Combinatorial Auctions with Online XOS Bidders

Shaddin Dughmi
Department of Computer Science
University of Southern California
shaddin@usc.edu

Bryan Wilder
Department of Computer Science
University of Southern California
bwilder@usc.edu

March 13, 2017

Abstract

In combinatorial auctions, a designer must decide how to allocate a set of indivisible items amongst a set of bidders. Each bidder has a valuation function which gives the utility they obtain from any subset of the items. Our focus is specifically on welfare maximization, where the objective is to maximize the sum of valuations that the bidders place on the items that they were allocated (the valuation functions are assumed to be reported truthfully). We analyze an online problem in which the algorithm is not given the set of bidders in advance. Instead, the bidders are revealed sequentially in a uniformly random order, similarly to secretary problems. The algorithm must make an irrevocable decision about which items to allocate to the current bidder before the next one is revealed. When the valuation functions lie in the class XOS (which includes submodular functions), we provide a black box reduction from offline to online optimization. Specifically, given an \( \alpha \)-approximation algorithm for offline welfare maximization, we show how to create a \( (0.199 \alpha) \)-approximation algorithm for the online problem. Our algorithm draws on connections to secretary problems; in fact, we show that the online welfare maximization problem itself can be viewed as a particular kind of secretary problem with nonuniform arrival order.

1 Introduction

Combinatorial auctions are a central problem at the intersection of computer science and economics which provide an abstract representation of resource allocation. A combinatorial auction is characterized by a set of bidders \( S = \{b_1...b_n\} \) and a set of items \( I = \{1...m\} \). Each bidder has a valuation function \( f_i \) over subsets of the items. In the welfare maximization problem for combinatorial auctions, we aim to construct an allocation of items to bidders \( A : S \to 2^I \) which maximizes the sum of the bidders’ valuations for their assigned items, \( \sum_{i=1}^n f_i(A(b_i)) \). This sum is referred to as the social welfare.

Computing the optimal allocation is hard in general, and a great deal of work has focused on providing approximation algorithms when the valuation functions lie in some restricted class. A common example is that of submodular valuations, which encode a natural diminishing returns property. Here, the welfare maximization problem can be seen as maximizing a submodular function over a partition matroid and the continuous greedy algorithm of Vondrák [14] provides a \( 1 - 1/e \) approximation to the optimal social welfare. More general classes include fractionally subadditive, or XOS valuations, as well as general subadditive valuations. Feige [4] provides a \( (1 - 1/e) \)-approximation for XOS valuations and a \( 1/2 \)-approximation for subadditive valuations.

Welfare maximization admits a natural online setting, in which the full problem is not known in advance. Previous work has studied the problem of online items, where \( I \) is revealed one item at
The algorithm must make an irrevocable decision about which bidder to assign the current item to before the next item is revealed. The objective is to achieve welfare competitive with the offline optimum that sees all of the items before making its allocation.

However, little attention has been paid to the case where the items are known in advance and the bidders arrive online. Devanur et al. study a variant of this problem with reassignment, in which items may be freely taken from an earlier bidder and given to one who arrives later. However, reassignment is clearly impossible in many domains where allocations must be truly final at the time they are made.

We study welfare maximization for online bidders with \( \mathcal{XOS} \) valuations (which includes submodular valuations as a special case). Our focus is on the random arrival model, reminiscent of secretary problems. Here the valuations of the bidders may be chosen adversarially, but the arrival order is a uniformly random permutation. Our main result is:

**Theorem 1.** There is a black-box reduction which converts an \( \alpha \)-approximation algorithm for offline welfare maximization with \( \mathcal{XOS} \) bidders into a \((0.199\alpha)\)-approximation algorithm for online bidders who arrive in random order.

Our online algorithm only interacts with the valuation functions via the offline approximation oracle. In particular, it is indifferent to the manner in which the valuation functions themselves can be accessed (e.g. whether value oracle or demand oracle queries are available).

The main idea behind our algorithm is inspired by connections to secretary problems. The algorithm waits for a fraction \( \rho \) of the bidders to arrive without assigning any items. Once this “sample” period has passed, each arriving bidder is processed as follows. The algorithm runs the offline oracle on the entire set of bidders who have arrived so far. Then, the most recently arrived bidder is given any items allocated to them by the offline oracle which are still available (i.e., were not assigned to a different bidder in a previous time step). Our analysis is centered on the fact that each item has a constant probability of remaining unassigned across all steps; intuitively, this means that with constant probability, it will still be available when the “right” bidder arrives to claim it.

### 2 Preliminaries

#### 2.1 Classes of valuations

We study welfare maximization in combinatorial auctions where each bidder’s valuation function lies in the class \( \mathcal{XOS} \), also referred to as fractionally subadditive. We start out by introducing the classes of modular and submodular functions, which \( \mathcal{XOS} \) functions generalize.

**Definition 1.** A set function \( f \) is modular if it can be represented as \( f(X) = \sum_{x \in X} a(x) \) for some \( a : I \to \mathbb{R} \) which assigns a value to each item.

**Definition 2.** A set function \( f \) is submodular if for any \( X, Y \subseteq I \), \( f(X \cup Y) + f(X \cap Y) \leq f(X) + f(Y) \).

Submodular functions arise in many economic and game-theoretic contexts where agents have diminishing marginal utility in the set of items that they receive.

The class \( \mathcal{XOS} \) can be equivalently characterized in several ways. We use the following definition:

**Definition 3.** A set function \( f \) is \( \mathcal{XOS} \) if it can be be written as the maximum over a set of modular functions \( \{a_i\}_{i=1}^K \), i.e., \( f(X) = \max_i \sum_{x \in X} a_i(x) \).
Each of these classes of valuations generalizes the one before: every modular function is submodular, and every submodular function is XOS. Throughout, we assume that all valuation functions are monotone. That is, for any \( j \in I \) and \( X \subseteq I \), \( f(X \cup \{j\}) \geq f(X) \).

We assume that each bidder’s valuation function is revealed truthfully when the bidder arrives. Computing an optimal allocation can be highly nontrivial even taking truthfulness for granted; here we focus just on the informational complications of online arrivals.

### 2.2 Online arrivals

We study combinatorial auctions in which the bidders arrive online. Each time a bidder arrives, the algorithm must decide which items to allocate to that bidder before the next bidder is revealed. All allocations are irrevocable; in particular, the algorithm may not reassign an item from an earlier bidder to a later one. The algorithm knows \( n \), the total number of bidders, but does not receive any information about the bidders’ valuation functions in advance (save that the valuations are XOS).

Online problems typically distinguish between various models of arrival order. The most general model is adversarial order, in which both the set of bidders and the order in which they arrive are entirely arbitrary. It is easy to see that no nontrivial guarantees for our problem can be given in this case. Accordingly, we study the random order model. Here, the set of bidders is chosen by an adversary, but the arrival order is a uniformly random permutation. A stronger assumption on arrival order is the distributional model, where each bidder who arrives is sampled IID from some (known or unknown) distribution. This is a special case of the random order model, so all of our results apply when the bidders are drawn from a distribution.

Our objective is to design an algorithm which competes with the offline optimum; i.e., does almost as well as if the entire set of bidders were known in advance. Let \( \text{OPT} \) be the highest possible social welfare achievable for the offline problem. Let \( A \) be the allocation produced by the algorithm (which is a random variable that depends on the arrival order). Then, we measure performance by the approximation ratio

\[
\frac{\mathbb{E} \left[ \sum_{i=1}^n f_i(A(b_i)) \right]}{\text{OPT}}
\]

where the expectation is taken over both the randomness in the arrival order and any randomization performed by the algorithm. This compares the algorithm’s expected utility to the offline optimum.

### 3 Additional related work

The most directly related work to ours is Devanaur et al.\(^3\) who introduce the Whole Page Optimization problem, which as a special case includes a welfare maximization problem with online bidders. However, their problem is significantly different in that they consider the adversarial arrival model. To offset this difficulty, they allow the algorithm to take an item away from an earlier bidder and reassign it to a later one. We analyze the more restricted random arrival model but require all of the algorithm’s assignments to be irrevocable. Further, their algorithm requires access to a demand oracle for the valuation functions, while our algorithm only accesses the valuation functions through an offline optimization oracle (without any assumptions about how this oracle in turn queries the valuations).

Previous work has also considered a different online welfare maximization problem in which the items arrive online, while the set of bidders is known in advance. Here, Kapralov et al.\(^9\) show that
a simple greedy algorithm obtains an optimal $\frac{1}{2}$-approximation in the adversarial model. Korula et al. [11] show that greedy obtains a 0.505-approximation in the random order model.

Our work also has interesting connections to secretary problems. In the classic secretary problem, the algorithm examines a set of items which are revealed in an online manner. As in our work, the set of items is adversarially chosen but the arrival order is uniformly random. Each item has an associated value and the objective is to select an item with as high value as possible. The optimal algorithm samples the first $\frac{1}{e}$ fraction of the items without making any selection. Then, it takes the first item whose value is greater than any item in the sample. This paradigm of sampling a portion of the input to protect the algorithm from committing too early is common in secretary problems, and informs the design of our algorithm.

Generalizations of the secretary problem have attracted a great deal of interest. In the matroid secretary problem, introduced by Babaioff et al. [1], the algorithm can select any subset of the items which form an independent set of a given matroid, with the aim of selecting items which have the greatest total value (the objective function is modular). Currently, the best algorithms have approximation ratio $\Omega\left(\frac{1}{\log \log r}\right)$ where $r$ is the rank of the matroid [12, 6]. The matroid secretary conjecture states that there is an algorithm with approximation ratio $\Omega(1)$; this conjecture is still open. More recent work has introduced the submodular secretary problem, in which the objective may be any submodular function. Constant-factor approximation algorithms are known for specific classes of matroids, such as uniform [2, 8, 5], partition [8, 5], and laminar and transversal [13]. Further, Feldman and Zenklusen [7] recently gave a black box mechanism for converting a constant factor approximation algorithm for the matroid secretary problem (over any given matroid) to a constant factor approximation algorithm for the submodular matroid secretary problem (over that same matroid).

4 Connection between online bidders and secretary problems

Welfare maximization with online bidders can be seen as an instantiation of the submodular secretary problem over a partition matroid in which the items arrive in random but nonuniform order. From an instance of our problem, we can create a corresponding secretary problem as follows: for each bidder $b$ and item $j$, we create a single item $(b, j)$ for the secretary problem. Choosing this item in the secretary problem corresponds to assigning $j$ to $b$. The feasible sets are the independent sets of a partition matroid. This constraint enforces that each item may be assigned to at most one bidder. However, the items of the secretary instance do not arrive in uniformly random order. Instead, all of the items corresponding a given bidder arrive all at once; i.e., the algorithm must decide which items to assign to that bidder before seeing any of the other choices.

Kesselheim et al. [10] recently studied secretary problems in which the items arrive in nonuniform order. They introduce two models of nonuniform arrival distributions and show that simple algorithms achieve an approximation ratio which depends on “how close” the distribution is to uniform. Their analysis focuses on the classic secretary problem and they also give results for the case of a linear objective function subject to uniform matroid (i.e., selecting up to $k$ items). Our work can be viewed as studying a particular submodular partition secretary problem with nonuniform arrivals (though the arrival distribution does not correspond to either of the models studied by Kesselheim et al.). Thus, our results can be taken as evidence that a wider class of secretary problems may be solvable under nonuniform arrivals.
Algorithm 1

Require: A random permutation $\pi$, offline oracle $\mathcal{O}$, items $\mathcal{I}$, sampling portion $\rho$

1: \text{\textbackslash\textbackslash sampling phase}
2: for $i = 1...\rho n$ do
3: $A(b_{\pi(i)}) = \emptyset$
4: end for
5: \text{\textbackslash\textbackslash allocation phase}
6: for $i = \rho n + 1...n$ do
7: $A_{off} = \mathcal{O\{b_{\pi(1)}, b_{\pi(i)}\}}$ \textbackslash\textbackslash offline allocation for the first $i$ bidders
8: for $j \in A_{off}(b_{\pi(i)})$ do
9: \quad if $j$ is not already allocated in $A$ then
10: \quad \quad $A(b_{\pi(i)}) = A(b_{\pi(i)}) \cup \{j\}$
11: \quad end if
12: end for
13: end for

5 Algorithm and analysis

We now proceed to prove Theorem 1, our main result. We give a black box reduction from offline to online welfare maximization. Specifically, our algorithm takes as input an oracle $\mathcal{O}$ for the online problem. For any set of bidders $B \subseteq S$, $\mathcal{O}_B$ is an allocation of the items $\mathcal{I}$ to the bidders in $B$. As an offline algorithm, $\mathcal{O}$ does not receive any information about the order in which the bidders in $B$ arrived. Algorithm 1 gives pseudocode for our online algorithm. The algorithm has two phases. The sampling phase consists of the first $\rho n$ bidders, where $\rho \in [0, 1]$ is a parameter which controls the length of sampling phase. These bidders are not assigned any items. Then, the allocation phase assigns items to bidders $\rho n + 1...n$. Roughly, at each iteration $i$, we run the offline oracle $\mathcal{O}$ on the set of bidders who have arrived so far. Then, we try to match the allocation produced by $\mathcal{O}$ as much as possible on the $i$th bidder. That is, we give the most recently arrived bidder every item which $\mathcal{O}$ assigns to them as long as that item has not already been allocated. The idea is that the sampling phase (which is also a common feature of secretary algorithms) protects the algorithm from making premature commitments: any bidder who is given an item in the allocation phase must have “wanted it” more than at least $\rho n$ other bidders.

To formalize this intuition, our proof analyzes the expected utility obtained by a bidder chosen uniformly at random from $S$. An important step is to show that, by construction, the algorithm allocates each item with probability at most $\ln \frac{1}{\rho}$ across all steps. Thus, with probability at least $1 - \ln \frac{1}{\rho}$, each item that the offline oracle attempts to give to a bidder in the allocation phase will still be available.

We first present the following common lemma:

Lemma 1. Let $X \subseteq \mathcal{I}$ be a set of items and let $f$ be any $XOS$ function. Let $Y$ be a random set of items which includes each element of $X$ with probability at least $p$ (not necessarily independently). Then, $E[f(Y)] \geq pf(X)$.

Proof. Since $f$ is in the category $XOS$, it can be represented as the maximum over a set of modular functions $\{a_i\}$. Let $a$ be the modular function attaining the maximum at $X$, so that $f(X) =
\[ \sum_{x \in X} a(x) \]. We have that
\[
\mathbb{E}[f(Y)] = \mathbb{E} \left[ \max_i \sum_{x \in Y} a_i(x) \right]
\geq \mathbb{E} \left[ \sum_{x \in Y} a(x) \right]
= \sum_{x \in X} \Pr[x \in Y] a(x)
\geq \sum_{x \in X} pa(x)
= pf(X).
\]

Interestingly, Lemma 1 is the only place that our proof uses the assumption that the valuations are XOS. Now, we move on to prove our main result:

**Theorem 1** (Restated). If \( \mathcal{O} \) is an \( \alpha \)-approximate offline oracle, then Algorithm 1 has an approximation ratio of at least \( \alpha(1 - \rho)(1 - \ln \frac{1}{\rho}) \). With the optimal choice of \( \rho \), the approximation ratio is at least \( \alpha \left( 1 - \frac{1}{W(e^2)} \right) (1 - \ln W(e^2)) \geq 0.199 \alpha \), where \( W \) is the Lambert \( W \) function.

**Proof.** Let \( f_i \) be the valuation function of bidder \( b_i \). Let \( \mathcal{A} \) be the allocation which is returned by the algorithm, where \( \mathcal{A}(b_i) \) gives the set of items which are allocated to bidder \( i \). The bidders arrive in random order, i.e. according to a uniformly random permutation \( \pi \).

The total expected value of \( \mathcal{A} \) is
\[
\mathbb{E}_\pi \left[ \sum_{i=\rho n+1}^{n} f_{\pi(i)}(\mathcal{A}(b_{\pi(i)})) \right] = \sum_{i=\rho n+1}^{n} \mathbb{E}_\pi \left[ f_{\pi(i)}(\mathcal{A}(b_{\pi(i)})) \right]. \tag{1}
\]

We start by examining the expected utility of a fixed bidder \( b_i \). Let \( \mathcal{O} \) be the \( \alpha \)-approximate offline allocation oracle which the algorithm runs at each step and let \( \mathcal{O}_B \) be the allocation which the oracle provides for the set of bidders \( B \).

**Lemma 2.** For any bidder \( b_i \), \( \mathbb{E}[f_i(\mathcal{A}(b_i))] \geq \left( 1 - \ln \frac{1}{\rho} \right) \mathbb{E}[f_i(\mathcal{O}_B(b_i))] \) where \( B \) is the random set consisting of \( b_i \) and all of the bidders who arrived before \( b_i \).

**Proof.** We couple the random variables \( f_i(\mathcal{A}(b_i)) \) and \( f_i(\mathcal{O}_B(b_i)) \) by fixing the set \( B \) (i.e., fixing the set of bidders who arrived before \( b_i \)), but not the order in which prior bidders arrived (which is still uniformly random). \( \mathcal{O}_B(b_i) \) is now a fixed set, since the offline oracle does not depend on order. We will show that each element of \( \mathcal{O}_B(b_i) \) is still available with probability at least \( 1 - \ln \left( \frac{1}{\rho} \right) \).

Fix an item \( j \in \mathcal{O}_B(b_i) \). Note that at each step \( k = \rho n + 1 \ldots |B| - 1 \), \( \mathcal{O} \) assigns \( j \) to one of the first \( k \) bidders who arrive, and \( j \) is actually allocated if the assigned bidder is the last one who arrives. Since the arrival order is uniformly random, \( j \) is allocated in each step with probability \( \frac{1}{k} \). Using union bound, the probability that \( j \) is allocated in steps \( \rho n + 1 \ldots |B| - 1 \) is at most

\[ \frac{1}{k} \leq \frac{1}{\rho n + 1} \]
\[
\sum_{k=\rho n+1}^{n} \frac{1}{k} \leq \sum_{k=\rho n+1}^{n} \frac{1}{k} \leq \int_{\rho n}^{n} \frac{1}{x} \, dx = \ln \left( \frac{1}{\rho} \right)
\]

Using Lemma 1, we conclude that \( E[f_i(A(b_i))] \geq \left( 1 - \ln \left( \frac{1}{\rho} \right) \right) E[f_i(O_B(b_i))] \). \( \square \)

Now we compare the value obtained in each step to the total value of the optimal solution. Define \( A^*_B \) to be the optimal allocation of the entire set of items to the set of bidders \( B \), and \( OPT(B) \) to be the value of this allocation (e.g. \( OPT = OPT(S) \) in this notation).

**Lemma 3.** Let \( B \) be a uniformly random set of \( |B| \) bidders and choose \( b_i \in B \) uniformly at random. Then \( E[f_i(O_B(b_i))] \geq \frac{\alpha}{n} OPT \).

**Proof.** Since \( b_i \) is a uniformly random element of \( B \), we have

\[
E[f_i(O_B(b_i))] = \frac{1}{|B|} E \left[ \sum_{k \in B} f_k(O_B(b_k)) \right]
\geq \frac{\alpha}{|B|} E[OPT(B)].
\]

Note that \( A^*_S \) must be at least as good as giving the bidders in \( B \) the items they receive in \( A^*_S \). Hence, for a uniformly random set \( B \),

\[
E[OPT(B)] \geq E \left[ \sum_{k \in B} f_k(A^*_S(b_k)) \right]
\geq \frac{|B|}{n} OPT
\]

Putting this together we have

\[
E[f_i(O_B(b_i))] \geq \frac{\alpha}{|B|} E[OPT(B)]
\geq \frac{\alpha}{n} OPT.
\]

\( \square \)

We can apply Lemma 3 to each term in the summation in Equation (1) since \( \{b_{\pi(1)} \ldots b_{\pi(i)}\} \) is a uniformly random subset of \( S \), and \( b_{\pi(i)} \) is a uniformly random element of this set. Hence
\[
\mathbb{E}_\pi \left[ \sum_{i=\rho n+1}^{n} f_{\pi(i)}(A(b_{\pi(i)})) \right] = \sum_{i=\rho n+1}^{n} \mathbb{E}_\pi [f_{\pi(i)}(A(b_{\pi(i)}))] \\
\geq \left(1 - \ln \left(\frac{1}{\rho}\right)\right) \sum_{i=\rho n+1}^{n} \mathbb{E}_\pi \left[ f_{\pi(i)}(O_{\{b_{\pi(1)} \ldots b_{\pi(i)}\}}(b_{\pi(i)})) \right] \quad \text{(Lemma 2)} \\
\geq \left(1 - \ln \left(\frac{1}{\rho}\right)\right) \sum_{i=\rho n+1}^{n} \frac{\alpha}{n} OPT \quad \text{(Lemma 3)} \\
\geq \alpha (1 - \rho) \left(1 - \ln \left(\frac{1}{\rho}\right)\right) OPT.
\]

The only remaining detail is to optimize the choice of \( \rho \). The expression \( (1 - \rho) \left(1 - \ln \left(\frac{1}{\rho}\right)\right) \) is maximized at \( \rho = \frac{1}{W(e^2)} \approx 0.642 \), where \( W \) is the Lambert \( W \) function. Taking this value for \( \rho \) yields the desired guarantee.

\[\square\]

6 Implications for efficient algorithms

Our focus has been on the informational challenges raised when the bidders arrive online; Algorithm 1 is entirely agnostic to how the underlying combinatorial optimization problem is solved or whether the offline oracle is computationally efficient. However, we note the following approximation ratios can be obtained by computationally efficient (i.e., polynomial time) algorithms using known offline algorithms as the oracle for Algorithm 1. These results depend on the manner in which the valuation functions may be accessed. Two models are commonly used in the literature. In the value oracle model, the algorithm may query (as a single computational step) \( f(X) \) for any set \( X \). In the demand oracle model, the algorithm may ask, for any vector of prices \( p \) on the items, for a set \( X \) which maximizes \( f(X) - \sum_{j \in X} p_j \). This has a natural economic interpretation as the bundle that the bidder would prefer to consume given a set of prices for the items.

**Theorem 2.** There exist efficient algorithms which obtain the following approximation ratios:

- For submodular valuations: \( 0.199(1 - \frac{1}{e}) \), with access to value oracle queries, using the algorithm of Vondrák [17].

- For general XOS valuations: \( 0.199(1 - \frac{1}{e}) \), with access to demand oracle queries, using the algorithm of Feige [4].

References

[1] Moshe Babaioff, Nicole Immorlica, and Robert Kleinberg. Matroids, secretary problems, and online mechanisms. In *Proceedings of the eighteenth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 434–443. Society for Industrial and Applied Mathematics, 2007.

[2] MohammadHossein Bateni, Mohammadtaghi Hajiaghayi, and Morteza Zadimoghaddam. Submodular secretary problem and extensions. *ACM Transactions on Algorithms (TALG)*, 9(4):32, 2013.
[3] Nikhil R Devanur, Zhiyi Huang, Nitish Korula, Vahab S Mirrokni, and Qiqi Yan. Whole-page optimization and submodular welfare maximization with online bidders. *ACM Transactions on Economics and Computation*, 4(3):14, 2016.

[4] Uriel Feige. On maximizing welfare when utility functions are subadditive. *SIAM Journal on Computing*, 39(1):122–142, 2009.

[5] Moran Feldman, Joseph Seffi Naor, and Roy Schwartz. Improved competitive ratios for submodular secretary problems. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 218–229. Springer, 2011.

[6] Moran Feldman, Ola Svensson, and Rico Zenklusen. A simple $O(\log \log (\text{rank}))$-competitive algorithm for the matroid secretory problem. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1189–1201. Society for Industrial and Applied Mathematics, 2015.

[7] Moran Feldman and Rico Zenklusen. The submodular secretory problem goes linear. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 486–505. IEEE, 2015.

[8] Anupam Gupta, Aaron Roth, Grant Schoenebeck, and Kunal Talwar. Constrained non-monotone submodular maximization: Offline and secretory algorithms. In *International Workshop on Internet and Network Economics*, pages 246–257. Springer, 2010.

[9] Michael Kapralov, Ian Post, and Jan Vondrák. Online submodular welfare maximization: Greedy is optimal. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1216–1225. SIAM, 2013.

[10] Thomas Kesselheim, Robert Kleinberg, and Rad Niazadeh. Secretary problems with non-uniform arrival order. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 879–888. ACM, 2015.

[11] Nitish Korula, Vahab Mirrokni, and Morteza Zadimoghaddam. Online submodular welfare maximization: Greedy beats 1/2 in random order. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 889–898. ACM, 2015.

[12] Oded Lachish. $O(\log \log \text{rank})$ competitive ratio for the matroid secretory problem. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 326–335. IEEE, 2014.

[13] Tengyu Ma, Bo Tang, and Yajun Wang. The simulated greedy algorithm for several submodular matroid secretory problems. *Theory of Computing Systems*, 58(4):681–706, 2016.

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