Mathematical Programming (2020) 183:149–169
https://doi.org/10.1007/s10107-019-01459-z

FULL LENGTH PAPER

Series B

Online submodular maximization: beating 1/2 made simple

Niv Buchbinder[1] · **Moran Feldman**[2,4] · **Yuval Filmus**[3] · **Mohit Garg**[2,5]

Received: 21 April 2019 / Accepted: 12 December 2019 / Published online: 3 January 2020

Abstract

The `Submodular Welfare Maximization` problem (SWM) captures an important subclass of combinatorial auctions and has been studied extensively. In particular, it has been studied in a natural online setting in which items arrive one-by-one and should be allocated irrevocably upon arrival. For this setting, Korula et al. (SIAM J Comput 47(3):1056–1086, 2018) were able to show that the greedy algorithm is 0.5052-competitive when the items arrive in a uniformly random order. Unfortunately, however, their proof is very long and involved. In this work, we present an (arguably) much simpler analysis of the same algorithm that provides a slightly better guarantee of 0.5096-competitiveness. Moreover, this analysis applies also to a generalization of online SWM in which the sets defining a (simple) partition matroid arrive online in a uniformly random order, and we would like to maximize a monotone submodular function subject to this matroid. Furthermore, for this more general problem, we prove an upper bound of 0.574 on the competitive ratio of the greedy algorithm, ruling out the possibility that the competitiveness of this natural algorithm matches the optimal offline approximation ratio of $1 - {}^1\!/_e$.

Keywords Submodular optimization · Online auctions · Greedy algorithms

Mathematics Subject Classification 68W27 · 68W40 · 90B99

An earlier version of this work appeared in IPCO 2019 [4].
We thank Nitish Korula, Vahab S. Mirrokni and Morteza Zadimoghaddam for sharing with us the full version of their paper [27] before it was officially published. The research of Niv Buchbinder was supported by Israel Science Foundation Grant 2233/19 and United States - Israel Binational Science Foundation Grant 2018352. The research of Moran Feldman and Mohit Garg was supported in part by Israel Science Foundation Grant 1357/16. Yuval Filmus is a Taub Fellow—supported by the Taub Foundations. His research was funded by Israel Science Foundation Grant 1337/16.

Extended author information available on the last page of the article

1 Introduction

The `Submodular Welfare Maximization` problem (SWM) captures an important subclass of combinatorial auctions and has been studied extensively from both computational and economic perspectives. In this problem we are given a set of m items and a set of n bidders, where each bidder has a non-negative monotone submodular utility function,[1] and the objective is to partition the items among the bidders in a way that maximizes the total utility of the bidders. Interestingly, SWM generalizes other extensively studied problems such as maximum (weighted) matching and budgeted allocation (see [30] for a comprehensive survey).

SWM is usually studied in the value oracle model (see Sect. 2 for definition). In this model the best approximation ratio for SWM is $1 - (1 - 1/n)^n \geq (1 - 1/e)$ [8,16,32]. A different line of work studies SWM in a natural online setting in which items arrive one-by-one and should be allocated irrevocably upon arrival. This setting generalizes, for example, online (weighted) matching and budgeted allocation [1,7,14,22,25,31,35]. It is well known that for this online setting the greedy approach that allocates each item to the bidder with the currently maximal marginal gain for the item is $1/2$-competitive, which is the optimal deterministic competitive ratio [18,23]. While randomization is known to be very helpful for many special cases of online SWM (e.g., matching), Kapralov et al. [23] proved that, unfortunately, this is not the case for online SWM itself—i.e., no (randomized) algorithm can achieve a competitive ratio better than $1/2$ for this problem (unless NP $=$ RP).

A common relaxation of the online setting is to assume that the items arrive in a random order rather than in an adversarial one [9,19]. This model was also studied extensively for special cases of SWM for which improved algorithms were obtained [19,24,28]. Surprisingly, unlike in the adversarial setting, Korula et al. [27] showed that the simple (deterministic) greedy algorithm achieves a competitive ratio of at least 0.5052 in the random arrival model. Unfortunately, the analysis of the greedy algorithm by Korula et al. [27] is very long and involves many tedious calculations, making it very difficult to understand why it works or how to improve it.

1.1 Our results

In this paper, we study the problem of maximizing a monotone submodular function over a (simple) partition matroid. This problem is a generalization of SWM (see Sect. 2 for exact definitions and a standard reduction between the problems) in which a ground set $\mathcal{N}$ is partitioned into disjoint non-empty sets $P_1, P_2, \ldots, P_m$. The goal is to choose a subset $S \subseteq \mathcal{N}$ that contains at most one element from each set P_i and maximizes a given non-negative monotone submodular function f.[2] We are interested in the performance of the greedy algorithm for this problem when the sets P_i are ordered uniformly at random. A formal description of the algorithm is given as Algorithm 1.

[1] A set function $f : 2^{\mathcal{N}} \to \mathbb{R}$ is *monotone* if $f(S) \leq f(T)$ for every two sets $S \subseteq T \subseteq \mathcal{N}$ and *submodular* if $f(S \cup \{u\}) - f(S) \geq f(T \cup \{u\}) - f(T)$ for every two such sets and an element $u \in \mathcal{N} \setminus T$.

[2] This constraint on the set of items that can be selected is equivalent to selecting an independent set of the partition matroid $\mathcal{M}$ defined by the partition $\{P_1, P_2, \ldots, P_m\}$.

Algorithm 1: Random Order Greedy(f, $\mathcal{M}$)

Initialize: $A_0 \leftarrow \varnothing$.
Let π be a uniformly random permutation of $[m]$.
for $i = 1$ **to** m **do**
  Let u_i be the element $u \in P_{\pi(i)}$ maximizing $f(u \mid A_{i-1}) \triangleq f(A_{i-1} \cup \{u\}) - f(A_{i-1})$.
  $A_i \leftarrow A_{i-1} \cup \{u_i\}$.
Return A_m.

It is well known that for a fixed (rather than random) permutation π, the greedy algorithm achieves exactly $1/2$-approximation [18]. We prove the following result.

Theorem 1 *Algorithm 1 achieves an approximation ratio of at least* 0.5096 *for the problem of maximizing a non-negative monotone submodular function subject to a partition matroid constraint.*

Through a standard reduction from SWM, this result yields the same guarantee also on the performance of the greedy algorithm for SWM in the random order model. Thus, the result both generalizes and improves over the previously known 0.5052-approximation [27]. Our analysis is also arguably simpler, giving a direct, clean, and short proof that avoids the use of factor revealing LPs.

It should also be mentioned that the result of Korula et al. [27] represents the first combinatorial algorithm for offline SWM achieving a better approximation ratio than $1/2$. Analogously, our result is a combinatorial algorithm achieving a better than $1/2$ approximation ratio for the more general problem of maximizing a non-negative monotone submodular function subject to a partition matroid constraint. We remark that in a recent work Buchbinder et al. [5] described a (very different) offline combinatorial algorithm which achieves a better than $1/2$ approximation for the even more general problem of maximizing a non-negative monotone submodular function subject to a general matroid constraint. However, the approximation guarantee achieved in [5] is worse, and the algorithm is more complicated and cannot be implemented in an online model.

The greedy algorithm in the random arrival model is known to be $(1 - 1/e)$-competitive for special cases of SWM [19]. For online SWM it is an open question whether the algorithm achieves this (best possible) ratio. However, for the more general problem of maximizing a monotone submodular function over a partition matroid, the following result answers this question negatively. In fact, the result shows that the approximation ratio obtained by the greedy algorithm is quite far from $1 - 1/e \approx 0.632$.

Theorem 2 *There exist a partition matroid $\mathcal{M}$ and a non-negative monotone submodular function f over the same ground set such that the approximation ratio of Algorithm 1 for the problem of maximizing f subject to the constraint defined by $\mathcal{M}$ is at most* $207/361 < 0.574$.[3]

[3] A slightly weaker bound of $19/33$, with a simpler proof, appeared in the conference version of this paper [4].

1.2 Our technique

The proof we describe for Theorem 1 consists of two parts. First, we show in Sect. 3.1 that in the random arrival model the greedy algorithm gains most of the value of its output set during its first iterations (Lemma 3). For example, after viewing 90% of the sets the algorithm already has 49.5% of the value of the optimal solution, which is 99% of its output guarantee according to the standard analysis. Thus, to prove that the greedy algorithm has a better than 1/2 approximation ratio, it suffices to show that it gets a non-negligible gain from its last iterations.

In the second part of our analysis (Sect. 3.2), we are able to show that this is indeed the case. Intuitively, in this part of the analysis we view the execution of Algorithm 1 as having three stages defined by two integer values $0 < r \leq r' < m$. The first stage consists of the first r iterations of the algorithm, the second stage consists of the next $r' - r$ iterations and the last stage consists of the remaining $m - r'$ iterations. As explained above, by Lemma 3 we get that if r' is large enough, then $f(A_{r'})$ is already very close to $f(OPT)/2$, where OPT is an optimal solution (recall that $A_{r'}$ is the set constructed in iteration r' of Algorithm 1). We use two steps to prove that $f(A_m)$ is significantly larger than $f(A_{r'})$, and thus, achieves a better than 1/2 approximation ratio. In the first step (Lemma 4), we use symmetry to argue that there are two independent sets of $\mathscr{M}$ that consist only of elements that Algorithm 1 can pick in its second and third phases, and in addition, the value of their union is large. One of these sets consists of the elements of OPT that are available in the final $m - r$ iterations, and the other set (which we denote by C) is obtained by applying an appropriately chosen function to these elements of OPT. In the second step of the analysis, implemented by Lemma 5, we use the fact that the final $m - r'$ elements of C are a random subset of C to argue that they have a large marginal contribution even with respect to the final solution A_m. Combining this with the observation that these elements represent a possible set of elements that Algorithm 1 could pick during its last stage, we get that the algorithm must have made a significant gain during this stage.

1.3 Additional related results

The optimal approximation ratio for the problem of maximizing a monotone submodular function subject to a partition matroid constraint (and its special case SWM) is obtained by an algorithm known as (Measured) Continuous Greedy [8,16]. Unfortunately, this algorithm is problematic from a practical point of view since it is based on a continuous relaxation and is quite slow. As discussed above, our first result can be viewed as an alternative simple combinatorial algorithm for this problem, and thus, it is related to a line of work that aims to find better alternatives for Continuous Greedy [3,6,17,33].

While the problem of maximizing a monotone submodular function subject to a partition matroid was studied almost exclusively in the value oracle model, the view of SWM as an auction has motivated its study also in an alternative model known as the demand oracle model. In this model a strictly better than $(1 - 1/e)$-approximation is known for the problem [13].

Another online model, that can be cast as a special case of the random arrival model and was studied extensively, is the i.i.d. stochastic model. In this model input items arrive i.i.d. according to a known or unknown distribution. In the i.i.d. model with a known distribution improved competitive ratios for special cases of SWM are known [2,15,21,29]. Moreover, for the i.i.d. model with an unknown distribution a $(1 - 1/e)$-competitive algorithm is known for SWM as well as for several of its special cases [10,11,23].

Finally, we would like to note that, beside the problem we consider, many other random order problems exhibit the property that the solution of the greedy algorithm tends to achieve most of its value early on. A few examples of algorithms exploiting this property can be found in [20,26,34].

2 Preliminaries

For every two sets $S, T \subseteq \mathcal{N}$ we denote the marginal contribution of adding T to S, with respect to a set function f, by $f(T \mid S) \triangleq f(T \cup S) - f(S)$. For an element $u \in \mathcal{N}$ we use $f(u \mid S)$ as shorthands for $f(\{u\} \mid S)$—note that we have already used this notation previously in Algorithm 1.

Following are two useful claims that we use in the analysis of Algorithm 1. The first of these claims is a rephrased version of a useful lemma which was first proved in [12], and the other is a well known technical observation that we prove here for completeness.

Lemma 1 (Lemma 2.2 of [12]) *Let $f : 2^{\mathcal{N}} \to \mathbb{R}$ be a submodular function, and let T be an arbitrary set $T \subseteq \mathcal{N}$. For every random set $T_p \subseteq T$ which contains every element of T with probability p (not necessarily independently),*

$$\mathbb{E}[f(T_p)] \geq (1 - p) \cdot f(\varnothing) + p \cdot f(T).$$

Observation 1 *For every sets two $S_1 \subseteq S_2 \subseteq \mathcal{N}$ and an additional set $T \subseteq \mathcal{N}$, it holds that*

$$f(S_1 \mid T) \leq f(S_2 \mid T) \quad \textit{and} \quad f(T \mid S_1) \geq f(T \mid S_2).$$

Proof The first inequality holds since the monotonicity of f implies that

$$f(S_1 \mid T) = f(S_1 \cup T) - f(T) \leq f(S_2 \cup T) - f(T) = f(S_2 \mid T),$$

and the second inequality holds since

$$\begin{aligned} f(T \mid S_1) &\geq f(T \mid S_1 \cup (S_2 \setminus T)) = f(T \cup S_2) - f(S_1 \cup (S_2 \setminus T)) \\ &\geq f(T \cup S_2) - f(S_2) = f(T \mid S_2), \end{aligned}$$

where the first inequality follows from submodularity and the second from monotonicity. □

The `Submodular Welfare Maximization` *problem* (SWM) In this problem we are given a set $\mathcal{N}$ of m items and a set B of n bidders. Each bidder i has a non-negative monotone submodular utility function $f_i : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$; and the goal is to partition the items among the bidders in a way that maximizes $\sum_{i=1}^{m} f_i(S_i)$ where S_i is the set of items allocated to bidder i.

Maximizing a monotone submodular function over a (simple) partition matroid In this problem we are given a partition matroid $\mathcal{M}$ over a ground set $\mathcal{N}$ and a non-negative monotone submodular function $f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$. A partition matroid is defined by a partition of its ground set into non-empty disjoint sets $P_1, P_2, \dots, P_m$. A set $S \subseteq \mathcal{N}$ is independent in $\mathcal{M}$ if $|S \cap P_i| \leq 1$ for every set P_i, and the goal in this problem is to find a set $S \subseteq \mathcal{N}$ that is independent in $\mathcal{M}$ and maximizes f.

In this work we make the standard assumption that the objective function f can be accessed only through a value oracle, i.e., an oracle that given a subset S returns the value $f(S)$.

A standard reduction between the above two problems Given an instance of SWM, we construct the following equivalent instance of maximizing a monotone submodular function subject to a partition matroid. For each item $u \in \mathcal{N}$ and bidder $i \in B$, we create an element (u, i) which represents the assignment of u to i. Additionally, we define a partition of these elements by constructing for every item u a set $P_u = \{(u, i) \mid i \in B\}$. Finally, for a subset S of the elements, we define

$$f(S) = \sum_{i \in B} f_i(\{u \in \mathcal{N} \mid (u, i) \in S\}).$$

One can verify that for every independent set S the value of f is equal to the total utility of the bidders given the assignment represented by S; and moreover, f is non-negative, monotone and submodular.

It is important to note that running a greedy algorithm that inspects the partitions in a random order after this reduction is the same as running the greedy algorithm on the original SWM instance in the random arrival model.

Additional technical reduction Our analysis of Algorithm 1 uses two integer parameters $0 < r \leq r' < m$. A natural way to choose these parameters is to set them to $r = \alpha m$ and $r' = \beta m$, where α and β are rational numbers. Unfortunately, not for every choice of α, β and m these values are integral. The following reduction allows us to bypass this technical issue.

Reduction 1 *For any fixed choice of two rational values* $\alpha, \beta \in (0, 1)$*, one may assume that* αm *and* βm *are both integral for the purpose of analyzing the approximation ratio of Algorithm 1.*

Proof Since α and β are positive rational numbers, they can be represented as ratios a_1/a_2 and b_1/b_2, where a_1, a_2, b_1 and b_2 are all natural numbers. This implies that we can make αm and βm integral by increasing m by some integer value $0 \leq m' < a_2 b_2$.

To achieve this increase, we introduce m' new dummy elements into the ground set and extend the objective function and partition matroid in the following way. Let D be the set of the m' dummy elements.

- For every set S that contains dummy elements, we define $f(S) = f(S \backslash D)$.
- For every dummy element $d \in D$, we introduce a new set that contains only d into the partition defining the matroid. Note that this implies that a set S that contains dummy elements is independent if and only if $S \backslash D$ is independent.

One can observe that this extension does not change the value of the optimal solution. Additionally, we observe that the extension does not affect the distribution of the value of the output set of Algorithm 1 because the fact that the algorithm added a dummy element to its solution does not affect either the current value of the solution or the marginals of elements considered later (in other words, the extension makes the algorithm have m' new meaningless iterations in which it picks dummy elements, but it does not affect the behavior of the algorithm in the other iterations).

The above observations imply that the approximation ratio of Algorithm 1 is not affected by the extension, and thus, the approximation that the algorithm has for the extended instance (in which αm and βm are integral) holds for the original instance as well. $\square$

3 Analysis of the approximation ratio

In this section, we analyze Algorithm 1 and lower bound its approximation ratio. The analysis is split between Sects. 3.1 and 3.2. In Sect. 3.1 we present a basic (and quite standard) analysis of Algorithm 1 which only shows that it is a 1/2-approximation algorithm, but proves along the way some useful properties of the algorithm. In Sect. 3.2 we use these properties to present a more advanced analysis of Algorithm 1 which shows that it is a 0.5096-approximation algorithm (and thus proves Theorem 1).

Let us now define some notation that we use in both parts of the analysis. Let OPT be an optimal solution (i.e., an independent set of $\mathcal{M}$ maximizing f). Note that since f is monotone we may assume, without loss of generality, that OPT is a base of $\mathcal{M}$ (i.e., it includes exactly one element of the set P_i for every $1 \leq i \leq m$). Additionally, for every set $T \subseteq \mathcal{N}$ we denote by $T^{(i)}$ the subset of T that excludes elements appearing in the first i sets out of $P_1, P_2, \ldots, P_m$ when these sets are ordered according to the permutation π. More formally,

$$T^{(i)} = T \backslash \bigcup_{j=1}^{i} P_{\pi(j)} = T \cap \bigcup_{j=i+1}^{m} P_{\pi(j)}.$$

Since π is a uniformly random permutation and OPT contains exactly one element of each set P_i (due to our assumption that it is a base of $\mathcal{M}$), we get the following observation as an immediate consequence.

Observation 2 *For every $0 \leq i \leq m$, $OPT^{(i)}$ is a uniformly random subset of OPT of size $m - i$.*

3.1 Basic analysis

In this section, we present a basic analysis of Algorithm 1. Following is the central lemma of this analysis which shows that the expression $f(A_i) + f(S \cup A_i \cup T^{(i)})$ is a non-decreasing function of i for every pair of set $S \subseteq \mathcal{N}$ and base T of $\mathcal{M}$ (recall that A_i is the set constructed by Algorithm 1 during its ith iteration). It is important to note that this lemma holds deterministically, i.e., it holds for **every** given permutation π.

Lemma 2 *For every subset $S \subseteq \mathcal{N}$, base T of $\mathcal{M}$ and $1 \leq i \leq m$,*

$$f(A_i) + f(S \cup A_i \cup T^{(i)}) \geq f(A_{i-1}) + f(S \cup A_{i-1} \cup T^{(i-1)}).$$

Proof Observe that

$$\begin{aligned} f(A_i) - f(A_{i-1}) &= f(u_i \mid A_{i-1}) \geq f(T \cap P_{\pi(i)} \mid A_{i-1}) \\ &\geq f(T \cap P_{\pi(i)} \mid S \cup A_{i-1} \cup T^{(i)}) \\ &= f(S \cup A_{i-1} \cup T^{(i-1)}) - f(S \cup A_{i-1} \cup T^{(i)}) \\ &\geq f(S \cup A_{i-1} \cup T^{(i-1)}) - f(S \cup A_i \cup T^{(i)}), \end{aligned}$$

where the first inequality follows from the greedy choice of the algorithm, the second inequality holds due to Observation 1 and the final inequality follows from the monotonicity of f. □

The following is an immediate corollary of the last lemma. Note that, like the lemma, it is deterministic and, thus, holds for **every** permutation π.

Corollary 1 *For every subset $S \subseteq \mathcal{N}$, base T of $\mathcal{M}$ and $0 \leq i \leq m$,*

$$f(A_m) + f(S \cup A_m) \geq f(A_i) + f(S \cup A_i \cup T^{(i)}) \geq f(S \cup T).$$

Proof Since $f(A_i) + f(S \cup A_i \cup T^{(i)})$ is a non-decreasing function of i by Lemma 2,

$$\begin{aligned} &f(A_m) + f(S \cup A_m \cup T^{(m)}) \\ &\quad \geq f(A_i) + f(S \cup A_i \cup T^{(i)}) \geq f(A_0) + f(S \cup A_0 \cup T^{(0)}). \end{aligned}$$

The corollary now follows by recalling that $A_0 = \varnothing$, observing that $f(A_0) \geq 0$ since f is non-negative and noticing that by definition $T^{(m)} = \varnothing$ and $T^{(0)} = T$. □

By choosing $S = \varnothing$ and $T = OPT$, the last corollary yields $f(A_m) \geq 1/2 \cdot f(OPT)$, which already proves that Algorithm 1 is a $1/2$-approximation algorithm as promised. The following lemma strengthens this result by showing a lower bound on the value of $f(A_i)$ for every $0 \leq i \leq m$. Note that this lower bound, unlike the previous one, holds only in expectation over the random choice of the permutation π. Let $g(x) \triangleq x - x^2/2$.

Lemma 3 *For every* $0 \leq i \leq m$, $\mathbb{E}[f(A_i)] \geq g(i/m) \cdot f(OPT)$.

We give a full Proof of Lemma 3 below for completeness. However, a very quick proof of it can be obtained by combining Lemmata 6 and 7 of [27].

Proof of Lemma 3 As explained above, for $i = m$ the lemma follows from Corollary 1. We prove the lemma for the other values of i by induction. For $i = 0$ the lemma holds, even without the expectation, due to the non-negativity of f since $g(0) = 0$. The rest of the proof is devoted to showing that the lemma holds for $1 \leq i < m$ assuming that it holds for $i - 1$.

Let π_{i-1} be an arbitrary injective function from $\{1, \ldots, i-1\}$ to $\{1, \ldots, m\}$, and let us denote by $\mathscr{E}(\pi_{i-1})$ the event that π_{i-1} agrees with the first $i-1$ parts of the random order π, that is, $\pi(j) = \pi_{i-1}(j)$ for every $1 \leq j \leq i-1$. Observe that conditioned on this event the sets A_{i-1} and $OPT^{(i-1)}$ become deterministic. For $OPT^{(i-1)}$ this follows from the definition, and for A_{i-1} this is true because Algorithm 1 uses the values of π only for the numbers in $\{1, \ldots, i-1\}$ for constructing A_{i-1}. Thus, conditioned on $\mathscr{E}(\pi_{i-1})$,

$$\begin{aligned}
\mathbb{E}[f(A_i) - f(A_{i-1})] &= \mathbb{E}[f(u_i \mid A_{i-1})] \geq \mathbb{E}[f(OPT \cap P_{\pi(i)} \mid A_{i-1})] \\
&= \frac{\sum_{j \in [m] \setminus \pi_{i-1}([i-1])} f(OPT \cap P_j \mid A_{i-1})}{m-i+1} \\
&= \frac{\sum_{u \in OPT^{(i-1)}} f(u \mid A_{i-1})}{m-i+1} \\
&\geq \frac{f(OPT^{(i-1)} \mid A_{i-1})}{m-i+1} \geq \frac{f(OPT) - 2f(A_{i-1})}{m-i+1},
\end{aligned}$$

where the first inequality follows from the greedy choice of Algorithm 1, the second equality holds since the conditioning on $\mathscr{E}(\pi_{i-1})$ implies that $OPT \cap P_{\pi(i)}$ is a uniformly random element of OPT_{i-1}, the second inequality follows from the submodularity of f and the last inequality hold because plugging $S = \varnothing$ and $T = OPT$ into the second inequality of Corollary 1 yields $f(A_{i-1}) + f(A_{i-1} \cup OPT^{(i-1)}) \geq f(OPT)$.

Now, taking expectation over all the possible choices of π_{i-1}, we get

$$\begin{aligned}
\mathbb{E}[f(A_i)] &\geq \mathbb{E}[f(A_{i-1})] + \frac{f(OPT) - 2\mathbb{E}[f(A_{i-1})]}{m-i+1} \\
&= \frac{m-i-1}{m-i+1} \cdot \mathbb{E}[f(A_{i-1})] + \frac{f(OPT)}{m-i+1} \\
&\geq \frac{m-i-1}{m-i+1} \cdot g\left(\frac{i-1}{m}\right) \cdot f(OPT) + \frac{f(OPT)}{m-i+1} \\
&= \left[g\left(\frac{i-1}{m}\right) + \frac{1 - 2g\left(\frac{i-1}{m}\right)}{m-i+1}\right] \cdot f(OPT),
\end{aligned}$$

where the second inequality follows from the induction hypothesis (since $i \leq m-1$). Using the observations that the derivative $g'(x) = 1-x$ of $g(x)$ is non-increasing and obeys $g'(x) = (1-2g(x))/(1-x)$, the last inequality yields

$$\frac{\mathbb{E}[f(A_i)]}{f(OPT)} \geq g\left(\frac{i-1}{m}\right) + \frac{1-2g\left(\frac{i-1}{m}\right)}{m-i+1} = g\left(\frac{i-1}{m}\right) + \frac{g'\left(\frac{i-1}{m}\right)}{m}$$
$$\geq g\left(\frac{i-1}{m}\right) + \int_{(i-1)/m}^{i/m} g'(x)dx = g(i/m).$$

□

3.2 Breaking 1/2: an improved analysis of Algorithm 1

In this section, we use the properties of Algorithm 1 proved in the previous section to derive a better than 1/2 lower bound on its approximation ratio and prove Theorem 1. As explained in Sect. 1.2, we view here an execution of Algorithm 1 as consisting of three stages, where the places of transition between the stages are defined by two integer parameters $0 < r \leq r' < m$ whose values are set later in this section to $0.4m$ and $0.76m$, respectively. The first lemma that we present (Lemma 4) uses a symmetry argument to prove that there are two (not necessarily distinct) independent sets of $\mathscr{M}$ that consist only of elements that Algorithm 1 can pick in its second and third stages (the final $m - r$ iterations), and in addition, the value of their union is large. One of these sets is $OPT^{(r)}$, and the other set is obtained by applying to $OPT^{(r)}$ an appropriately chosen function h. Interestingly, the guarantee of Lemma 4 is particularly strong when the algorithm makes little progress during the second and third stages (i.e., $f(A_m \mid A_{m-r})$ is small), which intuitively is the case in which the basic analysis (from Sect. 3.1) fails to guarantee more than 1/2-approximation.

Let c be the true (unknown) approximation ratio of Algorithm 1.

Lemma 4 *There exists a function* $h\colon 2^{\mathscr{N}} \to 2^{\mathscr{N}}$ *such that*

(a) *for every* $1 \leq i \leq m$ *and set* $S \subseteq \mathscr{N}$, $|P_i \cap h(S)| = |P_i \cap S|$.
(b) $\mathbb{E}[f(h(OPT^{(r)}) \cup OPT^{(r)})] \geq f(OPT) - c^{-1} \cdot \mathbb{E}[f(A_m \mid A_{m-r})]$.

Proof Given part (b) of the lemma, it is natural to define $h(S)$, for every set $S \subseteq \mathscr{N}$, as the set T maximizing $f(T \cup S)$ among all the sets obeying part (a) of the lemma (where ties are broken in an arbitrary way). In the rest of the proof we show that this function indeed obeys part (b).

Observe that

$$\begin{aligned}
&f(A_{m-r} \cup (OPT \backslash OPT^{(m-r)})) \\
&\quad = f(OPT \backslash OPT^{(m-r)} \mid A_{m-r}) + f(A_{m-r}) \\
&\quad \geq f(OPT \backslash OPT^{(m-r)} \mid A_{m-r} \cup OPT^{(m-r)}) + f(A_{m-r}) \\
&\quad = f(A_{m-r} \cup OPT) - f(OPT^{(m-r)} \mid A_{m-r}) \\
&\quad \geq f(OPT) - f(OPT^{(m-r)} \mid A_{m-r}),
\end{aligned}$$

where first inequality follows from Observation 1 and the second follows by the monotonicity of f. We now note that the last r iterations of Algorithm 1 can be viewed as a standalone execution of this algorithm on the partition matroid defined by the sets

$P_{m-r+1}, \ldots, P_m$ and the objective function $f(\cdot \mid A_{m-r})$. Thus, by the definition of c, the expected value of $f(OPT^{(m-r)} \mid A_{m-r})$ is at most $c^{-1} \cdot \mathbb{E}[f(A_m \backslash A_{m-r} \mid A_{m-r})] = c^{-1} \cdot \mathbb{E}[f(A_m \mid A_{m-r})]$. Combining this with the previous inequality, we get

$$\begin{aligned}
&\mathbb{E}[f(h(OPT \backslash OPT^{(m-r)}) \cup (OPT \backslash OPT^{(m-r)}))] \\
&\quad \geq \mathbb{E}[f(A_{m-r} \cup (OPT \backslash OPT^{(m-r)}))] \\
&\quad \geq \mathbb{E}[f(OPT) - f(OPT^{(m-r)} \mid A_{m-r})] \\
&\quad \geq f(OPT) - c^{-1} \cdot \mathbb{E}[f(A_m \mid A_{m-r})],
\end{aligned}$$

where the first inequality holds due to the definition of h since A_{m-r} obeys part (a) of the lemma (for $S = OPT \backslash OPT^{(m-r)}$).

To prove the lemma it remains to observe that by Observation 2 the random sets $OPT \backslash OPT^{(m-r)}$ and $OPT^{(r)}$ have the same distribution, which implies that $f(h(OPT \backslash OPT^{(m-r)}) \cup (OPT \backslash OPT^{(m-r)}))$ and $f(h(OPT^{(r)}) \cup OPT^{(r)})$ have the same expectation. □

Let us denote $C = h(OPT^{(r)})$. Note that C is a random set since $OPT^{(r)}$ is random. The following lemma uses the properties of C proved by Lemma 4 to show that Algorithm 1 must make a significant gain during its third stage. Intuitively, the guarantee of this lemma is useful because the basic analysis implies that when Algorithm 1 does not get much more than 1/2-approximation, its solution gains most of its value early in the execution. Thus, in this case both A_r and A_{m-r} should have a significant fraction of the value of the output set A_m, and therefore, the positive terms on the right hand side of the guarantee of the lemma can counter the negative term.

Lemma 5 *Let* $p = \frac{m-r'}{m-r}$, *then*

$$\begin{aligned}
\mathbb{E}[f(A_m \mid A_{r'})] \geq f(OPT) - (2 + p/c) \cdot \mathbb{E}[f(A_m)] + p \cdot \mathbb{E}[f(A_r)] \\
+ (p/c) \cdot \mathbb{E}[f(A_{m-r})].
\end{aligned}$$

Proof Observe that

$$\begin{aligned}
f(A_m \mid A_{r'}) &\geq f(C^{(r')} \mid A_m \cup OPT^{(r)}) \\
&= f(A_m \cup C^{(r')} \mid A_r \cup OPT^{(r)}) - f(A_m \mid A_r \cup OPT^{(r)}) \\
&\geq f(C^{(r')} \mid A_r \cup OPT^{(r)}) - f(A_m \mid A_r \cup OPT^{(r)}),
\end{aligned}$$

where the first inequality holds since plugging $i = r'$, $T = A_r \cup C$ and $S = A_m \cup OPT^{(r)}$ into the first inequality of Corollary 1 yields $f(A_m) + f(A_m \cup OPT^{(r)}) \geq f(A_{r'}) + f(A_m \cup OPT^{(r)} \cup C^{(r')})$, and the second inequality follows by Observation 1.

Similar to what we do in the Proof of Lemma 3, let us now denote by π_r an arbitrary injective function from $\{1, \ldots, r\}$ to $\{1, \ldots, m\}$ and by $\mathscr{E}(\pi_r)$ the event that $\pi(j) = \pi_r(j)$ for every $1 \leq j \leq r$. Observe that conditioned on $\mathscr{E}(\pi_r)$ the sets A_r and $OPT^{(r)}$ are deterministic, and thus so is the set C which is obtained from $OPT^{(r)}$

by the application of a deterministic function; but $C^{(r')}$ remains a random set that contains every element of C with probability p. Hence, by Lemma 1, conditioned on $\mathscr{E}(\pi_r)$, we get

$$\mathbb{E}[f(C^{(r')} \mid A_r \cup OPT^{(r)})] \geq p \cdot f(C \mid A_r \cup OPT^{(r)}).$$

Taking now expectation over all the possible events $\mathscr{E}(\pi_r)$, and combining with the previous inequality, we get

$$\begin{aligned}
\mathbb{E}[f(A_m \mid A_{r'})] &\geq p \cdot \mathbb{E}[f(C \mid A_r \cup OPT^{(r)})] - \mathbb{E}[f(A_m \mid A_r \cup OPT^{(r)})] \\
&= p \cdot \mathbb{E}[f(C \cup A_r \cup OPT^{(r)})] + (1-p) \cdot \mathbb{E}[f(A_r \cup OPT^{(r)})] \\
&\quad - \mathbb{E}[f(A_m \cup OPT^{(r)})] \\
&\geq p \cdot \mathbb{E}[f(C \cup OPT^{(r)})] + (1-p) \cdot \mathbb{E}[f(A_r \cup OPT^{(r)})] \\
&\quad - \mathbb{E}[f(A_m \cup OPT^{(r)})],
\end{aligned}$$

where the second inequality holds due to the monotonicity of f. We now need to bound all the terms on the right hand side of the last inequality. The first term is lower bounded by $p \cdot f(OPT) - (p/c) \cdot \mathbb{E}[f(A_m \mid A_{m-r})]$ due to Lemma 4. A lower bound of $f(OPT) - f(A_r)$ on the expression $f(A_r \cup OPT^{(r)})$ holds since plugging $i = r$, $T = OPT$ and $S = \varnothing$ into the second inequality of Corollary 1 gives $f(A_r) + f(A_r \cup OPT^{(r)}) \geq f(OPT)$. Finally, an upper bound of $2f(A_m) - f(A_r)$ on the expression $f(A_m \cup OPT^{(r)})$ holds since plugging $i = r$, $T = OPT$ and $S = A_m$ into the first inequality of the same corollary yields $2f(A_m) \geq f(A_r) + f(A_m \cup OPT^{(r)})$. Plugging all these bounds into the previous inequality yields

$$\begin{aligned}
\mathbb{E}[f(A_m \mid A_{r'})] &\geq p \cdot f(OPT) - (p/c) \cdot \mathbb{E}[f(A_m \mid A_{m-r})] \\
&\quad + (1-p) \cdot \mathbb{E}[f(OPT) - f(A_r)] - \mathbb{E}[2f(A_m) - f(A_r)] \\
&= f(OPT) - (2 + p/c) \cdot \mathbb{E}[f(A_m)] + p \cdot \mathbb{E}[f(A_r)] \\
&\quad + (p/c) \cdot \mathbb{E}[f(A_{m-r})].
\end{aligned}$$

□

We are now ready to prove Theorem 1.

Proof of Theorem 1 Let $q = r/m$. By Lemma 5,

$$\begin{aligned}
\mathbb{E}[f(A_m)] &= \mathbb{E}[f(A_m \mid A_{r'})] + \mathbb{E}[f(A_{r'})] \\
&\geq f(OPT) - (2 + p/c) \cdot \mathbb{E}[f(A_m)] + p \cdot \mathbb{E}[f(A_r)] \\
&\quad + (p/c) \cdot \mathbb{E}[f(A_{m-r})] + \mathbb{E}[f(A_{r'})].
\end{aligned}$$

Rearranging this inequality, and using the lower bound on $\mathbb{E}[f(A_i)]$ given by Lemma 3, we get

$$
\begin{aligned}
&(3 + p/c) \cdot \mathbb{E}[f(A_m)] \\
&\quad \geq [1 + p \cdot g(q) + (p/c) \cdot g(1-q) + g(1-p+pq)] \cdot f(OPT) \\
&\quad = [1 + pq(1-q/2) + (p/c)(1-q^2)/2 \\
&\qquad + (1 - p^2 + 2p^2q - p^2q^2)/2] \cdot f(OPT) \\
&\quad = \frac{1}{2}[3 + pq(2-q) + pc^{-1}(1-q^2) - p^2(1-q)^2] \cdot f(OPT).
\end{aligned}
$$

Thus, the approximation ratio of Algorithm 1 is at least

$$
\frac{3 + pq(2-q) + pc^{-1}(1-q^2) - p^2(1-q)^2}{6 + 2pc^{-1}}.
$$

Since c is the true approximation ratio of this algorithm by definition, we get

$$
c \geq \frac{3 + pq(2-q) + pc^{-1}(1-q^2) - p^2(1-q)^2}{6 + 2pc^{-1}}.
$$

We now choose $p = q = 0.4$. Notice that these values for p and q can be achieved by setting $r = 0.4m$ and $r' = 0.76m$, and moreover, we can assume that this is a valid choice for r and r' by Reduction 1. Plugging these values of p and q into the last inequality and simplifying, we get $6c^2 - 2.3984c - 0.336 \geq 0$. The roots of this quadratic are $c \approx -0.109886$ and $c \approx 0.509619$, and so all positive solutions for this inequality are larger than 0.5096, which completes the proof of the theorem. $\square$

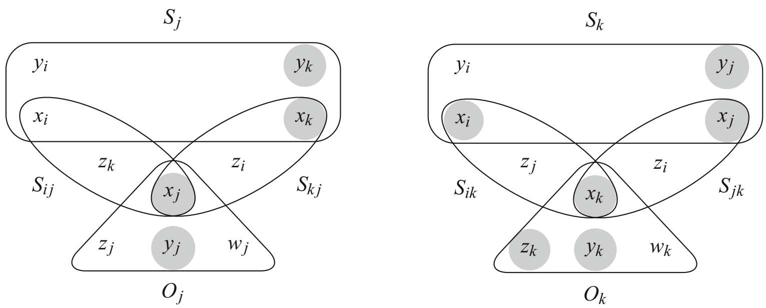
4 Upper bounding the approximation ratio

In this section, we prove Theorem 2. As a warm up, we first prove, in Sect. 4.1, a weaker form of Theorem 2 with a bound of $7/12 \approx 0.583$ instead of $207/361 < 0.574$. The proof of the theorem as stated appears in Sect. 4.2.

4.1 Warm-up: an upper bound of $7/12 \approx 0.583$

To prove this upper bound, we exhibit an instance on which the approximation ratio obtained by Algorithm 1 is $7/12$. We found this instance using a computer program, as we explain in detail below. It consists of a partition matroid over a ground set $\mathcal{N}$ consisting of twelve elements and a non-negative monotone submodular function $f : 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ over the same ground set. The partition matroid is defined by a partition of the ground set into three sets

$$
\begin{aligned}
&P_1 = \{O_1, S_1, S_{21}, S_{31}\}, \\
&P_2 = \{O_2, S_2, S_{12}, S_{32}\} \\
\text{and}\quad &P_3 = \{O_3, S_3, S_{13}, S_{23}\}.
\end{aligned}
$$



(a) Second step **(b)** Third step

Fig. 1 The available choices at the second and third steps of the algorithm when the order of partitions is $\langle P_i, P_j, P_k\rangle$. Elements already selected in previous steps are highlighted

To define the function f, we view each element of $\mathcal{N}$ as a subset of an underlying universe $\mathcal{U}$ consisting of 12 elements:

$$\mathcal{U} = \{x_i, y_i, z_i, w_i \mid 1 \leq i \leq 3\}.$$

The function f is then given as the coverage function $f(S) = |\bigcup_{u \in S} u|$ (coverage functions are known to be non-negative, monotone and submodular). The following table completes the definition of f by specifying the exact subset of $\mathcal{U}$ represented by each element of $\mathcal{N}$.

Elements of P_1	Elements of P_2	Elements of P_3
$O_1 = \{x_1, y_1, z_1, w_1\}$	$O_2 = \{x_2, y_2, z_2, w_2\}$	$O_3 = \{x_3, y_3, z_3, w_3\}$
$S_1 = \{x_2, y_2, x_3, y_3\}$	$S_2 = \{x_1, y_1, x_3, y_3\}$	$S_3 = \{x_1, y_1, x_2, y_2\}$
$S_{21} = \{x_1, x_2, z_3\}$	$S_{12} = \{x_1, x_2, z_3\}$	$S_{13} = \{x_1, z_2, x_3\}$
$S_{31} = \{x_1, z_2, x_3\}$	$S_{32} = \{z_1, x_2, x_3\}$	$S_{23} = \{z_1, x_2, x_3\}$

It is easy to verify that the optimal solution for this instance (i.e., the independent set of $\mathcal{M}$ maximizing f) is the set $\{O_1, O_2, O_3\}$, whose value is 12.

To analyze the performance of Algorithm 1 on this instance, we must set a tie-breaking rule. Here we assume that the algorithm always breaks ties in favor of the set which appears toward the bottom of the table, but it should be noted that a small perturbation of the values of f can be used to make the analysis independent of the tie-breaking rule used (at the cost of weakening the impossibility proved by an additive ε term for an arbitrary small constant $\varepsilon > 0$).

Suppose the partitions arrive in the order $\langle P_i, P_j, P_k\rangle$. In the first step, the algorithm has to choose one of the sets O_i, S_i, S_{ji}, S_{ki}. The sets O_i and S_i contain 4 elements each, while the sets S_{ji}, S_{ki} contain 3 elements each, and so the algorithm chooses the set $S_i = \{x_j, y_j, x_k, y_k\}$ in the first step.

In the second step, the algorithm has to choose one of the sets O_j, S_j, S_{ij}, S_{kj}. These sets add the following elements to the already chosen S_i (see Fig. 1a):

O_j	S_j	S_{ij}	S_{kj}
z_j, w_j	x_i, y_i	x_i, z_k	z_i

The first three add 2 elements, while the last one adds 1, so the algorithm chooses the set S_{ij} in the second step. Note that at this point, the elements covered are $\{x_i, x_j, y_j, x_k, y_k, z_k\}$.

In the final step, the algorithm has to choose one of the sets O_k, S_k, S_{ik}, S_{jk}. Each of these sets adds only a single new element (see Fig. 1b):

O_k	S_k	S_{ik}	S_{jk}
w_k	y_i	z_j	z_i

Therefore, regardless of the algorithm's choice, the output solution covers only 7 elements in total. Hence, the approximation ratio achieved by Algorithm 1 for the above instance (and any arrival order of the partitions) is $(4+2+1)/12 = 7/12$.

Remark It should be noted that by combining multiple independent copies of the above described instance, one can get an arbitrarily large instance for which the approximation ratio of Algorithm 1 is only 7/12. This rules out the possibility that the approximation ratio of Algorithm 1 approaches $1 - 1/e$ for large enough instances.

How we found this example The example described in this section, as well as the more complicated one described in Sect. 4.2, was constructed using a computer program. We now briefly discuss this process.

The set system described in this section is symmetric in the sense that if we permute the partitions P_1, P_2, P_3 then we get an isomorphic set system (the set system in Sect. 4.2 has fewer symmetries). Furthermore, we intend O_1, O_2, O_3 to be the optimal solution, S_i to be a valid choice in the first step if the first partition is P_i, and S_{ij} to be a valid choice in the second step if the first two partitions are P_i, P_j.

To describe how we find an example of the above form, let us denote by A, B, C, D, E, F an arbitrary collection of six sets over some domain Λ. Consider now three disjoint copies of this domain ($\Lambda_1, \Lambda_2, \Lambda_3$), and define X_i (for $X \in \{A, B, C, D, E, F\}$ and $i \in \{1, 2, 3\}$) to be the set X in the copy Λ_i. Using this notation, and taking into account the various required symmetries, we reach a general instance described by the following table.

Elements of P_1	Elements of P_2	Elements of P_3
$O_1 = A_1$	$O_2 = A_2$	$O_3 = A_3$
$S_1 = B_1 \cup C_2 \cup C_3$	$S_2 = B_2 \cup C_1 \cup C_3$	$S_3 = B_3 \cup C_1 \cup C_2$
$S_{21} = D_2 \cup E_1 \cup F_3$	$S_{12} = D_1 \cup E_2 \cup F_3$	$S_{13} = D_1 \cup E_3 \cup F_2$
$S_{31} = D_3 \cup E_1 \cup F_2$	$S_{32} = D_3 \cup E_2 \cup F_1$	$S_{23} = D_2 \cup E_3 \cup F_1$

Since the Λ_i are disjoint, we can compute the value of f in terms of the set system A, B, C, D, E, F alone. For example,

$$f(S_1 \cup S_{12}) = f(B_1 \cup C_2 \cup C_3 \cup D_1 \cup E_2 \cup F_3) = |B \cup D| + |C \cup E| + |C \cup F|.$$

Among all the instances captured by the above form, we would like to find an instance yielding the smallest possible approximation ratio. Towards this goal, we formulate a mathematical program with the following constraints.

1. For each order $\langle P_i, P_j, P_k\rangle$, S_i is a valid choice in the first step:
$$f(S_i) \geq \max\{f(O_i), f(S_{ji}), f(S_{ki})\}.$$
2. For each order $\langle P_i, P_j, P_k\rangle$, S_{ij} is a valid choice in the second step:
$$f(S_i \cup S_{ij}) \geq \max\{f(S_i \cup O_j), f(S_i \cup S_j), f(S_i \cup S_{kj})\}.$$
3. Normalization: $f(O_1 \cup O_2 \cup O_3) = 1$.

The objective of the mathematical program is to minimize

$$\max\{f(S_i \cup S_{ij} \cup O_k), f(S_i \cup S_{ij} \cup S_k), f(S_i \cup S_{ij} \cup S_{ik}), f(S_i \cup S_{ij} \cup S_{jk})\}.$$

It turns out that this mathematical program can be cast as a linear program by designating a variable for the number of elements in each one of the 64 cells of the Venn diagram of A, B, C, D, E, F (or, more generally, the weight of each cell). The optimal value of this program is $7/12$, and one of the several optimal solutions for it is

$$\begin{array}{lll} A = \{x, y, z, w\} & B = \varnothing & C = \{x, y\} \\ D = \{x\} & E = \{x\} & F = \{z\} \end{array}$$

Using this solution, we get the function f described above.

4.2 Stronger upper bound

In this section we prove Theorem 2 with the stated bound of $207/361$. The proof is similar to the one given in Sect. 4.1, but the set system we use here is more involved.

We consider a partition matroid consisting of partitions P_1, P_2, P_3, P_4 over a ground set $\mathcal{N}$ of size 44. For every $\alpha \in \{1, 2, 3, 4\}$, we define the partition

$$P_\alpha = \{O_\alpha, X_\alpha, Y_{\beta\alpha}, Z_{\gamma\beta\alpha} : \beta, \gamma \in \{1, 2, 3, 4\} \text{ and } \alpha, \beta, \gamma \text{ are distinct}\}.$$

Like in Sect. 4.1, to define the submodular function f, we view each element of $\mathcal{N}$ as a subset of an underlying universe $\mathcal{U}$ consisting of $4 \cdot 26$ elements.

$$\mathcal{U} = \{a_\alpha, \dots, z_\alpha \mid \alpha \in \{1, 2, 3, 4\}\}.$$

Next, we specify the sets that the elements of $\mathcal{N}$ correspond to. In contrast to Sect. 4.1, in which the construction was completely symmetric, here the symmetry is more complicated. We partition $\{1, 2, 3, 4\}$ into two blocks, $\{1, 2\}$ and $\{3, 4\}$. The set system constructed below is invariant under switching the elements inside a block, and under switching the two blocks.

– For any $\alpha \in \{1, 2, 3, 4\}$,

$$O_\alpha = \{a_\alpha, \dots, z_\alpha\}.$$

– For any $\alpha \in \{1, 2, 3, 4\}$, let β be the other member of the same block, and let γ, δ be the remaining two indices, comprising the other block.

$$X_\alpha = \{m_\beta, \dots, q_\beta, r_\gamma, r_\delta, \dots, x_\gamma, x_\delta, y_\beta, y_\gamma, y_\delta, z_\beta, z_\gamma, z_\delta\}.$$

– For any $\alpha, \beta \in \{1, 2, 3, 4\}$ in the same block, let γ, δ be the remaining indices.

$$\begin{aligned} Y_{\alpha\beta} = \{&e_\beta, e_\gamma, e_\delta, g_\gamma, g_\delta, h_\beta, h_\gamma, h_\delta, j_\alpha, j_\gamma, j_\delta, k_\gamma, k_\delta, m_\gamma, m_\delta, n_\gamma, n_\delta, o_\gamma, o_\delta, \\ &p_\alpha, p_\gamma, p_\delta, q_\alpha, q_\gamma, q_\delta, u_\gamma, u_\delta, v_\beta, w_\alpha, x_\alpha, x_\beta\}. \end{aligned}$$

– For any $\alpha, \beta \in \{1, 2, 3, 4\}$ in different blocks, let γ be α's block companion and δ be β's block companion.

$$\begin{aligned} Y_{\alpha\beta} = \{&e_\delta, f_\gamma, g_\gamma, h_\gamma, i_\gamma, i_\delta, j_\gamma, j_\delta, k_\beta, k_\gamma, k_\delta, l_\alpha, l_\beta, l_\gamma, l_\delta, m_\delta, n_\delta, o_\delta, p_\delta, \\ &q_\beta, q_\delta, r_\gamma, s_\gamma, t_\gamma, u_\gamma, v_\gamma, w_\gamma, x_\gamma\}. \end{aligned}$$

– For any $\alpha, \beta \in \{1, 2, 3, 4\}$ in the same block and $\gamma \in \{1, 2, 3, 4\}$ in the other block, let δ be the remaining index.

$$Z_{\alpha\beta\gamma} = \{c_\delta, d_\delta, f_\delta, i_\delta, l_\delta, o_\alpha, o_\beta, s_\alpha, s_\beta, t_\alpha, t_\beta, u_\alpha, u_\beta, w_\beta\}.$$

– For any $\alpha, \beta \in \{1, 2, 3, 4\}$ in different blocks and $\gamma \in \{1, 2, 3, 4\}$ in the same block as α, let δ be the remaining index.

$$Z_{\alpha\beta\gamma} = \{b_\delta, d_\delta, g_\gamma, h_\delta, i_\alpha, m_\beta, n_\beta, n_\delta, o_\alpha, o_\beta, p_\beta, r_\alpha, s_\alpha, t_\alpha, u_\beta, x_\alpha, z_\alpha\}.$$

– For any $\alpha, \beta \in \{1, 2, 3, 4\}$ in different blocks and $\gamma \in \{1, 2, 3, 4\}$ in the same block as β, let δ be the remaining index.

$$Z_{\alpha\beta\gamma} = \{a_\delta, d_\delta, e_\delta, h_\gamma, i_\alpha, m_\alpha, m_\beta, n_\alpha, n_\beta, o_\alpha, o_\beta, p_\alpha, q_\alpha, r_\alpha, t_\alpha, x_\alpha, z_\alpha\}.$$

Given the above construction, we now define f to be the weighted coverage function, where the weight of the individual elements of $\mathcal{U}$ are given by the following table.[4]

[4] A weighted coverage function f is defined as $f(S) = \sum_{e \in (\bigcup_{A \in S} A)} w(e)$ for every $S \subseteq \mathcal{N}$. We remark that such functions are well-known to be monotone and submodular.

v	a_α	b_α	c_α	d_α	e_α	f_α	g_α	h_α	i_α	j_α	k_α	l_α	m_α
$w(v)$	37	40	29	13	6	27	6	3	2	5	15	6	2

v	n_α	o_α	p_α	q_α	r_α	s_α	t_α	u_α	v_α	w_α	x_α	y_α	z_α
$w(v)$	19	11	5	17	2	11	3	10	2	18	1	69	2

One can verify that the optimal solution for the instance obtained this way is $\{O_1, O_2, O_3, O_4\}$, and that it achieves a total weight of 1444. In contrast, we show below that, by choosing an appropriate tie-breaking rule, Algorithm 1 can be made to pick the sets $X_\alpha, Y_{\alpha\beta}, Z_{\alpha\beta\gamma}, O_\delta$ (in this order) when the parts are presented in the order $\langle P_\alpha, P_\beta, P_\gamma, P_\delta\rangle$. This solution has total weight 828, and thus, yields an approximation ratio of $828/1444 = 207/361$.

We now get to verifying that Algorithm 1 can be made to pick $X_\alpha, Y_{\alpha\beta}, Z_{\alpha\beta\gamma}, O_\delta$ given that the parts arrive in the order $\langle P_\alpha, P_\beta, P_\gamma, P_\delta\rangle$. To do that, we need to verify the following inequalities.

1. X_α can be chosen from P_α: $f(\{X_\alpha\}) \geq f(\{S\})$ for all $S \in P_\alpha$.
2. $Y_{\alpha\beta}$ can be chosen from P_β: $f(\{X_\alpha, Y_{\alpha\beta}\}) \geq f(\{X_\alpha, S\})$ for all $S \in P_\beta$.
3. $Z_{\alpha\beta\gamma}$ can be chosen from P_γ: $f(\{X_\alpha, Y_{\alpha\beta}, Z_{\alpha\beta\gamma}) \geq f(\{X_\alpha, Y_{\alpha\beta}, S\})$ for all $S \in P_\gamma$.
4. O_δ can be chosen from P_δ: $f(\{X_\alpha, Y_{\alpha\beta}, Z_{\alpha\beta\gamma}, O_\delta\}) \geq f(\{X_\alpha, Y_{\alpha\beta}, Z_{\alpha\beta\gamma}, S\})$ for all $S \in P_\delta$.

Each of these steps requires verifying 10 different inequalities. The verification is tedious, but routine. Due to symmetries, it suffices to check three different orders: $\langle P_1, P_2, P_3, P_4\rangle$; $\langle P_1, P_3, P_2, P_4\rangle$; $\langle P_1, P_3, P_4, P_2\rangle$.

Here are the values of all the relevant sets for the order $\langle P_1, P_2, P_3, P_4\rangle$ (the values marked in boldface are the highest values in their respective rows).

S	$\mathbf{O_1}$	$\mathbf{X_1}$	Y_{21}	Z_{321}	Z_{421}	Y_{31}	Z_{231}	Z_{431}	Y_{41}	Z_{241}	Z_{341}
$f(\{S\})$	**361**	**361**	256	155	155	243	160	165	243	160	165

S	$\mathbf{O_2}$	X_2	$\mathbf{Y_{12}}$	Z_{312}	Z_{412}	Y_{32}	Z_{132}	Z_{432}	Y_{42}	Z_{142}	Z_{342}
$f(\{X_1, S\})$	**597**	486	**597**	508	508	540	511	460	540	511	460

S	$\mathbf{O_3}$	X_3	Y_{13}	Z_{213}	Z_{413}	Y_{23}	$\mathbf{Z_{123}}$	Z_{423}	Y_{43}	Z_{143}	Z_{243}
$f(\{X_1, Y_{12}, S\})$	**751**	740	722	733	694	704	**751**	684	705	688	660

S	$\mathbf{O_4}$	$\mathbf{X_4}$	$\mathbf{Y_{14}}$	$\mathbf{Z_{214}}$	Z_{314}	$\mathbf{Y_{24}}$	$\mathbf{Z_{124}}$	$\mathbf{Z_{324}}$	$\mathbf{Y_{34}}$	$\mathbf{Z_{134}}$	Z_{234}
$f(\{X_1, Y_{12}, Z_{123}, S\})$	**828**	**828**	**828**	**828**	827	**828**	**828**	**828**	**828**	**828**	811

Here are the values of all relevant sets for the order $\langle P_1, P_3, P_2, P_4\rangle$.

S	$\mathbf{O_1}$	$\mathbf{X_1}$	Y_{21}	Z_{321}	Z_{421}	Y_{31}	Z_{231}	Z_{431}	Y_{41}	Z_{241}	Z_{341}
$f(\{S\})$	**361**	**361**	256	155	155	243	160	165	243	160	165

S	$\mathbf{O_3}$	X_3	$\mathbf{Y_{13}}$	Z_{213}	Z_{413}	$\mathbf{Y_{23}}$	Z_{123}	Z_{423}	Y_{43}	Z_{143}	Z_{243}
$f(\{X_1, S\})$	**604**	580	**604**	515	483	**604**	515	465	541	516	460

S	$\mathbf{O_2}$	$\mathbf{X_2}$	Y_{12}	Z_{312}	Z_{412}	Y_{32}	$\mathbf{Z_{132}}$	Z_{432}	$\mathbf{Y_{42}}$	Z_{142}	Z_{342}
$f(\{X_1, Y_{13}, S\})$	**729**	**729**	722	725	692	722	**729**	686	**729**	711	686

S	$\mathbf{O_4}$	$\mathbf{X_4}$	Y_{14}	$\mathbf{Z_{214}}$	Z_{314}	$\mathbf{Y_{24}}$	Z_{124}	Z_{324}	$\mathbf{Y_{34}}$	$\mathbf{Z_{134}}$	Z_{234}
$f(\{X_1, Y_{13}, Z_{132}, S\})$	**828**	**828**	742	**828**	826	**828**	810	812	**828**	**828**	785

Here are the values of all relevant sets for the order $\langle P_1, P_3, P_4, P_2\rangle$.

S	$\mathbf{O_1}$	$\mathbf{X_1}$	Y_{21}	Z_{321}	Z_{421}	Y_{31}	Z_{231}	Z_{431}	Y_{41}	Z_{241}	Z_{341}
$f(\{S\})$	**361**	**361**	256	155	155	243	160	165	243	160	165

S	$\mathbf{O_3}$	X_3	$\mathbf{Y_{13}}$	Z_{213}	Z_{413}	$\mathbf{Y_{23}}$	Z_{123}	Z_{423}	Y_{43}	Z_{143}	Z_{243}
$f(\{X_1, S\})$	**604**	580	**604**	515	483	**604**	515	465	541	516	460

S	$\mathbf{O_4}$	$\mathbf{X_4}$	Y_{14}	Z_{214}	Z_{314}	$\mathbf{Y_{24}}$	Z_{124}	Z_{324}	Y_{34}	$\mathbf{Z_{134}}$	Z_{234}
$f(\{X_1, Y_{13}, S\})$	**759**	**759**	654	728	723	**759**	710	698	722	**759**	695

S	$\mathbf{O_2}$	$\mathbf{X_2}$	Y_{12}	Z_{312}	Z_{412}	Y_{32}	$\mathbf{Z_{132}}$	$\mathbf{Z_{432}}$	$\mathbf{Y_{42}}$	Z_{142}	$\mathbf{Z_{342}}$
$f(\{X_1, Y_{13}, Z_{134}, S\})$	**828**	**828**	813	816	815	818	**828**	**828**	**828**	826	**828**

5 Conclusions

In this paper we have studied monotone submodular maximization subject to a partition matroid constraint in the online setting of the random arrival model. We have considered the natural greedy algorithm, and showed that its approximation ratio in this model is between 0.5096 and 0.574. This suggests several natural open questions.

1. Determine the exact approximation ratio of the greedy algorithm in the random order model.
2. Determine the best approximation ratio achievable in this model.
3. Determine whether better guarantees are possible in the special case of Submodular Welfare Maximization. This question is relevent both for the greedy algorithm (since our upper bound does not apply in this special case) and for general algorithms.

The greedy algorithm of the random order model becomes the Random Order Greedy algorithm in the *offline* setting. The last algorithm provides a better than 1/2-approximation for the offline problem of maximizing a monotone submodular function subject to a partition matroid constraint, which beats the performence of the standard greedy algorithm for this setting. While better offline approximation algorithms are known [8,17], Random Order Greedy is a combinatorial algorithm which is simple to implement and fast to run. Thus, we can view it as a step in a research direction aiming to design simple and efficient combinatorial algorithms for this problem which outperform the natural 1/2-approximation barrier. The only other paper, we are aware of, that suggests such an algorithm is the recent paper [5] discussed above. However, some work has been done in the past on fast (and involved) algorithms for this problem [3,6].

References

1. Aggarwal, G., Goel, G., Karande, C., Mehta, A.: Online vertex-weighted bipartite matching and single-bid budgeted allocations. In: SODA, pp. 1253–1264 (2011)
2. Alaei, S., Hajiaghayi, M., Liaghat, V.: Online prophet-inequality matching with applications to ad allocation. In: EC, pp. 18–35 (2012)
3. Badanidiyuru, A., Vondrák, J.: Fast algorithms for maximizing submodular functions. In: SODA, pp. 1497–1514 (2014)
4. Buchbinder, N., Feldman, M., Filmus, Y., Garg, M.: Online submodular maximization: beating 1/2 made simple. In: IPCO, pp. 101–114 (2019)
5. Buchbinder, N., Feldman, M., Garg, M.: Deterministic $(1/2+\varepsilon)$-approximation for submodular maximization over a matroid. In: SODA, pp. 241–254 (2019)
6. Buchbinder, N., Feldman, M., Schwartz, R.: Comparing apples and oranges: query trade-off in submodular maximization. Math. Oper. Res. **42**(2), 308–329 (2017)
7. Buchbinder, N., Jain, K., Naor, J.: Online primal–dual algorithms for maximizing ad-auctions revenue. In: ESA, pp. 253–264 (2007)
8. Călinescu, G., Chekuri, C., Pál, M., Vondrák, J.: Maximizing a monotone submodular function subject to a matroid constraint. SIAM J. Comput. **40**(6), 1740–1766 (2011)
9. Devanur, N.R., Hayes, T.P.: The adwords problem: online keyword matching with budgeted bidders under random permutations. In: EC, pp. 71–78 (2009)
10. Devanur, N.R., Jain, K., Sivan, B., Wilkens, C.A.: Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In: EC, pp. 29–38 (2011)
11. Devanur, N.R., Sivan, B., Azar, Y.: Asymptotically optimal algorithm for stochastic adwords. In: EC, pp. 388–404 (2012)
12. Feige, U., Mirrokni, V.S., Vondrák, J.: Maximizing non-monotone submodular functions. SIAM J. Comput. **40**(4), 1133–1153 (2011)
13. Feige, U., Vondrák, J.: The submodular welfare problem with demand queries. Theory Comput. **6**(1), 247–290 (2010)
14. Feldman, J., Korula, N., Mirrokni, V.S., Muthukrishnan, S., Pál, M.: Online ad assignment with free disposal. In: WINE, pp. 374–385 (2009)
15. Feldman, J., Mehta, A., Mirrokni, V.S., Muthukrishnan, S.: Online stochastic matching: beating 1-1/e. In: FOCS, pp. 117–126 (2009)
16. Feldman, M., Naor, J., Schwartz, R.: A unified continuous greedy algorithm for submodular maximization. In: FOCS, pp. 570–579 (2011)
17. Filmus, Y., Ward, J.: Monotone submodular maximization over a matroid via non-oblivious local search. SIAM J. Comput. **43**(2), 514–542 (2014)
18. Fisher, M.L., Nemhauser, G.L., Wolsey, L.A.: An analysis of approximations for maximizing submodular set functions—II. Math. Program. Study **8**, 73–87 (1978)
19. Goel, G., Mehta, A.: Online budgeted matching in random input models with applications to adwords. In: SODA, pp. 982–991 (2008)
20. Guruganesh, G.P., Singla, S.: Online matroid intersection: beating half for random arrival. In: IPCO, pp. 241–253 (2017)
21. Haeupler, B., Mirrokni, V.S., Zadimoghaddam, M.: Online stochastic weighted matching: improved approximation algorithms. In: WINE, pp. 170–181 (2011)
22. Kalyanasundaram, B., Pruhs, K.: An optimal deterministic algorithm for online b-matching. Theor. Comput. Sci. **233**(1–2), 319–325 (2000)
23. Kapralov, M., Post, I., Vondrák, J.: Online submodular welfare maximization: greedy is optimal. In: SODA, pp. 1216–1225 (2013)
24. Karande, C., Mehta, A., Tripathi, P.: Online bipartite matching with unknown distributions. In: STOC, pp. 587–596 (2011)
25. Karp, R.M., Vazirani, U.V., Vazirani, V.V.: An optimal algorithm for on-line bipartite matching. In: STOC, pp. 352–358 (1990)
26. Konrad, C., Magniez, F., Mathieu, C.: Maximum matching in semi-streaming with few passes. In: APPROX, pp. 231–242 (2012)
27. Korula, N., Mirrokni, V.S., Zadimoghaddam, M.: Online submodular welfare maximization: greedy beats 1/2 in random order. SIAM J. Comput. **47**(3), 1056–1086 (2018)

28. Mahdian, M., Yan, Q.: Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In: STOC, pp. 597–606 (2011)
29. Manshadi, V.H., Gharan, S.O., Saberi, A.: Online stochastic matching: online actions based on offline statistics. Math. Oper. Res. **37**(4), 559–573 (2012)
30. Mehta, A.: Online matching and ad allocation. Found. Trends Theor. Comput. Sci. **8**(4), 265–368 (2013)
31. Mehta, A., Saberi, A., Vazirani, U.V., Vazirani, V.V.: Adwords and generalized online matching. J. ACM **54**(5), 22 (2007)
32. Mirrokni, V.S., Schapira, M., Vondrák, J.: Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions. In: EC, pp. 70–77 (2008)
33. Mirzasoleiman, B., Badanidiyuru, A., Karbasi, A., Vondrák, J., Krause, A.: Lazier than lazy greedy. In: AAAI, pp. 1812–1818 (2015)
34. Norouzi-Fard, A., Tarnawski, J., Mitrovic, S., Zandieh, A., Mousavifar, A., Svensson, O.: Beyond 1/2-approximation for submodular maximization on massive data streams. In: ICML, pp. 3826–3835 (2018)
35. Zadimoghaddam, M.: Online weighted matching: beating the 1/2 barrier. CoRR (2017). arXiv:1704.05384

Affiliations

Niv Buchbinder[1] · Moran Feldman[2,4] · Yuval Filmus[3] · Mohit Garg[2,5]

✉ Moran Feldman
moranfe@openu.ac.il; moranfe@cs.haifa.ac.il

Niv Buchbinder
niv.buchbinder@gmail.com

Yuval Filmus
filmus.yuval@gmail.com

Mohit Garg
mohitga@openu.ac.il; garg@idsia.ch

1 Department of Statistics and Operations Research, Tel Aviv University, P.O. Box 39040, 6997801 Tel Aviv, Israel

2 Department of Mathematics and Computer Science, The Open University of Israel, 1 University Road, P.O. Box 808, 4353701 Raanana, Israel

3 Department of Computer Science, Technion — Israel Institute of Technology, Technion City, 3200003 Haifa, Israel

4 Present Address: Department of Computer Science, University of Haifa, 199 Aba Khoushy Ave., Haifa 3498838, Israel

5 Present Address: IDSIA, USI-SUPSI, Lugano, Switzerland