# A Tight Competitive Ratio for Online Submodular Welfare Maximization

**Amit Ganz** ✉
The Henry and Marilyn Taub Faculty of Computer Science, Technion, Haifa, Israel

**Pranav Nuti** ✉
Department of Computer Science, Stanford University, CA, USA

**Roy Schwartz** ✉
The Henry and Marilyn Taub Faculty of Computer Science, Technion, Haifa, Israel

## Abstract

In this paper we consider the online Submodular Welfare (SW) problem. In this problem we are given $n$ bidders each equipped with a general non-negative (not necessarily monotone) submodular utility and $m$ items that arrive online. The goal is to assign each item, once it arrives, to a bidder or discard it, while maximizing the sum of utilities. When an adversary determines the items' arrival order we present a simple randomized algorithm that achieves a tight competitive ratio of $1/4$. The algorithm is a specialization of an algorithm due to [Harshaw-Kazemi-Feldman-Karbasi MOR'22], who presented the previously best known competitive ratio of $3 - 2\sqrt{2} \approx 0.171573$ to the problem. When the items' arrival order is uniformly random, we present a competitive ratio of $\approx 0.27493$, improving the previously known $1/4$ guarantee. Our approach for the latter result is based on a better analysis of the (offline) Residual Random Greedy (RRG) algorithm of [Buchbinder-Feldman-Naor-Schwartz SODA'14], which we believe might be of independent interest.

**2012 ACM Subject Classification** Theory of computation → Online algorithms

**Keywords and phrases** Online Algorithms, Submodular Maximization, Welfare Maximization, Approximation Algorithms

**Digital Object Identifier** 10.4230/LIPIcs.ESA.2023.52

**Related Version** *Full Version*: https://arxiv.org/abs/2308.07746

**Funding** Amit Ganz and Roy Schwartz have received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 852870-ERC-SUBMODULAR.

**Acknowledgements** The authors would like to thank the anonymous referees for helpful remarks.

## 1 Introduction

Submodularity is a mathematical notion that captures the concept of diminishing returns. Formally, a set function $f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ over a ground set $\mathcal{N}$ is submodular, if for all $A, B \subseteq \mathcal{N}$: $f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$. An equivalent definition, which is called the diminishing returns property, is the following: $f(A \cup \{u\}) - f(A) \geq f(B \cup \{u\}) - f(B)$, for every $A \subseteq B \subseteq \mathcal{N}$ and every $u \in \mathcal{N} \setminus B$. Submodular functions naturally arise in many different settings, e.g., combinatorics, graph theory, information theory and economics.

We consider the Submodular Welfare (SW) problem. In this problem we are given a set $\mathcal{N} = \{1, \ldots, m\}$ of $m$ unsplittable items and a set $B = \{1, \ldots, n\}$ of $n$ bidders. Each bidder $j$ has a non-negative (and not necessarily monotone) submodular utility function $f_j$ and the goal is to assign items to the bidders while maximizing the sum of the utilities: $\sum_{j=1}^{n} f_j(S_j)$. Here $S_j$ is the set of items allocated to bidder $j$, and the requirement is that $S_j \cap S_{j'} = \emptyset$ for every $j \neq j'$ (since the items are unsplittable) and $\cup_{j=1}^{n} S_j \subseteq \mathcal{N}$ (note that not all items must be assigned). SW with monotone utilities has been extensively studied for more than two decades, e.g., [13, 15, 19, 25, 43, 52, 62, 68, 69]. Submodular maximization


31st Annual European Symposium on Algorithms (ESA 2023).
Editors: Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman; Article No. 52; pp. 52:1–52:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

with general (not necessarily monotone) objectives is also the focus of extensive theoretical research, e.g., [24, 30, 32, 34, 36, 50, 51, 63, 68, 70]. Moreover, maximization of non-monotone submodular objectives has found numerous practical applications, e.g., network inference [39], mobile crowdsensing [55], summarization of documents and video [53, 54, 59], marketing in social networks [64], and even gang violence reduction [65], to name a few. In particular, non-monotone utilities in the context of `SW` can model soft budget constraints, where each bidder $j$ needs to pay a price $p_{i,j}$ when item $i$ is allocated to it. Additional related problems were also studied, including: `SW` with demand queries [26], and other utilities such as XOS and subadditive [18, 23].

When considering the offline version, `SW` is typically viewed as a special case of maximizing a submodular objective subject to a partition matroid independence constraint: (1) the ground set is $\mathcal{N} \times B$; (2) the partition matroid is defined over $\mathcal{N} \times B$ where each part in the partition corresponds to an item, *i.e.*, the part that corresponds to item $i \in \mathcal{N}$ is $\{(i, j)\}_{j=1}^{n}$; and (3) the objective $f : 2^{\mathcal{N} \times B} \to \mathbb{R}_{\geq 0}$ is defined as: $f(S) \triangleq \sum_{j=1}^{n} f_j(\{i : (i, j) \in S\})$. In case the utilities are monotone a tight (asymptotic) approximation of $(1 - 1/e)$ was given by [14] by introducing the celebrated continuous greedy algorithm, a tight approximation of $(1 - (1 - 1/n)^n)$ for any number $n$ of bidders was given by [31], and the matching hardness result was given by [60]. For a general and not necessarily monotone objective the current best known offline approximation is also based on the continuous approach and achieves an approximation of $(1/e) + 0.0171$ [8], which improved upon the previous works of [21, 31].

In the online version of `SW` items arrive one by one. Whenever an item arrives, one has to decide immediately and irrevocably whether to assign it to one of the bidders or not assign it at all (the latter decision is relevant only when the utilities are not necessarily monotone). There are two natural settings that differ in the order in which items arrive. First, in the online adversarial setting an adversary can choose the order in which items arrive (the adversary knows the algorithm and how it operates, however if the algorithm is randomized it does not know the outcome of its random choices). Second, in the online random order setting items arrive one by one in a uniform random order.

When considering the online version, as opposed to the offline version, worse results are known. For monotone utilities, in the adversarial setting, a $(1/2)$-competitive greedy algorithm is known and additionally it is the best possible algorithm for this setting [41]. However, if one assumes the online random order setting, it is known that one can achieve a competitive ratio of $(1/2) + 0.0096$ [9], which improved the result of [46] who were the first to break the $1/2$ barrier obtaining a competitive ratio of $(1/2) + 0.005$.

Unfortunately, when considering general submodular (and not necessarily monotone) utilities, worse results are known. In the adversarial setting, the special case of a single bidder is of particular interest since it is equivalent to online Unconstrained Submodular Maximization (`USM`): given a general submodular objective $f$ items arrive one by one in an online manner and once an item arrives the algorithm needs to decide whether to choose or discard it where the goal is to maximize $f(S)$ ($S$ denotes the chosen elements). An (implicit) competitive ratio of $1/4$ was given by [24] for online `USM` (the algorithm simply chooses every element independently with a probability of half). This result was proved to be tight [11]. For the general case of multiple bidders an algorithm achieving a competitive ratio of $3 - 2\sqrt{2} \approx 0.171573$ was given by [38] (this algorithm handles a general matroid independence constraint and assumes elements of the ground set arrive in an online manner).

In the random order setting, an (implicit) result follows from analyzing the offline Residual Random Greedy (`RRG`) algorithm of [10] for maximizing a non-monotone submodular objective subject to a matroid independence constraint. When considering `SW`, the `RRG` algorithm operates as follows: it chooses a uniform random order over the items and goes over the items in this order, and assigns each item to the bidder with the highest marginal value (if this marginal value is negative then the item is discarded). Thus, if the RRG algorithm achieves an approximation of $\alpha$ in the offline setting, then the deterministic greedy achives a competitive ratio of $\alpha$ in the online random order setting. [10] prove that the RRG algorithm achieves an approximation of $1/4$, therefore implying a competitive ratio of $1/4$ for SW in the random order setting. It is worth noting that better algorithms than RRG are known in the offline setting, e.g., [8, 21, 31], however, they do not apply to the online random order setting.

In this work, we assume the standard *value oracle* model: each submodular function $f$ is not given explicitly but rather the algorithm can query for every subset $S$ the value of $f(S)$. The running time of the algorithm is measured not only by the number of arithmetic operations, but also by the number of value queries. In the online version, for both adversarial and random order settings, the algorithm can query subsets $S$ only of items that already arrived. Thus, intuitively, the algorithm has no information regarding future items.

## 1.1 Our Results

Focusing first on the adversarial setting, we present the following positive result.

▶ **Theorem 1.** *There exists a randomized polynomial time algorithm achieving a competitive ratio of $1/4$ for online SW in the adversarial setting with general (not necessarily monotone) utilities.*

There are three things to note regarding Theorem 1. First, the competitive ratio of $1/4$ is tight as even for the special case of a single bidder (which is equivalent to online USM) [11] provide a matching hardness of $1/4$. Second, Theorem 1, to the best of our knowledge, improves the previous best known competitive ratio of $3 - 2\sqrt{2} \approx 0.171573$ [38]. Third, for the special case of a single bidder the competitive ratio of Theorem 1 matches the $1/4$ guarantee of the simple algorithm that just chooses independently for every item to include it in the solution with a probability of half [24]. However, we note that the algorithm we present in order to prove Theorem 1, even in the special case of a single bidder, differs from the algorithm that just chooses a uniform random subset.

We complement the above result by showing that the randomness of the online algorithm in Theorem 1 is needed. This is summarized in the following theorem that gives a hardness result that tends to zero.

▶ **Theorem 2.** *For every $M > 0$, no deterministic algorithm can achieve a competitive ratio better than $1/M$ for online SW in the adversarial setting with general (not necessarily monotone) utilities.*

Focusing on the random order setting, the following theorem proves that one can achieve an improved competitive ratio over the previously (implicit) known $1/4$ [10]. We note that the following theorem separates the random order and adversarial settings, since one should recall there is a hardness of $1/4$ in the adversarial setting even when only a single bidder is present.

▶ **Theorem 3.** *The deterministic greedy algorithm achieves a competitive ratio of $\approx 0.27493$ for online SW in the random order setting with general (not necessarily monotone) utilities.*

The above theorem is achieved by a better analysis of the offline randomized RRG algorithm of [10]. This improved analysis can be easily extended to a general matroid independence constraint, as the following theorem states (its proof is deferred to a full version of the paper).

▶ **Theorem 4.** *The* RRG *algorithm of [10] achieves an approximation guarantee of* $\approx 0.27493$ *for maximizing a general (not necessarily monotone) submodular function given a matroid independence constraint.*

## 1.2 Our Approach

The online algorithm for the adversarial setting adopts a simple randomized approach: once item $i$ arrives it defines a distribution over the bidders and assigns the item to a random bidder sampled from this distribution. Surprisingly, we prove that a remarkably simple distribution suffices to obtain a tight competitive ratio of $1/4$: item $i$ is assigned to the bidder with the highest marginal value with a probability of $1/2$, to the bidder with the second highest marginal with a probability of $1/4$, and so forth as long as the marginal value is non-negative. With the remainder probability item $i$ is discarded. It is important to note that the *ordering* of the bidders according to their marginal values (as long as the marginal values are non-negative) dictates the distribution. However, as long as the ordering remains the same, the distribution is independent of the actual marginal values themselves. We prove that the above simple approach suffices to obtain a tight competitive ratio of $1/4$ for the adversarial setting.

It should be noted that the above randomized approach is based on the streaming algorithm of [38], which obtains a competitive ratio of $3 - 2\sqrt{2} \approx 0.171573$. Intuitively, we specialize the algorithm of [38] to online SW in the adversarial setting. The reason is that once item $i$ arrives one can perform the following process in order to obtain the distribution over bidders that was defined above: sort the bidders in a non-increasing order of marginal values and assign the item to the first bidder with a probability of half, if the item was not assigned to this bidder then with a probability of half assign the item to the next bidder, and so forth as long as the marginal value is non-negative. This process describes how the algorithm of [38] operates in the special case the online order of elements of the ground set $\mathcal{N} \times B$ satisfies: (1) all elements $\{(i,j)\}_{j=1}^{n}$ are consecutive in the online order; and (2) elements $\{(i,j)\}_{j=1}^{n}$ are sorted in a non-increasing order of marginal values of the bidders.

Focusing on the random order setting, we present an improved analysis of the RRG algorithm for a general matroid independence constraint. As a preliminary step, which is not required but is mathematically convenient, we present a "smooth" version of the RRG algorithm. The difference between the original and smooth versions is that in the smooth version the distribution of the steps the algorithm can perform does not change as the algorithm progresses. However, in the original version this distribution evolves as the algorithm progresses. We note that the analysis of both versions is very similar (assuming the smooth version performs enough steps). Nonetheless, we believe it is easier to analyze the smooth version.[1]

A key insight in analyzing the RRG algorithm (as well as other closely related algorithms, e.g., the Random Greedy algorithm of [10]), is lower bounding the expected value of a fixed optimal solution $OPT$ when a random subset of elements $S$ is added to it. More precisely, if each element $u$ satisfies $\Pr[u \in S] \leq p$ then standard known arguments, e.g., Lemma 2.2 [10] which is based on Lemma 2.2 [24], imply that $\mathbb{E}[f(OPT \cup S)] \geq (1-p)f(OPT)$. It is important to note that this general insight works for *any* distribution of $S$ that satisfies $\Pr[u \in S] \leq p$ for every $u$. This general insight by itself enabled [10] to prove that the RRG algorithm achieves an approximation of $1/4$. We are able to improve upon the above

[1] For a detailed comparison of the two versions we defer to a full version of the paper.

by exploiting the specific probabilistic behavior of the smooth version of the algorithm, as opposed to the general insight which works for any distribution of $S$. This results in an improved analysis of the smooth version of the RRG algorithm via two jointly related recursive relations that together bound the performance of the algorithm.

## 1.3 Additional Related Work

The literature regarding submodular maximization, *i.e.*, problems of the form $\max\{f(S) : S \in \mathcal{F}\}$ ($\mathcal{F}$ is the collection of feasible solutions), is rich, e.g., [4, 7, 8, 10, 14, 21, 22, 24, 44, 47, 48, 60, 61, 66], and dates back to the late 70's. Moreover, the online SW problem naturally generalizes many online problems such as online matching [1, 29, 40, 42, 57], online weighted matching [1, 28], budgeted allocation [12, 16, 35, 58], and more general classes of online allocation problems [2, 17, 27, 67]. Other related problems include the extensively studied class of secretary problems, e.g., [3, 5, 6, 20, 31, 33, 37, 45, 49, 56], where the goal is to solve an optimization problem assuming the arrival order of the input is uniform and random.

## 1.4 Preliminaries

In our analysis we require the following known lemma.

▶ **Lemma 5** (Lemma 2.2 [10] which is based on Lemma 2.2 [24]). *Let $f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ be submodular. Denote by $A(p)$ a random subset of $A$ where each element appears with probability at most $p$ (not necessarily independently). Then $\mathbb{E}[f(A(p))] \geq (1-p)f(\emptyset)$.*

## 1.5 Paper Organization

Section 2 deals with the adversarial setting, proving Theorems 1 and 2. Section 3 focuses on the random order setting, proving Theorem 3 and 4.

# 2 Adversarial Setting

In this section we consider the adversarial setting, and present both a tight randomized algorithm proving Theorem 1 and hardness for any deterministic algorithm (Theorem 2).

## 2.1 Tight Randomized Algorithm

In this section we consider the adversarial setting, and start by presenting our algorithm which appears in Algorithm 1. For simplicity of presentation, we assume the items are numbered according to the order the adversary chooses, *i.e.*, item 1 is the first to arrive, item 2 is the second to arrive and so forth. Hence, the algorithm performs $m$ iterations where in iteration $i$ the algorithm chooses what to do with item $i$: with a probability of $1/2$ it assigns it to the bidder with highest marginal value, with a probability of $1/4$ is assigns it to the bidder with the second highest marginal value, and so forth as long as the marginal value is non-negative. With the remainder probability item $i$ is discarded and not assigned to any bidder. Therefore, if there are $\ell$ bidders with non-negative marginal with respect to item $i$, item $i$ is discarded with a probability of $2^{-\ell}$. In what follows we use the notation $f_j(i|S)$ to denote the marginal value of item $i$ with respect to bidder $j$ assuming bidder $j$ was already assigned a subset $S \subseteq \mathcal{N}$ of items, *i.e.*, $f_j(i|S) \triangleq f_j(S \cup \{i\}) - f_j(S)$.

**Algorithm 1** Online Adversarial Algorithm.

$\forall j = 1, \ldots, n : S_j^0 \leftarrow \emptyset$.
**for** $i = 1, ..., m$ **do**
  $\forall j = 1, \ldots, n$: $S_j^i \leftarrow S_j^{i-1}$.
  $\forall r = 1, \ldots, n$: let $j_r$ be the bidder with the $r^{\text{th}}$ highest marginal w.r.t item $i$:
    $f_{j_1}(i|S_{j_1}^{i-1}) \geq f_{j_2}(i|S_{j_2}^{i-1}) \geq \ldots \geq f_{j_n}(i|S_{j_n}^{i-1})$.
  let $j^i$ be a random bidder such that $\Pr[j^i = j_r] = 2^{-r}$.
  **if** $f_{j^i}(i|S_{j^i}^{i-1}) \geq 0$ **then**
    $S_{j^i}^i \leftarrow S_{j^i}^i \cup \{i\}$
  **end**
**end**
return $S_1^m, S_2^m, \ldots, S_n^m$.

Let $O$ denote an offline optimal allocation for the problem: $O = (O_1, \ldots, O_n)$, where $O_j$ is the collection of items assigned to bidder $j$ in the optimal solution. Formally, $O$ is defined as follows[2]:

$$\arg\max_{(O_1,\ldots,O_n)} \left\{ \sum_{j=1}^{n} f_j(O_j) : \forall j \neq r \quad O_j \cap O_r = \emptyset, \quad \forall j = 1, \ldots, n \quad O_j \subseteq \mathcal{N} \right\}.$$

Let $S^i$ be the allocation induced by the algorithm at the end of the $i^{\text{th}}$ iteration, *i.e.*, $S^i = (S_1^i, S_2^i, \ldots, S_n^i)$. For every bidder $j$, define $H_j^i$ as follows:

$$H_j^i \triangleq (O_j \cap \{1, \ldots, i\}) \cup S_j^i,$$

and let $H^i = (H_1^i, \ldots, H_n^i)$. It is important to note that $H^i$ is not necessarily a feasible solution, since an item might be assigned to up to two bidders. For simplicity of presentation we use the notation of $f(S^i)$ to denote the value of the allocation $S^i$, *i.e.*, $f(S^i) \triangleq \sum_{j=1}^n f_j(S_j^i)$. Similarly, we use $f(H^i)$ to denote $\sum_{j=1}^n f_j(H_j^i)$ (though $H^i$ is not an allocation since items might be assigned to more than a single bidder).

In order to analyze the algorithm we denote by $P^i$ the profit gained during the $i^{\text{th}}$ iteration: $P^i \triangleq f(S^i) - f(S^{i-1})$. Note that $P^1 + \ldots + P^m = f(S^m) - f(S^0)$, where $S^m$ is the output of Algorithm 1 and $S^0$ is the empty allocation that does not assign any item to any of the bidders. Finally, we define the sequence: $K^i \triangleq f(H^i) - f(H^{i-1})$. Note that $K^1 + \ldots + K^m = f(H^m) - f(S^0)$ (note that $H^0 = S^0$).

The analysis of the competitive ratio of Algorithm 1 is essentially based on a single observation, which for every iteration upper bounds $K^i$ as a function of the expected gained profit. This is summarized in Lemma 6.

▶ **Lemma 6.** *$\forall i = 1, \ldots, m$ the following holds:* $\mathbb{E}[K^i] \leq 2 \cdot \mathbb{E}[P^i]$.

We use Lemma 6 to prove Theorem 1. The proof is essentially by summation over all iterations of the algorithm.

[2] Note that formally speaking this is actually a set, but for notational convenience, in this paper, when we refer to something as equalling the arg max, we will always mean it equals an element of the arg max set.

**Proof of Theorem 1.** We prove that the competitive ratio of Algorithm 1 is 1/4. Lemma 6, when applied for every iteration $i$, implies that: $\mathbb{E}\left[\sum_{i=1}^{m} K^i\right] \leq 2 \cdot \mathbb{E}\left[\sum_{i=1}^{m} P^i\right]$. Hence, it follows that: $\mathbb{E}\left[f(H^m) - f(S^0)\right] \leq 2 \cdot \mathbb{E}\left[f(S^m) - f(S^0)\right]$. In fact, since $f(S^0)$ is non-negative, we have that:

$$\mathbb{E}\left[f(H^m)\right] \leq 2 \cdot \mathbb{E}\left[f(S^m)\right].$$

For every bidder $j$ and item $i$ we have that the probability that item $i$ is assigned to bidder $j$ is at most 1/2, *i.e.*, $\Pr[i \in S_j^m] \leq 1/2$. Therefore, using Lemma 5 applied to the submodular function $h_j$, where $h_j : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ and $h_j(S) \triangleq f_j(S \cup O_j)$ for every $S \subseteq \mathcal{N}$, we can conclude that: $\mathbb{E}[f_j(S_j^m \cup O_j)] = \mathbb{E}[h_j(S_j^m)] \geq (1/2) \cdot h_j(\emptyset) = (1/2) \cdot f_j(O_j)$. Therefore,

$$\mathbb{E}[f(H^m)] = \sum_{j=1}^{n} \mathbb{E}[f_j(H_j^m)] = \sum_{j=1}^{n} \mathbb{E}[f_j(S_j^m \cup O_j)] \geq \frac{1}{2} \sum_{j=1}^{n} f_j(O_j) = \frac{1}{2} \cdot f(O).$$

Combining the above, we conclude that: $(1/4) \cdot f(O) \leq \mathbb{E}[f(S^m)]$. ◀

All that remains is to prove Lemma 6.

**Proof of Lemma 6.** For proof simplicity, let us start by adding a dummy bidder whose utility is the zero function. Clearly, this does not change the algorithm's performance or the value of $f(O)$, but it lets us assume (without loss of generality) that: (1) for every item $i$ the optimal solution $O$ allocates item $i$ to some bidder $k$, *i.e.*, $i \in O_k$; and (2) for every item $i$ at least one bidder has a non-negative marginal value, *i.e.*, $f_j(i|S_j^{i-1}) \geq 0$ for some bidder $j$.

Fix an iteration $i = 1, \ldots, m$ and condition on any possible realization $R_{i-1}$ of the random choices of the algorithm in the first $i-1$ iterations. Thus, $S^{i-1}$, $H^{i-1}$, and $j_1$ up to $j_m$ are deterministic and fixed given this conditioning, whereas $j^i$ is a random variable.

Recall that (without loss of generality) item $i$ is assigned to bidder $k$ by the optimal solution and that $j_r$ denotes the bidder with the $r^{\text{th}}$ highest marginal value with respect to item $i$ given the elements previously assigned to the bidder. Let us assume that the first $\ell \geq 1$ bidders have a non-negative marginal value with respect to item $i$, *i.e.*, $f_{j_1}(i|S_{j_1}^{i-1}) \geq f_{j_2}(i|S_{j_2}^{i-1}) \geq \ldots \geq f_{j_\ell}(i|S_{j_\ell}^{i-1}) \geq 0$ and if $\ell < m$: $f_{j_{\ell+1}}(i|S_{j_{\ell+1}}^{i-1}) < 0$. Moreover, assume that bidder $k$ has the $t^{\text{th}}$ largest marginal, *i.e.*, $k = j_t$. Let us now bound the expected change from $f(H^{i-1})$ to $\mathbb{E}[f(H^i)|R_{i-1}]$, *i.e.*, $\mathbb{E}[K^i|R_{i-1}]$, given the assumption that $t \leq \ell$, *i.e.*, bidder $k$ to which the optimal solution assigned item $i$ is among the $\ell$ bidders who have a non-negative marginal value:

$$\begin{aligned}&\mathbb{E}[f(H^i)|R_{i-1}] - f(H^{i-1})\\ =&f_k(S_k^{i-1} \cup (O_k \cap \{1, \ldots, i-1\}) \cup \{i\}) - f_k(S_k^{i-1} \cup (O_k \cap \{1, \ldots, i-1\})) + \qquad (1)\\ &\sum_{r=1}^{\ell} \frac{\mathbf{1}_{\{j_r \neq k\}}}{2^r} \left\{ f_{j_r}((O_{j_r} \cap \{1, \ldots, i-1\}) \cup S_{j_r}^{i-1} \cup \{i\}) - f_{j_r}((O_{j_r} \cap \{1, \ldots, i-1\}) \cup S_{j_r}^{i-1}) \right\}.\end{aligned}$$

The equality in (1) follows from the definitions of $H^i$ and the algorithm, as well as the fact that $i \in O_k$. We note that that (1) can be upper bounded as follows:

$$\leq f_k(i|S_k^{i-1}) + \sum_{r=1}^{\ell} \frac{\mathbf{1}_{\{j_r \neq k\}} f_{j_r}(i|S_{j_r}^{i-1})}{2^r}. \qquad (2)$$

The inequality in (2) follows from the decreasing marginals property of the submodular utilities. Next, let us rewrite (2):

$$=f_{j_t}(i|S_{j_t}^{i-1}) \cdot \left(1 - \frac{1}{2^t}\right) + \sum_{r=1}^{\ell} \frac{f_{j_r}(i|S_{j_r}^{i-1})}{2^r} \tag{3}$$

$$=f_{j_t}(i|S_{j_t}^{i-1}) \cdot \left(\sum_{r=1}^{t} 2^{-r}\right) + \sum_{r=1}^{\ell} \frac{f_{j_r}(i|S_{j_r}^{i-1})}{2^r}. \tag{4}$$

The equality in (3) holds since bidder $k$ has the $t^{\text{th}}$ largest marginal, *i.e.*, $k = j_t$, and $t \leq \ell$. Also, the equality in (4) follows from the value of a geometric sum. Moreover, we note that (4) can be further upper bounded:

$$\leq \sum_{r=1}^{t} \frac{f_{j_r}(i|S_{j_r}^{i-1})}{2^r} + \sum_{r=1}^{\ell} \frac{f_{j_r}(i|S_{j_r}^{i-1})}{2^r}. \tag{5}$$

In the above, the inequality in (5) is true since bidder $j_r$ has the $r^{\text{th}}$ largest marginal value, *i.e.*, for every $r \leq t$: $f_{j_r}(i|S_{j_r}^{i-1})) \geq f_{j_t}(i|S_{j_t}^{i-1})$. Next, we upper bound (5) as follows:

$$\leq 2 \cdot \sum_{r=1}^{\ell} \frac{f_{j_r}(i|S_{j_r}^{i-1})}{2^r}. \tag{6}$$

We note that the inequality in (6) follows since we assumed $t \leq \ell$, *i.e.*, bidder $k$ is among the $\ell$ bidders with non-negative marginal values. Hence, the first sum in (5) can be extended to include all $\ell$ bidders with non-negative marginal value. Finally, we note that (6) equals the following by the definition of Algorithm 1:

$$=2 \cdot \mathbb{E}[P^i|R_{i-1}].$$

Hence, we can conclude that $\mathbb{E}[K^i|R_{i-1}] \leq 2 \cdot \mathbb{E}[P^i|R_{i-1}]$ as desired.

We note that if $t > \ell$ then the latter inequality trivially holds (the above proof works until (2) in which the first term is negative and thus can be dropped which implies that $\mathbb{E}[K^i|R_{i-1}] \leq \mathbb{E}[P^i|R_{i-1}]$ and hence that $\mathbb{E}[K^i|R_{i-1}] \leq 2 \cdot \mathbb{E}[P^i|R_{i-1}]$, since $P_i$ is non-negative). Thus, using the law of total expectation over all possible outcomes $R_{i-1}$ the proof is complete. ◀

## 2.2 Deterministic Hardness

**Proof of Theorem 2.** We present an instance for which any online deterministic algorithm cannot achieve a competitive ratio better than $1/M$ for every $M > 0$ versus an adversary. We consider an instance with a single bidder $B = \{b\}$ and two items $\mathcal{N} = \{v_1, v_2\}$. The items arrive according to their index in an online manner, *i.e.*, item $v_1$ is the first to arrive, and item $v_2$ is the second to arrive. Once item $v_1$ arrives, any deterministic algorithm can query $f$ on subsets of elements that can contain only $v_1$. Hence, we only need to define $f$ on $\emptyset$ and $\{v_1\}$. We define $f$ as follows: $f(\emptyset) = 0, f(\{v_1\}) = 1$. How this utility function is extended to a submodular function over all subsets of items depends on what the algorithm chooses to do once item $v_1$ arrives.

The first case is when the algorithm chooses not to assign $v_1$ to $b$. In this case we extend the above definition of $f$ by setting the contribution of $v_2$ to be linearly zero, *i.e*,, $f(\{v_2\}) \triangleq 0$ and $f(\{v_1, v_2\}) \triangleq 1$. One can verify that the resulting utility function $f$ is indeed submodular.

In this case, the value of an optimal solution to the instance equals 1, since item $v_1$ can be assigned to bidder $b$ by the optimal solution. However, every deterministic algorithm that does not assign $v_1$ to $b$ has a value of 0. Thus, we conclude a competitive ratio of 0 in this case.

The second case is when the algorithm chooses to assign $v_1$ to bidder $b$. In this case we extend the above definition of $f$ as follows: $f(\{v_2\}) \triangleq M$ and $f(\{v_1, v_2\}) \triangleq 0$. One can verify that the resulting utility function $f$ is indeed submodular. In this case, the value of an optimal solution to the instance equals $M$, since the optimal solution can choose to assign only $\{v_2\}$ to $b$. However, every deterministic algorithm that assigns $v_1$ to bidder $b$ achieves a value of at most 1. Thus, we conclude a competitive ratio of at most $1/M$ in this case.

In conclusion, for this instance, no deterministic algorithm can achieve any competitive ratio better than $1/M$ in the online adversarial setting, for every constant $M$. ◀

## 3 Uniform Random Order Setting

In this section we focus on the uniform random order setting. Recall that it is known that if the `RRG` algorithm provides an approximation of $\alpha$ for maximizing a general submodular function given a partition matroid independence constraint, then it also provides an online algorithm in the random order setting which achieves a competitive ratio of $\alpha$ (see brief discussion in Section 1).

To simplify the presentation of our improved analysis, we present a "smooth" version of the `RRG` algorithm of [10]. Though the analysis of the smoothed version is similar to the original `RRG` (assuming enough iterations are performed), we believe it is easier to analyze since all iterations have the same probabilistic distribution (whereas in the original `RRG` this is not the case). For simplicity of presentation, we focus on the case the matroid is a partition matroid (for a general matroid we defer to a full version of the paper). Recall that this special case already captures SW.

### 3.1 Partition Matroid

We are given a partition matroid $\mathcal{M} = (\mathcal{N}, \mathcal{I})$ over a ground set $\mathcal{N}$ which is partitioned into disjoint non-empty sets $P_1, P_2, \ldots, P_k$. The goal is to choose a subset $S \subseteq \mathcal{I}$, *i.e.*, $S$ contains at most one element from each set $P_j$, that maximizes a given non-negative (general) submodular function $f$.

We call our algorithm smooth since the random choices of the algorithm are always uniform, no matter how many iterations the algorithm performed so far. Specifically, the smooth algorithm chooses in every iteration a part uniformly at random from *all* $k$ parts of $\mathcal{N}$, and adds the best element in the chosen part assuming that part does not intersect what the algorithm chose so far. The original `RRG` chooses a part uniformly at random from parts that do not intersect what the algorithm chose so far, and adds the best element in the chosen part. Thus, the number of iterations the smooth algorithm can perform is unlimited. A formal description of the algorithm for partition matroids is given as Algorithm 2, and we note that the number of iterations $T$ is a parameter given to the algorithm. For a comparison of Algorithm 2 and the original `RRG` we defer to to a full version of the paper.

Given any $S \subseteq \mathcal{N}$, we denote the parts of the partition of $\mathcal{N}$ that do not intersect $S$ by $I(S)$ ,*i.e.*, $I(S) \triangleq \{j = 1, ..., k | P_j \cap S = \emptyset\}$. Without loss of generality, we assume every part $P_j$ is padded with a dummy element (a different dummy element for every $P_j$) that linearly contributes zero to the objective $f$. Hence, without loss of generality, $|OPT| = k$, where we denote by $OPT$ an optimal solution to the problem: $OPT \triangleq \arg\max\{f(S) | S \in \mathcal{I}\}$.

**Algorithm 2** Smooth Residual Random Greedy (Partition Matroid).

```
S_0 ← ∅.
for i = 1, ..., T do
    S_i ← S_{i-1}.
    M_i ← ⋃_{j∈I(S_{i-1})} {arg max {f(S_{i-1} ∪ {u}) − f(S_{i-1}) | u ∈ P_j}}.
    Let j be a uniformly random number from {1, ..., k}.
    if j ∈ I(S_{i-1}) then
        Let u_i be the element from P_j in M_i and S_i ← S_i ∪ {u_i}.
    end
end
Return S_T.
```

For every set $S \in \mathcal{I}$ we denote $O_S \triangleq \arg\max_A\{f(S \cup A) | S \cup A \in \mathcal{I}, A \subseteq \mathcal{N} \setminus S\}$, *i.e.*, $O_S$ is the best extension of $S$ to an independent set. Recalling that every part $P_j$ is padded with a dummy element that linearly contributes zero to the objective implies that $|O_S| = k - |S|$. One should note that $O_\emptyset = OPT$, and thus $O_{S_0} = OPT$. Our analysis tracks $f(O_{S_i} \cup S_i)$ as $S_i$ changes throughout the algorithm. Intuitively, $f(O_{S_i} \cup S_i)$ deteriorate as more elements are added to $S_i$. Building on the above intuition, the following two lemmas establish a system of joint recursive formulas for $\mathbb{E}[f(S_i)]$ and $\mathbb{E}[f(O_{S_i} \cup S_i)]$.

▶ **Lemma 7.** *For every $i = 1, \ldots, T$:*

$$\mathbb{E}[f(S_i)] - \mathbb{E}[f(S_{i-1})] \geq \frac{1}{k} \cdot \mathbb{E}\left[f\left(O_{S_{i-1}} \cup S_{i-1}\right) - f(S_{i-1})\right].$$

**Proof.** Fix $i = 1, \ldots, T$ and condition on any possible realization of the choices of the algorithm in the first $i-1$ iterations. Thus, $S_{i-1}$, $M_i$, and $O_{S_{i-1}}$ are deterministic and fixed given this conditioning, and $u_i$ and $S_i$ are the only random variables. For the remainder of the proof all the probabilities and expectations are conditioned on this possible realization.

$$\mathbb{E}[f(S_i)] - f(S_{i-1}) = \frac{1}{k} \sum_{u \in M_i} \{f(S_{i-1} \cup \{u\}) - f(S_{i-1})\} \tag{7}$$

$$\geq \frac{1}{k} \sum_{u \in O_{S_{i-1}}} \{f(S_{i-1} \cup \{u\}) - f(S_{i-1})\} \tag{8}$$

$$\geq \frac{1}{k} \{f(S_{i-1} \cup O_{S_{i-1}}) - f(S_{i-1})\} \tag{9}$$

In the above, the equality in (7) follows from the algorithm's definition. The inequality in (8) follows from the greedy choice of $M_i$, and the inequality in (9) follows from the submodularity of $f$. We conclude the proof by unfixing the conditioning and taking an expectation over all possible such events (the law of total expectation). ◀

The following lemma provides a recursive formula for $\mathbb{E}[f(O_{S_i} \cup S_i)]$ whose novelty is in the added contribution of $f(S_{i-1})$. Without the added $\mathbb{E}[f(S_{i-1})]/k$ term the resulting approximation will be 1/4 as in [10].

▶ **Lemma 8.** *For every $i = 1, \ldots, T$:*

$$\mathbb{E}[f(O_{S_i} \cup S_i)] \geq \left(1 - \frac{2}{k}\right) \cdot \mathbb{E}\left[f\left(O_{S_{i-1}} \cup S_{i-1}\right)\right] + \frac{1}{k} \cdot \mathbb{E}[f(S_{i-1})].$$

**Proof.** Fix $i = 1, \dots, T$ and condition on any possible realization of the choices of the algorithm in the first $i-1$ iterations. Thus, $S_{i-1}$, $M_i$, and $O_{S_{i-1}}$ are deterministic and fixed given this conditioning, and $u_i$, $S_i$, and $O_{S_i}$ are the only random variables. For the remainder of the proof all the probabilities and expectations are conditioned on this possible realization.

For every element $u \in M_i$ we denote by $P(u)$ the index of the part of the partition of $\mathcal{N}$ which $u$ belongs to. Formally, $P(u) = j$ if and only if $u \in P_j$. We define $h : M_i \to O_{S_{i-1}}$ to be a bijection mapping every element $u \in M_i$ to an element of $O_{S_{i-1}}$ in such a way that $P(u) = P(h(u))$. One should note that our assumption that every $P_j$ contains a dummy element that contributes zero to the objective implies that $I(S_{i-1}) = I(O_{S_{i-1}})$, and hence $h$ is well defined. Thus,

$$\begin{aligned}
&\mathbb{E}\left[f\left(O_{S_i} \cup S_i\right)\right] - f\left(O_{S_{i-1}} \cup S_{i-1}\right)\\
=&\frac{1}{k}\sum_{u \in M_i}\left\{f\left(S_{i-1} \cup \{u\} \cup O_{S_{i-1}\cup\{u\}}\right) - f\left(O_{S_{i-1}} \cup S_{i-1}\right)\right\} \quad (10)\\
\geq&\frac{1}{k}\sum_{u \in M_i}\left\{f\left(S_{i-1} \cup \{u\} \cup (O_{S_{i-1}} \setminus \{h(u)\})\right) - f\left(O_{S_{i-1}} \cup S_{i-1}\right)\right\}. \quad (11)
\end{aligned}$$

In the above, the equality in (10) follows from the algorithm's definition. The inequality in (11) follows from the observation that $O_{S_{i-1}\cup\{u\}}$ is the best extension of $S_{i-1} \cup \{u\}$ whereas $O_{S_{i-1}} \setminus \{h(u)\}$ is just an extension of $S_{i-1} \cup \{u\}$, *i.e.*,

$$f(S_{i-1} \cup \{u\} \cup O_{S_{i-1}\cup\{u\}}) \geq f(S_{i-1} \cup \{u\} \cup (O_{S_{i-1}} \setminus \{h(u)\})).$$

We note that (11) equals the following:

$$=\frac{1}{k}\sum_{u \in M_i | u \neq h(u)}\left\{f\left(S_{i-1} \cup \{u\} \cup (O_{S_{i-1}} \setminus \{h(u)\})\right) - f\left(O_{S_{i-1}} \cup S_{i-1}\right)\right\}. \quad (12)$$

The equality in (12) holds since if $u = h(u)$ then $\{u\} \cup (O_{S_{i-1}} \setminus \{h(u)\}) = O_{S_{i-1}}$, and therefore summation can be reduced to all candidate elements $u \in M_i$ satisfying $u \neq h(u)$. Moreover, we note that (12) can be lower bounded as follows:

$$\begin{aligned}
\geq&\frac{1}{k}\sum_{u \in M_i | u \neq h(u)}\left\{f\left(S_{i-1} \cup \{u\} \cup O_{S_{i-1}}\right) - f\left(S_{i-1} \cup O_{S_{i-1}}\right)\right\}+ \quad (13)\\
&\frac{1}{k}\sum_{u \in M_i | u \neq h(u)}\left\{f\left(S_{i-1} \cup (O_{S_{i-1}} \setminus \{h(u)\})\right) - f\left(S_{i-1} \cup O_{S_{i-1}}\right)\right\}.
\end{aligned}$$

The inequality in (13) follows from submodularity since summation is restricted only to candidates $u \in M_i$ satisfying $u \neq h(u)$ and hence: $f(S_{i-1} \cup \{u\} \cup (O_{S_{i-1}} \setminus \{h(u)\})) + f(S_{i-1} \cup O_{S_i-1}) \geq f(S_{i-1} \cup \{u\} \cup O_{S_{i-1}}) + f(S_{i-1} \cup (O_{S_{i-1}} \setminus \{h(u)\}))$.

We further lower bound (13) in the following way:

$$\begin{aligned}
\geq&\frac{1}{k}\sum_{u \in M_i}\left\{f\left(S_{i-1} \cup \{u\} \cup O_{S_{i-1}}\right) - f\left(S_{i-1} \cup O_{S_{i-1}}\right)\right\}+ \quad (14)\\
&\frac{1}{k}\sum_{u \in M_i}\left\{f\left(S_{i-1} \cup (O_{S_{i-1}} \setminus \{h(u)\})\right) - f\left(S_{i-1} \cup O_{S_{i-1}}\right)\right\}.
\end{aligned}$$

When examining the inequality in (14) let us start with the first sum. We note that if $u = h(u)$ for some $u \in M_i$ then $u \in O_{S_{i-1}}$, *i.e.*, $\{u\} \cup O_{S_{i-1}} = O_{S_{i-1}}$. Thus, extending the first sum to all $u \in M_i$ (regardless of whether $u$ equals $h(u)$ or not) does not change the first sum. Focusing on the second sum, we note that for every $u \in M_i$, regardless of whether $u$ equals $h(u)$ or not, the following holds: $f(S_{i-1} \cup (O_{S_{i-1}} \setminus \{h(u)\})) \leq f(S_{i-1} \cup O_{S_{i-1}})$. The reason for the latter is that $O_{S_{i-1}}$ is the best extension of $S_{i-1}$ whereas $O_{S_{i-1}} \setminus \{h(u)\}$ is some extension of $S_{i-1}$. Hence, adding to the second sum all terms corresponding to $u \in M_i$, where $u = h(u)$, can only decrease the second sum. Therefore, we conclude that the inequality in (14) holds. Finally, we lower bound (14) as follows:

$$\geq \frac{1}{k}\{f\left(S_{i-1} \cup O_{S_{i-1}} \cup M_i\right) - f\left(S_{i-1} \cup O_{S_{i-1}}\right)\}+ \tag{15}$$

$$\frac{1}{k}\{f\left(S_{i-1} \cup O_{S_{i-1}} \setminus O_{S_{i-1}}\right) - f\left(S_{i-1} \cup O_{S_{i-1}}\right)\}$$

$$\geq \frac{1}{k}\{f(S_{i-1}) - 2 \cdot \left(f\left(S_{i-1} \cup O_{S_{i-1}}\right)\right)\} \tag{16}$$

The inequality in (15) follows from submodularity which implies the following two:

$$\sum_{u \in M_i} \{f\left(S_{i-1} \cup \{u\} \cup O_{S_{i-1}}\right) - f\left(S_{i-1} \cup O_{S_{i-1}}\right)\} \geq f\left(S_{i-1} \cup O_{S_{i-1}} \cup M_i\right) - f\left(S_{i-1} \cup O_{S_{i-1}}\right)$$

$$\sum_{u \in M_i} \{f\left(S_{i-1} \cup O_{S_{i-1}} \setminus \{h(u)\}\right) - f\left(S_{i-1} \cup O_{S_{i-1}}\right)\} \geq f\left(S_{i-1} \cup O_{S_{i-1}} \setminus O_{S_{i-1}}\right) - f\left(S_{i-1} \cup O_{S_{i-1}}\right).$$

We note that the inequality in (16) follows from the non-negativity of $f$ which implies that: $f\left(S_{i-1} \cup O_{S_{i-1}} \cup M_i\right) \geq 0$ and the fact that $S_{i-1} \cup O_{S_{i-1}} \setminus O_{S_{i-1}} = S_{i-1}$.

We conclude the proof by unfixing the conditioning and taking an expectation over all possible such events (the law of total expectation). ◀

The following lemma lower bounds the solution to the system of joint recursive formulas presented in Lemmas 7 and 8 (its proof is deferred to a full version of the paper). For simplicity of presentation we introduce two absolute constants: $a \triangleq (3 - \sqrt{5})/2 \approx 0.381966$ and $b \triangleq (3 + \sqrt{5})/2 \approx 2.61803$.

▶ **Lemma 9.** *For every $i = 0, 1, \ldots, T$ the following hold:*

$$\mathbb{E}[f(S_i)] \geq \frac{f(OPT)}{\sqrt{5}} \left(\left(1 - \frac{a}{k}\right)^i - \left(1 - \frac{b}{k}\right)^i\right) \tag{17}$$

$$\mathbb{E}\left[f(O_{S_i} \cup S_i)\right] \geq \frac{f(OPT)}{2\sqrt{5}} \left((\sqrt{5} - 1)\left(1 - \frac{a}{k}\right)^i + (\sqrt{5} + 1)\left(1 - \frac{b}{k}\right)^i\right). \tag{18}$$

The following lemma establishes the approximation guarantee of Algorithm 2.

▶ **Lemma 10.** *Algorithm 2 achieves an approximation ratio of at least* 0.27493 *for the problem of maximizing a non-monotone submodular function subject to a partition matroid independence constraint.*

**Proof.** By our assumption of the existence of dummy elements, no element of $M_i$ has a negative marginal value, in any iteration $i$. We get that always for every $i$, $f(S_i) \geq f(S_{i-1})$ (note that this inequality holds for the random variables $S_i$ and $S_{i-1}$). Therefore, it suffices to show that $\mathbb{E}[f(S_i)] \geq 0.274 \cdot f(OPT)$ for some $i = 1, \ldots, T$.

Observe that setting $i = x^*k$, where $x^* = \ln(b/a)/\sqrt{5} = \ln\left((3 + \sqrt{5})/(3 - \sqrt{5})\right)/\sqrt{5} \approx 0.86$, alongside Lemma 9, implies that:

$$\mathbb{E}[f(S_{x^*k})] \geq \frac{f(OPT)}{\sqrt{5}} \left(\left(1 - \frac{a}{k}\right)^{x^*k} - \left(1 - \frac{b}{k}\right)^{x^*k}\right) \geq \frac{f(OPT)}{\sqrt{5}} \left(e^{-x^*a} - e^{-x^*b}\right). \tag{19}$$

The inequality above follows from the observation that $(1 - a/k)^{x^*k} - (1 - b/k)^{x^*k}$, as a function of $k$, is monotone decreasing for every $k \geq 3$ (the proof of this technical observation is deferred to a full version of the paper). Thus, the inequality follows by taking the limit $k \to \infty$. The lemma follows by plugging in the above values of $a$, $b$, and $x^*$. ◀

**Proof of Theorem 3.** Follows immediately from Lemma 10. ◀

## References


1 Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. *Online Vertex-Weighted Bipartite Matching and Single-bid Budgeted Allocations*, pages 1253–1264. SIAM, 2011. `doi:10.1137/1.9781611973082.95`.

2 Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. A dynamic near-optimal algorithm for online linear programming. *Operations Research*, 62(4):876–890, 2014.

3 Moshe Babaioff, Nicole Immorlica, and Robert Kleinberg. Matroids, secretary problems, and online mechanisms. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 434–443, USA, 2007. Society for Industrial and Applied Mathematics.

4 Ashwinkumar Badanidiyuru and Jan Vondrák. Fast algorithms for maximizing submodular functions. In *Proceedings of the Twenty-fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 1497–1514, 2014.

5 Siddharth Barman, Seeun Umboh, Shuchi Chawla, and David Malec. Secretary problems with convex costs. In Artur Czumaj, Kurt Mehlhorn, Andrew Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming*, pages 75–87, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

6 Mohammadhossein Bateni, Mohammadtaghi Hajiaghayi, and Morteza Zadimoghaddam. Submodular secretary problem and extensions. *ACM Trans. Algorithms*, 9(4), October 2013. `doi:10.1145/2500121`.

7 N. Buchbinder, M. Feldman, J. Naor, and R. Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. *SIAM Journal on Computing*, 44(5):1384–1402, 2015.

8 Niv Buchbinder and Moran Feldman. Constrained submodular maximization via a nonsymmetric technique. *Mathematics of Operations Research*, 44(3):988–1005, 2019.

9 Niv Buchbinder, Moran Feldman, Yuval Filmus, and Mohit Garg. Online submodular maximization: Beating 1/2 made simple. *Mathematical Programming*, 183(1-2):149–169, 2020.

10 Niv Buchbinder, Moran Feldman, Joseph (Seffi) Naor, and Roy Schwartz. Submodular maximization with cardinality constraints. In *Proceedings of the Twenty-fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 1433–1452, 2014.

11 Niv Buchbinder, Moran Feldman, and Roy Schwartz. Online submodular maximization with preemption. *ACM Trans. Algorithms*, 15(3), June 2019. `doi:10.1145/3309764`.

12 Niv Buchbinder, Kamal Jain, and Joseph (Seffi) Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In Lars Arge, Michael Hoffmann, and Emo Welzl, editors, *Algorithms – ESA 2007*, pages 253–264, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

13 Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint (extended abstract). In Matteo Fischetti and David P. Williamson, editors, *Integer Programming and Combinatorial Optimization*, pages 182–196, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

14 Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011.

15 Deeparnab Chakrabarty and Gagan Goel. On the approximability of budgeted allocations and improved lower bounds for submodular welfare maximization and gap. *SIAM Journal on Computing*, 39(6):2189–2211, 2010. `doi:10.1137/080735503`.

16 Nikhil R. Devanur and Thomas P. Hayes. The adwords problem: Online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM Conference on Electronic Commerce*, EC '09, pages 71–78, New York, NY, USA, 2009. Association for Computing Machinery. doi:10.1145/1566374.1566384.

17 Nikhil R Devanur, Zhiyi Huang, Nitish Korula, Vahab S Mirrokni, and Qiqi Yan. Whole-page optimization and submodular welfare maximization with online bidders. *ACM Transactions on Economics and Computation (TEAC)*, 4(3):1–20, 2016.

18 Shahar Dobzinski, Noam Nisan, and Michael Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. *Mathematics of Operations Research*, 35(1):1–13, 2010. doi:10.1287/moor.1090.0436.

19 Shahar Dobzinski and Michael Schapira. An improved approximation algorithm for combinatorial auctions with submodular bidders. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm*, SODA '06, pages 1064–1073, USA, 2006. Society for Industrial and Applied Mathematics.

20 E. B. Dynkin. The optimum choice of the instant for stopping a markov process, 1963.

21 Alina Ene and Huy L. Nguyen. Constrained submodular maximization: Beyond 1/e. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 248–257, 2016. doi:10.1109/FOCS.2016.34.

22 Alina Ene and Huy L. Nguyen. A nearly-linear time algorithm for submodular maximization with knapsack, partition and graphical matroid constraints. *CoRR*, abs/1709.09767, 2018.

23 Uriel Feige. On maximizing welfare when utility functions are subadditive. *SIAM Journal on Computing*, 39(1):122–142, 2009. doi:10.1137/070680977.

24 Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011.

25 Uriel Feige and Jan Vondrak. Approximation algorithms for allocation problems: Improving the factor of 1 - 1/e. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 667–676, 2006. doi:10.1109/FOCS.2006.14.

26 Uriel Feige and Jan Vondrák. The submodular welfare problem with demand queries. *Theory of Computing*, 6(11):247–290, 2010. doi:10.4086/toc.2010.v006a011.

27 Jon Feldman, Monika Henzinger, Nitish Korula, Vahab S Mirrokni, and Cliff Stein. Online stochastic packing applied to display ad allocation. In *European Symposium on Algorithms*, pages 182–194. Springer, 2010.

28 Jon Feldman, Nitish Korula, Vahab Mirrokni, S. Muthukrishnan, and Martin Pál. Online ad assignment with free disposal. In Stefano Leonardi, editor, *Internet and Network Economics*, pages 374–385, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

29 Jon Feldman, Aranyak Mehta, Vahab Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 117–126, 2009. doi:10.1109/FOCS.2009.72.

30 Moran Feldman. Maximizing symmetric submodular functions. *ACM Transactions on Algorithms (TALG)*, 13(3):1–36, 2017.

31 Moran Feldman, Joseph Naor, and Roy Schwartz. A unified continuous greedy algorithm for submodular maximization. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 570–579, 2011. doi:10.1109/FOCS.2011.46.

32 Moran Feldman, Joseph (Seffi) Naor, and Roy Schwartz. Nonmonotone submodular maximization via a structural continuous greedy algorithm. In *ICALP*, pages 342–353, 2011.

33 Moran Feldman, Ola Svensson, and Rico Zenklusen. *A Simple O(log log(rank))-Competitive Algorithm for the Matroid Secretary Problem*, pages 1189–1201. SIAM, 2015. doi:10.1137/1.9781611973730.79.

34 Shayan Oveis Gharan and Jan Vondrák. Submodular maximization by simulated annealing. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1098–1116. SIAM, 2011.

35 Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *SODA*, volume 8, pages 982–991. Citeseer, 2008.

36 Corinna Gottschalk and Britta Peis. Submodular function maximization on the bounded integer lattice. In *Approximation and Online Algorithms*. Springer International Publishing, 2015.

37 Anupam Gupta, Aaron Roth, Grant Schoenebeck, and Kunal Talwar. Constrained non-monotone submodular maximization: Offline and secretary algorithms. In Amin Saberi, editor, *Internet and Network Economics*, pages 246–257, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

38 Christopher Harshaw, Ehsan Kazemi, Moran Feldman, and Amin Karbasi. The power of subsampling in submodular maximization. *Mathematics of Operations Research*, 47(2):1365–1393, 2022.

39 Xinran He and Yan Liu. Not enough data? joint inferring multiple diffusion networks via network generation priors. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 465–474, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3018661.3018675.

40 Bala Kalyanasundaram and Kirk R. Pruhs. An optimal deterministic algorithm for online b-matching. *Theoretical Computer Science*, 233(1):319–325, 2000. doi:10.1016/S0304-3975(99)00140-1.

41 Mikhail Kapralov, Ian Post, and Jan Vondrák. Online and stochastic variants of welfare maximization. *CoRR*, abs/1204.1025, 2012. arXiv:1204.1025.

42 R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, STOC '90, pages 352–358, New York, NY, USA, 1990. Association for Computing Machinery. doi:10.1145/100216.100262.

43 Subhash Khot, Richard J. Lipton, Evangelos Markakis, and Aranyak Mehta. Inapproximability results for combinatorial auctions with submodular utility functions. In *Proceedings of the First International Conference on Internet and Network Economics*, WINE'05, pages 92–101, 2005.

44 Samir Khuller, Anna Moss, and Joseph (Seffi) Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999. doi:10.1016/S0020-0190(99)00031-9.

45 Robert Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pages 630–631, USA, 2005. Society for Industrial and Applied Mathematics.

46 Nitish Korula, Vahab Mirrokni, and Morteza Zadimoghaddam. Online submodular welfare maximization: Greedy beats 1/2 in random order. *SIAM Journal on Computing*, 47(3):1056–1086, 2018. doi:10.1137/15M1051142.

47 Andreas Krause and Carlos Guestrin. A note on the budgeted maximization of submodular functions. *Technical Report CMU-CALD-05-103*, 2005.

48 Ariel Kulik, Hadas Shachnai, and Tami Tamir. Approximations for monotone and nonmonotone submodular maximization with knapsack constraints. *Math. Oper. Res.*, 38(4):729–739, November 2013.

49 Oded Lachish. O(log log rank) competitive ratio for the matroid secretary problem. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 326–335, 2014. doi:10.1109/FOCS.2014.42.

50 Jon Lee, Vahab S Mirrokni, Viswanath Nagarajan, and Maxim Sviridenko. Maximizing nonmonotone submodular functions under matroid or knapsack constraints. *SIAM Journal on Discrete Mathematics*, 23(4):2053–2078, 2010.

51 Jon Lee, Maxim Sviridenko, and Jan Vondrák. Submodular maximization over multiple matroids via generalized exchange properties. *Math. Oper. Res.*, 35(4):795–806, November 2010.

52 Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. In *Proceedings of the 3rd ACM Conference on Electronic Commerce*, EC '01, pages 18–28, 2001.

53 Hui Lin and Jeff Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 912–920, 2010.

54 Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 510–520, 2011.

55 Shengzhong Liu, Zhenzhe Zheng, Fan Wu, Shaojie Tang, and Guihai Chen. Context-aware data quality estimation in mobile crowdsensing. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pages 1–9, 2017. doi:10.1109/INFOCOM.2017.8057033.

56 Tengyu Ma, Bo Tang, and Yajun Wang. The simulated greedy algorithm for several submodular matroid secretary problems. *Theory of Computing Systems*, 58(4):681–706, May 2016. doi:10.1007/s00224-015-9642-4.

57 Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: An approach based on strongly factor-revealing lps. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, STOC '11, pages 597–606, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/1993636.1993716.

58 Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *J. ACM*, 54(5):22–es, October 2007. doi:10.1145/1284320.1284321.

59 Baharan Mirzasoleiman, Stefanie Jegelka, and Andreas Krause. Streaming non-monotone submodular maximization: Personalized video summarization on the fly. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018.

60 G. L. Nemhauser and L. A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research*, 3(3):177–188, 1978. doi:10.1287/moor.3.3.177.

61 G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions–i. *Math. Program.*, 14(1):265–294, December 1978. doi:10.1007/BF01588971.

62 Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*, 129(1):192–224, 2006. doi:10.1016/j.jet.2004.10.007.

63 Xinghao Pan, Stefanie Jegelka, Joseph E Gonzalez, Joseph K Bradley, and Michael I Jordan. Parallel double greedy submodular maximization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 118–126. Curran Associates, Inc., 2014.

64 Ramakumar Pasumarthi, Ramasuri Narayanam, and Balaraman Ravindran. Near optimal strategies for targeted marketing in social networks. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pages 1679–1680, Richland, SC, 2015. International Foundation for Autonomous Agents and Multiagent Systems.

65 Paulo Shakarian, Joseph Salmento, William Pulleyblank, and John Bertetto. Reducing gang violence through network influence based targeting of social programs. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1829–1836, 2014.

66 Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Oper. Res. Lett.*, 32(1):41–43, January 2004.

**67** Erik Vee, Sergei Vassilvitskii, and Jayavel Shanmugasundaram. Optimal online assignment with forecasts. In *Proceedings of the 11th ACM conference on Electronic commerce*, pages 109–118, 2010.

**68** J. Vondrák. Symmetry and approximability of submodular maximization problems. *SIAM Journal on Computing*, 42(1):265–304, 2013.

**69** Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 67–74, 2008.

**70** Jan Vondrák, Chandra Chekuri, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 783–792, 2011.