Online Submodular Welfare Maximization: Greedy Beats 1/2 in Random Order

Nitish Korula
nitish@google.com

Vahab Mirrokni
mirrokni@google.com

Morteza Zadimoghaddam
zadim@google.com

ABSTRACT

In the Submodular Welfare Maximization (SWM) problem, the input consists of a set of n items, each of which must be allocated to one of m agents. Each agent ℓ has a valuation function v_ℓ, where $v_\ell(S)$ denotes the welfare obtained by this agent if she receives the set of items S. The functions v_ℓ are all submodular; as is standard, we assume that they are monotone and $v_\ell(\emptyset) = 0$. The goal is to partition the items into m disjoint subsets $S_1, S_2, \dots S_m$ in order to maximize the social welfare, defined as $\sum_{\ell=1}^{m} v_\ell(S_\ell)$. A simple greedy algorithm gives a 1/2-approximation to SWM in the offline setting, and this was the best known until Vondrak's recent $(1 - 1/e)$-approximation algorithm [34].

In this paper, we consider the online version of SWM. Here, items arrive one at a time in an online manner; when an item arrives, the algorithm must make an irrevocable decision about which agent to assign it to before seeing any subsequent items. This problem is motivated by applications to Internet advertising, where user ad impressions must be allocated to advertisers whose value is a submodular function of the set of users / impressions they receive. There are two natural models that differ in the order in which items arrive. In the fully *adversarial* setting, an adversary can construct an arbitrary / worst-case instance, as well as pick the order in which items arrive in order to minimize the algorithm's performance. In this setting, the 1/2-competitive greedy algorithm is the best possible. To improve on this, one must weaken the adversary slightly: In the *random order* model, the adversary can construct a worst-case set of items and valuations, but does not control the order in which the items arrive; instead, they are assumed to arrive in a random order. The random order model has been well studied for online SWM and various special cases, but the best known competitive ratio (even for several special cases) is $1/2 + 1/n$ [9, 10], barely better than the ratio for the adversarial order. Obtaining a competitive ratio of $1/2 + \Omega(1)$ for the random order model has been an important open problem for several years. We solve this open problem by demonstrating that the greedy algorithm has a competitive ratio of at least 0.505 for online SWM in the random order model. This is the first result showing a competitive ratio bounded above 1/2 in the random order model, even for special cases such as the weighted matching or budgeted allocation problems (without the so-called 'large capacity' assumptions). For special cases of submodular functions including weighted matching, weighted coverage functions and a broader class of "second-order supermodular" functions, we provide a different analysis that gives a competitive ratio of 0.51. We analyze the greedy algorithm using a factor-revealing linear program, bounding how the assignment of one item can decrease potential welfare from assigning future items. We also formulate a natural conjecture which, if true, would improve the competitive ratio of the greedy algorithm to at least 0.567.

In addition to our new competitive ratios for online SWM, we make two further contributions: First, we define the classes of *second-order* modular, supermodular, and submodular functions, which are likely to be of independent interest in submodular optimization. Second, we obtain an improved competitive ratio via a technique we refer to as *gain linearizing*, which may be useful in other contexts (see [26]): Essentially, we linearize the submodular function by dividing the gain of an optimal solution into gain from individual elements, compare the algorithm's gain when it assigns an element to the optimal solution's gain from the element, and, crucially, bound the extent to which assigning elements can affect the potential gain of other elements.

STOC'15, June 14–17, 2015, Portland, Oregon, USA.
ACM 978-1-4503-3536-2/15/06
http://dx.doi.org/10.1145/2746539.2746626.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; G.2.2 [**Discrete Mathematics**]: Combinatorics—*Combinatorial algorithms*

Keywords

Submodular Welfare Maximization; Online algorithms; Online SWM; random-order; submodular optimization; second-order submodular functions

1. INTRODUCTION

As a general abstraction of many economic resource-allocation problems, *submodular welfare maximization* (abbreviated as SWM) is a central optimization problem in combinatorial auctions and has attracted significant attention in the research area at the intersection of economics, game theory, and computer science. In this problem, an auctioneer sells a set N of n items to a set M of m agents. The value of agent $\ell \in M$ for any subset (bundle) of items is given by a submodular valuation set function $v_\ell : 2^N \to \mathbb{R}_+$, where $v_\ell(S)$ represents i's maximum willingness to pay for the bundle S. The two standard assumptions on each v_ℓ (besides submodularity) are that if $S \subseteq T$ then $v_\ell(S) \leq v_\ell(T)$ (monotonicity), and that $v_\ell(\emptyset) = 0$ (normalization). The objective is to partition N into m disjoint subsets $S_1, S_2, \ldots, S_m$, and give set S_ℓ to agent ℓ in a way that maximizes *the social welfare*, i.e. the expression $\sum_{\ell=1}^{m} v_\ell(S_\ell)$.

In the online version of the problem (also known as the *online submodular welfare maximization*), referred to as *online SWM*, items in N arrive one by one online, and upon arrival of an item, it must be assigned immediately and irrevocably to one of the agents. That is, the assignment of an item must be made before any subsequent items arrive, and it may not be changed later.

The online SWM problem is a natural generalization of the online matching [20, 17, 13, 1, 23], budgeted allocation [24, 4, 5, 15] and online weighted matching problems [1, 11], along with more general classes of online allocation / assignment problems [12, 33, 2, 6]. Besides being theoretically important, these problems have a number of practical applications including internet advertising, network routing, etc. These online allocation problems have been studied in both worst-case / *adversarial* and *stochastic* settings. In the adversarial arrival model, an adversary constructs a worst-case instance, and can order the items arbitrarily in order to minimize the algorithm's performance. In contrast, in the *random order* arrival model, the adversary can construct an arbitrary instance, but the order in which items arrive is considered to be chosen uniformly at random. Here, the performance of the algorithm is computed as the average over the random choice of the arrival order of the items.

The Submodular Welfare Maximization problem has been studied extensively as both an offline and an online optimization problem: For the offline optimization problem, one can easily observe that SWM is a special case of the monotone submodular maximization problem subject to a (partition) matroid constraint. As a result, an old result of Nemhauser, Wolsey, and Fisher [30] imply that a simple greedy algorithm achieves a $1/2$-approximation. Improving this approximation factor for the offline SWM was open until Vondrak [34] presented a new $1 - 1/e$-approximation algorithm for the problem; this is the best possible using a sub-exponential number of oracle calls [28].

For the online SWM problem, a simple online greedy algorithm (assign each item to the agent whose marginal valuation increases the most) achieves a competitive ratio of $1/2$ for the adversarial model [14, 21]. As online SWM and its special cases are of considerable theoretical and practical interest, there has been a large body of work trying to improve upon this competitive ratio in both the adversarial and random arrival models. For example, in the adversarial model, $1 - 1/e$-competitive algorithms have been achieved for the special case of the online matching problem, as well as the the budgeted allocation and online weighted matching problems under the so-called large capacity assumption [20, 24, 11]. However, such a result is not possible for the general online SWM problem in the adversarial setting, where the simple greedy algorithm is the best possible: A recent result by Kapralov, Post, and Vondrak [18] shows that this $1/2$-approximation is tight for online SWM unless NP=RP. This hardness result, however, does not rule out improving the approximation factor of $1/2$ for the random order model. In fact, getting a $1/2 + \Omega(1)$ approximation factor remains an open problem even for special cases of online SWM such as the budgeted allocation problem, and weighed matching with free disposal [11, 6].

Our Contributions and Techniques. In this paper, we resolve the open problem of obtaining an improved competitive ratio for random order online SWM:

Theorem 1. *The Greedy algorithm has competitive ratio at least* 0.5052 *for online SWM in the random order model.*

Prior to our work, the best known algorithm for this problem (even for special cases such as weighted matching or budgeted allocation) gave a $\frac{1}{2} + \frac{1}{n}$-approximation [9, 10]. Thus, our result is also the first $1/2 + \Omega(1)$-competitive algorithm for the budgeted allocation and the online weighted matching problems under the random order model. Beating the factor of $1/2$ for the online weighted matching and budgeted allocation problems in either the adversarial or the random order model (without the large capacity constraints) has remained a major open problem in the area.

Further, for a broad class of submodular functions, we can strengthen our analysis. To do this, we introduce the concept of *second-order supermodular* functions; to the best of our knowledge, this class of functions has not been explicitly studied before.

Definition 2. *For a submodular function f, let $MG(A, e) = f(A \cup \{e\}) - f(A)$ denote the marginal gain from adding element e to set A. For sets A, S, we define $GR(A, S, e) = MG(A, e) - MG(A \cup S, e)$ as the amount by which S reduces the marginal gain from adding e to A. (Here, GR stands for Gain Reduction.) Note that by definition of submodularity, $GR(A, S, e)$ is always non-negative.*

- *The function f is said to be* second-order modular *if, for all sets A, B, S such that $A \subseteq B$, and $S \cap B = \emptyset$, and all elements e, we have: $GR(A, S, e) = GR(B, S, e)$.*
- *The function f is* second-order supermodular *if, for all sets A, B, S such that $A \subseteq B$, and $S \cap B = \emptyset$, and all elements e, we have: $GR(A, S, e) \geq GR(B, S, e)$. Equivalently, $MG(A, e) - MG(B, e) \geq MG(A \cup S, e) - MG(B \cup S, e)$.*
- *The function f is said to be* second-order submodular *if, for all sets A, B, S such that $A \subseteq B$, and $S \cap B = \emptyset$, and all elements e, we have: $GR(A, S, e) \leq GR(B, S, e)$.*

It is well known that when considering the multilinear extension of f, submodularity corresponds to a non-positive second-order derivative. Our definition of second-order submodularity implies non-positive *third-order* derivatives, while the definition of second-order supermodularity implies a non-negative third-order derivative.

We believe these classes are likely to be of independent interest, as they help partition the space of submodular functions, and may refine our understanding of submodular optimization. Several natural submodular functions can be classified in this framework: For example, cut functions are second-order modular, while weighted coverage functions and weighted matching functions are second-order supermodular.

THEOREM 3. *The Greedy algorithm has competitive ratio at least* 0.5104 *for online SWM in the random order model if the valuation functions of agents are second-order supermodular functions.*

Note that our results also imply simple $1/2+\Omega(1)$ approximation algorithms for the offline SWM problems: simply permute the items randomly and apply the online Greedy algorithm.[1] We focus on the Greedy algorithm for several reasons: It is simple and natural, besides being easy to implement and likely to be used in practice. Further, it is optimal in the adversarial setting. Algorithms that perform well in both adversarial and stochastic settings (see, for instance, [27]) are of considerable practical utility; by showing that the Greedy algorithm achieves a ratio better than 1/2 in the random order model, we provide further justification for its practical importance. Moreover, the Greedy algorithm for online SWM has been extensively studied in strategic settings [22, 31, 32].

Our approach to analyzing the performance of the Greedy algorithm is to understand how item allocations interact via the technique we call *gain linearizing*: We first formulate some basic inequalities about *any* greedy allocation which yield another proof that the Greedy algorithm achieves a competitive ratio of 1/2. To go beyond this ratio, we note that the assignment of an item can reduce the expected gains from future items, but this reduction in future gains is bounded, and the random order ensures that it is 'evenly spread out' among future items. We formalize this intuition in Section 2; once we can bound such interactions, we derive constraints for a factor-revealing Linear Program, which we explicitly analyze to get a lower bound on the competitive ratio of the Greedy algorithm.

1.1 Related Work

The online submodular welfare maximization problem is a generalization of various well-studied online allocation problems, including problems with practical applications to Internet advertising. These include online weighted b-matching (with free disposal), also referred to as the *Display Ads Allocation* problem [11, 12, 2, 33], and budgeted allocation or the *AdWords* problem [24, 5]. In both of these problems, a publisher must assign online impressions to a set of agents corresponding to advertisers; the goal is to optimize efficiency or revenue of the allocation while respecting pre-specified contracts with advertisers. Both of these problems have been studied in the competitive adversarial model [24, 11, 4] and the stochastic random order model [5, 12, 2, 33].

In the online weighted b-matching problem with free disposal, motivated by display advertising, we are given a set of m advertisers; advertiser j has a set S_j of eligible impressions and demand of at most $N(j)$ impressions. The ad-serving algorithm must allocate the set of n impressions that arrive online. Each impression i has value $w(i,j) \geq 0$ for advertiser j. The goal of the algorithm is to assign each impression to at most one advertiser, while maximizing the value of all the assigned impressions and ensuring that advertiser j does not receive more than $N(j)$ impressions. Expressed combinatorially, this corresponds to finding a maximum weight b-matching online. This weighted b-matching problem is considered in [11], which showed that the problem is inapproximable without exploiting *free disposal*. When the *demand of each advertiser is large*, a $(1-\frac{1}{e})$-competitive algorithm exists [11], and this is the best possible. In the budgeted allocation problem, motivated by sponsored search advertising, the ad-serving algorithm allocates impressions resulting from search queries. Advertiser j has a budget $B(j)$ on the total spend instead of a bound $N(j)$ on the number of impressions. Assigning impression i to advertiser j consumes $w(i,j)$ units of j's budget instead of 1 of the $N(j)$ slots, as in the weighted b-matching problem. $1-\frac{1}{e}$-competitive algorithms have been designed for this problem under the assumption of *large budgets* [24, 4].

The random order model has also been studied extensively for these problems. In particular, a *dual technique* has been developed to solve problems in this setting: This approach is based on computing an offline optimal dual solution based on the first ϵ fraction of items / impressions, and using this solution online to assign the remaining vertices. Following the first such training-based dual algorithm of [5] for the budgeted allocation problem, training-based $(1-\epsilon)$-competitive algorithms have been developed for the weighted matching problem and its generalization to various packing linear programs [12, 33, 2, 29]. These papers develop $(1-\epsilon)$-competitive algorithms for online stochastic packing problems in the random order model, if $\frac{\mathbb{O}PT}{w_{ij}} \geq O(\frac{m \log n}{\epsilon^2})$ and the demand of each advertiser / agent is large. In a separate line of work, improved approximation algorithms have been proposed for the unweighted online bipartite matching problem (which is a special case of both the weighted matching and budgeted allocation problems) in the random order model [19, 23], even without the assumption that the demand of each agent is large.

i.i.d. Model. Other than the adversarial and random-order models studied in this paper, online ad allocation problems have been studied extensively in the *i.i.d. stochastic models* in which impressions arrive i.i.d. according to a known or an unknown distribution. A key technique in this area is the *primal approach* which is based on solving an offline allocation problem on the instance that we expect to arrive according to the stochastic information, and then applying this offline solution online. This technique has been applied to the online stochastic matching problem [20] and in the i.i.d. model with known distributions [13, 25, 16, 3], and resulted in improved competitive algorithms.

Interestingly, a new *hybrid* technique can be applied to obtain a $1-1/e$-competitive algorithm for the budgeted allocation problem in the i.i.d. model [7, 8], and a similar ratio can be obtained for the more general online SWM problem [18]. However, all results based on such techniques seem to only apply to the i.i.d. model [7, 18] and not the random order model. Generalizing such results to the random order model remains an interesting open problem in the area.

2. PRELIMINARIES AND KEY IDEAS

We study the following welfare maximization problem, referred to as online SWM: n items from a set N arrive sequentially online, and when each item arrives, it should be

[1] We note that improving on the ratio of 1/2 for the offline Submodular Welfare Maximization Problem was an open problem for nearly 40 years until the result of Vondrak [34].

irrevocably assigned to one of m agents from the set M. Each agent $1 \leq \ell \leq m$ has valuation function $v_\ell : 2^{[n]} \to \mathbb{R}_+$. We assume that each agent's valuation function v_ℓ is submodular, non-negative and monotone. The goal is to assign the items to agents in order to maximize the social welfare, defined as $\sum_{i=1}^{m} v_\ell(S_\ell)$ where S_ℓ is the set of items assigned to ℓ. We assume w.l.o.g. that all of the n items are distinct, but we sometimes consider the union of two or more allocations, as a result of which there may exist multiple copies of items: In this case, the set of items assigned to a single agent is the union of the sets assigned to it under the allocations, so it does not actually receive multiple copies of the same item. (Equivalently, one could extend the submodular function to multisets in the natural way so that the marginal value of an agent for the second copy of an item is 0.)

Algorithms for maximizing welfare in combinatorial auctions and submodular optimization are required to be polynomial in the natural parameters of the problem, m and n. However, since the "input" (the valuation functions) is of exponential size, one must specify how it can be accessed. Most works in this field have taken a "black box" approach in which bidders' valuation functions are accessed via oracles that can answer specific type of queries. The most natural and popular oracle query model is the *value query model* [9, 10, 21], which we also use: the query to a valuation function v_ℓ is in the form of a bundle $S \subseteq N$, and the response is $v_\ell(S)$.

As discussed above, no online algorithm can achieve a competitive ratio greater than 1/2 for online SWM in the adversarial setting, and hence we focus on the *random order* model. We measure the competitive ratio of an algorithm as the ratio of the expected social welfare achieved by the algorithm's assigment (where the expectation is taken over all $n!$ permutations) to the optimum social welfare. Our main contribution is to demonstrate that the simple and natural Greedy algorithm achieves a competitive ratio of $1/2 + \Omega(1)$.

What is the main difficulty in proving that the Greedy algorithm achieves a competitive ratio better than 1/2? It is obvious that the expected welfare that Greedy obtains by allocating the first item is at least as much as the optimum algorithm obtains by allocating this item. Of course, the disadvantage of the greedy allocation is that by assigning one item, it may severely reduce the potential welfare gain from items that arrive in the future. Our key insight is into how items interact and how allocating one item can affect the welfare *gain* that can be obtained from allocating other items. Unfortunately, due to space constraints, certain proofs are deferred to the full version, but we attempt to sketch / provide intuition for these proofs here.

2.1 Understanding Item Interactions and Gain

In order to study interactions between allocations, we need to introduce the following notation:

Definition 4. *An allocation of items to agents, denoted by A, consists of m subsets of items $A = \{A_\ell\}_{\ell=1}^{m}$ where A_ℓ is the set of items allocated to agent ℓ, and every item is assigned to at most one agent. We note that some items may remain unallocated in an allocation. We denote the total welfare or value of allocation A by $V(A) = \sum_{\ell=1}^{m} v_\ell(A_\ell)$.*

We let A^* be the allocation that maximizes social welfare $OPT = \sum_{\ell=1}^{m} v_\ell(A^*_\ell)$ where OPT is the maximum social welfare. Without loss of generality (by normalizing all values), we assume that OPT is equal to 1. For an item j, we define opt_j to be the agent that receives item j in the optimum allocation A^*, i.e. j belongs to $A^*_{opt_j}$. For a permutation σ on the items $[n]$, we define σ^i to be the set of first i items in σ, in other words $\{\sigma_1, \sigma_2, \cdots, \sigma_i\}$.

We can now introduce a central concept in our analysis, used to provide a lower bound on the welfare gains of Greedy. In standard analyses of greedy algorithms, a common approach is to show that one good option available to the greedy algorithm is the choice made by the optimal algorithm. In our setting, this would translate to showing that Greedy can obtain a good increase in welfare by assigning item j to opt_j. The other piece of this approach is to show that the greedy choice for allocating item j does not reduce the welfare gain that can be obtained from *future* items by more than the welfare increase from j. Together, these two ideas imply that Greedy is 1/2-competitive in the worst case. In order to formally argue about this, we introduce the following notation to describe the marginal gain of assigning item j to agent $i = opt_j$ based on a current allocation and a (partial) optimum allocation:

Definition 5. *Fix an arbitrary permutation σ (such as the identity permutation). For any item j and any allocation A let $\ell = opt_j$, and $i+1$ denote the index of item j in σ. We define $Gain(j, A)$ as:*

$$v_\ell\left(\{j\} \cup A_\ell \cup (A^*_\ell \cap \sigma^i)\right) - v_\ell\left(A_\ell \cup (A^*_\ell \cap \sigma^i)\right).$$

That is, $Gain(j, A)$ denotes the marginal gain we get from assigning j to the agent ℓ that receives it in the optimal solution, assuming that ℓ has already received all items in A_ℓ based on the allocation A, *as well as* those of the first i items (under σ) that the optimal solution allocates to ℓ (i.e. $A^*_\ell \cap \sigma^i$). It is important to note that in the definition, the permutation σ is fixed and the same permutation is used to define $Gain(j, A)$ for all items j and allocations A.

Intuitively, $Gain(j, A)$ captures the further marginal gain one can achieve from item j, given allocation A. As a simple example, consider the case when A is an empty allocation (no item is allocated in A). In this case, the sum $\sum_{j=1}^{n} Gain(j, A)$ is equal to OPT because for any permutation σ, the sum $\sum_{j=1}^{n} Gain(j, A)$ of marginal gains of the n items obtained by the optimal allocation is the definition of OPT. For any set S, we use $Gain(S, A)$ to denote $\sum_{j \in S} Gain(j, A)$.

How is the concept of $Gain$ useful? Let π denote the arrival order of the items. We denote the allocation of Greedy on the first i items (that is, on the sequence π^i) by A^i. The following lemma captures the marginal welfare Greedy achieves at each step by using the notion of $Gain$ variables.

Lemma 6. *If item j arrives at position $i+1$ under permutation π, the increase in welfare that Greedy achieves by allocating this item is at least $Gain(j, A^i)$.*

What does this have to do with interactions between items? For a fixed item j, the value $Gain(j, A^i)$ is non-increasing in i. In other words, when Greedy assigns a new item π_i, it may decrease the $Gain$ values for some items. We now prove that the total decrease in $Gain$ values for all n items is at most the increase in welfare that Greedy obtained by allocating item π_i. This next lemma allows us to keep track of changes in these *Gain* values throughout the algorithm.

LEMMA 7. *Greedy's increase in welfare from item π_i is at least $\sum_{j\in[n]} Gain(j, A^{i-1}) - Gain(j, A^i)$.*

Lemmas 6 and 7 (note that neither uses the fact that we have a random permutation) use the concept of *Gain* to give an alternate proof of the simple fact that the Greedy algorithm is 1/2-competitive in the adversarial model: We start with an empty allocation A^0, and Greedy assigns items one by one. At the beginning, the total $Gain = Gain(N, A^0)$ is equal to OPT. Let G denote the sum of the final gains $Gain(N, A^n)$. Lemma 7 implies that the value of the allocation A^n is at least $OPT - G$. On the other hand, Lemma 6 shows that if item j arrives at position $i + 1$, the algorithm's marginal increase in welfare is at least $Gain(j, A^i) \geq Gain(j, A^n)$. Therefore, the total welfare obtained by the algorithm is at least G. Since the welfare is at least $\max\{G, OPT - G\}$, it must be at least $OPT/2$.

2.2 Going Beyond 1/2: Our Techniques

Of course, we wish to argue that Greedy does *better* than half in the random order model. When we applied Lemma 6 in the previous paragraph, we could have lower bounded Greedy's marginal increase in welfare from allocating item j in position $i+1$ by $Gain(j, A^i)$; instead, we used the weaker lower bound $Gain(j, A^n) \leq Gain(j, A^i)$. When this inequality is not tight (that is, if the gain of item j is reduced *after* it arrives by subsequent Greedy allocations), we can obtain an improved competitive ratio. To make this formal, we define β as the total amount by which the Greedy allocation reduces the gain of items which have *already* arrived.

LEMMA 8. *The competitive ratio of Greedy is at least $1/2+\beta/2$.*

PROOF. If item j arrives in position $i+1$, define g_j as the reduction in the gain of j *after* it arrived; that is, $g_j = Gain(j, A^i) - Gain(j, A^n)$. By definition, $\beta = \sum_{j=1}^{n} g_j$. From Lemma 6, the increase in Greedy's welfare from allocating item j is at least $Gain(j, A^i) = Gain(j, A^n) + g_j$. Denoting $Gain(N, A^n)$ as G, we conclude that the total welfare of Greedy is at least $G + \beta$. Lemma 7 implies that the total welfare of Greedy is at least $1 - G$ (recall that we normalized OPT to 1. Hence, Greedy's total welfare is at least $\max\{G + \beta, 1 - G\} = 1/2 + \beta/2$. □

If the gains of items were consistently reduced by a substantial amount *after* they arrived, we would have a proof that Greedy has competitive ratio $1/2 + \Omega(1)$. But perhaps in the worst-case instances, Greedy's allocation of the ith item only reduces *Gain* values for future items? By taking advantage of the random arrival order, we can understand how the reduction in *Gain* is distributed among future items. In particular, the *second* item to arrive is unlikely to have its *Gain* reduced significantly (in expectation) by the first item , but when the *last* item arrives, it is quite likely that it has a very small *Gain* value, because some previous (mis)-allocations of the greedy algorithm mean that assigning it to its optimal agent produces little value.

A straightforward differential equation analysis based on Lemmas 6, and 7 (which ultimately yields a competitive ratio of 1/2) shows that when items arrive in a random order, the expected welfare from greedily assigning the first t fraction of items is at least $w(t) = t - \frac{t^2}{2}$ that of the optimal (more details can be found in the full version of this paper). Assuming this analysis is tight, the expected rate at which welfare increases is $w'(t) = 1 - t$ which is essentially *zero* for the last few items (as opposed to at least 1 for the first item). The key contribution we make, then, is to go beyond such analyses: We derive new lower bounds on the welfare obtained from items that arrive towards the end of the order. This allows us to show that the total welfare obtained by the greedy algorithm is strictly better than 1/2 of the optimal.

Proving such lower bounds is non-trivial, and the main technique we apply is lower bounding the welfare obtained from a single item by both *its own Gain* when it arrives (as in Lemma 6), and *how it reduces the Gain of other items* (as in Lemma 7). In particular, we apply these lower bounds to items arriving at the end of the random order, which allows us to show that Greedy obtains non-zero welfare even at the end. In this paper, we take three distinct approaches to obtain the needed lower bounds on *Gain* values from items that arrive at the end of the random order. Informally speaking, we do the following:

- First, we consider the special case of second-order supermodular functions. We show that Greedy gets good welfare from items that arrive towards the end by proving that there exists an allocation of these items that significantly reduces the *Gain*s of the first $n/2$ items. (Recall from Lemma 7 that Greedy's welfare from allocating an item is at least as much as its ability to reduce the *Gain*s of other items.) Our proof proceeds in several steps: We first show that there is an allocation of the first $n/2$ items that significantly reduces the *Gain*s of the last $n/2$ items; by symmetry, there must exist an allocation $\hat{A}$ of the last $n/2$ items that significantly reduces the *Gain*s of the first $n/2$. Next, we use second-order supermodularity to argue that a subset of the items that arrive at the end will be able to reduce the *Gain*s of the first $n/2$ items at least in proportion to the size of this subset. More formally, we show that for items that arrive at the end, the only way they might not significantly reduce the *Gain*s of the first $n/2$ items is if previous allocations among the last $n/2$ items already reduced these *Gains* significantly. But in this case, Greedy's allocations reduced the gains of items that arrived previously, which means that $\beta = \sum_i b_i$ is large, and Lemma 8 shows that Greedy must then have a competitive ratio better than 1/2. We precisely quantify this improvement over 1/2 by formulating a factor-revealing Linear Program, and analyzing it explicitly.
- Second, we consider the general submodular valuation case. The previous approach no longer applies, as without the second-order supermodularity property, allocating a single random item might significantly affect the collective gains of other items. Therefore, we consider the (simulated) effect of assigning the last quarter of items *three* times in succession, using three different allocations. Intuitively, this can be understood as showing three distinct allocations for these items and arguing that the (mis)-allocations of Greedy cannot harm all of these simultaneously. Therefore,

these allocations together obtain large welfare, and the first time these items arrive (corresponding to the 'real' arrival and allocation) is the best of these.

- Third, we consider duplicating a single random item and adding it to the end of the sequence of arriving items. We propose a natural conjecture on how the marginal gain of this item is related to the marginal gain of Greedy on the last item. If this conjecture holds, we obtain a much stronger lower bound on the marginal gain from Greedy at the end of the sequence, giving a competitive ratio of 0.567.

With these lower bounds on *Gain* values, we can prove Theorems 1 and 3 by constructing factor-revealing linear programs; the fact that items arrive in a random order allows us to constrain the values of the variables representing the interactions in item *Gain*s. We then formally analyze these linear programs, which requires some intricate computations. Though these calculations are problem-specific, we believe that our main technique of *gain linearizing* and bounding item interactions will be generally useful to improve analyses of submodular allocations and greedy approaches for random-order problems. (For one example, see [26].)

2.3 Formulating the Factor-Revealing LP

Recall from Lemmas 6 and 7 that when item j in position i arrives, Greedy has marginal welfare at least $Gain(j, A^{i-1})$. Further, assigning this item j reduces the $Gain$ values of other items (both those that have already arrived, and those which will arrive after j), but the total reduction is at most the marginal welfare of allocating item j.

Definition 9. *For each index $1 \le i \le n$, we define w_i to be the expected increase in welfare Greedy achieves by assigning the item in position i (that is, the ith item in the arrival order).*

Assigning item i possibly reduces the Gain values of items. We partition this effect into two parts; for any item i, we have two variables b_i and a_i defined as follows to capture the reduction in Gain values of other items:

- *b_i (we use b to denote* before*) is the expected reduction in Gain of items j that have already arrived. From our definition of β above, $\beta = \sum_{i=1}^{n} b_i$.*
- *a_i (a denotes* after*) is the expected reduction in Gain of items j that are going to arrive later.*

Clearly, allocating the item in position i reduces the total gain values of other items by $b_i + a_i$ in expectation.

By definition, Greedy's performance is simply $\sum_{i=1}^{n} w_i$. Our factor-revealing linear programs will consist of three separate lower bounds on w_i. Two of these are common to both the general submodular case as well as the case of second-order supermodular functions.

For the first constraint of our LP, we use the notation of Definition 9 to express Lemma 7 as $w_i \ge b_i + a_i$. (This follows because w_i is the increase in welfare that Greedy achieves from the ith item, and $b_i + a_i$ is how much this segment reduces the $Gain$ values of all items.)

In the next lemma, we prove another lower bound on w_i, yielding the next constraint of our linear program. Recall that we normalized our instance so that $OPT = 1$.

Lemma 10. *For any $1 \le i \le n$, if $OPT = 1$, we have $w_i \ge \frac{1}{n} - \sum_{j=1}^{i-1} \frac{a_j}{n-j}$.*

Proof. From Lemma 6, the increase in welfare of the algorithm from the ith item is at least its $Gain$ values at the time of its arrival; this can be written as the difference of the *original* $Gain$ value of this items (before the algorithm starts) and the reduction in $Gain$ value due to allocating items that arrived previously. Before the algorithm starts, the average $Gain$ value for the ith item is equal to $OPT/n = 1/n$, because the sum of original $Gain$ values is equal to $OPT = 1$.

Now it suffices to prove that the expected reduction in the $Gain$ value of the ith item does not exceed $\sum_{j=1}^{i-1} \frac{a_j}{k-j}$. Since we are bounding the reduction in the $Gain$ value of an item before it arrives, we can ignore any reduction that occurs after it arrives. If the $Gain$ value of the ith item is reduced before it arrives, it must have been reduced by the item in position j for some $j < i$, and hence this must have been included in a_j. However, the variable a_1, for instance, captures reductions in $Gain$ values for *all* items appearing after the first, not just the ith item. In general, for some $1 \le j < i$, a_j accounts for reduction in $Gain$ values of items which arrive in any of the positions $j+1, j+2, \cdots, n$. The probability that any of these items falls in position i and hence the reduction of its $Gain$ value becomes relevant in this lower bound is $1/(k-j)$ because there are $k-j$ future segments, and we have a random permutation of items. Therefore, for each $1 \le j < i$, we should deduct $\frac{a_j}{k-j}$ in this lower bound. □

We now have two constraints that lower bound w_i, but if we only use these two, we will not be able to show that Greedy has a competitive ratio better than 1/2. As described above, the further ingredient required is to show that Greedy obtains good welfare from the items that arrive towards the end of the sequence. In Section 3, we do this for the special case of second-order supermodular functions, proving Theorem 3. Then, in Section 4, we apply a more general technique to obtain a slightly weaker result for arbitrary submodular functions, proving Theorem 1. Finally, in Section 5, we discuss how these results can be greatly improved based on a new conjecture.

3. SECOND-ORDER SUPERMODULAR FUNCTIONS

Definition 11. *Let S_1 be the sequence of items with index $\le n/2$, S_2 the sequence of items with index i such that $n/2 < i \le 3n/4$, and S_3 the sequence of items with index $i > 3n/4$. All of S_1, S_2, and S_3 are random variables (sequences).*

We define $\langle S_{a_1}, S_{a_2}, \cdots, S_{a_\ell} \rangle$ to be the concatenation of sequences of items $S_{a_1}, S_{a_2}, \cdots, S_{a_\ell}$. For any sequence S_k, we abuse notation and use S_k to denote both a sequence of items and the set of these items, but the meaning will always be clear from context.

Let $A^G(S)$ be the allocation of items in S by the Greedy algorithm, where we assume that no other item has arrived prior to S, and then Greedy allocates items of S one by one. We also define $A^*(S)$ to be the optimum allocation of items in S.

We want to show that Greedy gets good welfare from items that arrive at the end. As described in Section 2, the straightforward analysis that obtains a competitive ratio of 1/2 shows that Greedy's welfare from assigning the first t fraction of the items is $\geq t - t^2/2$. Therefore, at $t = 0.8$, we have already obtained welfare 0.48, and at $t = 0.9$, we have welfare 0.495. That is, the remaining welfare from assigning the last 0.2 and 0.1 fraction of the items is 0.02 and 0.005 respectively. To improve on the competitive ratio of 1/2, we use better lower bounds on the value of items that arrive at the end.

We first show that Greedy's allocation of items in S_1 reduces the *Gain*s of items in $S_2 \cup S_3$ by at least 1/4; it follows that there exists an allocation of $S_2 \cup S_3$ that reduces the *Gain*s of S_1 by 1/4. However, Greedy's allocation of items in S_1 may also reduce the *Gain*s of other items in S_1 by up to 1/8. Hence, the further reduction of *Gains* of S_1 by allocating $S_2 \cup S_3$ is at least $1/4 - 1/8 = 1/8$. We then take advantage of second-order supermodularity to argue that the ability of a random subset of $S_2 \cup S_3$ to reduce the *Gain*s of items in S_1 is proportional to its size. That is, a subset of $S_2 \cup S_3$ of size k will be able to reduce the *Gain* of S_1 by at least $\frac{k}{n/2}\frac{1}{8}$. In particular, the last 0.1 fraction of the items will be able to reduce the *Gain* of S_1 by at least $\frac{0.1}{0.5} \cdot \frac{1}{8} = 0.025$; and hence there is an allocation of these items which could obtain welfare at least 0.025. (Note that this is 5 times better than the naive analysis of Greedy.) The only reason for Greedy to not get such a welfare increase would be if previous allocations of items in $\langle S_2, S_3 \rangle$ already reduced the *Gain* of S_1 significantly. But in this case, we use Lemma 8 to show that Greedy has a competitive ratio better than 1/2. More formally, instead of considering an explicit fraction such as the last 0.1 fraction of the items, we quantify the improvement over 1/2 by applying this idea to write a constraint in our factor-revealing Linear Program for every i, and then mathematically analyzing the LP. Please refer to the full version for details.

4. GENERAL SUBMODULAR FUNCTIONS

For arbitrary submodular functions, we can no longer use the technique of the previous section to argue that we obtain sufficient welfare from items in $\langle S_2, S_3 \rangle$. Instead, here we use a different constraint that provides a lower bound on $\sum_{i>3n/4} w_i$, which is Greedy's expected increase in welfare when assigning S_3, the last quarter of items. We obtain this constraint by considering the simulated sequence of items $S' = \langle S_1, S_2, S_3, S_2, S_3, S_2 \rangle$. In other words, we will analyze how one could assign items if after the items in S_1 arrive, items in S_2 and S_3 arrive *multiple* times. We assign items of S' using the following allocation $A' = A^G(\langle S_1, S_2, S_3 \rangle) \cup A^G(\langle S_2, S_3 \rangle) \cup A^*(S_2)$. That is, we first use Greedy to assign the items of $\langle S_1, S_2, S_3 \rangle$; then, we use the different allocation given by Greedy on $\langle S_2, S_3 \rangle$ assuming nothing has been assigned so far (that is, ignoring the previous allocation of Greedy on $\langle S_1, S_2, S_3 \rangle$). Finally we use the optimum allocation for S_2. A' is defined as the union of these three allocations.

To show that Greedy gets sufficient value from items in S_3, we will lower bound the value of A' on the last five segments $\langle S_2, S_3, S_2, S_3, S_2 \rangle$. This part of the value of A' can be formally written as $V(A') - V(A^G(S_1))$;[2] using Lemma 7, we can lower bound it by showing how much it reduces the *Gain* values of all items. That is, $\mathbb{E}[V(A') - V(A^G(S_1))] \geq \mathbb{E}[Gain(N, A^G(S_1)) - Gain(N, A')]$. Since $N = S_1 \cup S_2 \cup S_3$, we can lower bound the desired quantity by separately lower bounding $\mathbb{E}[Gain(S_k, A^G(S_k)) - Gain(S_k, A')]$ for each of $k = 1, 2, 3$.

Claim 1. $\mathbb{E}[Gain(S_1, A^G(S_1)) - Gain(S_1, A')] \geq \sum_{i=1}^{n/2} \frac{n/2}{n-i} a_i - \sum_{i=1}^{n/2} (\frac{n/2-i}{n-i} a_i + b_i)$.

Proof. For any item j, $Gain(j, A^G(\langle S_2, S_3 \rangle))$ is at least $Gain(j, A')$ by submodularity and the fact that $A^G(\langle S_2, S_3 \rangle) \subseteq A'$. So to lower bound $\mathbb{E}[Gain(S_1, A^G(S_1)) - Gain(S_1, A')]$, it suffices to lower bound
$\mathbb{E}[Gain(S_1, A^G(S_1)) - Gain(S_1, A^G(\langle S_2, S_3 \rangle))] =$

$$\mathbb{E}[Gain(S_1, \emptyset) - Gain(S_1, A^G(\langle S_2, S_3 \rangle))] -$$

$$\mathbb{E}[Gain(S_1, \emptyset) - Gain(S_1, A^G(S_1))]$$

It follows that $\mathbb{E}[Gain(S_1, \emptyset) - Gain(S_1, A^{Greedy}(\langle S_2, S_3 \rangle))]$ is equal to $\sum_{i=1}^{n/2} \frac{n/2}{n-i} a_i$ from symmetry and the definition of a_i variables. To see this, note that $\pi' = \langle S_2, S_3, S_1 \rangle$ is also a random permutation in which $\langle S_2, S_3 \rangle$ is the first half of items, and S_1 is the second half. Now we want to lower bound how much Greedy's allocation of first half items in the random permutation π' reduces the *Gain* variables of second half items of π'. We know that for any $i \leq n/2$, the ith item in π' reduces the *Gain* variables of items that appear in positions greater than i in π' by a_i, and (as in Lemma 10), in expectation $\frac{n/2}{n-i}$ fraction of this reduction is associated with items in positions greater than $n/2$. Similarly, we can consider the original random permutation $\pi = \langle S_1, S_2, S_3 \rangle$, and see that the allocation of items in S_1 reduces the *Gain* variables of other items in S_1 by a total of $\sum_{i=1}^{n/2} \frac{n/2-i}{n-i} a_i + b_i$. This proves the desired claim. □

Claim 2. $\mathbb{E}[Gain(S_2, A^G(S_1)) - Gain(S_2, A')] = \frac{1}{4} - \sum_{i=1}^{n/2} \frac{n/4}{n-i} a_i$.

Proof. Since S_2 is a random quarter of all items, we have that $\mathbb{E}[Gain(S_2, \emptyset)] = \frac{1}{4}$, and since A' concludes with the optimum allocation on items of S_2 ($A^*(S_2)$), $Gain(S_2, A')$ is zero. So $\mathbb{E}[Gain(S_2, A^G(S_1)) - Gain(S_2, A')]$ is equal to $\frac{1}{4} - (\mathbb{E}[Gain(S_2, \emptyset) - Gain(S_2, A^G(S_1))])$. The latter term can be exactly calculated from the fact that allocating the item in the ith position reduces the gains of subsequent items by a_i, and $n/4$ of the $n - i$ remaining items are in S_2; therefore, $\mathbb{E}[Gain(S_2, \emptyset) - Gain(S_2, A^G(S_1))] = \sum_{i=1}^{n/2} \frac{n/4}{n-i} a_i$. □

Claim 3. $\mathbb{E}[Gain(S_3, A^G(S_1)) - Gain(S_3, A')] \geq \sum_{i=n/2+1}^{3n/4} \frac{n/4}{n-i} a_i$.

Proof. Since $A^G(\langle S_1, S_2 \rangle)$ is a subset of A', it suffices to lower bound

$$\mathbb{E}[Gain(S_3, A^G(S_1)) - Gain(S_3, A^G(\langle S_1, S_2 \rangle))].$$

Similarly to the proof of the previous claim, this reduction in *Gain* of items in S_3 from allocating the ith item is $\frac{n/4}{n-i} a_i$, but now this reduction is achieved by allocating items of S_2. □

From the three preceding claims, we conclude that $\mathbb{E}[V(A') - V(A^G(S1))]$ is at least:

$$\sum_{i=1}^{n/2} \frac{n/2}{n-i} a_i - \sum_{i=1}^{n/2} \left(\frac{n/2-i}{n-i} a_i + b_i \right) + \frac{1}{4} - \sum_{i=1}^{n/2} \frac{n/4}{n-i} a_i + \sum_{i=n/2+1}^{3n/4} \frac{n/4}{n-i} a_i$$

[2]Recall from Definition 4 that $V(A)$ for some allocation A denotes the total welfare or value of the allocation.

$$= \frac{1}{4} + \sum_{i=1}^{n/2} \left(\frac{i - n/4}{n-i} a_i - b_i \right) + \sum_{i=n/2+1}^{3n/4} \frac{n/4}{n-i} a_i$$

Lemma 12. *The expected increase in welfare that Greedy achieves for the last quarter of items $\sum_{i>3n/4} w_i$ is at least $\frac{1}{24} + \sum_{i=1}^{n/2} (\frac{i-n/4}{6(n-i)} a_i - \frac{1}{6} b_i) + \sum_{i=n/2+1}^{3n/4} \frac{n/4}{6(n-i)} a_i - \frac{1}{6} \sum_{i=n/2+1}^{3n/4} w_i$.*

Proof. We use X to denote the lower bound obtained from the three preceding claims on how much A' increases the welfare from the rest of S' after items in S_1 have alrady arrived. In other words, this is how much it increases its welfare by allocating $\langle S_2, S_3, S_2, S_3, S_2 \rangle$. We use $W_3 = \sum_{i \in S_3} w_i$ to denote how much A' increases its welfare by allocating items in the first copy of S_3, and since items in the first copy are assigned greedily (in A'), their total increase in the welfare is at least as much as the increase in welfare A' obtains by allocating the second copy S_3 (this is easy to verify for each item). So A' in total increases its welfare from the two copies of S_3 by at most $2W_3$.

Let S'_2 be the sequence $\langle S_2, S_2, S_2 \rangle$ derived from S' by removing the initial copy of S_1 and the two copies of S_3. Let A'_2 be the projection of allocation A' on sequence S'_2; in other words, A'_2 is an allocation of sequence S'_2, and it is consistent with A' on this sequence. It is clear that after the allocation $A^G(S_1)$, allocating the three copies of S_2 using A'_2 increases the welfare by at least $X - 2W_3$ since removing the two copies of S_3 will not reduce the welfare by more than $2W_3$.

We now appeal to symmetry and switch the argument from S_2 to S_3. Given set S_1, S_2 is a random sequence of half of the items in $N \setminus S_1$, and so is S_3; that is, given sequence S_1, S_2 and S_3 have the same distribution. Suppose S_1 is allocated by Greedy; as argued above, we know that after this allocation it is possible to allocate three consecutive copies of S_2 using A'_2 and increase the welfare by at least $X - 2W_3$ in expectation. Therefore we can claim that after allocation $A^G(S_1)$, it is possible to allocate three copies of S_3 and increase the welfare by the same amount of at least $X - 2W_3$. Formally, there exists an allocation A'_3 of sequence $S'_3 = \langle S_3, S_3, S_3 \rangle$ such that $\mathbb{E}[V(A^G(S_1) \cup A'_3) - V(A^G(S_1))]$ is at least $X - 2W_3$.

Now consider the sequence $\langle S_1, S_2, S_3, S'_3 \rangle$, which begins with the original sequence $S = \langle S_1, S_2, S_3 \rangle$ and then has three copies of S_3. For this sequence, consider the allocation $A^{Final} = A^{Greedy}(\langle S_1, S_2, S_3 \rangle) \cup A'_3$; that is, we first assign the original sequence $\langle S_1, S_2, S_3 \rangle$ according to Greedy, and then assign the three copies of S_3 using A'_3. The increase in welfare by allocating the last three copies of S_3 in A^{Final} is at least $X - 2W_3 - W_2 - W_3$ where $W_2 = \sum_{i \in S_2} w_i$. Now there are four copies of S_3 in the sequence $< S_1, S_2, S_3, S'_3 >$. Since the first copy of S_3 is allocated greedily in A^{Final}, its increase in welfare which is W_3 is at least as much as the increase in welfare of any the other copies of S_3 in A^{Final}, and hence also at least as much as the average of increase in welfare by these three copies. We conclude that:

$$W_3 \geq \frac{X - 2W_3 - W_2 - W_3}{3}$$

By definition of X and W_2, we have: $W_3 \geq \frac{X - W_2}{6}$

$$= \frac{\frac{1}{4} + \sum_{i=1}^{n/2} (\frac{i-n/4}{n-i} a_i - b_i) + \sum_{i=n/2+1}^{3n/4} \frac{n/4}{n-i} a_i - \sum_{i=n/2+1}^{3n/4} w_i}{6}$$

which concludes the proof. □

We now have all the building blocks we need to provide a lower bound on the total welfare obtained by Greedy in terms of w_i, a_i, and b_i, and consequently prove that Greedy achieves a competitive ratio better than $\frac{1}{2}$.

Lemma 13. *The expected welfare obtained by Greedy is at least $\frac{1}{24} + \sum_{i=1}^{n/2} \Big((1 + \frac{i-n/4}{6(n-i)}) a_i + \frac{5}{6} b_i \Big) + \sum_{i=n/2+1}^{3n/4} \Big((\frac{5}{6} + \frac{n/4}{6(n-i)}) a_i + \frac{5}{6} b_i \Big) + \sum_{i=1}^{3n/4} \frac{5}{6} (w_i - a_i - b_i)$.*

Proof. We can write the expected increase in welfare by Greedy as $\sum_{i=1}^{n/2} w_i + \sum_{i=n/2+1}^{3n/4} w_i + \sum_{i=3n/4+1}^{n} w_i$. Using Lemma 12 to lower bound the last term $\sum_{i>3n/4} w_i$, we conclude that the welfare is at least $\frac{1}{24} + \sum_{i=1}^{n/2} (\frac{i-n/4}{6(n-i)} a_i - \frac{1}{6} b_i) + \sum_{i=n/2+1}^{3n/4} \frac{n/4}{6(n-i)} a_i + \frac{5}{6} \sum_{i=n/2+1}^{3n/4} w_i + \sum_{i=1}^{n/2} w_i$.

Since the coefficient of each w_i is positive (either $\frac{5}{6}$ or 1) in this lower bound, we can apply Lemma 7 to replace each w_i with $a_i + b_i$, and add the sum $\sum_{i=1}^{3n/4} \frac{5}{6} (w_i - a_i - b_i)$ while still having a valid lower bound. It suffices to merge the sums to conclude the claim of this lemma. □

We can now complete our proof of Theorem 1:

Proof of Theorem 1. Using Lemma 13, we know that for any instance the expected gain of greedy is at least:

$$LB = \frac{1}{24} + \sum_{i=1}^{n/2} \left((1 + \frac{i - n/4}{6(n-i)}) a_i + \frac{5}{6} b_i \right) +$$

$$\sum_{i=n/2+1}^{3n/4} \left((\frac{5}{6} + \frac{n/4}{6(n-i)}) a_i + \frac{5}{6} b_i \right) + \sum_{i=1}^{3n/4} \frac{5}{6} (w_i - a_i - b_i)$$

for some set of $\frac{9n}{4}$ numbers $\{w_i, a_i, b_i\}_{i=1}^{3n/4}$ all in $[0, 1]$. From Lemmas 7, and 10, we know they satisfy the inequalities $w_i \geq a_i + b_i$, and $w_i \geq \frac{1}{n} - \sum_{j=1}^{i-1} \frac{a_j}{n-j}$ for any $1 \leq i \leq \frac{3n}{4}$. The minimum value of LB among the feasible solutions of these linear constraints lower bounds the competitive ratio of Greedy algorithm. We prove that to minimize LB given these linear constraints, one can assume w.l.og. that $w_i = a_i + b_i$ for any $i \leq \frac{3n}{4}$ as follows. Suppose for some $i \leq \frac{3n}{4}$, the gap $\delta = w_i - (a_i + b_i)$ is positive. We increase b_i by δ. The value of LB is intact because on one hand $\frac{5}{6} b_i$ appears in either the first or second summation in the lower bound, and $-\frac{5}{6} b_i$ appears in the third summation. The inequalities $w_i \geq a_i + b_i$ and $w_i \geq \frac{1}{n} - \sum_{j=1}^{i=1} \frac{a_j}{n-j}$ are also still satisfied, and w_i is now equal to $a_i + b_i$. Performing this update for every constraint $w_i \geq a_i + b_i$ that is not tight, we can assume that $w_i = a_i + b_i$ for any $i \leq \frac{3n}{4}$. Therefore it suffices to lower bound the following simplified expression LB':

$$LB' = \frac{1}{24} + \sum_{i=1}^{n/2} \left((1 + \frac{i - n/4}{6(n-i)}) a_i + \frac{5}{6} b_i \right) +$$

$$\sum_{i=n/2+1}^{3n/4} \left((\frac{5}{6} + \frac{n/4}{6(n-i)}) a_i + \frac{5}{6} b_i \right)$$

for some set of $\frac{3n}{2}$ numbers $\{a_i, b_i\}_{i=1}^{3n/4}$ all in $[0, 1]$ with linear constraints $a_i + b_i \geq \frac{1}{n} - \sum_{j=1}^{i-1} \frac{a_j}{n-j}$ for any $1 \leq i \leq \frac{3n}{4}$.

We find the minimum value of LB' among all feasible solutions of these linear constraints. First, we prove that there exists an optimum solution (minimizing LB', and satisfying the constraints) in which $a_i + b_i = \frac{1}{n} - \sum_{j=1}^{i-1} \frac{a_j}{n-j}$ for any $1 \leq i \leq \frac{3n}{4}$. In other words, the linear constraints should be tight. Second, we show that there exists an optimal solution in which, furthermore, all b_i values are 0. This then allows us to explicitly find the values of a_i in this optimal solution, and we can evaluate LB' explicitly.

Due to space constraints, we are unable to include the complete analysis to lower bound LB'; in the full version of this paper, we show that LB' is at least 0.5052. □

5. TOWARDS AN IMPROVED COMPETITIVE RATIO

While the techniques of the previous section show that the competitive ratio of the Greedy algorithm is strictly better than 1/2, we believe that the performance is considerably better. In this section, we present a natural conjecture on the marginal increases in welfare obtained by Greedy (Conjecture 15), that captures much of the difficulty of this problem. Theorem 16, proved in the full version of this paper, shows that if the conjecture holds, the Greedy algorithm achieves at least a competitive ratio of 0.567. In order to state the conjecture, we need to introduce some notation:

DEFINITION 14. *Fix a permutation $\pi = (\pi_1, \pi_2, \cdots, \pi_n)$. We define two new permutations based on π:*

- *Let $\pi^{Move,i}$ be the permutation $(\pi_1, \pi_2, \cdots, \pi_{i-1}, \pi_{i+1}, \pi_{i+2}, \cdots, \pi_n, \pi_i)$ (which is achieved from π by moving π_i to the end).*
- *Let $\pi^{Copy,i}$ be the permutation $(\pi_1, \pi_2, \cdots, \pi_n, \pi_i)$ (achieved from π by copying π_i to the end without removing the original π_i). We note that $\pi^{Copy,i}$ is a sequence of $n+1$ items with two copies of item π_i.*

Let $MG(i, \pi)$ denote the marginal gain that the algorithm Greedy gets by allocating the ith online item π_i in permutation π, and let $\mathbb{S}_n$ be the set of all $n!$ permutations on the n items. The algorithm Greedy achieves welfare $Welfare(Greedy) = \mathbb{E}_{\pi\sim\mathbb{S}_n}[\sum_{i=1}^{n} MG(i, \pi)]$ where the expectation is taken over permutations π chosen uniformly at random from $\mathbb{S}_n$. It is clear that $MG(i, \pi) \geq MG(n, \pi^{Move,i})$ for any π since the first $i-1$ items of both π and $\pi^{Move,i}$ are the same, and therefore Greedy's allocation on them is the same; the claim is implied by the non-increasing marginals property of submodular functions. Now we can present the conjecture, and the improved competitive ratio for greedy based on it:

CONJECTURE 15. *For any instance of the online submodular welfare maximization problem, we conjecture that*

$$\mathbb{E}_{\pi\sim\mathbb{S}_n}\left[\sum_{i=1}^{n} MG(n+1, \pi^{Copy,i})\right] \leq \mathbb{E}_{\pi\sim\mathbb{S}_n}\left[\sum_{i=1}^{n} MG(n, \pi^{Move,i})\right].$$

We also note that the latter is equal to $n\mathbb{E}_{\pi\sim\mathbb{S}_n}[MG(n, \pi)]$ as both π and $\pi^{Move,i}$ have the same distribution.

THEOREM 16. *Assuming Conjecture 15 holds, the competitive ratio of the Greedy algorithm for online SWM is at least* 0.567 *in the random order model.*

6. CONCLUSIONS AND OPEN PROBLEMS

As we have seen, the Greedy algorithm, which achieves an optimal competitive ratio of 1/2 for online SWM in the adversarial setting, does strictly better in the random order setting. We showed that the competitive ratio of this algorithm is at least 0.5052, and defined the new and interesting class of *second-order* supermodular functions (including weighted matching and weighted coverage functions), for which the ratio is at least 0.5104. Further, under Conjecture 15, the competitive ratio is considerably better, at least 0.567, which is more than halfway between 0.5 and $1 - 1/e$. This work motivates several open problems, which are interesting directions for future research:

- First, and most obviously, can one prove Conjecture 15? If true, as discussed above, this gives an immediate improvement to the competitive ratio of the Greedy algorithm.
- We believe it should be possible to improve on the competitive ratios of both Theorems 1 and 3. Our work broke the barrier of 1/2, but further improvements may be possible via a more careful analysis.
- A natural question is whether the Greedy algorithm does in fact achieve a ratio of $1 - 1/e$ in the random order model. A hardness result showing that this ratio is impossible would be extremely interesting, yielding one of the first provable separations between the random order and i.i.d. models.
- Finally, the new classes of second-order modular, second-order supermodular and second-order submodular functions that we defined are likely to be of independent interest. We may be able to refine our understanding of submodular optimization by determining which problems become more tractable for submodular functions in these classes.

References

[1] Gagan Agarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocation. In *SODA*. SIAM, 2011.

[2] Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. A dynamic near-optimal algorithm for online linear programming. *Computing Research Repository*, 2009.

[3] Saeed Alaei, MohammadTaghi Hajiaghayi, and Vahid Liaghat. Online prophet-inequality matching with applications to ad allocation. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 18–35. ACM, 2012.

[4] Niv Buchbinder, Kamal Jain, and Joseph Seffi Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *ESA*, pages 253–264. Springer, 2007.

[5] Nikhil Devanur and Thomas Hayes. The adwords problem: Online keyword matching with budgeted bidders under random permutations. In *EC*, pages 71–78, 2009.

[6] Nikhil R. Devanur, Zhiyi Huang, Nitish Korula, Vahab S. Mirrokni, and Qiqi Yan. Whole-page optimization and submodular welfare maximization with online bidders. In *ACM Conference on Electronic Commerce*, pages 305–322, 2013.

[7] Nikhil R. Devanur, Kamal Jain, Balasubramanian Sivan, and Christopher A. Wilkens. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *EC*, pages 29–38. ACM, 2011.

[8] Nikhil R Devanur, Balasubramanian Sivan, and Yossi Azar. Asymptotically optimal algorithm for stochastic adwords. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 388–404. ACM, 2012.

[9] Shahar Dobzinski, Noam Nisan, and Michael Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. In *STOC*, pages 610–618, 2005.

[10] Shahar Dobzinski and Michael Schapira. An improved approximation algorithm for combinatorial auctions with submodular bidders. In *SODA*, pages 1064–1073, 2006.

[11] J. Feldman, N. Korula, V. Mirrokni, S. Muthukrishnan, and M. Pal. Online ad assignment with free disposal. In *WINE*, 2009.

[12] Jon Feldman, Monika Henzinger, Nitish Korula, Vahab S. Mirrokni, and Cliff Stein. Online stochastic packing applied to display ad allocation. In *ESA*, pages 182–194. Springer, 2010.

[13] Jon Feldman, Aranyak Mehta, Vahab Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating 1 - 1/e. In *FOCS*, pages 117–126, 2009.

[14] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. An analysis of the approximations for maximizing submodular set functions II. *Mathematical Programming Study*, 8:73–87, 1978.

[15] Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *SODA*, pages 982–991, 2008.

[16] B. Haeupler, V. Mirrokni, and M. ZadiMoghaddam. Online stochastic weighted matching: Improved approximation algorithms. In *WINE*, pages 170–181, 2011.

[17] Bala Kalyanasundaram and Kirk R. Pruhs. An optimal deterministic algorithm for online b -matching. *Theoretical Computer Science*, 233(1–2):319–325, 2000.

[18] Michael Kapralov, Ian Post, and Jan Vondrák. Online submodular welfare maximization: Greedy is optimal. In *SODA*, pages 1216–1225, 2013.

[19] Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. Online bipartite matching with unknown distributions. In *STOC*, pages 587–596, 2011.

[20] R.M. Karp, U.V. Vazirani, and V.V. Vazirani. An optimal algorithm for online bipartite matching. In *STOC*, pages 352–358, 1990.

[21] Lehman, Lehman, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behaviour*, pages 270–296, 2006.

[22] Renato Paes Leme, Vasilis Syrgkanis, and Éva Tardos. Sequential auctions and externalities. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 869–886. SIAM, 2012.

[23] Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: A strongly factor revealing lp approach. In *STOC*, pages 597–606, 2011.

[24] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *J. ACM*, 54(5):22, 2007.

[25] Vahideh H. Menshadi, Shayan Oveis Gharan, and A. Saberi. Online stochastic matching: Online actions based on offline statistics. In *SODA*, pages 1285–1294, 2011.

[26] Vahab Mirokni and Morteza Zadimoghaddam. Randomized composable core-sets for distributed submodular maximization. In *Proceedings of the 47th Annual Symposium on the Theory of Computing*, 2015. To appear.

[27] Vahab S. Mirrokni, Shayan Oveis Gharan, and Morteza ZadiMoghaddam. Simultaneous approximations of stochastic and adversarial budgeted allocation problems. In *SODA*, pages 1690–1701, 2011.

[28] Vahab S. Mirrokni, Michael Schapira, and Jan Vondrák. Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions. In *ACM Conference on Electronic Commerce*, pages 70–77, 2008.

[29] Marco Molinaro and R Ravi. The geometry of online packing linear programs. *Mathematics of Operations Research*, 2013.

[30] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions. I. *Math. Programming*, 14(3):265–294, 1978.

[31] Vasilis Syrgkanis and Eva Tardos. Bayesian sequential auctions. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 929–944. ACM, 2012.

[32] Vasilis Syrgkanis and Eva Tardos. Composable and efficient mechanisms. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pages 211–220. ACM, 2013.

[33] Erik Vee, Sergei Vassilvitskii, and Jayavel Shanmugasundaram. Optimal online assignment with forecasts. In *EC*, pages 109–118, 2010.

[34] J. Vondrak. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC*, pages 67–74, 2008.