MATHEMATICS OF OPERATIONS RESEARCH
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0364-765X | EISSN 1526-5471 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000

Budget-Feasible Mechanism Design for Non-Monotone Submodular Objectives: Offline and Online*

Georgios Amanatidis
University of Essex, United Kingdom, georgios.amanatidis@essex.ac.uk

Pieter Kleer
Tilburg University, The Netherlands, P.S.Kleer@tilburguniversity.edu

Guido Schäfer
Centrum Wiskunde & Informatica (CWI) and University of Amsterdam, The Netherlands, G.Schaefer@cwi.nl

The framework of budget-feasible mechanism design studies procurement auctions where the auctioneer (buyer) aims to maximize his valuation function subject to a hard budget constraint. We study the problem of designing truthful mechanisms that have good approximation guarantees and never pay the participating agents (sellers) more than the budget. We focus on the case of general (non-monotone) submodular valuation functions and derive the first truthful, budget-feasible and $O(1)$-approximation mechanisms that run in polynomial time in the value query model, for both offline and online auctions. Prior to our work, the only $O(1)$-approximation mechanism known for non-monotone submodular objectives required an exponential number of value queries.

At the heart of our approach lies a novel greedy algorithm for non-monotone submodular maximization under a knapsack constraint. Our algorithm builds two candidate solutions simultaneously (to achieve a good approximation), yet ensures that agents cannot jump from one solution to the other (to implicitly enforce truthfulness). The fact that in our mechanism the agents are not ordered according to their marginal value per cost, allows us to appropriately adapt these ideas to the online setting as well.

To further illustrate the applicability of our approach, we also consider the case where additional feasibility constraints are present, e.g., at most k agents can be selected. We obtain $O(p)$-approximation mechanisms for both monotone and non-monotone submodular objectives, when the feasible solutions are independent sets of a p-system. With the exception of additive valuation functions, no mechanisms were known for this setting prior to our work. Finally, we provide lower bounds suggesting that, when one cares about non-trivial approximation guarantees in polynomial time, our results are asymptotically best possible.

Key words: budget-feasible mechanism design; procurement auctions; non-monotone submodular maximization; submodular knapsack secretary
MSC2000 subject classification: Primary: 91A68; secondary: 68Q25, 90C27
OR/MS subject classification: Primary: Games/group decisions: Bidding/auctions; secondary: Analysis of algorithms: suboptimal algorithms, computational complexity

1. Introduction

We consider the problem of designing *budget-feasible mechanisms* for a natural model of procurement auctions. In this model, an auctioneer is interested in buying services (or

* A preliminary conference version of this work appeared in the *Proceedings of the 2019 ACM Conference on Economics and Computation (EC 2019) [4]*.

goods) from a set of agents A. Each agent $i \in A$ specifies a cost c_i to be paid by the buyer for using his service; crucially, these costs are assumed to be private information. The auctioneer has a budget B and a valuation function $v(\cdot)$, where $v(S)$ specifies the value derived from the services of the agents in $S \subseteq A$. Given the (reported) costs of the agents, the goal of the auctioneer is to choose a *budget-feasible* subset $S \subseteq A$ of the agents, such that the valuation $v(S)$ is maximized. Budget-feasibility here means that $\sum_{i\in S} p_i \leq B$, where p_i is the payment issued from the mechanism to agent i.

Note that the agents might try to extract larger payments from the mechanism by misreporting their actual costs—which of course is undesirable from the auctioneer's perspective. The goal, therefore, is to design budget-feasible mechanisms that *(i)* elicit truthful reporting of the costs by all agents, and *(ii)* achieve a good approximation with respect to the optimal value for the auctioneer. What makes the problem so intriguing is the fact that truthfulness and budget-feasibility are two directly conflicting goals, since the former is achieved by paying as much as needed to make agents indifferent to lying (see Lemma 1). Indicatively, the use of the celebrated truthful VCG mechanism in this setting completely fails with respect to keeping the payments bounded [45].

The problem of designing budget-feasible mechanisms was introduced by Singer [45] and has received a lot of attention, both because of its theoretical appeal and of its relevance to several emerging application domains. A prominent such application is in crowdsourcing marketplaces (such as Mechanical Turk, Figure Eight and Clickworker) which provide online platforms to procure workforce (see [5, 29, 35]). Another application is in the context of influence maximization in social networks, where one seeks to select influential users (see [46, 1]).

We focus on the design of budget-feasible mechanisms for the general class of *non-monotone submodular* valuation functions. Submodular objectives constitute an important class of valuation functions as they satisfy the property of diminishing returns, which naturally arises in many settings. Most existing works make the assumption that the valuation functions are *monotone* (non-decreasing), i.e., $v(S) \leq v(T)$ for $S \subseteq T$. Although the monotonicity assumption makes sense in certain applications, there are several examples where it is violated. For example, in the context of influence maximization in social networks, adding more users to the selected set may sometimes result in negative influence (see [15]). The most prominent example of a non-monotone submodular objective studied in our setting is the *budgeted max-cut problem* [21, 2], where $v(\cdot)$ is determined by the cuts of a given graph.

A natural generalization of this framework is to assume that the space of feasible sets has some structure, e.g., the feasible sets form a matroid. This variant has been studied only for additive valuation functions [1, 39], despite its wide range of applications varying from team formation to spectrum markets (see [39]). Here we study the problem for monotone and non-monotone submodular objectives under p-system constraints.

The purely algorithmic versions of these mechanism design problems ask for the maximization of a (non-monotone) submodular function subject to the constraint that the total cost of the selected agents does not exceed the budget; often referred to as a *knapsack constraint*. These problems are typically NP-hard, hence our focus is on approximation algorithms that compute a close to optimal solution in polynomial time. From an algorithmic point of view, most of these problems are well-understood and admit good approximations. However, it is not clear how to appropriately convert these algorithms into truthful, budget-feasible mechanisms and, up to this work, this goal had been elusive for non-monotone submodular objectives. Our results illustrate that for the mechanism design problems it is possible to achieve the same asymptotic guarantees that are known for their algorithmic counterparts in polynomial time.

It should be stressed that we are interested in *computationally efficient* mechanisms that only use *value queries* (see Section 2). This adds an extra layer of difficulty to the task at hand. Due to the challenges of dealing with incentives in this line of work, often the computational efficiency requirement is dropped completely and it is further assumed that the mechanisms have access to demand queries [20, 13, 2]. Note that, in general, a demand query cannot be simulated by a polynomial number of value queries (see, e.g., [14]).

1.1. Our Contributions Since the introduction of the problem, obtaining computationally efficient mechanisms for objectives that go beyond the class of monotone submodular functions has been open. We derive the first budget-feasible and $O(1)$-approximate mechanisms for non-monotone submodular objectives, both for the offline and the online setting. Our results for the online setting hold for the well-studied *random-arrival model*, where the agents arrive in a uniformly random order, used in the numerous variants of the *secretary problem*. Our mechanisms run in polynomial time in the value query model. The highlights of this work are as follows:

- We obtain the first universally truthful, budget-feasible $O(1)$-approximation mechanism for non-monotone submodular objectives in the value query model.
- We derive the first universally truthful, budget-feasible $O(1)$-approximation *online* mechanism for non-monotone submodular objectives. As a consequence, we obtain an $O(1)$-approximation algorithm for the non-monotone *Submodular Knapsack Secretary Problem*, a budget constrained variant of the famous Secretary Problem.
- We give universally truthful, budget-feasible $O(p)$-approximation mechanisms for both monotone and non-monotone submodular objectives, when the feasible solutions are independent sets of a p-system. Beyond the additive case, nothing was known for this constrained setting.
- We provide lower bounds illustrating that asymptotically our results are as general as one could hope for. On a high level, only trivial guarantees can be achieved with value queries in polynomial time if one imposes constraints beyond downward-closed systems or goes to a broader class of objectives like XOS functions.

1.2. Technical Challenges It should be noted that for monotone submodular objectives all known mechanisms essentially use the same greedy subroutine introduced by Singer [45]: Sort all agents in decreasing order of marginal value per cost and pick as many agents as possible before hitting some carefully selected threshold. This is a simplified version of the optimal greedy algorithm of Sviridenko [49] and indeed gives non-trivial approximation guarantees. Further, due to its simplicity it also has the other desired properties of truthfulness, individual rationality, and budget-feasibility. While this whole framework might feel somewhat straightforward, the existing literature on budget-feasible mechanisms suggests that there is a frail balance between simplicity and performance here. Only “naive” algorithmic ideas, like greedy, seem to have any hope generating truthful mechanisms that are robust subject to cost changes and, thus, budget-feasible.

Unfortunately, it is easy to construct examples where running such a greedy algorithm for a non-monotone objective results in a solution of arbitrarily poor quality. The algorithmic state-of-the-art for non-monotone submodular maximization under a knapsack constraint, e.g., [27, 36, 19], provides us with quite involved algorithms on continuous relaxations of the problem that seem very unlikely to yield monotone allocation rules, and thus truthful mechanisms. The only simple (and deterministic) exception is the two-pass greedy algorithm of Gupta et al. [32], where it is shown that running Sviridenko’s greedy algorithm twice and then maximizing without the knapsack constraint is sufficient to get a deterministic 6-approximation algorithm.[1] Despite being significantly simpler, however, this two-pass greedy algorithm still suffers with respect to monotonicity.

More recently, several simple randomized greedy approaches for maximizing non-monotone submodular objectives have been proposed for a cardinality or a matroid constraint [28, 18, 17, 25] and even for a knapsack constraint [3]. However, these approaches are also not applicable here. In its

[1] In fact, that algorithm has an approximation ratio of $4+\alpha$, where α is the approximation ratio of any deterministic algorithm for the unconstrained maximization of non-monotone submodular functions. Recently, Buchbinder and Feldman [16] suggested a deterministic 2-approximation algorithm for the unconstrained problem, hence the ratio of 6.

simplest version such a random greedy algorithm would initially randomly discard half of the agents and then run a greedy algorithm for monotone submodular objectives. The issue is that even if a random greedy algorithm directly worked for a knapsack constraint in terms of approximate optimality and truthfulness—something that is not straightforward—budget-feasibility crucially depends on the monotonicity of the objective function [20, 45]. So, one still needs to deal with the fact that for non-monotone objectives the payments of simple greedy algorithms (like the one by Singer [45]) can be unbounded.

At the heart of our approach lies a novel deterministic greedy algorithm for non-monotone submodular maximization under a knapsack constraint. Our algorithm builds two candidate solutions *simultaneously*, yet prevents agents to jump from one solution to the other by changing their cost. To do the latter we offer each agent a take-it-or-leave-it price based on an estimate of the optimal value which we obtain by sampling. A crucial property of the resulting mechanism is that the agents are not ordered with respect to their marginal value per cost. While the latter is a very simple property, this is the first mechanism using only value queries where the ordering of the agents is independent of their cost. This further allows us to appropriately modify the algorithm and adapt it to the online secretary setting and to settings with additional feasibility constraints, while maintaining all its desired properties.

All of our mechanisms are randomized and, in fact, random sampling is an essential building block in our approach. Obtaining a good estimate of the optimal value via random sampling has been crucial in previous works on budget-feasible mechanism design for monotone objectives as well [13, 8, 2, 39]. Designing *deterministic* budget-feasible mechanisms seems very challenging. Beyond additive valuation functions [45, 20], no deterministic, polynomial-time $O(1)$-approximation mechanisms are known, except for some specific well-behaved objectives [46, 1, 33, 21, 2]. In order to obtain a constant approximation ratio while maintaining truthfulness, one would need to compare the single most valuable agent to an easy-to-calculate estimate of the optimal value that is also non-increasing to each agent's cost. Obtaining deterministic, budget-feasible, $O(1)$-approximation mechanisms is an intriguing topic for future research.

1.3. Related Work As mentioned above, the study of budget-feasible mechanisms was initiated by Singer [45], who gave a randomized $O(1)$-approximation mechanism for monotone submodular functions. Later, Chen et al. [20] significantly improved the approximation ratio and also suggested a deterministic $O(1)$-approximation mechanism, albeit with superpolynomial running time. Several follow-up results modified this deterministic mechanism so that it runs in polynomial time for special cases, including coverage functions [46, 1] and information gain functions [33]. For subadditive functions, Dobzinski et al. [21] suggested a $O(\log^2 n)$-approximation mechanism, and gave the first constant factor mechanisms for a special case of non-monotone objectives, namely cut functions. The factor for subadditive functions was later improved to $O(\log n/\log\log n)$ by Bei et al. [13], who also gave a randomized $O(1)$-approximation mechanism for XOS functions, albeit in exponential time in the value query model (see Remark 1), initiated the Bayesian analysis in this setting, and gave an existential result for an $O(1)$-approximation mechanism for subadditive valuations. Amanatidis et al. [2] suggested $O(1)$-approximation mechanisms for a subclass of non-monotone submodular objectives, namely symmetric submodular objectives, however their approach does not seem to generalize beyond this subclass. For settings with additional combinatorial constraints, Amanatidis et al. [1] and Leonardi et al. [39] gave $O(1)$-approximation mechanisms for additive valuation functions subject to independent system constraints. There is also a line of related work under the *large market* assumption (where no participant can significantly affect the market outcome), which allows for mechanisms with improved performance (see, e.g., [48, 5, 29, 10, 35]). Very recently, Gravin et al. [30] almost resolved the additive case by designing an optimal randomized mechanism and a near-optimal deterministic mechanism.

The online version of the problem was introduced and studied by Badanidiyuru et al. [8] who give an $O(1)$-approximation mechanism for monotone submodular functions. Singer and Mittal [47] also studied an online version of the problem for a cardinality objective, i.e., for the case one wants to maximize the number of winning agents. The problem as introduced by Badanidiyuru et al. [8] is closely related to the purely algorithmic version of the problem (i.e., without the incentives), namely the Submodular Knapsack Secretary Problem introduced by Bateni et al. [11] as a generalization of the Knapsack Secretary Problem [7]. Bateni et al. studied the problem for monotone and non-monotone submodular objectives, although they provide a complete proof only for the former case. While the monotone submodular case has been improved [26] and generalized [34], there is no follow-up work on the non-monotone case to the best of our knowledge.

On maximization of submodular functions subject to knapsack or other type of constraints, there is a vast literature, going back several decades (see, e.g., [43, 50]). Focusing on knapsack constraints, there is a rich line of recent work on developing algorithms on continuous relaxations of the problem (see, e.g., [27, 36, 19, 23] and references therein) achieving an e-approximation for non-monotone objectives. However, the most relevant recent work to ours is that of Gupta et al. [32] who proposed a deterministic 6-approximation algorithm for the non-monotone case, related on a high level to our main approach. Gupta et al. also gave algorithms for certain constrained secretary problems, although not with knapsack constraints. When ℓ knapsack constraints and a p-system constraint are both present, the algorithmic state-of-the-art is a $(p+2\ell+1)$-approximation algorithm for the monotone submodular case due to Badanidiyuru and Vondrák [9] and a $(p+1)(2p+2\ell+1)/p$-approximation algorithm for the non-monotone submodular case due to Mirzasoleiman et al. [41].

As mentioned above, there is a line of work that uses random greedy algorithms for maximizing non-monotone submodular objectives subject to other combinatorial constraints [28, 18, 17, 25]. Although not directly related to our work, there are underlying similarities as the algorithms developed are simple, greedy and often extend to online settings. Additionally, if one could resolve the issue of the payments being unbounded, a random greedy version of Singer's mechanism could lead to significantly improved approximation guarantees in our setting.

Remark 1 (On the $O(1)$-approximation mechanism of Bei, Chen, Gravin, and Lu). Bei et al. [13] propose an $O(1)$-approximation mechanism for *non-decreasing* XOS objectives that runs in polynomial time in the much stronger *demand query* model. However, they briefly discuss how to extend their result to general XOS functions via the use of $\hat{v}(S) = \max_{T\subseteq S} v(T)$. It is easy to see that $\hat{v}$ is non-decreasing and that S is an optimal solution of v if and only if it is a minimal optimal solution for $\hat{v}$. Moreover, Gupta et al. [31] proved that if v is general XOS then $\hat{v}$ is monotone XOS. It should be noted that this transformation does not work in the submodular case, i.e., when v is submodular, $\hat{v}$ is not necessarily submodular [2]. Therefore, known results for monotone submodular functions do not extend to the non-monotone case, even in the demand query model.

2. Preliminaries

We use $A = [n] = \{1, 2, \ldots, n\}$ to denote a set of n agents. Each agent i is associated with a private cost c_i, denoting the cost for participating in the solution. We consider a procurement auction setting, where the auctioneer is equipped with a valuation function $v : 2^A \to \mathbb{Q}_{\geq 0}$ and a budget $B > 0$. For $S \subseteq A$, $v(S)$ is the value derived by the auctioneer if the set S is selected (for singletons, we will often write $v(i)$ instead of $v(\{i\})$). Therefore, the algorithmic goal in all the problems we study is to select a set S that maximizes $v(S)$ subject to the constraint $\sum_{i\in S} c_i \leq B$. We assume oracle access to v via *value queries*, i.e., we assume the existence of a polynomial time value oracle that returns $v(S)$ when given as input a set S.

A function v is *non-decreasing* (often referred to as *monotone*), if $v(S) \leq v(T)$ for any $S \subseteq T \subseteq A$. We consider general (i.e., not necessarily monotone), normalized (i.e., $v(\emptyset) = 0$), non-negative submodular valuation functions. Since marginal values are extensively used, we adopt the shortcut $v(i \mid S)$ for the marginal value of agent i with respect to the set S, i.e., $v(i \mid S) = v(S \cup \{i\}) - v(S)$. The following three definitions of submodularity are equivalent. While definition *(i)* is the most standard, the other two alternative definitions will be useful later on.

Definition 1. A function v, defined on 2^A for some set A, is *submodular* if and only if

(i) $v(i \mid S) \geq v(i \mid T)$ for all $S \subseteq T \subseteq A$, and $i \notin T$.

(ii) $v(S) + v(T) \geq v(S \cup T) + v(S \cap T)$ for all $S, T \subseteq A$.

(iii) $v(T) \leq v(S) + \sum_{i \in T \setminus S} v(i \mid S) - \sum_{i \in S \setminus T} v(i \mid S \cup T \setminus \{i\})$ for all $S, T \subseteq A$.

In the special case where $v(i \mid S) = v(i \mid \emptyset)$, for all $i \in A$ and all $S \subseteq A$, then we say that v is *additive*.

In Section 6 we also deal with valuation functions that come from a superclass of submodular functions, namely *XOS* or *fractionally subadditive* functions. In particular, it is known that non-negative (monotone) submodular functions are a strict subset of (monotone) XOS functions [37, 31].

Definition 2. A function v, defined on 2^A for some set A, is *XOS* or *fractionally subadditive*, if there exist additive functions $\alpha_1, \ldots, \alpha_r$, for some finite r, such that $v(S) = \max_{i \in [r]} \alpha_i(S)$.

We often need to argue about optimal solutions of sub-instances of the original instance $(A, v, \mathbf{c}, B)$. Given a cost vector $\mathbf{c}$, and a subset $X \subseteq A$, we denote by $\mathbf{c}_X$ the projection of $\mathbf{c}$ on X, and by $\mathbf{c}_{-X}$ the projection of $\mathbf{c}$ on $A \setminus X$. By $\text{OPT}(X, v, \mathbf{c}_X, B)$ we denote the value of an optimal solution to the problem restricted on X. Similarly, $\text{OPT}(X, v, \infty)$ denotes the value of an optimal solution to the unconstrained version of the problem restricted on X. For the sake of readability, we usually drop the valuation function and the cost vector, and write $\text{OPT}(X, B)$ and $\text{OPT}(X, \infty)$, respectively.

2.1. Mechanism Design In the strategic version that we consider here, every agent $i \in A$ only has his true cost c_i as private information. Hence, this is a *single-parameter environment*. A mechanism $\mathcal{M} = (f, p)$ in our context consists of an outcome rule f and a payment rule p. Given a vector of cost declarations, $\mathbf{b} = (b_i)_{i \in A}$, where b_i denotes the cost reported by agent i, the outcome rule of the mechanism selects the set $f(\mathbf{b}) \subseteq A$. At the same time, it computes payments $p(\mathbf{b}) = (p_i(\mathbf{b}))_{i \in A}$ where $p_i(\mathbf{b})$ denotes the payment issued to agent i. Hence, the final utility of agent i is $p_i(\mathbf{b}) - c_i$.

Unless stated otherwise, our mechanisms run in polynomial time in the value query model. Further properties we want to enforce in our mechanism design problem are the following.

Definition 3. A mechanism $\mathcal{M} = (f, p)$ is

- *truthful*, if reporting c_i is a dominant strategy for every agent i.
- *individually rational*, if $p_i(\mathbf{b}) \geq 0$ for every $i \in A$, and $p_i(\mathbf{b}) \geq c_i$, for every $i \in f(\mathbf{b})$.
- *budget-feasible*, if $\sum_{i \in A} p_i(\mathbf{b}) \leq B$ for every $\mathbf{b}$.

For our randomized mechanisms we use the strong notion of *universal truthfulness*, which means that the mechanism is a probability distribution over deterministic truthful mechanisms. As all the mechanisms we suggest are universally truthful, we will consistently use $\mathbf{c} = (c_i)_{i \in A}$ rather than $\mathbf{b} = (b_i)_{i \in A}$ for the declared costs in their description and analysis.

To design truthful mechanisms for single-parameter environments, we use a characterization by Myerson [42]. We say that an outcome rule f is *monotone*, if for every agent $i \in A$, and any vector of cost declarations $\mathbf{b}$, if $i \in f(\mathbf{b})$, then $i \in f(b'_i, \mathbf{b}_{-i})$ for $b'_i \leq b_i$. That is, if an agent i is selected by declaring cost b_i, then he should still be selected by declaring a lower cost. Myerson's lemma, below, implies that monotone algorithms admit truthful payment schemes (often referred to as *threshold payments*). This greatly simplifies the design of truthful mechanisms, as one may focus on constructing monotone algorithms rather than having to worry about the payment scheme. For all of our mechanisms, we assume that the underlying payment scheme is given by Myerson's lemma.

LEMMA 1 (Myerson [42]). *Given a monotone algorithm f, there is a unique payment scheme p, such that (f,p) is a truthful and individually rational mechanism, given by*

$$p_i(\mathbf{b}) = \begin{cases} \sup_{b'_i \in [c_i,\infty)} \{b'_i : i \in f(b'_i, \mathbf{b}_{-i})\}\,, & \text{if } \ i \in f(\mathbf{b}), \\ 0\,, & \text{otherwise.} \end{cases}$$

2.2. Technical Assumptions We may assume, without loss of generality, that in any given instance all the costs are upper bounded by the budget. To see this notice that neither our mechanisms nor the optimal offline solution will ever consider any agent with cost higher than B. Furthermore, no agent has an incentive to misreport a very high true cost. Indeed, due to budget-feasibility, if agent i reports a cost $b_i \le B$ instead of his true cost $c_i > B$ and is selected, then he has utility $p_i(\mathbf{b}) - c_i < B - B = 0$. Thus, in all of our mechanisms we implicitly assume a preprocessing step that removes all the agents with declared costs exceeding B. The resulting instance (given as input to the corresponding mechanism) has the same set of optimal solutions subject to the budget constraint as the original one. Note that in the case of the online mechanism GenSm-Online rejecting such agents as they arrive suffices.

We should stress that wherever tie-breaking is needed (e.g., in lines 3 and 10 of Simultaneous Greedy, during the execution of the auxiliary algorithms $\textsc{alg}_1$, $\textsc{alg}_2$ and $\textsc{alg}_3$, etc.), we assume the consistent use of a tie-breaking rule that is *independent of the declared costs*. An obvious such choice would be a deterministic lexicographic

In our mechanisms we often use randomized approximation algorithms for constrained submodular maximization as subroutines. In particular, $\textsc{alg}_1$ in Sample-then-Greedy and GenSm-Online, $\textsc{alg}_3$ in MonSm-Constrained, and $\textsc{alg}_4$ in GenSm-Constrained are all randomized. In our analyses, we need variants of these algorithms that almost achieve their guarantees with probability close to 1. In particular, we use the fact that for any constants δ,η, a randomized ρ-approximation algorithm $\textsc{alg}$ can be modified so that *with probability at least $1-\delta$ it returns a solution of value at least $\left(\frac{1}{\rho}-\eta\right)\textsc{opt}$ in polynomial time*. This is achieved by simply running $\textsc{alg}$ $\Theta\left(\frac{1}{\eta}\log\left(\frac{1}{\delta}\right)\right)$ times and keeping the best solution. For completeness we prove this simple fact here.

LEMMA 2. *Let $\textsc{alg}$ be a randomized ρ-approximation algorithm for a constrained submodular maximization problem. Also, let $\textsc{alg}'$ be the algorithm that runs $\textsc{alg}$ $\frac{3}{\eta}\log\left(\frac{1}{\delta}\right)$ times and outputs the best among these solutions, where $\delta,\eta \in (0,1)$. Then for any instance I, with probability at least $1-\delta$, $v(\textsc{alg}'(I)) \ge \left(\frac{1}{\rho}-\eta\right)\cdot\textsc{opt}(I)$, where $\textsc{opt}(\cdot)$ is the value of an optimal solution of the corresponding problem.*

Proof. Let I be an arbitrary instance. By $\mathcal{E}_<$ we denote the event $v(\textsc{alg}'(I)) < \left(\frac{1}{\rho}-\eta\right)\cdot\textsc{opt}(I)$ and by $\mathcal{E}_\ge$ its complement. We have

$$\begin{aligned}\frac{1}{\rho}\textsc{opt}(I) &\le \mathbb{E}(v(\textsc{alg}(I))) = \mathrm{P}(\mathcal{E}_<)\cdot\mathbb{E}\big(v(\textsc{alg}(I))\,|\,\mathcal{E}_<\big) + \mathrm{P}(\mathcal{E}_\ge)\cdot\mathbb{E}\big(v(\textsc{alg}(I))\,|\,\mathcal{E}_\ge\big)\\ &\le 1\cdot\left(\frac{1}{\rho}-\eta\right)\textsc{opt}(I) + \mathrm{P}(\mathcal{E}_\ge)\cdot\textsc{opt}(I)\,.\end{aligned}$$

Then it is easy to see that $\mathrm{P}(\mathcal{E}_\ge) \ge \eta$, and thus, the probability that $\textsc{alg}'$ fails to produce a solution of value $\left(\frac{1}{\rho}-\eta\right)\cdot\textsc{opt}(I)$ is

$$\mathrm{P}\Big(v(\textsc{alg}'(I)) < \big(\frac{1}{\rho}-\eta\big)\cdot\textsc{opt}(I)\Big) \le (1-\eta)^{\frac{3}{\eta}\log\left(\frac{1}{\delta}\right)} < (1-\eta)^{\log\left(\frac{1}{\delta}\right)/\log\left(\frac{1}{1-\eta}\right)} = \delta\,,$$

where verifying the last inequality for $\eta \in (0,1)$ is just a matter of simple calculations. ∎

Whenever we say that we use a known algorithm as a subroutine, we mean its *concentrated version* suggested by Lemma 2 for appropriately small positive constants δ and η. Note that as long as ALG runs in polynomial time and δ, η are constants, ALG$'$ also runs in polynomial time.

As it will become apparent by the proof of Theorem 1 (in particular by the proofs of Lemma 7 and Corollary 2), technical nuances aside, ALG_1—or any other concentrated randomized algorithm—can be used like a deterministic approximation algorithm in our analysis. To facilitate the presentation, in the proofs of Theorems 2 and 4 we treat ALG_1, ALG_3, and ALG_4 as if there were deterministic. This does not affect the achieved guarantees, as there is some slack in the approximation ratios derived in this work and, for small enough δ and η, any resulting increase can, in fact, be "hidden" in the current ratios.

3. An Efficient Mechanism for Submodular Objectives The main result of this section is the first $O(1)$-approximation mechanism (termed GenSm-Main below) for non-monotone submodular valuation functions.

Theorem 1. *GenSm-Main is a universally truthful, individually rational, budget-feasible, $O(1)$-approximation mechanism.*

At the heart of our approach lies a novel greedy algorithm for non-monotone submodular maximization under a knapsack constraint (Simultaneous Greedy below). As we mentioned in the Introduction, all known mechanisms use the same greedy subroutine: sort all agents in decreasing order of marginal value per cost and pick as many agents as possible before hitting some threshold. While for monotone submodular objectives this gives a non-trivial approximation guarantee, for non-monotone objectives may result in arbitrarily bad solutions. Moreover, continuous algorithmic approaches for non-monotone submodular maximization under a knapsack constraint [27, 36] seem very unlikely to yield monotone allocation rules, and thus truthful mechanisms. The only algorithm that is conceptually close to our approach is the two-pass greedy algorithm of Gupta et al. [32], that runs Sviridenko's greedy algorithm twice and then maximizes without the knapsack constraint to get a deterministic 6-approximate solution. The intuition behind this approach is that submodularity prevents the greedy algorithm from getting stuck in consecutive "bad" local maxima. Despite being significantly simpler, however, this two-pass greedy algorithm still suffers irreparably with respect to monotonicity, as it allows agents to jump from one solution to the other by changing their cost.

The presentation of our mechanism resembles the presentation of other algorithms in the related literature, e.g, [8, 13], as it has a similar high-level structure (randomization between best singleton and a greedy solution which needs a sampling preprocessing step).

First we introduce Simultaneous Greedy, a greedy mechanism that builds two candidate solutions *simultaneously*. While the analysis of Gupta et al. [32] does not apply here (our solutions are neither built sequentially nor according to the standard greedy algorithm), the way we obtain our approximation guarantee is of the same flavor: at least one of the solutions will contain an approximately optimal set. At the same time Simultaneous Greedy prevents agents to choose their favorite candidate solution by misreporting their cost. To achieve that, we offer each agent a take-it-or-leave-it price based on an estimate x of the optimal value which we obtain by sampling. It is crucial that in our mechanism the agents are *not* ordered with respect to their marginal value per cost. This will further allow us to appropriately modify Simultaneous Greedy for the *online* setting of Section 4 while maintaining all its desired properties.

Note that in line 4 of the algorithm we introduce the parameter β. We later set β equal to 9.185 in order to get the approximation factor of Corollary 2 but, otherwise, our analysis is independent of β's value. An analogous parameter will be used in all of our mechanisms in later sections, and each time it will be tuned differently. ALG_2 in line 9 can be any approximation algorithm for *unconstrained* non-monotone submodular maximization. In particular, we may use the best known approximation algorithm, i.e., the 2-approximation algorithm of Buchbinder and Feldman [16].

Simultaneous Greedy$(D, v, \mathbf{c}_D, B, x)$

$S_1 = S_2 = \emptyset$; $B_1 = B_2 = B$; $U = D$ /* each S_j has its own budget B_j */
while $\max_{i \in U, j \in \{1,2\}} v(i|S_j) > 0$ **do**
  Let $(\hat{\imath}, \hat{\jmath}) \in \arg\max_{i \in U, j \in \{1,2\}} v(i|S_j)$
  if $c_{\hat{\imath}} \le \frac{\beta B}{x} v(\hat{\imath}|S_{\hat{\jmath}}) \le B_{\hat{\jmath}}$ **then**
    $S_{\hat{\jmath}} = S_{\hat{\jmath}} \cup \{\hat{\imath}\}$
    $B_{\hat{\jmath}} = B_{\hat{\jmath}} - \frac{\beta B}{x} v(\hat{\imath}|S_{\hat{\jmath}})$
  $U = U \setminus \{\hat{\imath}\}$
for $j \in \{1,2\}$ **do**
  $T_j = \mathrm{ALG}_2(S_j)$ /* a 2-approximate solution with respect to $\mathrm{OPT}(S_j, v, \mathbf{c}_{S_j}, \infty)$ */
Let S be the best solution among S_1, S_2, T_1, T_2
return S

Ideally, we would like the rate parameter x to be close to $\mathrm{OPT}(A,B)$ and also to be robust in the sense that no single agent can significantly affect its value. To achieve that, Sample-then-Greedy randomly partitions the set of agents into two sets A_1 and A_2, then approximately solves the problem on A_1 to obtain an estimate of $\mathrm{OPT}(A_1, B)$, and finally uses this x to set the threshold rate for Simultaneous Greedy on A_2.

ALG_1 in line 2 can be the concentrated version (suggested by Lemma 2; see the discussion around the lemma) of any approximation algorithm for non-monotone submodular maximization subject to a knapsack constraint. Again, we may use the best known approximation algorithm, i.e., the e-approximation algorithm of Kulik et al. [36]. The constants δ, η used for its concentrated version are given right before Lemma 7.

Sample-then-Greedy$(A, v, \mathbf{c}, B)$

1 Put each agent of A in either A_1 or A_2 independently at random with probability $\frac{1}{2}$
2 $x = v(\mathrm{ALG}_1(A_1))$ /* (the concentrated version of) an e-approximation of $\mathrm{OPT}(A_1, v, \mathbf{c}_{A_1}, B)$ */
3 **return** Simultaneous Greedy$(A_2, v, \mathbf{c}_{A_2}, B, x)$

Lemma 6 in Subsection 3.1, due to Bei et al. [12] and Leonardi et al. [38], guarantees that with high probability both A_1 and A_2 contain enough value subject to the budget constraint for things to work, as long as no agent is too valuable. The latter leads to the final mechanism GenSm-Main (GENeral SubModular-MAIN) that randomizes between all the above and just returning a best singleton.

GenSm-Main$(A, v, \mathbf{c}, B)$

With probability $p = 0.2$ **: return** $i^* \in \arg\max_{i \in A} v(i)$
With probability $1 - p$ **: return** Sample-then-Greedy$(A, v, \mathbf{c}, B)$
Pay the agents according to Myerson's lemma

Remark 2 (Turning this into a (non-truthful) algorithm). Here it is necessary that Simultaneous Greedy uses thresholds in order to achieve the properties stated in Theorem 1. While this goes beyond the point of this work, one can follow the same approach of greedily building two solutions at the same time (using a variant of Sviridenko's algorithm [49]), in order to design a deterministic 7-approximation algorithm for maximizing non-monotone submodular functions subject to a knapsack constraint. While this does not improve the state of the art, the overall approach might be of independent interest for certain variants of the problem (like it was here for mechanism design).

3.1. Proving the Properties of GenSm-Main We fix some additional notation to facilitate the presentation of the proofs. In particular, we want to be able to argue about S_1, S_2, B_1, B_2 on a per iteration basis. We use $(D, v, \mathbf{c}, B, x)$ for a generic instance given to Simultaneous Greedy and S for the set returned. By $i_1, i_2, \dots, i_t$ we denote the sequence of agents of D that were examined (i.e., appeared in line 3 of the algorithm) during this execution of Simultaneous Greedy in this exact order. All the agents of S clearly appear within this sequence, so for any particular $\ell \in S$ we have that $\ell = i_k$ for some k. By j_k we denote the index $\hat{\jmath}$ picked during the kth execution of line 3 of Simultaneous Greedy, while by $S_{j_k}^{(k)}$ and $B_{j_k}^{(k)}$ we denote the set S_{j_k} and its remaining budget, respectively, at that time. Conventionally, we use notation like $S_{j_k}^{(k+1)}$ to denote S_{j_k} right after the kth execution of line 7, even if line 3 is never executed more than k times. Recall that we use a tie-breaking rule that is independent of the costs, as mentioned in Section 2.

Lemma 3. *The allocation rule defined by* Simultaneous Greedy *is monotone. Thus, using the threshold payments of Myerson's lemma, the resulting mechanism is truthful and individually rational.*

Proof. We need to show that a winning agent remains a winner if he decreases his cost; then the statement follows by Lemma 1. In fact, we show something stronger, namely that no winning agent can affect the output of Simultaneous Greedy by lowering his bid.

Let S be the set returned when the input is $(D, v, \mathbf{c}, B, x)$ and fix some agent $i_k \in S$. That is, during the kth execution of line 3, $(\hat{\imath}, \hat{\jmath}) = (i_k, j_k)$. Fix the vector $\mathbf{c}_{-i_k}$ for the other agents, and suppose that agent i_k declares $c'_{i_k} < c_{i_k}$. Clearly, the execution of Simultaneous Greedy$(D, v, (\mathbf{c}_{-i_k}, c'_{i_k}), B, x)$ will be exactly the same as before for agents $i_1, \dots, i_{k-1}$. Further, $\hat{\jmath}$ will again be j_k. Thus, i_k will again be added to the $S_{j_k}^{(k)}$ since

$$c'_{i_k} < c_{i_k} \le \frac{\beta B}{x} v\left(i_k \,\middle|\, S_{j_k}^{(k)}\right) \le B_{j_k}.$$

After updating B_{j_k} to $B_{j_k}^{(k)} - \frac{\beta B}{x} v\big(i_k | S_{j_k}^{(k)}\big)$, everything is exactly the same as in the beginning of the $(k+1)$th iteration of the original execution of Simultaneous Greedy$(D, v, \mathbf{c}, B, x)$ and, therefore, the algorithm will proceed in exactly the same way to produce the same output S. In particular, agent i_k will still be a winner. ∎

In all the following statements, when we refer to mechanisms, we always assume threshold payments. Before we study the total payment, we should point out that enforcing budget-feasibility has been the main source of technical difficulties in the budget-feasible mechanism design literature. A significant advantage of the *posted price* approach used in threshold mechanisms like Simultaneous Greedy is that the budget-feasibility becomes much more manageable. To some extent this comes at the expense of the approximation guarantee and its analysis, but also offers some additional flexibility that will be explored in Sections 4 and 5.

Lemma 4. *The mechanism* Simultaneous Greedy *is budget-feasible.*

Proof. Let S be the set returned given the instance $(D, v, \mathbf{c}, B, x)$ and fix $i_k \in S$. We claim that the payment $p_{i_k}(\mathbf{c})$ is exactly $\pi_k = \frac{\beta B}{x} v\big(i_k | S_{j_k}^{(k)}\big)$, i.e., $i_k \in S$ if and only if he bids $c'_{i_k} \le \pi_k$. First note that i_k cannot affect the time when he is examined by the mechanism or which agents come before him. So, since $\mathbf{c}_{-i_k}$ is fixed, during the kth execution of line 3, he is always "offered" π_k; either he accepts, i.e., $c'_{i_k} \le \pi_k$, and the algorithm proceeds in the exact same way as with $c'_{i_k} = c_{i_k}$ (see also the proof of Lemma 3) or he rejects, i.e., $c'_{i_k} > \pi_k$, and he is removed from the active set of agents. Once an agent is removed, however, he is never reexamined and thus, if $c'_{i_k} > \pi_k$ then i_k is not in the winning set.

Recall that S can be any of S_1, S_2, T_1, T_2. We will show that all four sets are budget-feasible. Let $T_1 = \{i_{a_1}, i_{a_2}, \dots, i_{a_{|T_1|}}\}$ and $S_1 = \{i_{b_1}, i_{b_2}, \dots, i_{b_{|S_1|}}\}$, where $(a_i)_{i=1}^{|T_1|}$ is a subsequence of $(b_i)_{i=1}^{|S_1|}$ which is a subsequence of $1, 2, \dots, t$. Recall that t is the total number of agents examined during this particular execution of SIMULTANEOUS GREEDY. Also notice that the budget B_1 for S_1 never becomes negative. We have

$$\sum_{\tau=1}^{|T_1|}\pi_{a_\tau} \le \sum_{\tau=1}^{|S_1|}\pi_{b_\tau} = \sum_{\tau=1}^{|S_1|}\frac{\beta B}{x}v\left(i_{b_\tau}|S_{j_{b_\tau}}^{(b_\tau)}\right) = B - B_1^{(|S_1|+1)} \le B\,.$$

The first and the second sum represent the total payment when $S=T_1$ and when $S=S_1$, respectively. The budget-feasibility of T_2 and S_2 is proved in the exact same way. ∎

COROLLARY 1. *The mechanism* GENSM-MAIN *is universally truthful, individually rational and budget-feasible.*

Proof. Given that $\textsc{alg}_1$ and $\textsc{alg}_2$ run in polynomial time, and that it is straightforward to determine the payments (Lemma 4), it is clear that GENSM-MAIN is a polynomial-time mechanism.

Further, GENSM-MAIN is a probability distribution over the mechanism that returns $i^* \in \arg\max_{i\in A} v(i)$ and SIMULTANEOUS GREEDY$(D, v, \mathbf{c}_D, B, v(\textsc{alg}_1(A\setminus D)))$ for all $D\subseteq A$. The simple mechanism that returns i^* and pays him the threshold payment is truthful and individually rational. Also, it is clear that the threshold payment is exactly B, so this mechanism is budget-feasible as well.

The desired properties of GENSM-MAIN now follow from Lemmata 3 and 4 and the above observations. ∎

LEMMA 5. *If there is a positive integer ℓ such that $\max_{i\in D} v(i) < \frac{x}{\ell\cdot\beta}$, then* SIMULTANEOUS GREEDY$(D, v, \mathbf{c}, B, x)$ *outputs a set S such that*

$$v(S) \ge \min\left\{\frac{\ell x}{(\ell+1)\beta}\,,\,\frac{1}{6}\left(\textsc{opt}(D,B) - \frac{2x}{\beta}\right)\right\}.$$

Proof. Let t be the number of times line 3 was executed. At the end of the tth iteration, U is the set of agents never examined. That is, U only contains agents that have non-positive marginal utilities with respect to $S_1^{(t+1)}$ and $S_2^{(t+1)}$. For the sake of readability, we henceforth use S_1 and S_2 to denote $S_1^{(t+1)}$ and $S_2^{(t+1)}$, respectively. Let $R = D\setminus(U\cup S_1\cup S_2)$ be the agents i_k that were considered at some point by the mechanism but were *rejected*, i.e., not added to either $S_1^{(k)}$ or $S_2^{(k)}$. We first partition R into two sets depending on *why* the corresponding agents were rejected. The set

$$R_{\mathbf{c}} = \left\{ i_k \,\middle|\, \frac{\beta B}{x} v\left(i_k \,\middle|\, S_{j_k}^{(k)}\right) < c_{\hat{\imath}} \right\}$$

contains the agents rejected because the first inequality in line 4 was violated during the corresponding iteration. Similarly, the set

$$R_B = \left\{ i_k \,\middle|\, B_{j_k}^{(k)} < \frac{\beta B}{x} v\left(i_k \,\middle|\, S_{j_k}^{(k)}\right) \right\}$$

contains the agents rejected because the second inequality in line 4 was violated. Clearly, $R = R_{\mathbf{c}}\cup R_B$. We consider two cases, depending on whether R_B is empty or not.

Case 1. Assume that $R_B\neq\emptyset$ and let $i_k\in R_B$. That is, during the kth execution of line 3, $(\hat{\imath},\hat{\jmath}) = (i_k, j_k)$, but $\frac{\beta B}{x}v\big(i_k|S_{j_k}^{(k)}\big) > B_{j_k}^{(k)}$. Let $S_{j_k}^{(k)} = \{i_{a_1}, i_{a_2},\dots,i_{a_s}\}$, where $(a_i)_{i=1}^s$ is a subsequence of $1,2,\dots,t$. Further, notice that, by its definition, $B_{j_k}^{(k)} = B - \sum_{\tau=1}^{s}\frac{\beta B}{x}v\big(i_{a_\tau}|S_{j_k}^{(a_\tau)}\big)$. We have

$$\begin{aligned} v\left(S_{j_k}^{(k)}\right) &= \sum_{\tau=1}^{s} v\left(i_{a_\tau}|S_{j_k}^{(a_\tau)}\right) = \frac{x}{\beta B}\left(B - B_{j_k}^{(k)}\right) \\ &> \frac{x}{\beta B}\left(B - B_{j_k}^{(k)}\right) + \frac{x}{\beta B}B_{j_k}^{(k)} - v\left(i_k|S_{j_k}^{(k)}\right) = \frac{x}{\beta} - v\left(i_k|S_{j_k}^{(k)}\right). \end{aligned} \tag{1}$$

By submodularity and the way the agents in $S_{j_k}^{(k)}$ are chosen, we have

$$v(i_{a_1}) = v\left(i_{a_1}|S_{j_k}^{(a_1)}\right) \geq v\left(i_{a_2}|S_{j_k}^{(a_2)}\right) \geq \ldots \geq v\left(i_{a_s}|S_{j_k}^{(a_s)}\right) \geq v\left(i_k|S_{j_k}^{(k)}\right).$$

Yet, each one of these values is at most $\max_{i\in D} v(i) < \frac{x}{\ell\cdot\beta}$. Combining with (1), we have

$$\ell\cdot\max_{i\in D} v(i) < \frac{x}{\beta} \leq v\left(S_{j_k}^{(k)}\right) + v\left(i_k|S_{j_k}^{(k)}\right) \leq \sum_{\tau=1}^{s} v\left(i_{a_\tau}|S_{j_k}^{(a_\tau)}\right) + v\left(i_k|S_{j_k}^{(k)}\right) \leq (s+1)\cdot v\left(i_{a_1}\right),$$

and therefore, we conclude that $\left|S_{j_k}^{(k)}\right| = s \geq \ell$. Now we repeat the same argument for the average marginal value in the sum $\sum_{\tau=1}^{s} v\big(i_{a_\tau}|S_{j_k}^{(a_\tau)}\big)$. Using the simple observation that the smallest term of a sum cannot exceed the average of the remaining terms, we get

$$\frac{x}{\beta} \leq v\left(S_{j_k}^{(k)}\right) + v\left(i_k|S_{j_k}^{(k)}\right) \leq v\left(S_{j_k}^{(k)}\right) + \frac{1}{s}\sum_{\tau=1}^{s} v\left(i_{a_\tau}|S_{j_k}^{(a_\tau)}\right) \leq \frac{s+1}{s} v\left(S_{j_k}^{(k)}\right) \leq \frac{\ell+1}{\ell} v\left(S_{j_k}^{(k)}\right), \quad (2)$$

where the last inequality follows from the fact that $f(z) = \frac{z+1}{z}$ is decreasing.

Finally, to get the approximation guarantee for this case, we combine (2) with the fact that S is at least as good as each greedy solution:

$$v(S) \geq v(S_{j_k}) \geq v\left(S_{j_k}^{(k)}\right) \geq \frac{\ell}{\ell+1}\cdot\frac{x}{\beta}.$$

Case 2. Now assume that $R_B = \emptyset$, i.e., $R = R_{\mathbf{c}}$. Let C^* be an optimal solution for the given instance and define $C_1 = C^* \cap S_1$, $C_2 = C^* \cap S_2$ and $C_3 = C^* \setminus (C_1 \cup C_2)$. By subadditivity, we have

$$\textsc{opt}(D,B) = v(C^*) \leq v(C_1) + v(C_2) + v(C_3). \quad (3)$$

Recall that $T_j = \textsc{alg}_2(S_j), j \in \{1,2\}$, is a 2-approximate solution with respect to $\textsc{opt}(S_j,\infty)$. Thus, $v(C_j) \leq \textsc{opt}(S_j,B) \leq 2\cdot v(T_j)$, for $j\in\{1,2\}$, and inequality (3) gives

$$\textsc{opt}(D,B) \leq 2v(T_1) + 2v(T_2) + v(C_3). \quad (4)$$

Upper bounding $v(C_3)$ in terms of S_1, S_2, T_1, T_2 and x is somewhat more involved. We begin by invoking the non-negativity of v, as well as its submodularity (as defined in Definition 1*(ii)*) on $S_1 \cup C_3$ and $S_2 \cup C_3$. We have

$$v(C_3) \leq v(C_3) + v(S_1 \cup C_3 \cup S_2) \leq v(S_1 \cup C_3) + v(S_2 \cup C_3). \quad (5)$$

In order to upper bound $v(S_1 \cup C_3)$ we again use the submodularity of v, together with a couple of facts about the marginal utilities of agents outside of S_1. Since the mechanism stopped after t iterations, $\max_{i\in D\setminus(S_1\cup S_2\cup R)} v(i|S_1) \leq 0$. Also, given that $R = R_{\mathbf{c}}$, for all agents that got rejected at some point, we know that they had very low marginal value per cost ratio with respect to both S_1 and S_2. In particular, if $i_k \in R$, then $c_{i_k} > \frac{\beta B}{x} v\big(i_k|S_j^{(k)}\big)$, for both $j\in\{1,2\}$. We may now rely on Definition 1*(iii)* to get

$$\begin{aligned} v(S_1 \cup C_3) &\leq v(S_1) + \sum_{i_k\in C_3} v(i_k|S_1) \\ &\leq v(S_1) + \sum_{i_k\in C_3\cap R} v(i_k|S_1) \qquad (\ v(i_k|S_1) \leq 0 \text{ for } i_k \in C_3\setminus R\ ) \end{aligned}$$

$$\leq v(S_1) + \sum_{i_k \in C_3 \cap R} v\left(i_k | S_1^{(k)}\right) \quad \left(\text{ by submodularity, } v(i_k|S_1) \leq v\left(i_k|S_1^{(k)}\right) \text{ for } i_k \in D\ \right)$$
$$\leq v(S_1) + \sum_{i_k \in C_3 \cap R} \frac{x}{\beta B} c_{i_k} \quad \left(\ \tfrac{\beta B}{x} v\big(i_k|S_1^{(k)}\big) < c_{i_k} \text{ for } i_k \in R\ \right)$$

Similarly, $v(S_2 \cup C_3) \leq v(S_2) + \sum_{i_k \in C_3 \cap R} \frac{x}{\beta B} c_{i_k}$. Also, recall that $\sum_{i \in C^*} c_i \leq B$ to get

$$v(S_j \cup C_3) \leq v(S_j) + \frac{x}{\beta}, \quad \text{for} \quad j \in \{1,2\}. \tag{6}$$

Finally, we may combine (4), (5) and (6) to get

$$\begin{aligned}\textsc{opt}(D,B) &\leq 2v(T_1) + 2v(T_2) + v(S_1 \cup C_3) + v(S_2 \cup C_3) \\ &\leq 2v(T_1) + 2v(T_2) + v(S_1) + v(S_2) + \frac{2x}{\beta} \leq 6 \cdot v(S) + \frac{2x}{\beta},\end{aligned}$$

or, equivalently, $v(S) \geq \frac{1}{6}\big(\textsc{opt}(D,B) - \frac{2x}{\beta}\big)$.

Combining Case 1 and Case 2, we obtain the claimed guarantee. ∎

So far, unless $x = \Theta(\textsc{opt}(D,B))$, the approximation guarantee seems to be rather weak. In fact, the way Simultaneous Greedy is used within Sample-then-Greedy requires that both $x = v(\textsc{alg}_1(A_1))$ and $\textsc{opt}(A_2,B)$ are $\Theta(\textsc{opt}(A,B))$. The next technical lemma guarantees that this happens with high probability, unless there is an extremely valuable agent; it follows from Lemma 2.1 of Bei et al. [13] or Lemmata 6.1 and 6.2 of Leonardi et al. [38].

Lemma 6 (Bei et al. [13], Leonardi et al. [38]). *Consider any submodular function $v(\cdot)$. For any given subset $T \subseteq A$ and a positive integer k assume that $v(T) \geq k \cdot \max_{i \in T} v(i)$. Further, suppose that T is divided uniformly at random into two subsets T_1 and T_2. Then with probability at least $\frac{1}{2}$, we have that $v(T_1) \geq \frac{k-1}{4k} v(T)$ and $v(T_2) \geq \frac{k-1}{4k} v(T)$.*

We are now ready to lower bound the approximation guarantee of Sample-then-Greedy. This is the first step where Lemma 2 is necessary. We set δ and η of Lemma 2 to be appropriate constants and, thus, $\textsc{alg}_1$ runs in polynomial time. For the sake of presentation we use $\delta = 2\varepsilon$ and $\eta = \frac{\xi}{e(e+\xi)}$ in the lemma below, where $\varepsilon = \xi = 10^{-4}$ as discussed later in Corollary 2.

Lemma 7. *Let $\varepsilon, \xi \in (0,1)$ and assume that for some positive integer k, $\textsc{opt}(A,B) > k \cdot \max_{i \in A} v(i)$. Then with probability at least $\frac{1}{2} - \varepsilon$, Sample-then-Greedy$(A, v, \mathbf{c}, B)$ outputs a set S such that*

$$v(S) \geq \min\left\{ \frac{\left\lfloor \frac{k-1}{4(e+\xi)\beta} \right\rfloor (k-1)}{(e+\xi)\left(\left\lfloor \frac{k-1}{4(e+\xi)\beta} \right\rfloor + 1\right)}, \frac{\beta(k-1) - 8k}{6} \right\} \cdot \frac{1}{4\beta k} \cdot \textsc{opt}(A,B).$$

Proof. We choose $\delta = 2\varepsilon$ and $\eta = \frac{\xi}{e(e+\xi)}$ in defining $\textsc{alg}_1$ as described in Lemma 2. Let C^* be an optimal solution for the given instance. The random partition of A into A_1 and A_2 induces a uniformly random partition of C^* into $A_1 \cap C^*$ and $A_2 \cap C^*$. As a result, Lemma 6 applies for $T = C^*$; thus, with probability at least $\frac{1}{2}$ it holds that $v(A_i \cap C^*) \geq \frac{k-1}{4k} v(C^*)$ for both $i \in \{1,2\}$. Independently, with probability at least $1 - 2\varepsilon$, we have that $v(\textsc{alg}_1(A_1)) \geq \big(\frac{1}{e} - \frac{\xi}{e(e+\xi)}\big) \cdot \textsc{opt}(A_1,B) = \frac{1}{e+\xi}\textsc{opt}(A_1,B)$. Therefore, with probability at least $\frac{1}{2}(1-2\varepsilon) = \frac{1}{2} - \varepsilon$ both these "good" events happen simultaneously. In what follows we assume that this is indeed the case. Thus,

$$\textsc{opt}(A,B) \geq x = v(\textsc{alg}_1(A_1)) \geq \frac{1}{e+\xi}\textsc{opt}(A_1,B) \geq \frac{k-1}{4(e+\xi)k}\textsc{opt}(A,B)$$

and also $\textsc{opt}(A_2, B) \geq \frac{k-1}{4k}\,\textsc{opt}(A, B)$.

The lower bound on x paired with the upper bound on $\max_{i\in A} v(i)$, imply that

$$\max_{i\in A} v(i) < \frac{1}{k}\cdot \textsc{opt}(A,B) \leq \frac{1}{k}\cdot\frac{4(e+\xi)k\beta}{k-1}\cdot\frac{x}{\beta} \leq \frac{1}{\left\lfloor\frac{k-1}{4(e+\xi)\beta}\right\rfloor}\cdot\frac{x}{\beta}\,.$$

Thus, we can use Lemma 5 with $D = A_2$, $x = v(\textsc{alg}_1(A_1))$ and $\ell = \left\lfloor\frac{k-1}{4(e+\xi)\beta}\right\rfloor$. Therefore, Simultaneous Greedy$(A_2, v, \mathbf{c}_{A_2}, B, x)$ outputs an S such that

$$\begin{aligned}
v(S) &\geq \min\left\{\frac{\left\lfloor\frac{k-1}{4(e+\xi)\beta}\right\rfloor x}{\left(\left\lfloor\frac{k-1}{4(e+\xi)\beta}\right\rfloor+1\right)\beta}, \frac{1}{6}\left(\textsc{opt}(A_2,B)-\frac{2x}{\beta}\right)\right\}\\
&\geq \min\left\{\frac{\left\lfloor\frac{k-1}{4(e+\xi)\beta}\right\rfloor (k-1)}{4(e+\xi)k\left(\left\lfloor\frac{k-1}{4(e+\xi)\beta}\right\rfloor+1\right)\beta}\,\textsc{opt}(A,B), \frac{1}{6}\left(\frac{k-1}{4k}\,\textsc{opt}(A,B)-\frac{2}{\beta}\,\textsc{opt}(A,B)\right)\right\}\\
&\geq \min\left\{\frac{\left\lfloor\frac{k-1}{4(e+\xi)\beta}\right\rfloor (k-1)}{(e+\xi)\left(\left\lfloor\frac{k-1}{4(e+\xi)\beta}\right\rfloor+1\right)}, \frac{\beta(k-1)-8k}{6}\right\}\cdot\frac{1}{4\beta k}\cdot\textsc{opt}(A,B)\,. \qquad\blacksquare
\end{aligned}$$

Corollary 2. *The set S returned by* GenSm-Main$(A, v, \mathbf{c}, B)$ *satisfies*

$$505\cdot\mathbb{E}(v(S)) \geq \textsc{opt}(A,B)\,.$$

Proof. Suppose that $\max_{i\in A} v(i) \geq \frac{1}{101}\cdot\textsc{opt}(A,B)$. Then, with probability p at least $1/101$ of the optimal value is returned. Hence,

$$\mathbb{E}(v(S)) \geq p\cdot\max_{i\in A} v(i) \geq \frac{0.2}{101}\cdot\textsc{opt}(A,B) = \frac{1}{505}\cdot\textsc{opt}(A,B)\,.$$

Next suppose that $\max_{i\in A} v(i) < \frac{1}{101}\cdot\textsc{opt}(A,B)$. We may apply Lemma 7 with $k = 101$ and $\varepsilon = \xi = 10^{-4}$. As discussed after the description of mechanism Simultaneous Greedy, the parameter β is equal to 9.185. This implies that $\left\lfloor\frac{k-1}{4e\beta}\right\rfloor = 1$. By substituting the values of ε, ξ, k and β to the bound of Lemma 7, we get that with probability at least $(1-p)(0.5-10^{-4})$

$$v(S) \geq \min\left\{\frac{50}{e+10^{-4}}, \frac{110.5}{6}\right\}\cdot\frac{1}{3710.74}\cdot\textsc{opt}(A,B) \geq \frac{1}{201.75}\cdot\textsc{opt}(A,B)\,,$$

and thus,

$$\mathbb{E}(v(S)) \geq (1-0.2)(0.5-10^{-4})\cdot\frac{1}{201.75}\cdot\textsc{opt}(A,B) \geq \frac{1}{505}\cdot\textsc{opt}(A,B)\,. \qquad\blacksquare$$

Notice that Corollaries 1 and 2 complete the proof of Theorem 1.

4. Online Procurement Note that the mechanism presented in the last section already bares some resemblance to online algorithms for variants of the *secretary problem* (although truthfulness is rarely a requirement there). Namely, a part of the input is only used to estimate the quality of the optimal solution and then, based on that estimation, some threshold is set for the remaining instance. On a high level, this is straightforward to adjust for the *random-arrival model*; we use the first (roughly) half of the stream of agents to find an estimate of $\textsc{opt}(A, B)$ and then set a threshold similar to the one in Simultaneous Greedy. However, there are a few issues one has to deal with.

First, Simultaneous Greedy goes through the agents in a specific order (in decreasing order of the maximum marginal value with respect to either one of the two constructed sets). Even though this fact is indeed used in the proof of Lemma 5, we show that even examining agents in arbitrary order works well, albeit with a somewhat worse approximation factor. Note that this is not true when there are other constraints on top of the budget-feasibility requirement, as in Section 5.

Second, towards the end, in line 9, SIMULTANEOUS GREEDY runs an unconstrained submodular maximization algorithm on S_1 and S_2 to possibly reveal a subset of them with much higher value. While this is a critical step, we rely on a very elegant result of Feige et al. [24]: a uniformly random set gives a 4-approximation for the unconstrained problem. Thus, every agent that passes the threshold and is added to S_j is only accepted to T_j with probability 1/2. The actual output of the mechanism is a random choice S between S_1, S_2, T_1 and T_2, made *before* the arrival of the first agent. So, while the four sets are built obliviously with respect to the choice of S, the agents added to S are irrevocably chosen while everyone else is irrevocably discarded.

One last issue is that we want the mechanism to occasionally return the single most valuable agent. This, however, is easily resolved by running *Dynkin's algorithm* [22] with constant probability instead. This mechanism samples the first n/e agents and then it picks the first agent i', among the remaining agents, who is at least as good as the best agent in the sample, i.e., $v(i') \geq \max_{k \leq n/e} v(i_k)$. This guarantees that $\mathbb{E}(v(i')) \geq \frac{1}{e} \max_{i \in A} v(i)$, where the expectation is over the order of the agents.

The mechanism GENSM-ONLINE below incorporates all these adjustments, yet maintains all the good properties of GENSM-MAIN. We assume a secretary setting, where the agents arrive *uniformly at random*. In particular, agents have no control over their arrival time, so this is still a single-parameter environment and truthfulness still means *universal truthfulness*, i.e., if we fix the random bits of the mechanism, then *for any arrival order* no agent has an incentive to lie. Moreover, note that GENSM-ONLINE is *order oblivious* [6], and thus, it does not fully exploit the randomness in the arrival of agents. Roughly, this means that *after* A_1 and A_2 are determined *but not yet observed*, i.e., right after randomly selecting ξ, an adversary is allowed to determine the order in which the mechanism observes the elements of A_1 and, separately, the order in which it observes the elements of A_2. A somewhat weaker, but intuitively more clear, interpretation is that the agents arrive uniformly at random up until the sampling phase is over, and after that the order is adversarial.

Again, ALG_1 is the e-approximation algorithm of Kulik et al. [36]. The parameter β is set to 8.725 and, like the parameter in SIMULTANEOUS GREEDY, is only relevant for the approximation factor.

THEOREM 2. GENSM-ONLINE *is a universally truthful, individually rational, budget-feasible online mechanism and achieves an* $O(1)$*-approximation in the random-arrival model.*

Proof. Fix any particular arrival order $i_1, i_2, \ldots, i_n$ of the agents.

By fixing the sequence ρ of the random bits of the mechanism, we get a deterministic allocation rule GENSM-ONLINE(ρ). In the case where this is Dynkin's algorithm, it is straightforward that—coupled with the threshold payment of B to the possible winner—it is truthful, individually rational and budget-feasible. Otherwise, i.e., if lines 4-16 are executed, the proof of monotonicity, and thus of truthfulness and individual rationality, of GENSM-ONLINE(ρ) is virtually identical to the proof of Lemma 3. Similarly, the budget-feasibility is proved exactly like the budget-feasibility of SIMULTANEOUS GREEDY in Lemma 4.

Since GENSM-ONLINE is a probability distribution over GENSM-ONLINE(ρ) for all possible ρ, we conclude that it is universally truthful, individually rational and budget-feasible. Also, given that Dynkin's algorithm and ALG_1 run in polynomial time and that the payments are easily determined, GENSM-ONLINE runs in polynomial time.

It remains to show that the solution returned by the mechanism is a constant approximation of the offline optimum. It is not hard to see that when the most valuable agent is comparable to the optimal solution, then Dynkin's algorithm suffices to guarantee an overall good performance. In particular, suppose that $\max_{i \in A} v(i) \geq \frac{1}{250} \cdot \text{OPT}(A, B)$. Then, with probability q at least $1/e$ of the $1/250$ of

GenSm-Online$(A, v, \mathbf{c}, B)$

1 *With probability* $q = 0.4$ **:**
2 Run Dynkin's algorithm and **return** the winner
3 *With probability* $1 - q$ **:**
4 $S_1 = S_2 = T_1 = T_2 = \emptyset$; $B_1 = B_2 = B$
5 $S = \begin{cases} S_j\,, & \text{with probability } 1/10, \text{ for each } j \in \{1,2\} \\ T_j\,, & \text{with probability } 2/5, \text{ for each } j \in \{1,2\} \end{cases}$
6 Draw ξ from the binomial distribution $\mathcal{B}(n, 0.5)$
7 Let A_1 be the set of the first ξ agents, and $A_2 = A \setminus A_1$
8 Reject all the agents in A_1 and calculate $x = v(\textsc{alg}_1(A_1))$
9 **for** *each* $i \in A_2$ *as he arrives* **do**
10 Let $\hat{\jmath} \in \arg\max_{j \in \{1,2\}} v(i|S_j)$
11 **if** $c_i \le \frac{\beta B}{x} v(i|S_{\hat{\jmath}}) \le B_{\hat{\jmath}}$ **then**
12 $S_{\hat{\jmath}} = S_{\hat{\jmath}} \cup \{i\}$
13 $B_{\hat{\jmath}} = B_{\hat{\jmath}} - \frac{\beta B}{x} v(i|S_{\hat{\jmath}})$
14 With probability $1/2$, $T_{\hat{\jmath}} = T_{\hat{\jmath}} \cup \{i\}$ (otherwise, $T_{\hat{\jmath}}$ remains the same)
15 Update S `/* the update is consistent to the choice made in line 5 */`
16 **return** S
17 Pay the agents according to Myerson's lemma

the optimal value is returned in expectation (with respect to the arrival order). Hence, if X is the (possibly empty) set returned by GenSm-Online

$$\mathbb{E}(v(X)) \ge \frac{q}{e} \cdot \max_{i \in A} v(i) \ge \frac{q}{e} \cdot \frac{1}{250} \cdot \textsc{opt}(A, B) \ge \frac{1}{1710} \cdot \textsc{opt}(A, B)\,.$$

For the case where $\max_{i \in A} v(i) < \frac{1}{250} \cdot \textsc{opt}(A, B)$, we are going to prove the analog of Lemma 7. First, notice that randomly ordering the elements of A and then picking the first ξ, where ξ follows the binomial distribution $\mathcal{B}(n, 0.5)$, is equivalent to just picking each element of A with probability $1/2$. This simple observation is crucial, because it allows to still use Lemma 6. So, assume it is the case that $\textsc{opt}(A_i, B) \ge \frac{k-1}{4k} \textsc{opt}(A, B)$ for $i \in \{1, 2\}$, where $k = 250$. Unless otherwise stated, all expectations below are conditioned on this fact. Recall that this happens with probability at least $1/2$ as discussed in the beginning of the proof of Lemma 7.

We will follow a similar case analysis as in the proof of Lemma 5, depending on whether the set R_B, defined below, is empty or not. Similarly to the notation used in Section 3, let $i_1, i_2, \ldots, i_{n-\xi}$ be the agents of A_2 ordered according to their arrival. Also, let $S_1^{(k)}$, $B_1^{(k)}$, $S_2^{(k)}$, $B_2^{(k)}$ denote S_1, B_1, S_2, B_2, respectively, at the time i_k arrives. We will use S_1 and S_2 exclusively for their final versions. Let $R = A_2 \setminus (S_1 \cup S_2)$ be the agents i_k that were rejected, i.e., not added to either $S_1^{(k)}$ or $S_2^{(k)}$. We again partition R depending on why the agents where rejected, i.e., $R_{\mathbf{c}}$ (resp. R_B) contains everyone rejected because the first (resp. the second) inequality in line 11 was violated.

Case 1. Assume that $R_B \ne \emptyset$ and let $i_k \in R_B$. If j_k is the value of $\hat{\jmath}$ chosen in line 10, then $\frac{\beta B}{x} v\left(i_k | S_{j_k}^{(k)}\right) > B_{j_k}^{(k)}$. Using the exact same argument leading to (1) (see proof of Lemma 5), we get

$$v\left(S_{j_k}^{(k)}\right) \ge \frac{x}{\beta} - v\left(i_k | S_{j_k}^{(k)}\right) \ge \frac{x}{\beta} - \max_{i \in A} v(i)\,.$$

Given the known lower bound on x and upper bound on $\max_{i\in A} v(i)$, this leads to

$$v(S_{j_k}) \geq \left(\frac{k-1}{4ek\beta} - \frac{1}{k}\right) \cdot \textsc{opt}(A,B)\,. \tag{7}$$

Before we lower bound $\mathbb{E}(v(S))$, it not hard to see that $\mathbb{E}(v(T_j)) \geq \frac{1}{2}v(S_j)$, where the expectation is over the random choices made in line 14. In fact, this is a direct corollary of the non-negativity of v and the following well-known probabilistic property of submodular functions.

Lemma 8 (Feige et al. [24]). *Let $g: 2^X \to \mathbb{R}$ be submodular. Denote by $A[p]$ a random subset of A where each element appears with probability p. Then $\mathbb{E}(g(A[p])) \geq (1-p)\cdot g(\emptyset) + p\cdot g(A)$.*

By taking the expectation of $v(S)$ over the random choices made in lines 14 and 5, we get

$$\begin{aligned}
\mathbb{E}(v(S)) &= \frac{1}{10}\cdot v(S_1) + \frac{1}{10}\cdot v(S_2) + \frac{2}{5}\cdot \mathbb{E}(v(T_1)) + \frac{2}{5}\cdot \mathbb{E}(v(T_2)) \\
&\geq \frac{1}{10}\cdot v(S_{j_k}) + \frac{2}{5}\cdot\frac{1}{2}\cdot v(S_{j_k}) \\
&\geq \frac{3}{10}\cdot\left(\frac{k-1}{4ek\beta} - \frac{1}{k}\right)\cdot \textsc{opt}(A,B) \\
&\geq \frac{1}{512}\cdot \textsc{opt}(A,B)\,.
\end{aligned} \tag{8}$$

Case 2. Assume that $R_B = \emptyset$. Let C^* be an optimal solution for the instance $(A_2, v, \mathbf{c}_{A_2}, B)$ and $C_1 = C^* \cap S_1$, $C_2 = C^* \cap S_2$, $C_3 = C^* \setminus (C_1 \cup C_2)$. Recall inequality (3) (see proof of Lemma 5):

$$\textsc{opt}(A_2,B) = v(C^*) \leq v(C_1) + v(C_2) + v(C_3)\,. \tag{3}$$

To upper bound the value of C_1 and C_2 we need the following result by Feige et al. [24].

Theorem 3 (Feige et al. [24]). *Let $v: 2^A \to \mathbb{R}_{\geq 0}$ be a submodular function and let T denote a random subset of A, where each element is sampled independently with probability $1/2$. Then $\mathbb{E}(v(T)) \geq \frac{1}{4}\,\textsc{opt}(A,\infty)$.*

By the definition of T_1, T_2 and Theorem 3, we get

$$v(C_j) \leq \textsc{opt}(S_j, B) = \textsc{opt}(S_j,\infty) \leq 4\cdot \mathbb{E}(v(T_j))\,, \quad \text{for} \quad j\in\{1,2\}\,. \tag{9}$$

For upper bounding $v(C_3)$ recall inequality (5) (see proof of Lemma 5):

$$v(C_3) \leq v(S_1 \cup C_3) + v(S_2 \cup C_3)\,. \tag{5}$$

Using the same arguments leading to (6) (see proof of Lemma 5), we get

$$v(S_j \cup C_3) \leq v(S_j) + \frac{x}{\beta}\,, \quad \text{for} \quad j\in\{1,2\}\,. \tag{10}$$

We may now combine (3), (9), (5) and (10). Note that $\mathbb{E}(v(T_j)), j\in\{1,2\}$, below is over the random choices in line 14, while $\mathbb{E}(v(S_j)), j\in\{1,2\}$, is over the random choices in both line 14 and line 5.

$$\begin{aligned}
\frac{k-1}{4k}\,\textsc{opt}(A,B) &\leq \textsc{opt}(A_2,B) \\
&\leq 4\cdot\mathbb{E}(v(T_1)) + 4\cdot\mathbb{E}(v(T_2)) + v(S_1) + v(S_2) + \frac{2x}{\beta}
\end{aligned}$$

$$= 10 \cdot \mathbb{E}(v(S)) + \frac{2x}{\beta}$$
$$\leq 10 \cdot \mathbb{E}(v(S)) + \frac{2}{\beta} \cdot \textsc{opt}(A, B)\,,$$

or, equivalently,

$$\mathbb{E}(v(S)) \geq \frac{1}{10}\left(\frac{k-1}{4k} - \frac{2}{\beta}\right) \cdot \textsc{opt}(A, B)$$
$$\geq \frac{1}{513} \cdot \textsc{opt}(A, B)\,. \tag{11}$$

Therefore, given that both A_1 and A_2 contain a good fraction of the optimal budget-feasible solution, the expectation of $v(S)$ is always at least $\frac{1}{513} \cdot \textsc{opt}(A, B)$. Coupled with Lemma 6, this means that the unconditional expectation of $v(S)$ is at least $\frac{1}{2} \cdot \frac{1}{513} \cdot \textsc{opt}(A, B)$.

Hence, if X is the set returned by GenSm-Online, by the law of total expectation, we have

$$\mathbb{E}(v(X)) \geq (1-p) \cdot \frac{1}{2} \cdot \frac{1}{513} \cdot \textsc{opt}(A, B) = \frac{1}{1710} \cdot \textsc{opt}(A, B)\,.$$

We conclude that GenSm-Online achieves, in expectation, an 1710-approximation. ∎

One immediate consequence of Theorem 2 is the existence of an $O(1)$-approximation algorithm for the non-monotone *Submodular Knapsack Secretary Problem* (SKS). Bateni et al. [11] proposed an $O(1)$-approximation algorithm for the SKS. While they give a proof for the monotone case, the non-trivial details of extending their analysis to the non-monotone case are omitted. Thus, we think that Corollary 3, which provides a complete proof of the existence of an $O(1)$-approximation algorithm for the non-monotone SKS, is of independent interest.

Formally, an instance of SKS consists of a ground set $A = [n]$, a non-negative submodular objective $v : A \to \mathbb{R}_+$ and a given budget B. The elements of A arrive in a uniformly random order and each element must be accepted or rejected immediately upon arrival. An algorithm for SKS has access to $n = |A|$, to the costs of elements that have arrived (i.e., each cost is revealed upon arrival) and to a value oracle that, given a subset $S \subseteq A$ of elements that have already arrived, returns $v(S)$. The objective is to accept a set of elements maximizing v without exceeding the budget.

It is straightforward to see that the only difference of SKS with the online procurement problem studied in this section is the information about the costs. In SKS there is no notion of misreporting a cost and thus it can be seen as a special case of our online problem where agents are guaranteed to always reveal their true costs.

Corollary 3. *There is an $O(1)$-approximation algorithm for the non-monotone SKS.*

5. Adding Combinatorial Constraints To illustrate the applicability of our approach, we turn to the case where the solution has to satisfy some additional combinatorial constraint. With the exception of additive valuation functions [1, 39], even for *monotone* submodular objectives no polynomial-time mechanisms using only value queries are known. Here we show that the general approach of GenSm-Main can be utilized to achieve an $O(p)$-approximation when the set of feasible solutions—even before taking budget feasibility into consideration—forms a p-system, i.e., an independence system with *rank quotient* at most p. In particular, as stated in Corollary 4, this implies constant factor approximation for cardinality, matroid and bipartite matching constraints. As it is shown in Section 6, going beyond independence systems is hindered by strong impossibility results. For an example that makes this distinction more concrete, suppose that a given instance had a graph representation. Requiring that the solution forms a *spanning forest* is an example of a matroid constraint and admits a constant approximation (Corollary 4), while requiring that the solution forms a *spanning tree* instead does not admit any bounded approximation (Theorem 8).

Definition 4. An *independence system* or a *downward-closed system* is a pair $(U,\mathcal{I})$, where U is a finite set and $\mathcal{I} \subseteq 2^U$ is a family of subsets, whose members are called the *independent sets* of U and satisfy:

(i) $\emptyset \in \mathcal{I}$, and

(ii) if $B \in \mathcal{I}$ and $A \subseteq B$, then $A \in \mathcal{I}$.

Given a set $S \subseteq U$, a maximal independent set contained in S is called a *basis* of S. The *upper rank* $\mathrm{ur}(S)$ (resp. the *lower rank* $\mathrm{lr}(S)$) is defined as the cardinality of a largest (resp. smallest) basis of S. A *p-system* $(U,\mathcal{I})$ is an independence system such that $\max_{S\subseteq U} \frac{\mathrm{ur}(S)}{\mathrm{lr}(S)} \le p$.

Special cases of p-systems. A number of well-studied combinatorial constraints are special cases of p-systems for small values of p. A *cardinality* constraint requires that the solution contains at most k agents for a given $k \in \mathbb{N}$. It is easy to see that a cardinality constraint induces a 1-system, as all maximal independent sets in this case have size exactly $\min\{k,n\}$. A *matroid* constraint requires that the solution belongs to a given matroid. A matroid is an independence system that also has the *exchange property*: if $A,B \in I$ and $|A| < |B|$, then there exists $x \in B \setminus A$ such that $A \cup \{x\} \in I$. The exchange property ensures that all maximal independent sets have the same size, and thus a matroid is a 1-system; a cardinality constraint is a special case of a matroid constraint. A *bipartite matching* constraint requires that the solution is a matching in a bipartite graph representation of the instance (where agents are edges). It is not hard to see that a bipartite matching constraint induces a 2-system, as the sizes of any two maximal matchings in a bipartite graph are always within a factor of 2 of each other. In fact, a bipartite matching is an example of an intersection of two matroids [44]. More generally, the constraint imposed by the *intersection of k matroids* is a k-system.

For the sake of readability, we present the case of *monotone* submodular objectives here; the non-monotone case is deferred to Appendix 7. A technical highlight of our analysis, later used for the non-monotone case as well, is Claim 1. The claim crucially depends on the order in which we consider the agents, in order to bound the value lost because of the p-system constraint.

As usual, we assume the existence of an independence oracle. In particular, when we write that $\mathcal{I}$ is part of the input of the mechanism, we mean that the mechanism has access to a membership oracle for $\mathcal{I}$. The parameter β is later set to 13/3. $\textsc{alg}_3$ in line 5 can be any polynomial time approximation algorithm for monotone submodular maximization subject to a knapsack and a p-system constraint. Here we assume the $(p+3)$-approximation algorithm of Badanidiyuru and Vondrák [9].

Theorem 4. *Assuming that the solution has to be an independent set of a p-system, there is a universally truthful, individually rational, budget-feasible, $O(p)$-approximation mechanism that runs in polynomial time for (non-monotone) submodular objectives.*

Proof. The proof of the theorem for the non-monotone case is deferred to Appendix 7. Here we prove that MonSm-Constrained below has all the stated properties for monotone submodular objectives. First, we observe that S starts as an independent set, namely the empty set, and it is expanded only if it remains an independent set. Hence, at the end MonSm-Constrained does return a feasible solution, i.e., S is in $\mathcal{I}$.

At this point, following the same reasoning used for GenSm-Main and GenSm-Online, it should be easy to see that MonSm-Constrained is universally truthful, individually rational, budget-feasible, and runs in polynomial time.

Next we show that the solution returned by the mechanism is an $O(p)$-approximation of the optimum. First, suppose that $\max_{i\in A} v(i) \ge \textsc{opt}(A,B)/(26(p+10))$. Then, for the set S returned by MonSm-Constrained, we have $\mathbb{E}(v(S)) \ge q \cdot \max_{i\in A} v(i) \ge \frac{1}{5} \cdot \textsc{opt}(A,B)/(26(p+10)) \ge \textsc{opt}(A,B)/(138(p+10))$.

MonSm-Constrained$(A, \mathcal{I}, v, \mathbf{c}, B)$

With probability $q = 0.2$ **:**
  return $i^* \in \arg\max_{i \in A} v(i)$
With probability $1 - q$ **:**
  Put each agent of A in either A_1 or A_2 independently at random with probability $\frac{1}{2}$
  $x = v(\text{ALG}_3(A_1))$ /* a $(p+3)$-approximation of $\text{OPT}(A_1, v, \mathbf{c}_{A_1}, B)$ */
  $S = \emptyset$; $B_R = B$; $U = A_2$
  while $U \neq \emptyset$ **do**
    Let $\hat{\imath} \in \arg\max_{i \in U} v(i|S)$
    if $c_{\hat{\imath}} \leq \frac{\beta B}{x} v(\hat{\imath}|S) \leq B_R$ **and** $S \cup \{\hat{\imath}\} \in \mathcal{I}$ **then**
      $S = S \cup \{\hat{\imath}\}$
      $B_R = B_R - \frac{\beta B}{x} v(\hat{\imath}|S)$
    $U = U \setminus \{\hat{\imath}\}$
  return S
Pay the agents according to Myerson's lemma

For the case where $\max_{i \in A} v(i) < \text{OPT}(A, B)/(26(p+10))$, we follow the same notation and the same high level approach as with the approximation guarantees of GenSm-Main and GenSm-Online. So, $i_1, i_2, \ldots, i_{|A_2|}$ are the agents of A_2 in the order considered by the mechanism. By $S^{(k)}$ and $B_R^{(k)}$ we denote S and B_R, respectively, at the time i_k arrives, and we only use S for the final set returned. The set $R = A_2 \setminus S$ contains the agents i_k that were not added to $S^{(k)}$ and it is further partitioned to

$$R_{\mathbf{c}} = \left\{ i_k \,\middle|\, \frac{\beta B}{x} v\Big(i_k \,|\, S^{(k)}\Big) < c_{\hat{\imath}} \right\}, \quad R_B = \left\{ i_k \,\middle|\, B_R^{(k)} < \frac{\beta B}{x} v\Big(i_k \,|\, S^{(k)}\Big) \right\} \quad \text{and} \quad R_{\mathcal{I}} = R \setminus (R_{\mathbf{c}} \cup R_B).$$

Assume that $\text{OPT}(A_i, B) \geq \frac{k-1}{4k} \text{OPT}(A, B)$ for $i \in \{1, 2\}$, where $k = 26(p+10)$. Thus, $x = v(\text{ALG}_1(A_1)) \geq \frac{k-1}{4(p+3)k} \text{OPT}(A, B)$. Recall that this does happen with probability at least $\frac{1}{2}$, as discussed in the beginning of the proof of Lemma 7.

Case 1. Assume that $R_B \neq \emptyset$. Let $i_k \in R_B$, i.e., $\frac{\beta B}{x} v\left(i_k | S^{(k)}\right) > B_R^{(k)}$. Using the same argument as in the proof of Lemma 5, we get $v\big(S^{(k)}\big) \geq \frac{x}{\beta} - \max_{i \in A} v(i)$ and, given the known bounds on x and $\max_{i \in A} v(i)$, this leads to $v(S) \geq \left(\frac{k-1}{4(p+3)k\beta} - \frac{1}{k}\right) \cdot \text{OPT}(A, B)$.

By substituting $k = 26(p+10)$ and $\beta = \frac{13}{3}$, it is a matter of simple calculations to get

$$v(S) \geq \frac{5}{276(p+10)} \cdot \text{OPT}(A, B). \tag{12}$$

Case 2. Assume that $R_B = \emptyset$ and let C^* be an optimal solution for the instance $(A_2, v, \mathbf{c}_{A_2}, B)$. By monotonicity, we have

$$\text{OPT}(A_2, B) = v(C^*) \leq v(S \cup C^*). \tag{13}$$

Because of the p-system constraint, however, deriving the analog of inequality (6) needs some extra work. By Definition 1*(iii)*, we have

$$v(S \cup C^*) \leq v(S) + \sum_{i_k \in C^* \setminus S} v(i_k|S) \leq v(S) + \sum_{i_k \in C^* \cap R_{\mathbf{c}}} v(i_k|S) + \sum_{i_k \in C^* \cap R_{\mathcal{I}}} v(i_k|S). \tag{14}$$

We may upper bound the first sum using the fact that all agents involved got rejected because they had very low marginal value per cost ratio. That is,

$$\sum_{i_k \in C^* \cap R_{\mathbf{c}}} v(i_k|S) \le \sum_{i_k \in C^* \cap R_{\mathbf{c}}} v\left(i_k|S^{(k)}\right) < \frac{x}{\beta B} \sum_{i_k \in C^* \cap R_{\mathbf{c}}} c_{i_k} \le \frac{x}{\beta} \le \frac{\textsc{opt}(A,B)}{\beta}. \tag{15}$$

For the second sum we prove the following result that crucially relies on the fact that agents are examined in decreasing marginal value.

Claim 1. $\sum_{i_k \in C^* \cap R_{\mathcal{I}}} v(i_k|S) \le p \cdot v(S)$.

Proof of Claim 1 Recall that when we index agents we follow the ordering imposed by the mechanism, i.e., i_k is always the agent picked at the kth execution of line 8 of MonSm-Constrained.

Suppose that there is a mapping $f : C^* \cap R_{\mathcal{I}} \to S$ such that

(i) if $f(i_k) = i_\ell$, then $v\left(i_k|S^{(k)}\right) \le v\left(i_\ell|S^{(\ell)}\right)$ for all $i_k \in C^* \cap R_{\mathcal{I}}$, and

(ii) $|f^{-1}(i_\ell)| \le p$ for all $i_\ell \in S$.

We slightly abuse the notation and write $S^{f(i_k)}$ instead of $S^{(\ell)}$ when $f(i_k) = i_\ell$. The existence of f implies that

$$\sum_{i_k \in C^* \cap R_{\mathcal{I}}} v(i_k|S) \le \sum_{i_k \in C^* \cap R_{\mathcal{I}}} v\left(i_k|S^{(k)}\right) \le \sum_{i_k \in C^* \cap R_{\mathcal{I}}} v\left(f(i_k)\,|\,S^{f(i_k)}\right) \le p \cdot \sum_{i_\ell \in S} v\left(i_\ell|S^{(\ell)}\right) = p \cdot v(S).$$

The first inequality follows from the submodularity of v, while the second and third inequalities follow from *(i)* and *(ii)*, respectively.

Next, we are going to construct such an f. Let $S = \{i_{a_1}, i_{a_2}, \ldots, i_{a_s}\}$ and $C^* \cap R_{\mathcal{I}} = \{i_{b_1}, i_{b_2}, \ldots, i_{b_t}\}$, where both $(a_i)_{i=1}^s$ and $(b_i)_{i=1}^t$ are subsequences of $1, 2, \ldots, |A_2|$. We are going to map the first p elements of $C^* \cap R_{\mathcal{I}}$, $i_{b_1}, \ldots, i_{b_p}$, to i_{a_1}, the next p elements $i_{b_{p+1}}, \ldots, i_{b_{2p}}$, to i_{a_2}, and so on. That is, $f(i_{b_j}) = i_{a_{\lceil j/p \rceil}}$.

It is straightforward that f satisfies property *(ii)*. In order to prove property *(i)*, it suffices to show that for all $j \in \{1, 2, \ldots, t\}$, agent i_{b_j} is considered by MonSm-Constrained after agent $f(i_{b_j})$. Indeed, if that was the case, by the definition of $\hat{\imath}$ in line 8 and submodularity, we would get

$$v\left(i_{a_{\lceil j/p \rceil}}\,|\,S^{(a_{\lceil j/p \rceil})}\right) \ge v\left(i_{b_j}\,|\,S^{(a_{\lceil j/p \rceil})}\right) \ge v\left(i_{b_j}\,|\,S^{(b_j)}\right),$$

for all $i_{b_j} \in C^* \cap R_{\mathcal{I}}$, as desired. Suppose, towards a contradiction, that there is some $k \in \{1, 2, \ldots, t\}$, such that $b_k < a_{\lceil k/p \rceil}$; in fact, suppose k is the smallest such index. Consider the sets $T = \{i_{a_1}, i_{a_2}, \ldots, i_{a_{\lceil k/p \rceil - 1}}\} \subseteq S$ and $Q = \{i_{b_1}, i_{b_2}, \ldots, i_{b_k}\} \subseteq C^* \cap R_{\mathcal{I}}$. By construction, $T \in \mathcal{I}$. Moreover, we claim that T is maximally independent in $T \cup Q$. Indeed, each $i_{b_\tau} \in Q$ was rejected because $S^{(b_\tau)} \cup \{i_{b_\tau}\} \notin \mathcal{I}$, and since $S^{(b_\tau)} \subseteq T$ we get $T \cup \{i_{b_\tau}\} \notin \mathcal{I}$. This implies that $\mathrm{lr}(T \cup Q) \le |T|$. On the other hand, $Q \in \mathcal{I}$ because $Q \subseteq C^* \in \mathcal{I}$. As a result $\mathrm{ur}(T \cup Q) \ge |Q|$. However, notice that

$$p \cdot |T| = p\left(\lceil k/p \rceil - 1\right) < p\left(k/p + 1 - 1\right) = k = |Q|.$$

Thus, $\frac{\mathrm{ur}(T \cup Q)}{\mathrm{lr}(T \cup Q)} \ge \frac{|Q|}{|T|} > p$, contradicting the fact that $(A, \mathcal{I})$ is a p-system. We conclude that f satisfies both *(i)* and *(ii)*, and therefore, $\sum_{i_k \in C^* \cap R_{\mathcal{I}}} v(i_k|S) \le p \cdot v(S)$. ◁

Now, combining (13), (14), (15), and Claim 1, we have $\textsc{opt}(A_2, B) \le (p+1) \cdot v(S) + \frac{\textsc{opt}(A,B)}{\beta}$, and using the lower bound on $\textsc{opt}(A_2, B)$, $v(S) \ge \frac{1}{p+1} \cdot \left(\frac{k-1}{4k} - \frac{1}{\beta}\right) \textsc{opt}(A, B)$. Again, by substituting k and β, it is a matter of calculations to get

$$v(S) \ge \frac{5}{276(p+10)} \cdot \textsc{opt}(A, B). \tag{16}$$

By Lemma 6, both (12) and (16) hold with probability at least 1/2. Hence,

$$\mathbb{E}(v(S)) \geq (1-q)\cdot\frac{1}{2}\cdot\frac{5}{276(p+10)}\cdot\textsc{opt}(A,B) = \frac{1}{138(p+10)}\cdot\textsc{opt}(A,B)\,. \qquad \blacksquare$$

As we already mentioned, for matroid constraints we have $p=1$ and for bipartite matching constraints $p=2$. Since cardinality constraints are a special case of matroid constraint, we directly get the following.

Corollary 4. *For cardinality, matroid and bipartite matching constraints, there is a universally truthful, budget-feasible $O(1)$-approximation mechanism for (non-monotone) submodular objectives.*

6. Lower Bounds In the value query model there is a strong lower bound on the number of queries for deterministic algorithms for monotone XOS objectives due to Singer [45]. This result is based on a lower bound of Mirrokni et al. [40] on welfare maximization in combinatorial auctions. As the latter also holds for randomized algorithms, so does Singer's result as well, essentially with the same proof. We restate it here for completeness. Note that it holds even when the costs are public knowledge.

Theorem 5 (Singer [45]). *For any fixed $\varepsilon > 0$, any (randomized) $n^{\frac{1}{2}-\varepsilon}$-approximation algorithm for monotone XOS function maximization subject to a budget constraint requires exponentially many value queries (in expectation).*

When one moves to non-monotone objectives, as it is the case in this work, it is possible to prove even stronger lower bounds. Below we show that for general XOS objectives, exponentially many value queries are needed for *any* non-trivial approximation even without the budget constraint. As this result applies to the purely algorithmic setting, it is of independent interest.

It is known that in many settings there is a separation between the power of value and demand queries of polynomial size, see, e.g., [14]. To stress this difference in our setting, recall that in the demand query model, the class of XOS objectives admits a truthful $O(1)$-approximation mechanism with a polynomial number of queries.

Theorem 6. *For any fixed $\varepsilon > 0$, any (randomized) $n^{1-\varepsilon}$-approximation algorithm for XOS function maximization requires exponentially many value queries (in expectation).*

Proof. We follow, on a high level, the approach in [40]. Recall that $A=[n]$ and choose a set R of size $|R| = \rho = n/4$ uniformly at random amongst all the subsets of A of size ρ. We are going to construct two XOS functions, v_1 and v_2, that are hard to tell apart, i.e., to distinguish between them with constant probability, an exponential number of value queries will be required.

For any $T \subseteq A$, let α_T be the additive function that assigns the value 1 to each $i \in T$ and the value 0 to each $i \notin T$. For $\tau = n^{\varepsilon/2}/4$, we define v_1 as the maximum over all such additive functions on sets of size τ:

$$v_1(S) = \max_{T\subseteq A:|T|=\tau} \alpha_T(S), \quad \text{for all } S \subseteq A\,.$$

Further, let β be the additive function that assigns the value 1 to each $i \in R$ and the value $-\rho$ to each $i \notin R$. We define v_2 as the maximum between v_1 and β:

$$v_2(S) = \max\{v_1(S), \beta(S)\}\,, \quad \text{for all } S \subseteq A\,.$$

Clearly, both v_1 and v_2 are XOS functions since each of them is defined as the maximum of a finite number of additive functions. Also notice that for any $S \nsubseteq R$ we have $v_2(S) = v_1(S)$. However, $\textsc{opt}(A, v_1, \infty) = \tau$ and $\textsc{opt}(A, v_2, \infty) = \rho = n^{1-\varepsilon/2}\cdot\tau > n^{1-\varepsilon}\cdot\tau$. Hence, any (possibly randomized) algorithm that achieves an approximation ratio smaller or equal to $n^{1-\varepsilon}$ can distinguish between the two functions.

Consider a value query for some set S. This query can distinguish between v_1 and v_2 if and only if $S \subseteq R$ and $|S| > \tau$, and otherwise it will reveal no information about R. We will call such an S a *distinguishing set*. For a given S with $|S| > \tau$, the probability that $S \subseteq R$, over the random choice of R, is

$$\frac{\binom{\rho}{|S|}}{\binom{n}{|S|}} \leq \frac{\left(\frac{e\rho}{|S|}\right)^{|S|}}{\left(\frac{n}{|S|}\right)^{|S|}} = \left(\frac{e}{4}\right)^{|S|} < \left(\frac{e}{4}\right)^{\frac{n^{\varepsilon/2}}{4}}, \tag{17}$$

using the well-known fact that for $1 \leq k \leq m$ we have $\left(\frac{m}{k}\right)^k \leq \binom{m}{k} \leq \left(\frac{em}{k}\right)^k$.

Now, let $q(\cdot)$ be a polynomial and $p \in (0,1]$ be a constant. Suppose first that there is a deterministic algorithm that asks queries $S_1, S_2, \ldots, S_{q(n)}$ and distinguishes between v_1 and v_2 with probability at least p. Note that the choice for S_j can depend on all previous queries $S_1, \ldots, S_{j-1}$ as well as the answers of the value query oracle obtained for those sets. Also, the choices made by the algorithm are the same for all non-distinguishing queries regardless of whether we present v_1 or v_2 to the algorithm. Using a union bound, it then follows that the probability that we distinguish between v_1 and v_2 is at most

$$\sum_{i=1}^{q(n)} \frac{\binom{\rho}{|S_i|}}{\binom{n}{|S_i|}} < q(n)\left(\frac{e}{4}\right)^{\frac{n^{\varepsilon/2}}{4}} = o(1).$$

which contradicts p being constant. In case of a randomized algorithm, we can condition on the random bits of the algorithm. Averaging over the choices of the random bits, we are still only able to distinguish between v_1 and v_2 with exponentially small probability. ∎

One immediate consequence of Theorem 6 is that when we care for constant approximation ratios, the result of Theorem 1 is (asymptotically) the *best possible* for budget-feasible mechanism design. General submodular objectives is the broadest class of well studied non-monotone functions one could hope for, even for randomized mechanisms.

6.1. Combinatorial Constraints We now turn to the problem of maximizing subject to additional constraints on top of the budget constraint. To further motivate our restriction to p-system constraints, we restate here a lower bound of Badanidiyuru and Vondrák [9]: for independence system constraints one cannot achieve an approximation factor better than $\max_{S\subseteq U} \frac{\mathrm{ur}(S)}{\mathrm{lr}(S)}$ with a polynomial number of queries. Thus, the result of Theorem 4 is asymptotically optimal.

Theorem 7 (Badanidiyuru and Vondrák [9]). *For any fixed $\varepsilon > 0$, any (randomized) $(p+\varepsilon)$-approximation algorithm for additive function maximization subject to p-system constraints requires exponentially many independence oracle queries (in expectation).*

As we mentioned in the beginning of Section 5, we cannot really go beyond independence systems and have any non-trivial approximation guarantee in polynomial time. This is illustrated in Theorem 8 and Corollary 5 below. Theorem 8 generalizes Singer's [45] strong impossibility result for deterministically "hiring a team of agents" to *any* constraint that is not downward-closed below. Note that it holds even for super-constant approximation ratios, even for the special case of additive objectives, irrespectively of any complexity assumptions.

Theorem 8. *Let $\mathcal{F} \subseteq 2^A$ be any collection of feasible sets that is not downward-closed. Then there is no deterministic, truthful, individually rational, budget-feasible mechanism achieving a bounded approximation when restricted on $\mathcal{F}$, even for additive objectives.*

Proof. Since $\mathcal{F}$ is not downward-closed, there is some $F \in \mathcal{F}$ with $|F| \geq 2$ which is minimally feasible, i.e., if $S \subseteq F$ and $S \in \mathcal{F}$, then $S = F$.

Towards a contradiction, suppose that there is a deterministic, truthful, budget-feasible, α-approximation mechanism ALG for additive objectives, where $\alpha = \alpha(n) > 1$. Consider the following instance on A where v is additive: for each agent $i \in F$, $v(i) = 1/|F|, c_i = \varepsilon \ll B/|F|$, while for each agent $i \in A \setminus F$, $v(i) = \delta < 1/\alpha, c_i = B$. All the $\mathcal{F}$-feasible and budget-feasible solutions are F and, possibly, some of the singletons outside of F. If ALG returns any solution other than F, then $v(\text{ALG}(A, v, \mathbf{c}, B)) \leq \delta < \frac{1}{\alpha} = \frac{1}{\alpha} \cdot \text{OPT}(A, B)$, which contradicts the approximation guarantee of ALG. So, ALG should return F.

However, the latter is true even if we slightly modify the instance, so that for a specific $j \in F$, $c_j = B - (|F| - 1) \cdot \varepsilon$. Therefore, in the original instance, the threshold payment for j is at least $B - (|F| - 1) \cdot \varepsilon$. In fact, due to symmetry, all the threshold payments in the original instance should be at least $B - (|F| - 1) \cdot \varepsilon$. Since $|F| \geq 2$ and $B - (|F| - 1) \cdot \varepsilon \approx B$, this contradicts the budget-feasibility of ALG. ∎

The next corollary of Theorem 7 states that under general combinatorial constraints it is not possible to achieve any non-trivial approximation with polynomially many queries. While it is not hard to prove it directly, given Theorem 7 it suffices to notice that such a lower bound holds even for general independence systems. Indeed, there are cases where $\frac{\text{ur}(U)}{\text{lr}(U)}$ is $\Theta(n)$ like the $(n-1)$-systems of independent sets of star graphs.

COROLLARY 5. *For any fixed $\varepsilon > 0$, any (randomized) $n^{1-\varepsilon}$-approximation algorithm for additive function maximization subject to general feasibility constraints requires exponentially many queries (in expectation).*

7. Discussion

We already discussed in the Introduction that designing *deterministic* budget-feasible mechanisms has been elusive. Positive results are only known for specific well-behaved objectives [45, 20, 46, 47, 1, 33, 21, 2] and, even worse, beyond monotone submodular valuation functions no deterministic $O(1)$-approximation mechanism is known, irrespectively of time or query complexity. We consider obtaining deterministic, budget-feasible, $O(1)$-approximation mechanisms—or showing that they do not exist—the most intriguing related open problem.

While our results provide a proof of concept with respect to what is asymptotically possible with polynomial-time, truthful mechanisms, the constants involved are very far from being practical. Although we do not claim that the different parameters appearing in the description and the analysis of our mechanisms are optimized, they had to be carefully chosen and we suspect there is not much room for improvement. Bringing down these approximation factors is another interesting direction.

Finally, it is mentioned in Remark 2 that the high level approach of SIMULTANEOUS GREEDY can be turned into a deterministic 7-approximation algorithm. We believe that it is worth exploring other possible applications of the high level approach of SIMULTANEOUS GREEDY, both in mechanism design and in constrained non-monotone submodular maximization.

Acknowledgments. Georgios Amanatidis and Pieter Kleer are supported by the NWO Gravitation Project NETWORKS, Grant Number 024.002.003. We would also like to thank the anonymous reviewers of EC 2019 for their valuable comments and suggestions.

Appendix. Proof of Theorem 4 for the Non-Monotone Case

For the reader's convenience, we repeat the statement of the theorem.

THEOREM 4. *Assuming that the solution has to be an independent set of a p-system, there is a universally truthful, individually rational, budget-feasible, $O(p)$-approximation mechanism that runs in polynomial time for (non-monotone) submodular objectives.*

Proof. Here we move on to the case of *non-monotone* submodular objectives. GENSM-CONSTRAINED is a modification of GENSM-MAIN that maintains a set F of "feasible" pairs, i.e., of pairs (i,j) such that $S_j \cup \{i\}$ is an independent set. In each step, the best such pair $(\hat{\imath},\hat{\jmath})$ is chosen and, given that $v(\hat{\imath}|S_{\hat{\jmath}})$ is neither too high nor too low, $\hat{\imath}$ is added to $S_{\hat{\jmath}}$. The parameter β is 8.5 and ALG_4 in line 5 can be any polynomial time approximation algorithm for non-monotone submodular maximization subject to a knapsack and a p-system constraint. Here we assume the $\frac{(1+\varepsilon)(p+1)(2p+3)}{p}$-approximation algorithm of Mirzasoleiman et al. [41] for $\varepsilon = 10^{-3}$.

GENSM-CONSTRAINED$(A, \mathcal{I}, v, \mathbf{c}, B)$

1 *With probability* $q = 1/3$ **:**
2   **return** $i^* \in \arg\max_{i \in A} v(i)$
3 *With probability* $1 - q$ **:**
4   Put each agent of A in either A_1 or A_2 independently at random with probability $\frac{1}{2}$
5     $x = v(\text{ALG}_4(A_1))$ `/* a` $(1+\varepsilon)(p+1)(2p+3)/p$`-approximation of` $\text{OPT}(A_1, v, \mathbf{c}_{A_1}, B)$ `*/`
6     $S_1 = S_2 = \emptyset$; $B_1 = B_2 = B$; $U = A_2$
7     $F = \{(i,j) \,|\, i \in U, j \in \{1,2\}$ and $S_j \cup \{i\} \in \mathcal{I}\}$ `/* all ``feasible'' pairs */`
8   **while** $F \neq \emptyset$ **do**
9   Let $(\hat{\imath},\hat{\jmath}) \in \arg\max_{(i,j)\in F} v(i|S_j)$
10   **if** $c_{\hat{\imath}} \leq \frac{\beta B}{x} v(\hat{\imath}|S_{\hat{\jmath}}) \leq B_{\hat{\jmath}}$ **then**
11             $S_{\hat{\jmath}} = S_{\hat{\jmath}} \cup \{\hat{\imath}\}$
12             $B_{\hat{\jmath}} = B_{\hat{\jmath}} - \frac{\beta B}{x} v(\hat{\imath}|S_{\hat{\jmath}})$
13         $U = U \setminus \{\hat{\imath}\}$
14   Update F
15   **for** $j \in \{1,2\}$ **do**
16         $T_j = \text{ALG}_2(S_j)$ `/* a 2-approximate solution with respect to` $\text{OPT}(S_j, v, \mathbf{c}_{S_j}, \infty)$ `*/`
17   Let S be the best solution among S_1, S_2, T_1, T_2
18   **return** S
19 Pay the agents according to Myerson's lemma

Clearly, S_1, S_2 start as independent sets and they are expanded only if they remain independent sets. As subsets of independent sets, T_1, T_2 are independent sets as well. Hence, GENSM-CONSTRAINED does return a solution $S \in \mathcal{I}$.

Like in the monotone case, following the reasoning used for GENSM-MAIN and GENSM-ONLINE, it is easy to prove universal truthfulness, individual rationality, budget-feasibility, and—given polynomial-time oracles—polynomial running time. What is left to show is that $\mathbb{E}(v(S))$ is an $O(p)$-approximation of $\text{OPT}(A, B)$.

First, suppose that $\max_{i\in A} v(i) \geq \frac{1}{136(p+6)} \cdot \textsc{opt}(A,B)$. Then, for the set S returned by GenSm-Constrained,

$$\mathbb{E}(v(S)) \geq q \cdot \max_{i\in A} v(i) \geq \frac{1}{3} \cdot \frac{1}{136(p+6)} \cdot \textsc{opt}(A,B) > \frac{1}{410(p+6)} \cdot \textsc{opt}(A,B)\,.$$

When $\max_{i\in A} v(i) < \frac{1}{136(p+6)} \cdot \textsc{opt}(A,B)$, we may follow the same approach as with the our other proofs. Recall the notation. That is, $i_1, i_2, \ldots, i_{|A_2|}$ are the agents of A_2 in the order considered by the mechanism and $j_1, \ldots, j_{|A_2|}$ are the corresponding $\hat{\jmath}$ selected in the kth execution of line 9.[2] By $S_j^{(k)}$ and $B_j^{(k)}$ we denote S_j and B_j, respectively, at the time i_k is selected. We only use S_1, S_2, B_1, B_2 for the final version of the corresponding set or quantity. The set $R = A_2 \setminus (S_1 \cup S_2)$ contains the agents i_k that were not added to $S_{j_k}^{(k)}$ and it is further partitioned to $R_{\mathbf{c}} = \big\{i_k \mid \frac{\beta B}{x} v\big(i_k|S_{i_k}^{(k)}\big) < c_{\hat{\imath}}\big\}$, $R_B = \big\{i_k \mid B_{i_k}^{(k)} < \frac{\beta B}{x} v\big(i_k|S_{i_k}^{(k)}\big)\big\}$, and $R_{\mathcal{I}} = R \setminus (R_{\mathbf{c}} \cup R_B)$.

Recall that Lemma 6 guarantees that $\textsc{opt}(A_i, B) \geq \frac{k-1}{4k} \textsc{opt}(A,B)$ for $i \in \{1,2\}$, where $k = 136(p+6)$, happens with probability at least $1/2$. Assume this is indeed the case. Therefore, $x = v(\textsc{alg}_1(A_1)) \geq \frac{(k-1)p}{4k(1+\varepsilon)(p+1)(2p+3)} \textsc{opt}(A,B)$.

Case 1. Assume that $R_B \neq \emptyset$. By repeating the analysis of Case 1 in the proof of Lemma 5, we get

$$v(S) \geq \left(\frac{(k-1)p}{4k(1+\varepsilon)(p+1)(2p+3)\beta} - \frac{1}{k}\right) \cdot \textsc{opt}(A,B)\,.$$

By substituting $k = 136(p+6)$, $\beta = 8.5$ and $\varepsilon = 10^{-3}$, it is a matter of simple calculations to get

$$v(S) \geq \frac{1}{136(p+6)} \cdot \textsc{opt}(A,B)\,. \tag{18}$$

Case 2. Next, assume that $R_B = \emptyset$. Let C^* be an optimal solution for the instance $(A_2, v, \mathbf{c}_{A_2}, B)$ and $C_1 = C^* \cap S_1$, $C_2 = C^* \cap S_2$, $C_3 = C^* \setminus (C_1 \cup C_2)$. By subadditivity (recall inequality (3)) and the fact that $T_j = \textsc{alg}_2(S_j), j \in \{1,2\}$, is a 2-approximate solution with respect to $\textsc{opt}(S_j, \infty)$ we get

$$\textsc{opt}(A_2, B) = v(C^*) \leq v(C_1) + v(C_2) + v(C_3) \leq 2v(T_1) + 2v(T_2) + v(C_3)\,. \tag{19}$$

For $v(C_3)$ recall inequality (5) (see proof of Lemma 5):

$$v(C_3) \leq v(S_1 \cup C_3) + v(S_2 \cup C_3)\,. \tag{5}$$

To upper bound $v(S_j \cup C_3)$ we work like in the proof of Theorem 4 because of the p-system constraint. By Definition 1*(iii)*, we have

$$\begin{aligned} v(S_j \cup C_3) &\leq v(S_j) + \sum_{i_k \in C_3} v(i_k|S_j) \\ &\leq v(S_j) + \sum_{i_k \in C_3 \cap R_{\mathbf{c}}} v(i_k|S_j) + \sum_{i_k \in C_3 \cap R_{\mathcal{I}}} v(i_k|S_j)\,. \end{aligned} \tag{20}$$

We upper bound the first sum exactly as in (15):

$$\sum_{i_k \in C_3 \cap R_{\mathbf{c}}} v(i_k|S_j) \leq \sum_{i_k \in C_3 \cap R_{\mathbf{c}}} v\left(i_k|S_j^{(k)}\right) < \frac{x}{\beta B} \sum_{i_k \in C_3 \cap R_{\mathbf{c}}} c_{i_k} \leq \frac{x}{\beta} \leq \frac{\textsc{opt}(A,B)}{\beta}\,. \tag{21}$$

For the second sum we have the analog of Claim 1. Recall that we never used the monotonicity of v in the proof of Claim 1. With just minor changes in notation, we can prove the following.

[2] In case not all agents are considered, what remains in F is arbitrarily indexed and paired with some $\hat{\jmath}$. This is as if we had a few dummy iterations at the end of the **while** loop in order to exhaust all agents by rejecting them one by one.

Claim 2. *For both $j \in \{1,2\}$, $\sum_{i_k \in C_3 \cap R_{\mathcal{I}}} v(i_k|S_j) \le p \cdot v(S_j)$.*

Now, combining (19), (5), (20), (21), and Claim 2, we have

$$\mathrm{OPT}(A_2, B) \le 2v(T_1) + 2v(T_2) + (p+1)v(S_1) + (p+1)v(S_2) + 2\frac{\mathrm{OPT}(A,B)}{\beta},$$

and, using the definition of S and the lower bound on $\mathrm{OPT}(A_2, B)$,

$$v(S) \ge \frac{1}{2p+6} \cdot \left(\frac{k-1}{4k} - \frac{2}{\beta}\right) \mathrm{OPT}(A,B).$$

By substituting k and β, it is a matter of calculations to get

$$v(S) \ge \frac{1}{136(p+6)} \cdot \mathrm{OPT}(A,B). \tag{22}$$

Since, due to Lemma 6, both (18) and (22) hold with probability at least 1/2, we have

$$\mathbb{E}(v(S)) \ge (1-q) \cdot \frac{1}{2} \cdot \frac{1}{136(p+6)} \cdot \mathrm{OPT}(A,B) > \frac{1}{410(p+10)} \cdot \mathrm{OPT}(A,B),$$

thus concluding the proof. ∎

References

[1] Amanatidis G, Birmpas G, Markakis E (2016) Coverage, matching, and beyond: New results on budgeted mechanism design. *Web and Internet Economics - 12th International Conference, WINE 2016, Montreal, Canada, December 11-14, 2016, Proceedings*, volume 10123 of *LNCS*, 414–428 (Springer).

[2] Amanatidis G, Birmpas G, Markakis E (2017) On budget-feasible mechanism design for symmetric submodular objectives. *Web and Internet Economics - 13th International Conference, WINE 2017, Bangalore, India, December 17-20, 2017, Proceedings*, volume 10660 of *LNCS*, 1–15 (Springer).

[3] Amanatidis G, Fusco F, Lazos P, Leonardi S, Reiffenhäuser R (2020) Fast adaptive non-monotone submodular maximization subject to a knapsack constraint. *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

[4] Amanatidis G, Kleer P, Schäfer G (2019) Budget-feasible mechanism design for non-monotone submodular objectives: Offline and online. *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019.*, 901–919 (ACM).

[5] Anari N, Goel G, Nikzad A (2014) Mechanism design for crowdsourcing: An optimal 1-1/e competitive budget-feasible mechanism for large markets. *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, 266–275.

[6] Azar PD, Kleinberg R, Weinberg SM (2014) Prophet inequalities with limited information. *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*, 1358–1377 (SIAM).

[7] Babaioff M, Immorlica N, Kempe D, Kleinberg R (2007) A knapsack secretary problem with applications. *APPROX-RANDOM*, volume 4627 of *LNCS*, 16–28 (Springer).

[8] Badanidiyuru A, Kleinberg R, Singer Y (2012) Learning on a budget: posted price mechanisms for online procurement. *EC*, 128–145 (ACM).

[9] Badanidiyuru A, Vondrák J (2014) Fast algorithms for maximizing submodular functions. *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, 1497–1514 (SIAM).

[10] Balkanski E, Hartline JD (2016) Bayesian budget feasibility with posted pricing. *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, 189–203 (ACM).

[11] Bateni M, Hajiaghayi MT, Zadimoghaddam M (2013) Submodular secretary problem and extensions. *ACM Trans. Algorithms* 9(4):32:1–32:23.

[12] Bei X, Chen N, Gravin N, Lu P (2012) Budget feasible mechanism design: from prior-free to bayesian. *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, 449–458.

[13] Bei X, Chen N, Gravin N, Lu P (2017) Worst-case mechanism design via bayesian analysis. *SIAM J. Comput.* 46(4):1428–1448.

[14] Blumrosen L, Nisan N (2009) On the computational power of demand queries. *SIAM J. Comput.* 39(4):1372–1391.

[15] Borodin A, Filmus Y, Oren J (2010) Threshold models for competitive influence in social networks. *Proceedings of the 6th International Workshop on Internet and Network Economics, WINE 2010*, 539–550.

[16] Buchbinder N, Feldman M (2018) Deterministic algorithms for submodular maximization problems. *ACM Trans. Algorithms* 14(3):32:1–32:20.

[17] Buchbinder N, Feldman M, Naor J, Schwartz R (2014) Submodular maximization with cardinality constraints. *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, 1433–1452 (SIAM).

[18] Chekuri C, Gupta S, Quanrud K (2015) Streaming algorithms for submodular function maximization. *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, volume 9134 of *LNCS*, 318–330 (Springer).

[19] Chekuri C, Vondrák J, Zenklusen R (2014) Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM J. Comput.* 43(6):1831–1879.

[20] Chen N, Gravin N, Lu P (2011) On the approximability of budget feasible mechanisms. *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, 685–699.

[21] Dobzinski S, Papadimitriou CH, Singer Y (2011) Mechanisms for complement-free procurement. *Proceedings 12th ACM Conference on Electronic Commerce (EC-2011), San Jose, CA, USA, June 5-9, 2011*, 273–282.

[22] Dynkin EB (1963) Optimal choice of the stopping moment of a markov process. *Doklady Akademii Nauk*, volume 150, 238–240 (Russian Academy of Sciences).

[23] Ene A, Nguyen HL (2019) A nearly-linear time algorithm for submodular maximization with a knapsack constraint. *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece.*, volume 132 of *LIPIcs*, 53:1–53:12 (Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik).

[24] Feige U, Mirrokni VS, Vondrák J (2011) Maximizing non-monotone submodular functions. *SIAM J. Comput.* 40(4):1133–1153.

[25] Feldman M, Harshaw C, Karbasi A (2017) Greed is good: Near-optimal submodular maximization via greedy optimization. *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017*, volume 65 of *Proceedings of Machine Learning Research*, 758–784 (PMLR).

[26] Feldman M, Naor J, Schwartz R (2011) Improved competitive ratios for submodular secretary problems (extended abstract). *APPROX-RANDOM*, volume 6845 of *LNCS*, 218–229 (Springer).

[27] Feldman M, Naor J, Schwartz R (2011) A unified continuous greedy algorithm for submodular maximization. *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, 570–579 (IEEE Computer Society).

[28] Feldman M, Zenklusen R (2018) The submodular secretary problem goes linear. *SIAM J. Comput.* 47(2):330–366.

[29] Goel G, Nikzad A, Singla A (2014) Mechanism design for crowdsourcing markets with heterogeneous tasks. *Proceedings of the Seconf AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2014, November 2-4, 2014, Pittsburgh, Pennsylvania, USA.*

[30] Gravin N, Jin Y, Lu P, Zhang C (2019) Optimal budget-feasible mechanisms for additive valuations. *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019.*, 887–900 (ACM).

[31] Gupta A, Nagarajan V, Singla S (2017) Adaptivity gaps for stochastic probing: Submodular and XOS functions. *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, 1688–1702 (SIAM).

[32] Gupta A, Roth A, Schoenebeck G, Talwar K (2010) Constrained non-monotone submodular maximization: Offline and secretary algorithms. *Internet and Network Economics - 6th International Workshop, WINE 2010, Stanford, CA, USA, December 13-17, 2010. Proceedings*, volume 6484 of *LNCS*, 246–257 (Springer).

[33] Horel T, Ioannidis S, Muthukrishnan S (2014) Budget feasible mechanisms for experimental design. *LATIN 2014: Theoretical Informatics - 11th Latin American Symposium, Montevideo, Uruguay, March 31 - April 4, 2014. Proceedings*, 719–730.

[34] Kesselheim T, Tönnis A (2017) Submodular secretary problems: Cardinality, matching, and linear constraints. *APPROX-RANDOM*, volume 81 of *LIPIcs*, 16:1–16:22 (Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik).

[35] Khalilabadi PJ, Tardos É (2018) Simple and efficient budget feasible mechanisms for monotone submodular valuations. *Web and Internet Economics - 14th International Conference, WINE 2018, Oxford, UK, December 15-17, 2018, Proceedings*, volume 11316 of *LNCS*, 246–263 (Springer).

[36] Kulik A, Shachnai H, Tamir T (2013) Approximations for monotone and nonmonotone submodular maximization with knapsack constraints. *Math. Oper. Res.* 38(4):729–739.

[37] Lehmann B, Lehmann DJ, Nisan N (2006) Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior* 55(2):270–296.

[38] Leonardi S, Monaco G, Sankowski P, Zhang Q (2016) Budget feasible mechanisms on matroids. *CoRR* abs/1612.03150.

[39] Leonardi S, Monaco G, Sankowski P, Zhang Q (2017) Budget feasible mechanisms on matroids. *Integer Programming and Combinatorial Optimization - 19th International Conference, IPCO 2017, Waterloo, ON, Canada, June 26-28, 2017, Proceedings*, volume 10328 of *LNCS*, 368–379 (Springer).

[40] Mirrokni VS, Schapira M, Vondrák J (2008) Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions. *Proceedings 9th ACM Conference on Electronic Commerce (EC-2008), Chicago, IL, USA, June 8-12, 2008*, 70–77 (ACM).

[41] Mirzasoleiman B, Badanidiyuru A, Karbasi A (2016) Fast constrained submodular maximization: Personalized data summarization. *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR*, 1358–1367 (JMLR.org).

[42] Myerson R (1981) Optimal auction design. *Mathematics of Operations Research* 6(1).

[43] Nemhauser GL, Wolsey LA, Fisher ML (1978) An analysis of approximations for maximizing submodular set functions - I. *Math. Program.* 14(1):265–294.

[44] Schrijver A (2003) *Combinatorial optimization: polyhedra and efficiency*, volume 24 (Springer Science & Business Media).

[45] Singer Y (2010) Budget feasible mechanisms. *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, 765–774 (IEEE Computer Society).

[46] Singer Y (2012) How to win friends and influence people, truthfully: influence maximization mechanisms for social networks. *Proceedings of the Fifth International Conference on Web Search and Web Data Mining, WSDM 2012, Seattle, WA, USA, February 8-12, 2012*, 733–742.

[47] Singer Y, Mittal M (2013) Pricing mechanisms for crowdsourcing markets. *22nd International World Wide Web Conference, WWW '13*, 1157–1166 (International World Wide Web Conferences Steering Committee / ACM).

[48] Singla A, Krause A (2013) Incentives for privacy tradeoff in community sensing. *Proceedings of the First AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2013, November 7-9, 2013, Palm Springs, CA, USA* (AAAI).

[49] Sviridenko M (2004) A note on maximizing a submodular set function subject to a knapsack constraint. *Oper. Res. Lett.* 32(1):41–43.

[50] Wolsey LA (1982) Maximising real-valued submodular functions: Primal and dual heuristics for location problems. *Math. Oper. Res.* 7(3):410–425.