explore in the broader field of TCS. To exhibit this breadth of possibilities, we mention a couple of preliminary results that we have for other problems that fit into our new model:

- Submodular maximization subject to *knapsack constraint*. While the optimal $1 - 1/e$ factor can again be recovered in polynomial time, the state-of-the-art algorithms for this problem are still not completely satisfying [46, 19, 39], and the greedy algorithm does not provide any non-trivial approximation guarantee. Our preliminary results show that with budget-smoothed analysis, greedy guarantees a constant factor approximation with knapsack constraint, and in fact to date we haven't been able to rule out $1 - 1/e$ approximation (or better).
- Budget-feasible mechanism design: this is a well-studied problem in algorithmic game theory [43, 13, 17, 5, 44, 10, 20, 24, 28, 7, 11, 38, 52, 53, 34, 3, 31, 2, 25, 35]. Under a large market assumption [3] obtain a mechanism with optimal approximation guarantee, incidentally also $1 - 1/e$. Our preliminary results show that this mechanism does *not* improve at all under budget-smoothed analysis. However, the budget-smoothed analysis

[3] For comparison, [45]'s original polynomial upper bound for smoothed analysis (of a non-trivial variant) of the Simplex algorithm was $\tilde{O}(d^{55}n^{86}\sigma^{-30} + d^{70}n^{86})$ iterations (d is the number of variables, n is the number of constraints, and σ^2 is the variance of the perturbation), which is significantly improved now [15]. That said, in light of Result 3, we do not expect the positive side of budget-smoothed analysis to fully explain the success of greedy in practice, but explaining a greater fraction of success or showing a more robust hardness result would still be interesting.

inspires a *new mechanism* that is not only optimal for every budget distribution but also instance-optimal among a canonical class of mechanisms, and it also significantly outperforms [3]'s mechanism on realistic distributions empirically.

In hindsight, although the performance improvement guaranteed by budget-smoothed analysis is relatively moderate, we believe that the optimality of an algorithm in the model of budget-smoothed analysis is a theoretical evidence that the algorithm is not just worst-case optimal but also likely to perform favorably on realistic instances. In other words, budget-smoothed analysis offers an analytically approachable beyond-worst-case performance test for the worst-case optimal algorithms of budget-constrained problems, which helps us identify (or design) the "right" algorithm among various worst-case optimal algorithms.

1.4 Roadmap

In Section 1.5 we survey several other approaches to beyond-worst-case submodular maximization. In Section 3 we prove our core technical result, namely that greedy obtains optimal approximation factors for any distribution; in the appendix of the full version, we extend this result to other related algorithms. Henceforth, we build on these techniques; in particular we simply analyze the approximation factors of the greedy algorithm. In Section 4, we prove that the optimal approximation factor is bounded away from 1 for any distribution. In Section 5, we characterize the optimal approximation factor by a program, which we then use to simulate several exemplary distributions.

1.5 Beyond-worst-case submodular functions

Due to the popularity of submodular maximization in practice, there is a lot of interest in understanding and designing algorithms for "typical" cases. We discuss a few approaches below. We note that our model of beyond-worst-case *cardinality constraint* is orthogonal to any assumptions about the submodular function, and in principle could be combined with any of them to obtain even stronger results.

The model most closely in spirit to our smoothed-analysis-like approach is to take a worst-case submodular function and perturb it with random noise. The most straightforward way of doing this is independently perturbing the value of the function for each set. Unfortunately, this breaks the submodularity, which makes the problem significantly *harder*, even for small perturbations: [27] barely recovers the $1 - 1/e$ approximation factor in this setting (under further restrictions and with a technically involved algorithm).

Another approach is to consider *coverage functions*, an important special class of monotone submodular functions. This restriction has been successful for learning submodular functions [6, 4, 23], but Feige's NP-hard instance already rules out efficient algorithms with improved approximation ratios for this case. One may combine this restriction with perturbations of the weights of the elements of the ground set; but it is not hard to show that Feige's instance can be made robust even to very large amounts of noise. Another alternative is to consider special classes of graphs, e.g., power law, small-world, or triangle-dense that are common for social networks [50, 26]. But again Feige's instance either already satisfies all of those, or can be adapted to do so.

Another popular restriction of monotone submodular functions is *bounded curvature* [14], which restricts the extent to which ground elements interact; this indeed allows for better algorithms with applications to e.g. maximum entropy sampling [14, 47, 9, 29, 51]. But bounded curvature seems too restrictive for applications like influence in social networks and consumers' valuations with diminishing returns[4].

[4] If, for example, we already selected all of a node's neighbors, the marginal contribution of adding

To cope with the limited applicability of curvature, the original paper of [14] also defined a relaxed notion of *greedy curvature*, which only restricts the interaction between elements selected by the greedy algorithm and elements in the optimal solution. In exciting recent work, [48] define various notions of *sharpness* which only restricts the interactions of the average element of the optimal solution. Both greedy curvature and sharpness parameters suffer from the disadvantage that they may be intractable to compute (both are assumptions about interaction of elements with the optimal solutions, and if we knew the optimal solution...). Moreover, due to their complicated form it's hard to heuristically reason about their fit for any particular application. Nevertheless, on the positive side both are more realistic than vanilla curvature assumption, and combining them with our budget-smoothed analysis model is an interesting direction for future research.

[12] study submodular maximization under a *stability* assumption, i.e. they assume that the optimal subset does not change when the function is perturbed. [48] argue that in the context of submodular maximization, stable instances may fail to capture significant interaction between elements. As in the case of greedy curvature and sharpness, it is also not clear how to compute the stability of a function, or reason about instances that we expect to be stable.

Finally, one setting that is both natural and allows for improved approximation factors is influence maximization in undirected graphs [32, 41, 42]. Specifically, [32] prove that the greedy algorithm obtains a $(1 - 1/e + \varepsilon)$-approximation (for some small unspecified constant $\varepsilon > 0$) for the independent cascade model on undirected graph. [42] show that in the linear threshold model the greedy algorithm does not beat the $(1 - 1/e)$-approximation factor (by any constant, in the worst case).

2 Preliminaries

▶ **Definition 2.** *A function $f : 2^V \to \mathbb{R}_{\geq 0}$ is submodular if for all $S \subseteq T \subseteq V$ and $i \in V \setminus T$, $f(S \cup \{i\}) - f(S) \geq f(T \cup \{i\}) - f(T)$, where V is called ground set. Moreover, we denote the marginal gain by $f(X \mid S) := f(X \cup S) - f(S)$.*

We make the following conventions in this paper – When we say **"efficient algorithm"**, we mean polynomial time algorithms in the general computation model assuming $\mathsf{P} \neq \mathsf{NP}$, or algorithms using sub-exponential number of function queries in the oracle query model. Moreover, we consider **continuous distribution of budget perturbations** $\mathcal{D}$, and we let $\mathcal{D}(k)$ denote the distribution of budgets in which a budget is sampled by multiplying a random perturbation factor $\rho \sim \mathcal{D}$ with k (if $\rho \cdot k$ is fractional, we can round it to an integer). Furthermore, following Definition 1, we denote $\mathcal{R}_{ALG}(\mathcal{D}(k)) := \min_f \mathcal{R}_{ALG}(f, \mathcal{D}(k))$ s.t. f is monotone and submodular.

The hard instances in our analysis can be built on top of either Feige's max-k-cover instances [22] or Vondrák's hard instances [49]. In the following theorem, we summarize useful properties of these two hardness results.

▶ **Theorem 3.** *There exists a class of monotone submodular functions $\mathcal{C}$ such that for every $\epsilon > 0$, for any efficient algorithm $\mathcal{A}$ that given a submodular function f and an integer l outputs a set X_l of cardinality l, for every sufficiently large k that grows with size of instance, there is a submodular function $f_k \in \mathcal{C}$ such that*

this node is diminished to zero. For consumers' valuations, the marginal contribution of, e.g. the one-thousandth apple, is again diminished to essentially zero. This means that curvature is unbounded in both settings (see [14, 47] for formal definitions).

i *for all $l \le k$, $f_k(O_l) = (l/k) f_k(O_k)$, where O_l is the optimal set that maximizes f_k among all cardinality-l sets, and*

ii *for all l, $f_k(X_l) \le (1 - e^{-l/k} + \epsilon) f_k(O_k)$.*

Next, we state a standard lemma for greedy analysis.

▶ **Lemma 4.** *Given a monotone submodular f, we let X_k and O_k denote the greedy solution and the optimal solution of cardinality k, respectively. Then, for all $i, k > 0$, $f(X_i) - f(X_{i-1}) \ge \frac{1}{k}(f(O_k) - f(X_{i-1}))$.*

Proof. Let x_i denote the i-th element selected by greedy. It holds that

$$\begin{aligned} f(X_i) - f(X_{i-1}) = f(x_i \mid X_{i-1}) &\ge \frac{1}{k} \cdot \sum_{o \in O_k} f(o \mid X_{i-1}) \\ &\ge \frac{1}{k} \cdot f(O_k \mid X_{i-1}) \ge \frac{1}{k}(f(O_k) - f(X_{i-1})), \end{aligned}$$

where the first inequality is by greedy selection, the second is by submodularity, and the third is by monotonicity. ◀

3 Greedy is Optimal

In this section, we prove our core technical result: greedy is optimal for submodular maximization with respect to arbitrary distribution of budget perturbations.

▶ **Theorem 5.** *For any distribution of budget perturbations $\mathcal{D}$, for every $\epsilon' > 0$, for any efficient algorithm $\mathcal{A}$, for every sufficiently large*[5] *k that grows with the size of instance, it holds that $\mathcal{R}_{\mathcal{A}}(\mathcal{D}(k)) \le (1 + \epsilon')\mathcal{R}_{greedy}(\mathcal{D}(k))$.*

Theorem 5 follows directly from Theorem 6 using a discretization argument.

▶ **Theorem 6.** *For any perturbation factors $0 < \rho_1 < \rho_2 < \cdots < \rho_m$, for every $\epsilon > 0$, there exists a sufficiently large k that grows with the size of instance such that given m budgets $k_1 = \rho_1 \cdot k, \ldots, k_m = \rho_m \cdot k$, for any monotone submodular function $f^{(bad\text{-}for\text{-}greedy)}$, for any efficient algorithm $\mathcal{A}$, there exists a monotone submodular function f such that*

(i) Greedy is (almost) no worse than $\mathcal{A}$ on f: *for all $i \in [m]$, the solution Y_{k_i} computed by $\mathcal{A}$ for budget k_i has value $f(Y_{k_i}) \le (1 + \epsilon) f(X_{k_i})$, where X_{k_i} is the greedy solution for budget k_i,*

(ii) f is as hard as $f^{(bad\text{-}for\text{-}greedy)}$ for greedy: *for all $i \in [m]$, given budget k_i, the approximation ratio of greedy on f is at most $1 + \epsilon$ times the approximation ratio of greedy on $f^{(bad\text{-}for\text{-}greedy)}$.*

Proof of Theorem 5. For arbitrarily small $\tau > 0$, let $\rho_{\min}$ and $\rho_{\max}$ be such that the mass of $\mathcal{D}$ on $[\rho_{\min}, \rho_{\max}]$ is at least $1 - \tau$. We discretize $\{\rho_{\min} \cdot k, \rho_{\min} \cdot k + 1, \ldots, \rho_{\max} \cdot k\}$ into $\{\rho_{\min} \cdot k, (1+\delta)\rho_{\min} \cdot k, (1+\delta)^2 \rho_{\min} \cdot k, \ldots, \rho_{\max} \cdot k\}$. Without loss of generality, we assume that there exists m such that $(1 + \delta)^{m-1} \rho_{\min} = \rho_{\max}$ and every $k_i := (1 + \delta)^{i-1} \rho_{\min} \cdot k$ is integral. Let f^* be the worst-case monotone submodular function for which greedy achieves only $\mathcal{R}_{\text{greedy}}(\mathcal{D}(k))$ approximation in expectation. By Theorem 6, for any efficient algorithm $\mathcal{A}$, there is a monotone submodular function f such that for all $i \in [m]$, the solution Y_{k_i}

[5] The implicit dependence of k on ϵ in Theorem 3 carries over to the dependence of k on ϵ' in this statement, and therefore, we keep such dependence implicit, and we are mostly interested in the asymptotic result.

outputted by $\mathcal{A}$ for budget k_i only achieves $f(Y_{k_i}) \leq (1+\epsilon)f(X_{k_i})$, where X_{k_i} is the greedy solution for budget k_i, and moreover, for every budget k_i,

$$\frac{f(X_{k_i})}{f(O_{k_i})} \leq (1+\epsilon)\frac{f^*(X^*_{k_i})}{f^*(O^*_{k_i})} \tag{1}$$

where O_{k_i} and $O^*_{k_i}$ denote optimal size-k_i sets of f and f^* respectively, and $X^*_{k_i}$ denotes the size-k_i greedy solution for f^*.

Besides, because marginal gain in each iteration of greedy is non-increasing, we have that $(1+\delta)f(X_{k_{i-1}}) \geq f(X_{k_i})$. Furthermore, without loss of generality, we assume that $f(Y_b)$ is non-decreasing in b, since otherwise, for budget b, we can let the algorithm choose the best solution among Y_l for all $l \leq b$ instead. For any $2 \leq i \leq m$ and any budget b such that $k_{i-1} \leq b \leq k_i$, it follows that

$$\begin{aligned}
\frac{f(Y_b)}{f(O_b)} &\leq \frac{f(Y_{k_i})}{f(O_b)} && \text{(Since } b \leq k_i\text{)} \\
&\leq (1+\epsilon)\frac{f(X_{k_i})}{f(O_b)} && \text{(By Theorem 6)} \\
&\leq (1+\epsilon)(1+\delta)\frac{f(X_{k_{i-1}})}{f(O_b)} && \text{(Non-increasing marginal gain)} \\
&\leq (1+\epsilon)(1+\delta)\frac{f(X_{k_{i-1}})}{f(O_{k_{i-1}})} && \text{(Since } b \geq k_{i-1}\text{)} \\
&\leq (1+\epsilon)^2(1+\delta)\frac{f^*(X^*_{k_{i-1}})}{f^*(O^*_{k_{i-1}})} && \text{(By Eq. (1))} \\
&\leq (1+\epsilon)^2(1+\delta)\frac{f^*(X^*_b)}{f^*(O^*_{k_{i-1}})} && \text{(Since } X^*_{k_{i-1}} \subseteq X^*_b\text{)} \\
&\leq (1+\epsilon)^2(1+\delta)^2\frac{f^*(X^*_b)}{f^*(O^*_b)}. && \text{(}f^*(O^*_b) \leq \frac{b}{k_{i-1}} f^*(O^*_{k_{i-1}})\text{ by submodularity)}
\end{aligned}$$

Therefore, for every budget b in $\{\rho_{\min} \cdot k,\ \rho_{\min} \cdot k + 1, \ldots, \rho_{\max} \cdot k\}$, $\mathcal{A}$ can achieve on f in expectation at most a factor of $(1+\epsilon)^2(1+\delta)^2$ times what greedy achieves on f^*. The proof finishes since δ, ϵ, τ can be arbitrarily small. ◀

3.1 Proof of Theorem 6

In our proof we will not derive the analytic formula of the approximation ratio, but instead, the proof works in a black-box way – First, we introduce an array of parameters such that every instance can be characterized by these parameters, and we can show a parameterized guarantee of the marginal gain for each iteration of greedy. Then, we construct a hard instance characterized by the same parameters such that the best possible marginal gains for this instance always match the parametrized guarantees from greedy. It follows that the performance of greedy is optimal for every budget. Our hard instance has the following nice structure: it is a convex combination of disjoint-support copies of the classic hard instances guaranteed by Theorem 3.

Proof of Theorem 6. Proof setup: bounding a single step of greedy performance

We first lower bound the single-step performance of greedy solutions. By Lemma 4, we have the following performance guarantees for each iteration of greedy,

$$\forall\, l \in [m], \quad f(X_i) - f(X_{i-1}) \geq \frac{1}{k_l} \cdot (f(O_{k_l}) - f(X_{i-1})) \quad \textbf{(}\boldsymbol{l}\textbf{-th guarantee)},$$

where O_{k_l} denotes the optimal solution of cardinality k_l, and we call the inequality associated with O_{k_l} the l-th guarantee. Given any $1 \le l_1 < l_2 \le m$, if the l_2-th guarantee dominates (i.e., is at least as large as the l_1-th guarantee) at some iteration i, then the l_2-th guarantee will keep dominating the l_1-th guarantee for all the iterations $i' \ge i$, because the two guarantees are linear functions with variable $f(X_{i-1})$, and the l_2-th guarantee decreases slower than the l_1-th guarantee. Therefore, as $f(X_i)$ increases, the best guarantee can only transit from some l to some $l' > l$. Given an instance, we let $t \le m-1$ be the number of times such transition occurs until k_m-th iteration and let $l_1 < l_2 < \cdots < l_t$ be the indices of the corresponding best guarantees.

For $j \le t-1$, let F_j be the lowest possible value of $f(X_{i-1})$, for which the j-th transition occurs,

$$\underbrace{\frac{1}{k_{l_j}} \cdot (f(O_{k_{l_j}}) - F_j)}_{(l_j\text{-th guarantee})} = \underbrace{\frac{1}{k_{l_{j+1}}} \cdot (f(O_{k_{l_{j+1}}}) - F_j)}_{(l_{j+1}\text{-th guarantee})}. \tag{2}$$

We will be particularly interested in the quantity $r_j := F_j / f(O_{k_{l_j}})$. Plugging into Eq. (2), we have that

$$r_j = \frac{1 - (k_{l_j}/k_{l_{j+1}}) \cdot (f(O_{k_{l_{j+1}}})/f(O_{k_{l_j}}))}{1 - (k_{l_j}/k_{l_{j+1}})}. \tag{3}$$

Lower bounding the total value of the greedy solution recursively

For $q \ge 0$, we denote by $f^{(\text{greedy-lb})}(q)$ the best lower bound induced by the union of "l-th guarantees" on the value of the q-th iterate of the greedy algorithm, namely

$$f^{(\text{greedy-lb})}(q) := f^{(\text{greedy-lb})}(q-1) + \max_l \left\{ \frac{1}{k_l} \cdot (f(O_{k_l}) - f^{(\text{greedy-lb})}(q-1) \right\}.$$

Now we analyze $f^{(\text{greedy-lb})}(q)$ specifically for the instance with before-mentioned guarantee transitions. We start from the l_1-th guarantee and let $f^{(\text{greedy-lb})}(0) = 0$. Inductively, suppose that in the current iteration q, the l_j-th guarantee dominates the others, we apply the l_j-th guarantee $f^{(\text{greedy-lb})}(q) - f^{(\text{greedy-lb})}(q-1) = (f(O_{k_{l_j}}) - f^{(\text{greedy-lb})}(q-1))/k_{l_j}$ and continue iteratively until we reach some i_j-th iteration such that $f^{(\text{greedy-lb})}(i_j - 1) \le r_j \cdot f(O_{k_{l_j}}) < f^{(\text{greedy-lb})}(i_j)$. At the i_j-th iteration, l_{j+1}-th guarantee starts dominating, and thus, we switch to the l_{j+1}-th guarantee and continue like above.

Approximation ratio based on $f^{(\text{greedy-lb})}(q)$'s is determined by r_j's

We claim that the parameters r_j fully determine the ratio between the greedy lower bound $f^{(\text{greedy-lb})}(k_{l_i})$ and $f(O_{k_{l_i}})$ for all $i \in [t]$. To see this, first observe that by 3 we can infer $f(O_{k_{l_{j+1}}})/f(O_{k_{l_j}})$ from r_j. We can assume that $f(O_{k_{l_1}})$ is fixed without loss of generality, and then the parameters r_j determine all the remaining $f(O_{k_{l_j}})$, i.e., for all $1 < j \le t$,

$$f(O_{k_{l_j}}) = f(O_{k_{l_1}}) \cdot \prod_{j' \le j-1} \left(\frac{k_{l_{j'+1}}}{k_{l_{j'}}} - \left(\frac{k_{l_{j'+1}}}{k_{l_{j'}}} - 1 \right) r_{j'} \right). \tag{4}$$

Moreover, the greedy lower bound $f^{(\text{greedy-lb})}(k_{l_i})$ by definition is a linear combination of the $f(O_{k_{l_j}})$'s. Therefore, the ratio between any $f^{(\text{greedy-lb})}(k_{l_i})$ and $f(O_{k_{l_i}})$ is fully characterized by r_j. In the other words, given an instance, we can get the approximation ratios of the greedy algorithm that depend only on its parameters r_j.

By definition of r_1, any feasible r_1 has to satisfy $r_1 \leq 1$, and by our assumption of the transitions, any feasible r_j should satisfy $r_{j-1} \cdot f(O_{k_{l_{j-1}}}) \leq r_j \cdot f(O_{k_{l_j}})$ for all $1 < j \leq t$, which is equivalent to

$$\begin{aligned} r_{j-1} &\leq r_j \cdot (f(O_{k_{l_j}})/f(O_{k_{l_{j-1}}})) && (5)\\ &= r_j \cdot \left(\frac{k_{l_j}}{k_{l_{j-1}}} - \left(\frac{k_{l_j}}{k_{l_{j-1}}} - 1\right) r_{j-1}\right) && \text{(By Eq. (4))}\\ &= r_j \cdot \left(1 + \frac{k_{l_j} - k_{l_{j-1}}}{k_{l_{j-1}}} - \left(\frac{k_{l_j} - k_{l_{j-1}}}{k_{l_{j-1}}}\right) r_{j-1}\right)\\ &= r_j \cdot \left(1 + \left(\frac{k_{l_j} - k_{l_{j-1}}}{k_{l_{j-1}}}\right)(1 - r_{j-1})\right). && (6) \end{aligned}$$

Next, for any feasible r_j's (and in particular the r_j's that correspond to the arbitrary instance $f^{(\text{bad-for-greedy})}$ in the theorem statement), we construct a hard instance f that is characterized by the same r_j's (i.e., it satisfies Eq. (3) for the given r_j's), such that for the hard instance f and for every budget k_i, up to an arbitrarily small multiplicative error, (i) the aforementioned approximation ratio determined by the r_j's is also an upper bound of the approximation ratio of greedy (which implies the second item in the theorem statement, because the approximation ratio determined by the particular r_j's corresponding to $f^{(\text{bad-for-greedy})}$ is a lower bound of the approximation ratio of greedy on $f^{(\text{bad-for-greedy})}$), and (ii) greedy performs at least (almost) as good as the efficient $\mathcal{A}$ (which implies the first item in the theorem statement).

Construction of hard instance

Let $\Delta_1 = k_{l_1}$, $\Delta_j = k_{l_j} - k_{l_{j-1}}$, $\forall 1 < j < t$, and $\Delta_t = k_m - k_{l_{t-1}}$. We apply Theorem 3 to create t hard (with respect to greedy and $\mathcal{A}$) functions $f_{\Delta_1}, \ldots, f_{\Delta_t}$ over disjoint ground sets $V_1, \ldots, V_t$. We normalize these functions such that they have the same optimal value 1 (i.e., $f_{\Delta_j}(O^{(j)}) = 1$, where $O^{(j)}$ denotes the optimal size-Δ_j solution for f_{Δ_j}) and extend them to the ground set $V := \cup_{i=1}^{t} V_i$. The final submodular function is

$$f(X) := \sum_{j=1}^{t} \alpha_j \cdot f_{\Delta_j}(X),$$

where $\alpha_1 := 1$ and

$$\alpha_j := (\Delta_j / \sum_{s=1}^{j-1} \Delta_s) \cdot (\sum_{s=1}^{j-1} \alpha_s) \cdot (1 - r_{j-1}), \quad \text{for all } 1 < j \leq t.$$

▷ Claim 7. For any r_j that satisfy Eq. (5), $r_j \cdot \sum_{s=1}^{j} \alpha_s$ is non-decreasing in j.

Proof of Claim 7.

$$\begin{aligned}
r_j \cdot \sum_{s=1}^{j} \alpha_s &= r_j \cdot \left(\sum_{s=1}^{j-1} \alpha_s + \alpha_j \right) \\
&= r_j \cdot \left(\sum_{s=1}^{j-1} \alpha_s + \frac{\Delta_j}{\sum_{s=1}^{j-1} \Delta_s} \cdot \left(\sum_{s=1}^{j-1} \alpha_s \right) \cdot (1 - r_{j-1}) \right) && \text{(Definition of } \alpha_s\text{)} \\
&= r_j \cdot \left(\sum_{s=1}^{j-1} \alpha_s + \frac{k_{l_j} - k_{l_{j-1}}}{k_{l_{j-1}}} \cdot \left(\sum_{s=1}^{j-1} \alpha_s \right) \cdot (1 - r_{j-1}) \right) && \text{(Telescoping sum)} \\
&= \left(1 + \frac{k_{l_j} - k_{l_{j-1}}}{k_{l_{j-1}}} \cdot (1 - r_{j-1}) \right) r_j \cdot \sum_{s=1}^{j-1} \alpha_s \\
&\geq r_{j-1} \cdot \sum_{s=1}^{j-1} \alpha_s, && \text{(Ineq. (5))}
\end{aligned}$$

◁

Because any feasible r_j's that we need to consider satisfy $r_1 \leq 1$ and Eq. (5), it follows by Claim 7 and $\sum_{s=1}^{j-1} \alpha_s - (\sum_{s=1}^{j-1} \Delta_s / \Delta_j)\alpha_j = r_{j-1} \cdot \sum_{s=1}^{j-1} \alpha_s$ that α_j / Δ_j is decreasing as j increases. Hence, $f(O_{k_{l_j}}) = \sum_{i=1}^{j} \alpha_i$ for all $j \in [t]$. Moreover, it is easy to verify that the r_j's indeed characterize the f constructed above in the sense that Eq. (3) holds for the f constructed above.

Upper bounding greedy performance on the hard instance: a single step

First, we analyze the best possible improvement of a single step of greedy on this instance. Suppose that greedy has chosen some size-i set $X_i^{(j)} \subset V_j$, if it chooses another element from V_j, then we claim that the marginal gain is almost always $(f_{\Delta_j}(O^{(j)}) - f_{\Delta_j}(X_i^{(j)}))/\Delta_j$ (it is at least this amount by greedy guarantee). Assume otherwise, for some $\gamma, \epsilon_1, \epsilon_2 > 0$, in the first $\gamma \cdot \Delta_j$ iterations when greedy chooses elements from V_j, there are more than $\epsilon_1 \cdot \Delta_j$ iterations i in which the marginal gain is larger than $((1+\epsilon_2)/\Delta_j) \cdot (f_{\Delta_j}(O^{(j)}) - f_{\Delta_j}(X_i^{(j)}))$. Suppose that at $(\gamma \cdot \Delta_j)$-th iteration $f_{\Delta_j}(X_{\gamma \cdot \Delta_j}^{(j)}) = c \cdot f_{\Delta_j}(O^{(j)})$, then each of those $\epsilon_1 \cdot \Delta_j$ iterations gets at least an extra $(\epsilon_2/\Delta_j) \cdot (1-c) f_{\Delta_j}(O^{(j)})$ in addition to basic greedy guarantee, which implies that $f_{\Delta_j}(X_{\Delta_j}^{(j)}) \geq (1 - e^{-\gamma} + \epsilon_1 \cdot \epsilon_2 \cdot (1-c)) f_{\Delta_j}(O^{(j)})$. Then, for some $\epsilon_3 > 0$, $f_{\Delta_j}(X_{\gamma \cdot \Delta_j}^{(j)}) \geq (1 - e^{-\gamma} + \epsilon_3) f_{\Delta_j}(O^{(j)})$, which is impossible by Theorem 3. Henceforth, we can assume that the marginal gain for f_{Δ_j} is always $(f_{\Delta_j}(O^{(j)}) - f_{\Delta_j}(X_i^{(j)}))/\Delta_j$ for the i-th iteration when greedy chooses elements from V_j, and this will only decrease all the values of interest by an arbitrarily small multiplicative error.

Upper bounding greedy performance on the hard instance: total value

When we start running the greedy algorithm, for a while it only select elements from V_1 since those have the highest marginal contribution. Specifically, suppose that at the beginning of the q-th step, greedy has selected $X_{q-1} \subset V_1$. Then the best achievable marginal gain of an element from V_1 for f is $\alpha_1(f_{\Delta_1}(O^{(1)}) - f_{\Delta_1}(X_{q-1}))/\Delta_1$. In comparison, the best singleton value of an element in V_2 is $(\alpha_2/\Delta_2) f_{\Delta_2}(O^{(2)})$, which is dominated by

$\alpha_1(f_{\Delta_1}(O^{(1)}) - f_{\Delta_1}(X_{q-1}))/\Delta_1$, when $f_{\Delta_1}(X_{q-1}) \le r_1 \cdot f_{\Delta_1}(O^{(1)})$, because

$$\begin{aligned}
&\frac{\alpha_1}{\Delta_1}(f_{\Delta_1}(O^{(1)}) - f_{\Delta_1}(X_{q-1})) \\
\ge&\frac{\alpha_1}{\Delta_1}(f_{\Delta_1}(O^{(1)}) - r_1 \cdot f_{\Delta_1}(O^{(1)})) && (\text{By } f_{\Delta_1}(X_{q-1}) \le r_1 \cdot f_{\Delta_1}(O^{(1)})) \\
=&\frac{\alpha_1}{\Delta_1}(1 - r_1) && (\text{By } f_{\Delta_1}(O^{(1)}) = 1) \\
=&\frac{\alpha_2}{\Delta_2} && (\text{By definition of } \alpha_2) \\
=&\frac{\alpha_2}{\Delta_2} f_{\Delta_2}(O^{(2)}) && (\text{By } f_{\Delta_2}(O^{(2)}) = 1). \qquad (7)
\end{aligned}$$

Thus, when $f(X_{q-1}) = \alpha_1 \cdot f_{\Delta_1}(X_{q-1}) < r_1 \cdot f(O^{(1)})$, greedy should always prefer choosing elements from V_1 over V_2 (and other V_i's), and the single step improvement is $f(X_q) - f(X_{q-1}) = (f(O^{(1)}) - f(X_{q-1}))/\Delta_1 = (f(O^{(1)}) - f(X_{q-1}))/k_1$ (this matches how $f^{(\text{greedy-lb})}(q)$ changes).

We now analyze what happens when, after running greedy for a while, the marginal contribution from V_1-elements decays so that greedy may prefer V_2-elements. By Eq. (7), it is when $f_{\Delta_1}(X_{q-1}) = r_1 \cdot f_{\Delta_1}(O^{(1)})$ that the best singleton value of V_2-elements $(\alpha_2/\Delta_2)f_{\Delta_2}(O^{(2)})$ becomes equal to the best marginal contribution of a V_1-element $\alpha_1(f_{\Delta_1}(O^{(1)}) - f_{\Delta_1}(X_{q-1}))/\Delta_1$. Therefore, once $r_2 \cdot f(O^{(1)} \cup O^{(2)}) > f(X_{q-1}) \ge r_1 \cdot f(O^{(1)})$, greedy should start choosing elements from V_1 and V_2 alternatively to keep the identity $\alpha_1(f_{\Delta_1}(O^{(1)}) - f_{\Delta_1}(X_{q-1}))/\Delta_1 = \alpha_2(f_{\Delta_2}(O^{(2)}) - f_{\Delta_2}(X_{q-1}))/\Delta_2$ (up to negligible error), it follows that

$$\begin{aligned}
&\frac{\alpha_1(f_{\Delta_1}(O^{(1)}) - f_{\Delta_1}(X_{q-1}))}{\Delta_1} \\
=&\frac{\alpha_2(f_{\Delta_2}(O^{(2)}) - f_{\Delta_2}(X_{q-1}))}{\Delta_2} \\
=&\frac{\alpha_1(f_{\Delta_1}(O^{(1)}) - f_{\Delta_1}(X_{q-1})) + \alpha_2(f_{\Delta_2}(O^{(2)}) - f_{\Delta_2}(X_{q-1}))}{\Delta_1 + \Delta_2} \\
=&\frac{f(O^{(1)} \cup O^{(2)}) - f(X_{q-1})}{k_{l_2}}.
\end{aligned}$$

Thus, there is a transition of the best marginal gain when $f(X_{q-1}) = r_1 \cdot f(O^{(1)})$ with $X_{q-1} \subseteq V_1$, and after that the best achievable marginal gain is characterized by $(f(O^{(1)} \cup O^{(2)}) - f(X_{q-1}))/k_{l_2}$ (this matches the guarantee transition for $f^{(\text{greedy-lb})}(q)$), which is larger than $(\alpha_3/\Delta_3)f_{\Delta_3}(O^{(3)})$ by definition of α_3.

Similarly, for every $3 \le p \le t$, when $f(X_{q-1}) = r_{p-1} \cdot f(\cup_{j=1}^{p-1} O^{(j)})$ with $X_{q-1} \subseteq \cup_{j \le p-1} V_j$, it holds that $(f(\cup_{j=1}^{p-1} O^{(j)}) - f(X_{q-1}))/k_{l_{p-1}} = (\alpha_p/\Delta_p)f_{\Delta_p}(O^{(p)})$ by definition of α_p, and hence, greedy starts to choose elements from $V_1, \ldots, V_p$ to keep $\alpha_j(f_{\Delta_j}(O^{(j)}) - f_{\Delta_j}(X_{q-1}))/\Delta_j$ for all $j \le p$ approximately equal to each other. Hence, for all $j \le p$,

$$\begin{aligned}
\frac{\alpha_j(f_{\Delta_j}(O^{(j)}) - f_{\Delta_j}(X_{q-1}))}{\Delta_j} &= \frac{\sum_{j \le p} \alpha_j(f_{\Delta_j}(O^{(j)}) - f_{\Delta_j}(X_{q-1}))}{\sum_{j \le p} \Delta_j} \\
&= \frac{f(\cup_{j=1}^{p} O^{(j)}) - f(X_{q-1})}{k_{l_p}},
\end{aligned}$$

and this is a transition of the best marginal gain from $(f(\cup_{j=1}^{p-1} O^{(j)}) - f(X_{q-1}))/k_{l_{p-1}}$ to $(f(\cup_{j=1}^{p} O^{(j)}) - f(X_{q-1}))/k_{l_p}$ (this matches the guarantee transition for $f^{(\text{greedy-lb})}(q)$). Therefore, we have shown that the greedy performance $f(X_q)$ changes in exactly the same way as $f^{(\text{greedy-lb})}(q)$, and hence, the approximation ratio based on $f^{(\text{greedy-lb})}(q)$'s is tight for greedy on the hard instance.

How greedy spends the budget

Finally, following the above derivation, we emphasize how greedy spends the budget. As we have shown, for any $1 \le p \le t$, when $r_{p-1} \cdot f(\cup_{j=1}^{p} O^{(j)}) \le f(X_{q-1}) \le r_p \cdot f(\cup_{j=1}^{p} O^{(j)})$, greedy splits its budget on $V_1, \ldots, V_p$ to keep all the $\alpha_j(f_{\Delta_j}(O^{(j)}) - f_{\Delta_j}(X_{q-1}))/\Delta_j$ approximately equal to each other. Moreover, for any $j' > p$, the best singleton value $(\alpha_{j'}/\Delta_{j'})f_{\Delta_{j'}}(O^{(j')})$ of $V_{j'}$ is smaller than $\alpha_j(f_{\Delta_j}(O^{(j)}) - f_{\Delta_j}(X_{q-1}))/\Delta_j$ for any $j \le p$. Suppose that greedy has spent budget $\hat{b}_j$ on V_j for each j, which implies that $f_{\Delta_j}(X_{q-1}) = 1 - e^{-\hat{b}_j/\Delta_j}$. Then, we have that $\alpha_j(f_{\Delta_j}(O^{(j)}) - f_{\Delta_j}(X_{q-1}))/\Delta_j = \frac{d\alpha_j(1-e^{-x/\Delta_j})}{dx}|_{x=\hat{b}_j}$, and thus, $\frac{d\alpha_j(1-e^{-x/\Delta_j})}{dx}|_{x=\hat{b}_j}$ for all $j \le p$ are equal to each other. Moreover, since $(\alpha_{j'}/\Delta_{j'})f_{\Delta_{j'}}(O^{(j')}) = \frac{d\alpha_{j'}(1-e^{-x/\Delta_{j'}})}{dx}|_{x=0}$, $\frac{d\alpha_j(1-e^{-x/\Delta_j})}{dx}|_{x=\hat{b}_j} \ge \frac{d\alpha_{j'}(1-e^{-x/\Delta_{j'}})}{dx}|_{x=0}$ for all $j \le p$ and $j' > p$.

Greedy spends the budget optimally on the hard instance

By Theorem 3 and the design of our hard instances, for any budget b_j, the best possible value the efficient algorithm $\mathcal{A}$ can get by spending budget b_j on f_{Δ_j} is $u_j(b) = (1 - e^{-b_j/\Delta_j})\alpha_j$. Suppose $\mathcal{A}$ spends budget b_j on each f_{Δ_j}, where $\sum_{j=1}^{t} b_j = b$ for some b, then in this case, the best possible value in total is $\sum_{j=1}^{t} u_j(b_j)$, and hence, in general, the best possible value for budget b is upper bounded by the maximum of the following program:

$$\max \sum_{j=1}^{t} u_j(b_j) \quad \text{s.t.} \ \sum_{j=1}^{t} b_j = b \ \text{ and } \forall j \ b_j \ge 0.$$

We observe that for an arbitrary fixed b, the maximizer b_j^*'s for this program should satisfy that for all positive b_j^*, the derivatives of u_j's at b_j^*'s are equal (notice that the way greedy spends the budget also satisfies this property), and moreover, they are not smaller than the derivative of $u_{j'}$'s at 0 for any j' such that $b_{j'}^* = 0$. Otherwise, there must exist $\frac{du_{j_1}(x)}{dx}|_{x=b_{j_1}^*} < \frac{du_{j_2}(x)}{dx}|_{x=b_{j_2}^*}$ where $b_{j_1}^*$ is strictly positive, then increasing $b_{j_2}^*$ by δ and decreasing $b_{j_1}^*$ by δ for sufficiently small δ will increase the objective value while preserving the feasibility of b_j^*'s.

Now we prove that for any fixed b, the b_j^*'s satisfying the above mentioned property are unique. (Then, it follows that the maximizer matches exactly how greedy spends the budget, and moreover, greedy attains the optimal value of the program.) Suppose that besides b_j^*'s, $\widetilde{b}_j$'s also satisfy the property. Let $\mathrm{supp}(b^*)$ be the set of j such that $b_j^* > 0$. We first argue if $j' \notin \mathrm{supp}(b^*)$, then $\widetilde{b}_{j'} = 0$. Suppose otherwise, $\widetilde{b}_{j'} > 0$, then $\sum_{j \in \mathrm{supp}(b^*)} \widetilde{b}_j < t$, and hence, there must exist a $j \in \mathrm{supp}(b^*)$ such that $\widetilde{b}_j < b_j^*$. By strict concavity of u_j, $\frac{du_j(x)}{dx}|_{x=\widetilde{b}_{j'}} > \frac{du_j(x)}{dx}|_{x=b_j^*}$. However, since the b_j^*'s satisfy the above mentioned property and $b_{j'}^* = 0$, $\frac{du_j(x)}{dx}|_{x=b_j^*} \ge \frac{du_{j'}(x)}{dx}|_{x=0}$, and by strict concavity of $u_{j'}$, $\frac{du_{j'}(x)}{dx}|_{x=0} > \frac{du_{j'}(x)}{dx}|_{x=\widetilde{b}_{j'}}$, which gives a contradiction. Furthermore, we can argue that for all $j \in \mathrm{supp}(b^*)$, $\widetilde{b}_{j'} = b_j^*$, because otherwise, there must exist $j, j' \in \mathrm{supp}(b^*)$ such that $b_j^* > \widetilde{b}_j$ and $b_{j'}^* < \widetilde{b}_{j'}$, and hence $\frac{du_j(x)}{dx}|_{x=b_j^*} < \frac{du_j(x)}{dx}|_{x=\widetilde{b}_j}$ and $\frac{du_{j'}(x)}{dx}|_{x=b_{j'}^*} > \frac{du_{j'}(x)}{dx}|_{x=\widetilde{b}_{j'}}$, which contradicts the property $\frac{du_j(x)}{dx}|_{x=\widetilde{b}_j} = \frac{du_{j'}(x)}{dx}|_{x=\widetilde{b}_{j'}}$. ◀

3.2 Two Remarks for Theorem 6

One might wonder whether the transitions of the greedy guarantees in the above analysis of Theorem 6 always occur in the order $1, 2, \ldots, m$ but never in any proper subsequence, namely whether $r_l \cdot f(O_{k_l}) \leq r_{l-1} \cdot f(O_{k_{l-1}})$, which is equivalent to

$$\frac{f(O_{k_l}) - (k_l/k_{l+1}) f(O_{k_{l+1}})}{1 - (k_l/k_{l+1})} \geq \frac{f(O_{k_{l-1}}) - (k_{l-1}/k_l) f(O_{k_l})}{1 - (k_{l-1}/k_l)},$$

This is equivalent to

$$f(O_{k_{l+1}}) - f(O_{k_l}) \leq \frac{k_{l+1} - k_l}{k_l - k_{l-1}} \cdot (f(O_{k_l}) - f(O_{k_{l-1}})),$$

which is actually true for our instances but not in general. See Example 8.

▶ **Example 8.** Consider the function f on the ground set $\{1, 2, 3, 4\}$ with values $f(\emptyset) = 0$, $f(\{1\}) = 1$, $f(\{2\}) = f(\{3\}) = f(\{4\}) = 1/2$, $f(\{1, 2\}) = f(\{1, 3\}) = f(\{1, 4\}) = 7/6$, $f(\{2, 3\}) = f(\{2, 4\}) = f(\{3, 4\}) = 1$, $f(\{2, 3, 4\}) = 3/2$, $f(\{1, 2, 3\}) = f(\{1, 2, 4\}) = f(\{1, 3, 4\}) = 4/3$ and $f(\{1, 2, 3, 4\}) = 3/2$. It is straightforward to check that f is submodular and monotone, and that $f(O_3) - f(O_2) > f(O_2) - f(O_1)$.

Finally, we end this section with the following observation. In the appendix of the full version, we give the proof of this observation and show that many practical algorithms satisfy the condition of this observation.

▶ **Observation 9.** *For any perturbation factors $0 < \rho_1 < \rho_2 < \cdots < \rho_m$, there exists a sufficiently large k that grows with the size of instance such that given m budgets $k_1 = \rho_1 \cdot k$, $\ldots$, $k_m = \rho_m \cdot k$, the optimality described in Theorem 6 actually holds for a general class of algorithms such that:*

- **i** *Given budget k_i, the algorithm $\mathcal{A}$ runs in T rounds (T is sufficiently large), of which each round selects about k_i/T elements.*
- **ii** *For any $\epsilon > 0$, it holds for all $t \in [T]$, for all $j \in [m]$, that $f(X^{\mathcal{A}}_{tk_i/T}) - f(X^{\mathcal{A}}_{(t-1)k_i/T}) \geq ((1-\epsilon)\rho_i/(\rho_j T)) \cdot (f(O_{k_j}) - f(X^{\mathcal{A}}_{tk_i/T}))$, where $X^{\mathcal{A}}_s$ is the s-th element chosen by $\mathcal{A}$.*

4 A Lower Bound for Every Distribution

The main thesis of this paper is that worst case instances of submodular maximization are really tailored to a specific budget constraint. It is natural to hope that as the distribution of budget perturbation becomes arbitrarily spread (aka arbitrarily far from the worst case single budget), the approximation factor approaches 1. In this section, we give a negative answer to this question.

▶ **Theorem 10.** *For any distribution of budget perturbations $\mathcal{D}$, for any efficient algorithm $\mathcal{A}$, for every sufficiently large k that grows with the size of instance, $\mathcal{R}_{\mathcal{A}}(\mathcal{D}(k)) \leq 0.9087$.*

(We did not seriously try to optimize the constant 0.9087. Computing the optimal constant is an interesting open problem for future work.)

Proof. For arbitrarily small $\tau > 0$, let $\rho_{\min}$ and $\rho_{\max}$ be such that the mass of $\mathcal{D}$ on $[\rho_{\min}, \rho_{\max}]$ is at least $1 - \tau$. Let $q = 50$. Let $K_1 = q^{-(i^*-1)} \cdot k$ where i^* is the largest i such that $q^{-(i-1)} \cdot k \leq \rho_{\min} \cdot k$. Let N be the smallest i such that $q^{(i-1)} \cdot K_1 \geq \rho_{\max} \cdot k$. We first construct the hard instances, and by Theorem 6 it suffices to upper bound the approximation ratio achieved by greedy on these instances.

Construction of hard instances

Let $K_i = q^{(i-1)} \cdot K_1$. We use Theorem 3 to create hard (with respect to greedy algorithm) functions f_{K_i} for all $i \in [N]$ over disjoint ground sets. We normalize these functions such that they have the same optimal value 1 (i.e., $f_{K_i}(O^{(i)}) = 1$, where $O^{(i)}$ denotes the optimal size-K_i solution for f_{K_i}) and extend them to the union of all the ground sets. The final submodular function is $f(X) = \sum_{i=1}^{N} (q/e)^{i-1} \cdot f_{K_i}(X)$.

Upper bounding the approximation ratio on the hard instances

Consider a budget K between $\sum_{j=1}^{i} K_j$ and $\sum_{j=1}^{i+1} K_j$, for any $i \leq N-1$. We first show that the contribution of f_{K_j} with $j \leq i-1$ is negligible. Notice that the best singleton value of f_{K_j} is $(q/e)^{j-1}/(q^{j-1} \cdot K_1)$, which is decreasing in j. Hence, we can generously assume that the algorithm spends a budget of size $\sum_{j=1}^{i-1} K_j$ getting all the utilities from f_{K_j} with $j \leq i-1$, which is the best one can hope for. The total value of these f_{K_j}'s is $\sum_{j=1}^{i-1} (q/e)^{j-1} = ((q/e)^{i-1} - 1)/(q/e - 1)$, which is less than $1/(q/e-1) < 0.0575$ fraction of the value of the f_{K_i}. Therefore, the best possible approximation ratio for budget K is at most the best possible approximation ratio for budget $K - \sum_{j=1}^{i-1} K_j$ on the f_{K_j}'s with $j \geq i$ plus 0.0575.

Furthermore, the best singleton value of $f_{K_{i+1}}$ is at most $(q/e)^i/(q^i \cdot K_1)$. On the other hand, with budget K_i on f_{K_i}, greedy can achieve approximation ratio at most $1 - 1/e$ by Theorem 3, and thus, at the (K_i+1)-th iteration, greedy has marginal gain at least $(1-(1-1/e)) \cdot (q/e)^{i-1}/(q^{(i-1)} \cdot K_1)$, which is equal to the best singleton value of $f_{K_{i+1}}$. Hence, greedy will not choose anything from $f_{K_{i+1}}$ until it has selected K_i elements from f_{K_i}. The remaining budget $K - \sum_{j=1}^{i} K_j$ is at most K_{i+1}, and it follows for the same reason that greedy will not spend remaining budget on any f_{K_j}'s with $j \geq i+2$.

It remains to show how greedy performs on f_{K_i} and $f_{K_{i+1}}$ with budget $K' = K - \sum_{j=1}^{i} K_j$. Let $a = K'/K_i$. Notice that greedy splits its budget in the way that the marginal gain of choosing the next element from f_{K_i} is approximately equal to that of choosing the next element from $f_{K_{i+1}}$. This can be expressed as the following equations:

$$a_1 K_i + a_2 K_i = a K_i,$$
$$e^{-a_1} \cdot \frac{e/q}{K_i} = e^{-\frac{a_2 K_i}{K_{i+1}}} \cdot \frac{1}{K_{i+1}},$$

where $a_1 K_i$ and $a_2 K_i$ are the budgets spent on f_{K_i} and $f_{K_{i+1}}$ respectively. The solution is $a_1 = (a+q)/(q+1)$ and $a2 = (q(a-1))/(q+1)$. Hence the approximation ratio of greedy on f_{K_i} and $f_{K_{i+1}}$ with budget K' is at most

$$\frac{(1 - e^{-a_1 K_i/K_i}) \cdot \frac{e}{q} + (1 - e^{-a_2 K_i/K_{i+1}})}{\frac{e}{q} + \frac{aK_i - K_i}{K_{i+1}}} = \frac{(1 - e^{-(a+q)/(q+1)}) \cdot \frac{e}{q} + (1 - e^{-(a-1)/(q+1)})}{\frac{e}{q} + \frac{a-1}{q}}.$$

The maximum is approximately 0.85114 achieved by $a \approx 9.2199$. Hence, in total, the approximation ratio for the entire instance is less than $0.8512 + 0.0575 = 0.9087$. We have proved that the optimal approximation ratio for the hard instance is less than 0.9087 for any budget in $[\rho_{\min} \cdot k, \rho_{\max} \cdot k]$, and this finishes the proof because τ is arbitrarily small. ◀

5 Numerical Simulation

We formulate a mathematical program that computes the worst possible optimal expected approximation ratio, for any fixed distribution on any fixed choice of m budgets $\rho_1 k < \rho_2 k < \cdots < \rho_m k = k$ (we also let $\rho_0 = 0$). We denote the probability of budget $\rho_i k$ by p_i for each i.

Reducing the hard instances to a standard form

Recall that the hard instances in the proof of Theorem 6 have following form – $f^*(X) = \sum_{j=1}^{t} \alpha_j \cdot f_{(\rho_{l_j}-\rho_{l_{j-1}})k}$, where $l_1, \ldots, l_t$ is a subsequence of $1, \ldots, m$, and $f_{(\rho_{l_j}-\rho_{l_{j-1}})k}(X)$ is the hard submodular function from Theorem 3, and it is normalized such that its optimal value for budget $(\rho_{l_j} - \rho_{l_{j-1}})k$ is 1. Moreover, $\alpha_j/(\rho_{l_j} - \rho_{l_{j-1}})$ is increasing in j. We show that there is a submodular function that is as hard as f^* to approximately maximize in the following **standard form**[6] – $f(X) = \sum_{i=1}^{m} \beta_i \cdot f_{(\rho_i-\rho_{i-1})k}(X)$, where β_i's satisfy

$$\frac{\beta_i}{\rho_i - \rho_{i-1}} \geq \frac{\beta_{i+1}}{\rho_{i+1} - \rho_i}, \quad \forall\, i < m, \tag{8}$$

where $f_{(\rho_i-\rho_{i-1})k}$ is defined analogously to $f_{(\rho_{l_j}-\rho_{l_{j-1}})k}$, and we denote the ground set of $f_{(\rho_i-\rho_{i-1})k}$ by V_i.

For $i \in [m]$ and j such that $l_{j-1} < i \leq l_j$, we define

$$\lambda_i := \frac{\rho_i - \rho_{i-1}}{\rho_{l_j} - \rho_{l_{j-1}}}.$$

▷ Claim 11. Given budget $x \cdot k$ for any $x \geq 0$, the best achievable approximation ratio for $\sum_{i=l_{j-1}+1}^{l_j} \lambda_i \cdot f_{(\rho_i-\rho_{i-1})k}(X)$ is equal to that for $f_{(\rho_{l_j}-\rho_{l_{j-1}})k}(X)$.

Proof. For any budget $x \cdot k$, the best achievable approximation ratio for $\sum_{i=l_{j-1}+1}^{l_j} \lambda_i \cdot f_{(\rho_i-\rho_{i-1})k}(X)$ is

$$\max_{x_i\text{'s}} \sum_{i=l_{j-1}+1}^{l_j} \lambda_i (1 - e^{-\frac{x_i}{\rho_i-\rho_{i-1}}}) \quad \text{s.t.} \quad \sum_{i=l_{j-1}+1}^{l_j} x_i = x \text{ and } x_i\text{'s are non-negative.}$$

For any feasible x_i's,

$$\begin{aligned}
&\sum_{i=l_{j-1}+1}^{l_j} \lambda_i (1 - e^{-\frac{x_i}{\rho_i-\rho_{i-1}}}) \\
\leq &1 - e^{-\sum_{i=l_{j-1}+1}^{l_j} \frac{\lambda_i \cdot x_i}{\rho_i-\rho_{i-1}}} && \text{(Jensen's inequality and } \sum_{i=l_{j-1}+1}^{l_j} \lambda_i = 1) \\
= &1 - e^{-\sum_{i=l_{j-1}+1}^{l_j} \frac{x_i}{\rho_{l_j}-\rho_{l_{j-1}}}} && \text{(By definition of } \lambda_i) \\
= &1 - e^{-\frac{x}{\rho_{l_j}-\rho_{l_{j-1}}}} && \text{(By } \sum_{i=l_{j-1}+1}^{l_j} x_i = x).
\end{aligned}$$

Moreover, when $\frac{x_i}{\rho_i-\rho_{i-1}}$ for all i are equal to each other, we have that

$$\frac{x_i}{\rho_i - \rho_{i-1}} = \frac{\sum_{i=l_{j-1}+1}^{l_j} x_i}{\sum_{i=l_{j-1}+1}^{l_j} \rho_i - \rho_{i-1}} = \frac{x}{\rho_{l_j} - \rho_{l_{j-1}}},$$

and then $\sum_{i=l_{j-1}+1}^{l_j} \lambda_i (1 - e^{-\frac{x_i}{\rho_i-\rho_{i-1}}}) = 1 - e^{-\frac{x}{\rho_{l_j}-\rho_{l_{j-1}}}}$. Hence, $1 - e^{-\frac{x}{\rho_{l_j}-\rho_{l_{j-1}}}}$ is exactly the best achievable approximation ratio for $\sum_{i=l_{j-1}+1}^{l_j} \lambda_i \cdot f_{(\rho_i-\rho_{i-1})k}(X)$. Notice that it is also the best achievable approximation ratio for $f_{(\rho_{l_j}-\rho_{l_{j-1}})k}(X)$. ◁

[6] The difference between f^* and the standard form f is that in the f there is a sub-instance for every budget

Henceforth, we can replace each $f_{(\rho_{l_j}-\rho_{l_{j-1}})k}$ with $\sum_{i=l_{j-1}+1}^{l_j} \lambda_i \cdot f_{(\rho_i-\rho_{i-1})k}(X)$ in f^*, which reduces f^* to the standard form. Then, Eq. (8) follows by definition of λ_i's and the monotonicity of $\alpha_j/(\rho_{l_j} - \rho_{l_{j-1}})$. Finally, we note that Eq. (8) implies that optimal value of the f^* for budget $\rho_i k$ is $\sum_{j=1}^{i} \beta_j$ and that for any i, whenever V_i is used by the greedy algorithm, so should the $V_{i'}$'s for any $i' \le i$.

Formulating the mathematical program

For each budget $\rho_i \cdot k$, the best possible approximation ratio is achieved by choosing elements from the first l subsets $V_1, \ldots, V_l$ for certain $l \le m$ (which we do not know a priori), and the budget should be split in a way such that the marginal contribution from the next element is (approximately) equal among $V_1, \ldots, V_l$. That is,

$$\frac{d(\beta_1(1-e^{-x_1^{(i,l)}/\rho_1}))}{dx_1^{(i,l)}} = \frac{d(\beta_j(1-e^{-x_j^{(i,l)}/(\rho_j-\rho_{j-1})}))}{dx_j^{(i,l)}},\ \forall\, j \le l,$$
$$\sum_{j\le l} x_j^{(i,l)} = \rho_i,\ \text{and } x_j^{(i,l)} \ge 0,\ \forall j \le l.$$

Solving the system of equations in the above constraint gives us

$$\frac{x_1^{(i,l)}}{\rho_1} = \frac{\rho_i}{\rho_l} - \sum_{j=1}^{l} \ln\left(\frac{\beta_j\rho_1}{\beta_1(\rho_j-\rho_{j-1})}\right)\cdot\frac{\rho_j-\rho_{j-1}}{\rho_l},$$
$$\frac{x_j^{(i,l)}}{\rho_j-\rho_{j-1}} = \frac{x_1^{(i,l)}}{\rho_1} + \ln\left(\frac{\beta_j\rho_1}{\beta_1(\rho_j-\rho_{j-1})}\right),\ \forall\, j \le l. \tag{9}$$

We let $h^{(i,l)}(\beta_1, \ldots, \beta_m)$ denote the approximation ratio achieved by $x_j^{(i,l)}$'s, then it is given by

$$\begin{aligned}
&h^{(i,l)}(\beta_1,\ldots,\beta_m)\\
=&\frac{\sum_{j=1}^{l}\beta_j(1-e^{-\frac{x_j^{(i,l)}}{\rho_j-\rho_{j-1}}})}{\sum_{j=1}^{i}\beta_j}\\
=&\frac{\sum_{j=1}^{l}\beta_j(1-e^{-\frac{x_1^{(i,l)}}{\rho_1}\cdot\frac{\beta_1(\rho_j-\rho_{j-1})}{\beta_j\rho_1}})}{\sum_{j=1}^{i}\beta_j} && \text{(Solution of } x_j^{(i,l)}\text{)}\\
=&\frac{\sum_{j=1}^{l}\beta_j-\sum_{j=1}^{l}\frac{\beta_1(\rho_j-\rho_{j-1})}{\rho_1}\cdot e^{-\frac{x_1^{(i,l)}}{\rho_1}}}{\sum_{j=1}^{i}\beta_j}\\
=&\frac{\left(\sum_{j=1}^{l}\beta_j\right)-\beta_1\cdot\frac{\rho_l}{\rho_1}\cdot e^{-\frac{x_1^{(i,l)}}{\rho_1}}}{\sum_{j=1}^{i}\beta_j} && \text{(Telescoping sum)}\\
=&\frac{\left(\sum_{j=1}^{l}\beta_j\right)-\beta_1\cdot\frac{\rho_l}{\rho_1}\cdot e^{-\frac{\rho_i}{\rho_l}+\sum_{j=1}^{l}\ln\left(\frac{\beta_j\rho_1}{\beta_1(\rho_j-\rho_{j-1})}\right)\cdot\frac{\rho_j-\rho_{j-1}}{\rho_l}}}{\sum_{j=1}^{i}\beta_j} && \text{(Solution of } x_1^{(i,l)}\text{)},
\end{aligned}$$

where the nominator is the value achieved by $x_j^{(i,l)}$'s, and the denominator is the optimal value. Since we do not know the right choice of l a priori, we will enumerate all possible

Table 2 Campaign Budgets (in millions).

Candidate	Bennet	Biden	Bloomberg	Buttigieg	Gabbard
Budget	2.6	23.3	188.4	34.1	2.9
Klobuchar	Patrick	Sanders	Steyer	Warren	Yang
10.1	0.9	50.1	153.7	33.7	19.2

choices of l and pick the best. Moreover, for every $l \le m$, we consider l as a candidate choice only if the solutions of $x_j^{(i,l)}$'s by Eq. (9) are non-negative, because this holds for the right choice of l. (Note that $l = 1$ is always a candidate choice, because it means that all budget are spent on the first sub-instance, and hence $x_1^{(i,1)} = \rho_i \ge 0$.) Therefore, we let $h^{(i)}$ denote the best approximation ratio for budget $\rho_i k$, then it is given by

$$\begin{aligned} h^{(i)}(\beta_1, \ldots, \beta_m) &= \max_{1 \le l \le m} h^{(i,l)}(\beta_1, \ldots, \beta_m) \cdot \mathbb{1}[x_l^{(i,l)} \ge 0] \\ &= \max_{1 \le l \le m} h^{(i,l)}(\beta_1, \ldots, \beta_m) - C \cdot \mathbb{1}[x_l^{(i,l)} < 0] \quad (C \text{ is a large constant}), \end{aligned}$$

where $x_l^{(i,l)}$ can be represented as a function of β_i's. Note that we only restrict $x_l^{(i,l)}$ to be non-negative, which actually implies that every $x_j^{(i,l)}$ for $j \le l$ is non-negative, by Eq.(8) and Eq.(9). Finally, the expected approximation ratio h is given by $h(\beta_1, \ldots, \beta_m) = \sum_{i=1}^m p_i \cdot h^{(i)}(\beta_1, \ldots, \beta_m)$. Given any fixed ρ_i's, the worst possible optimal average approximation ratio is the result of the following program

$$\min_{\beta_1, \ldots, \beta_m \ge 0} h(\beta_1, \ldots, \beta_m) \text{ s.t. Eq.(8) and Eq.(9).}$$

5.1 Empirical results

We solve this program numerically for various distributions of budget perturbations (ρ_i's); the results are summarized in Table 1.

Canonical distributions It is natural to ask what is the expected approximation factor when the budget is drawn from uniform over $[x, 10x]$. Since we don't know how to compute this value exactly, we take discretization of this distribution namely 25 budgets[7] evenly spaced between x and $10x$. Similarly, we experiment with discretizations of log-scale uniform distributions over $[x, 10x]$ and $[x, 600x]$.

Top social/political campaigns on Facebook With the application of influence maximization on social networks in mind, we use the budgets of the top ten campaigns on Facebook's database of social/political campaigns[8].

2020 Democratic Party presidential candidates We use reported total campaign budgets by candidates in the 2020 Democratic Party primary elections during months October-December 2019 [1] (see Table 2).

References

1 Sarah Almukhtar, Thomas Kaplan, and Rachel Shorey. 2020 democrats went on a spending spree in the final months of 2019. `https://www.nytimes.com/interactive/2020/02/01/us/elections/democratic-q4-fundraising.html`, 2020.

[7] We use a somewhat sparse discretization since the program is non-convex.
[8] Top ten amount spent during the month before Mar. 26, 2020 in the Facebook Ad report [21].

2 Georgios Amanatidis, Pieter Kleer, and Guido Schäfer. Budget-feasible mechanism design for non-monotone submodular objectives: Offline and online. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, pages 901–919, 2019. doi:10.1145/3328526.3329622.

3 Nima Anari, Gagan Goel, and Afshin Nikzad. Budget feasible procurement auctions. *Operations Research*, 66(3):637–652, 2018.

4 Ashwinkumar Badanidiyuru, Shahar Dobzinski, Hu Fu, Robert Kleinberg, Noam Nisan, and Tim Roughgarden. Sketching valuation functions. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1025–1035, 2012. doi:10.1137/1.9781611973099.81.

5 Ashwinkumar Badanidiyuru, Robert Kleinberg, and Yaron Singer. Learning on a budget: posted price mechanisms for online procurement. In *Proceedings of the 13th ACM Conference on Electronic Commerce, EC 2012, Valencia, Spain, June 4-8, 2012*, pages 128–145, 2012. doi:10.1145/2229012.2229026.

6 Maria-Florina Balcan, Florin Constantin, Satoru Iwata, and Lei Wang. Learning valuation functions. In *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland*, pages 4.1–4.24, 2012. URL: http://proceedings.mlr.press/v23/balcan12b/balcan12b.pdf.

7 Eric Balkanski and Jason D. Hartline. Bayesian budget feasibility with posted pricing. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 189–203, 2016. doi:10.1145/2872427.2883032.

8 Eric Balkanski, Sharon Qian, and Yaron Singer. Instance specific approximations for submodular maximization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 609–618. PMLR, 2021.

9 Eric Balkanski, Aviad Rubinstein, and Yaron Singer. The power of optimization from samples. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4017–4025, 2016. URL: http://papers.nips.cc/paper/6447-the-power-of-optimization-from-samples.

10 Hau Chan and Jing Chen. Truthful multi-unit procurements with budgets. In *Web and Internet Economics - 10th International Conference, WINE 2014, Beijing, China, December 14-17, 2014. Proceedings*, pages 89–105, 2014. doi:10.1007/978-3-319-13129-0_7.

11 Hau Chan and Jing Chen. Budget feasible mechanisms for dealers. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, May 9-13, 2016*, pages 113–122, 2016. URL: http://dl.acm.org/citation.cfm?id=2936945.

12 Vaggos Chatziafratis, Tim Roughgarden, and Jan Vondrák. Stability and recovery for independence systems. In *25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017, Vienna, Austria*, pages 26:1–26:15, 2017. doi:10.4230/LIPIcs.ESA.2017.26.

13 Ning Chen, Nick Gravin, and Pinyan Lu. On the approximability of budget feasible mechanisms. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 685–699. Society for Industrial and Applied Mathematics, 2011.

14 Michele Conforti and Gérard Cornuéjols. Submodular set functions, matroids and the greedy algorithm: Tight worst-case bounds and some generalizations of the rado-edmonds theorem. *Discret. Appl. Math.*, 7(3):251–274, 1984. doi:10.1016/0166-218X(84)90003-9.

15 Daniel Dadush and Sophie Huiberts. A friendly smoothed analysis of the simplex method. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 390–403. ACM, 2018. doi:10.1145/3188745.3188826.

16 Abhimanyu Das and David Kempe. Submodular meets spectral: greedy algorithms for subset selection, sparse approximation and dictionary selection. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 1057–1064, 2011.

17 Shahar Dobzinski, Christos H. Papadimitriou, and Yaron Singer. Mechanisms for complement-free procurement. In *Proceedings 12th ACM Conference on Electronic Commerce (EC-2011), San Jose, CA, USA, June 5-9, 2011*, pages 273–282, 2011. doi:10.1145/1993574.1993615.

18 Ethan R Elenberg, Rajiv Khanna, Alexandros G Dimakis, and Sahand Negahban. Restricted strong convexity implies weak submodularity. *The Annals of Statistics*, 46(6B):3539–3568, 2018.

19 Alina Ene and Huy L. Nguyen. A nearly-linear time algorithm for submodular maximization with a knapsack constraint. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 53:1–53:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ICALP.2019.53.

20 Ludwig Ensthaler and Thomas Giebe. A dynamic auction for multi-object procurement under a hard budget constraint. *Research Policy*, 43(1):179–189, 2014.

21 Facebook. Facebook ad library report. https://www.facebook.com/ads/library/report/, 2020. Accessed: 2020-03-26.

22 Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

23 Vitaly Feldman and Pravesh Kothari. Learning coverage functions and private release of marginals. In *Proceedings of The 27th Conference on Learning Theory, COLT 2014, Barcelona, Spain, June 13-15, 2014*, pages 679–702, 2014. URL: http://proceedings.mlr.press/v35/feldman14a.html.

24 Gagan Goel, Afshin Nikzad, and Adish Singla. Mechanism design for crowdsourcing markets with heterogeneous tasks. In *Proceedings of the Seconf AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2014, November 2-4, 2014, Pittsburgh, Pennsylvania, USA*, 2014. URL: http://www.aaai.org/ocs/index.php/HCOMP/HCOMP14/paper/view/8968.

25 Nick Gravin, Yaonan Jin, Pinyan Lu, and Chenhao Zhang. Optimal budget-feasible mechanisms for additive valuations. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 887–900. ACM, 2019.

26 Rishi Gupta, Tim Roughgarden, and C. Seshadhri. Decompositions of triangle-dense graphs. *SIAM J. Comput.*, 45(2):197–215, 2016. doi:10.1137/140955331.

27 Avinatan Hassidim and Yaron Singer. Submodular optimization under noise. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017*, pages 1069–1122, 2017. URL: http://proceedings.mlr.press/v65/hassidim17a.html.

28 Thibaut Horel, Stratis Ioannidis, and S. Muthukrishnan. Budget feasible mechanisms for experimental design. In *LATIN 2014: Theoretical Informatics - 11th Latin American Symposium, Montevideo, Uruguay, March 31 - April 4, 2014. Proceedings*, pages 719–730, 2014. doi:10.1007/978-3-642-54423-1_62.

29 Thibaut Horel and Yaron Singer. Maximization of approximately submodular functions. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3045–3053, 2016. URL: http://papers.nips.cc/paper/6236-maximization-of-approximately-submodular-functions.

30 David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11:105–147, 2015. doi:10.4086/toc.2015.v011a004.

31 Pooya Jalaly Khalilabadi and Éva Tardos. Simple and efficient budget feasible mechanisms for monotone submodular valuations. In *Web and Internet Economics - 14th International Conference, WINE 2018, Oxford, UK, December 15-17, 2018, Proceedings*, pages 246–263, 2018. doi:10.1007/978-3-030-04612-5_17.

32 Sanjeev Khanna and Brendan Lucier. Influence maximization in undirected networks. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms,*

SODA 2014, Portland, Oregon, USA, January 5-7, 2014, pages 1482–1496, 2014. doi:10.1137/1.9781611973402.109.

33 Andreas Krause and Daniel Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*. Cambridge University Press, February 2014.

34 Stefano Leonardi, Gianpiero Monaco, Piotr Sankowski, and Qiang Zhang. Budget feasible mechanisms on matroids. In *Integer Programming and Combinatorial Optimization - 19th International Conference, IPCO 2017, Waterloo, ON, Canada, June 26-28, 2017, Proceedings*, pages 368–379, 2017. doi:10.1007/978-3-319-59250-3_30.

35 Juan Li, Yanmin Zhu, and Jiadi Yu. Redundancy-aware and budget-feasible incentive mechanism in crowd sensing. *Comput. J.*, 63(1):66–79, 2020. doi:10.1093/comjnl/bxy139.

36 George L Nemhauser and Laurence A Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of operations research*, 3(3):177–188, 1978.

37 George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.

38 Besmira Nushi, Adish Singla, Andreas Krause, and Donald Kossmann. Learning and feature selection under budget constraints in crowdsourcing. In *Proceedings of the Fourth AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2016, 30 October - 3 November, 2016, Austin, Texas, USA*, pages 159–168, 2016. URL: http://aaai.org/ocs/index.php/HCOMP/HCOMP16/paper/view/14032.

39 Zeev Nutov and Elad Shoham. Practical budgeted submodular maximization. *CoRR*, abs/2007.04937, 2020. arXiv:2007.04937.

40 Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi:10.1007/s11263-015-0816-y.

41 Grant Schoenebeck and Biaoshuai Tao. Influence maximization on undirected graphs: Towards closing the (1-1/e) gap. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, pages 423–453, 2019. doi:10.1145/3328526.3329650.

42 Grant Schoenebeck, Biaoshuai Tao, and Fang-Yi Yu. Limitations of greed: Influence maximization in undirected networks re-visited. In *AAMAS 2020*, 2020. To appear.

43 Yaron Singer. Budget feasible mechanisms. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 765–774. IEEE, 2010.

44 Yaron Singer and Manas Mittal. Pricing mechanisms for crowdsourcing markets. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 1157–1166, 2013. doi:10.1145/2488388.2488489.

45 Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. ACM*, 51(3):385–463, 2004. doi:10.1145/990308.990310.

46 Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Oper. Res. Lett.*, 32(1):41–43, 2004. doi:10.1016/S0167-6377(03)00062-2.

47 Maxim Sviridenko, Jan Vondrák, and Justin Ward. Optimal approximation for submodular and supermodular optimization with bounded curvature. *Math. Oper. Res.*, 42(4):1197–1218, 2017. doi:10.1287/moor.2016.0842.

48 Alfredo Torrico, Mohit Singh, and Sebastian Pokutta. On the unreasonable effectiveness of the greedy algorithm: Greedy adapts to sharpness. *CoRR*, abs/2002.04063, 2020. arXiv:2002.04063.

49 Jan Vondrák. Symmetry and approximability of submodular maximization problems. *SIAM J. Comput.*, 42(1):265–304, 2013. doi:10.1137/110832318.

50 Duncan Watts and Steven Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 1998.

51 Yuichi Yoshida. Maximizing a monotone submodular function with a bounded curvature under a knapsack constraint. *CoRR*, abs/1607.04527, 2016. arXiv:1607.04527.

52 Dong Zhao, Xiang-Yang Li, and Huadong Ma. Budget-feasible online incentive mechanisms for crowdsourcing tasks truthfully. *IEEE/ACM Trans. Netw.*, 24(2):647–661, 2016. doi:10.1109/TNET.2014.2379281.

53 Zhenzhe Zheng, Fan Wu, Xiaofeng Gao, Hongzi Zhu, Shaojie Tang, and Guihai Chen. A budget feasible incentive mechanism for weighted coverage maximization in mobile crowdsensing. *IEEE Trans. Mob. Comput.*, 16(9):2392–2407, 2017. doi:10.1109/TMC.2016.2632721.