Selling Ad Campaigns: Online Algorithms with Cancellations

Moshe Babaioff
Microsoft Research
Mountain View, CA 94043
moshe@microsoft.com

Jason D. Hartline
Electrical Engineering and Computer Science
Northwestern University
Evanston, IL 60208
hartline@northwestern.edu

Robert D. Kleinberg*
Computer Science
Cornell University
Ithaca, NY 14853
rdk@cs.cornell.edu

ABSTRACT

We study online pricing problems in markets with cancellations, i.e., markets in which prior allocation decisions can be revoked, but at a cost. In our model, a seller receives requests online and chooses which requests to accept, subject to constraints on the subsets of requests which may be accepted simultaneously. A request, once accepted, can be canceled at a cost which is a fixed fraction of the request value. This scenario models a market for web advertising campaigns, in which the buyback cost represents the cost of canceling an existing contract.

We analyze a simple constant-competitive algorithm for a single-item auction in this model, and we prove that its competitive ratio is optimal among deterministic algorithms, but that the competitive ratio can be improved using a randomized algorithm. We then model ad campaigns using knapsack domains, in which the requests differ in size as well as in value. We give a deterministic online algorithm that achieves a bi-criterion approximation in which both approximation factors approach 1 as the buyback factor and the size of the maximum request approach 0. We show that the bi-criterion approximation is unavoidable for deterministic algorithms, but that a randomized algorithm is capable of achieving a constant competitive ratio. Finally, we discuss an extension of our randomized algorithm to matroid domains (in which the sets of simultaneously satisfiable requests constitute the independent sets of a matroid) as well as present results for domains in which the buyback factor of different requests varies.

Categories and Subject Descriptors

F.1.2 [**Modes of Computation**]: Online Computation; F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity; J.4 [**Social and Behavioral Sciences**]: Economics; K.4.4 [**Computers and Society**]: Electronic commerce

General Terms

Algorithms, Economics, Theory

Keywords

Selling Advertisements, Online Algorithms, Costly Decision Revocation, Knapsack, Matroids

*Supported by NSF grants CCF-0643934 and CCF-0729102, an Alfred P. Sloan Foundation Fellowship, and a Microsoft Research New Faculty Fellowship.

1. INTRODUCTION

The problem we study is motivated by the following real scenario in the Display Advertising industry. The large banner ads shown on a popular website such as MSN are sold by negotiated contract up to a year in advance. The impression inventory is diverse, the advertisers are looking to meet campaign limits in terms of demographics targets (who an ad is shown to), temporal targets (when an ad is shown), and location targets (where an ad is shown). Any particular advertiser request may be able to be met in a variety of ways. The ad seller has a complex online optimization task. As advertiser requests arrive the ad seller must decide whether the requests should be accepted or not. When formulated most naturally as an online optimization problem, two aspects make this a nearly impossible task. It is a small market; display advertisers on a major website number in the thousands but big advertisers number in the tens. It is an online market; inventory may be preallocated to advertisers at a low value early. It may not ever be learned that advertisers arriving later would have paid a premium for the same inventory. Absent this knowledge, there is little basis for accepting or declining ad orders placed by advertisers. This reality of the industry is mirrored by online algorithm theory which suggests that online algorithms for limited supply allocation problems suffer from the worst possible competitive ratios — linear in the number of requests.

This paper addresses both the theoretical problem in online algorithms analysis for allocation problems and the practical problem in settings such as Display Advertising, above, and gives algorithms with competitive ratios that are a constant, parameterized by a coefficient representing a required level of commitment. Notice that the scenarios that bring bad competitive ratios are ones where the algorithm commits to serve a request with low value, say $1, and then an incompatible request arrives with high value, say $1M.

A committed algorithm is stuck. Practically speaking, we cannot think of one real scenario where it is impossible to bump out the $1 request (perhaps at some cost) to take the $1M request, unless the $1 request has already been served.[1] Thus, we are motivated to explicitly model the revocation of a commitment into the model of online allocation and design algorithms that make use of it. Our resulting work pursues a theory of online algorithms with buyback. While our results are theoretical, their motivation is directly relevant to electronic commerce. In fact, the problem studied here was described to us by the ad marketing group at a major Internet software company as we have presented it.

Of course this kind of contract revocation is not new and in many real settings a possibility of revocation is explicitly written into agreed upon contracts. Notable instances of this are the airline industry that routinely over-sells economy seats when premium last minute bookings are made, bumps travelers on over-booked flights, and cancels under-booked flights. Even in our motivating example of Display Advertising, it is possible that the actual impression inventory does not meet forecasted estimates. In this case the actual inventory may be over-sold. There are clauses in the advertisers' contracts that specify that the seller will "make good" by substituting alternative, comparable inventory. Taking these kinds of contracts as given, our paper studies the effect of structural constraints, similar to those that arise in the Display Advertising setting, on optimal online admission and revocation policies.

We consider the following model for studying algorithms for online admission with costly revocation. An admitted request of value v can be revoked at any time at a cost proportional to v. Let f represent this *buyback factor*. Then the cost of revoking a v-valued request is $f \cdot v$. Our algorithm now, faced with a sequence of requests arriving online, must make admission and revocation decisions satisfying the conditions that (a) at any time the set of admitted requests is feasible, (b) a request may only be admitted at the time it first arrives, and (c) a request may be revoked (a.k.a., bought back) at a penalty of the buyback factor at any time, but once revoked it may not subsequently be readmitted. After the last request arrives and is processed, the set of admitted requests that have not been revoked are served. The total payoff of the algorithm is the cumulative value of these served requests less the cumulative buyback cost, i.e., the cumulative value of the revoked requests scaled by the buyback factor.

Prior work on buyback algorithms.

As far as we are aware, the first published work on online algorithms with buyback costs consisted of two independent, concurrently written papers that appeared in the 2008 Workshop on Ad Auctions [3, 10]. One of those papers subsequently appeared in SODA 2009 [11], while the other has evolved into the present work, which contains some of the results from the original workshop paper along with several new results. The starting point for all of these papers is the observation that there exist constant-competitive deterministic algorithms for selling a single item in the buyback model. They identify a one-parameter family of such algorithms, and they find the optimum competitive ratio within this one-parameter family. These papers [3, 10, 11] also proceed to analyze a generalization of this deterministic single-item algorithm to *matroid domains*, in which the sets of requests that can be simultaneously satisfied form the independent sets of a matroid.

The SODA paper [11] and its precursor [10] additionally consider incentive issues relating to the design of buyback algorithms. When one regards these algorithms as mechanisms for selfish bidders, the mechanisms are not dominant-strategy truthful, because bidders who are not going to win any items have an incentive to act as *speculators*, declaring a bid greater than their true value to maximize their buyback payment. However, because all the algorithms for single-item and matroid domains in [3, 10, 11] — as well as in the present paper — can be implemented as posted-price mechanisms, any strategy that bids below the agent's true value is dominated by the strategy of truthful bidding. We can therefore conclude that agents who play undominated strategies will bid at least their true value, which leads to provable revenue guarantees for these algorithms even in the presence of speculators. (The revenue guarantee bounds the ratio between the mechanism's revenue and the VCG revenue; see [11] for details.) It is more difficult to provide welfare guarantees because the action of speculators may lead to a very inefficient allocation; however, it is shown in [11] that the mechanism approximates the optimal social welfare in any execution such that the speculators have non-negative total utility.

As mentioned above, the algorithms developed for single-parameter domains in this paper (such as the single-item and matroid domains) can be implemented as posted-price mechanisms and therefore inherit the good incentive properties of the mechanisms in [11]: agents who are playing undominated strategies will not bid below their true value. For this reason, in subsequent sections of this paper we will not explicitly address incentive issues and will instead focus on the design and analysis of competitive online buyback algorithms.

Our contributions.

One type of structural constraint that arises in Display Advertising is a knapsack-type constraint. Here advertisers' ad campaigns may require different quantities of inventory. An advertiser may want its request to be all fulfilled or all denied, as either they run their full campaign or no campaign. We make the simplifying assumption that there is a single commodity for sale, e.g., a fixed quantity of impressions for a single type of banner ad slots (web traffic that is not demographically or temporally segregated.) The quantities and values of the requests form an instance of the online knapsack problem. We give a deterministic online buyback algorithm that achieves a bi-criterion approximation result. With an assumption on the size of the largest demand relative to the total supply, we can turn this bi-criterion result into a full constant approximation (for constant f) that is not much worse than the best possible by any deterministic online algorithm with buyback. With no assumption, standard randomization approaches for knapsack can be used to convert our approximation algorithm to one that has a constant approximation factor (for constant f) that is bounded away from 1; however all online algorithms, even randomized ones, have a competitive ratio that is bounded away from 1 in this case.

[1]The situations modeled in this paper are those in which a seller makes a sequence of commitments to provide service at some later time after the end of the request sequence. All commitments are thus revocable in our model.

Another type of structural constraint we may have in the Display Advertising setting is a matching constraint. Advertisers' requests could potentially be satisfied from different impression inventory. Supply constraints imply a matching structure on the set of served requests. We make the simplifying assumption that each request is for a unit of supply which implies that the sets of compatible requests are the *independent sets* of a *transversal matroid*. We present several new results on this problem, including the special case of designing online buyback algorithms for selling a single indivisible item. We give a lower bound of $1+2f+2\sqrt{f(1+f)}$ for the competitive ratio of deterministic single-item buyback algorithms, which matches the competitive ratio achieved by the algorithms described in [3, 10, 11] and in Section 2.1.[2] We then consider randomized algorithms for the single-item buyback problem, and we prove that they can achieve a superior competitive ratio at least when $0 < f < \frac{1}{2}$, which is the range of values appropriate to the Display Advertising application. Our algorithm's competitive ratio approaches 1 as the buyback factor f tends to 0. Our results on randomized algorithms also extend to the matroid case, with no deterioration in the competitive ratio bound. Finally, we consider the question of what happens when advertisers can request a buyback factor other than f. We provide an algorithm for the single-item case whose competitive ratio is $1 + 2f + 2\sqrt{f(1+f)}$ against the maximum bid in a subset of the requests that contains all those whose buyback factor is at most f, but possibly others as well.

Because our algorithms are based on simple threshold rules, we believe they strongly resemble the decision procedures that would realistically be used by practitioners when deciding which bids to accept in a market with buyback. The theoretical analysis of these algorithms thus sheds light on important questions about a realistic class of decision procedures. For instance, should the "threshold value" increase additively or multiplicatively? (Answer: multiplicatively) Should the rate of increase of the threshold be roughly proportional to the buyback factor? (Answer: no, for small values of f it should be proportional to the square root.) Can we prove useful revenue guarantees for threshold rules, or are the theoretical worst-case bounds too large to be of practical importance? (Answer: The worst-case bounds are quite mild. Our competitive ratio tends to 1 as f and relative request size tend to zero.)

Related Work.. Many authors have studied Internet advertising in the context of online algorithms or in the context of knapsack problems. Most of this work deals with sponsored search auctions whereas our work is motivated primarily by problems that arise in selling banner advertisements. There are clear relations between the underlying algorithmic issues in both applications, but also some interesting differences: in sponsored search, the inventory (i.e., queries) arrives online, whereas in banner advertising the demand (i.e., requests from advertisers) arrives online. With the exception of the papers [3, 10, 11] discussed above, none of this prior work models markets with online requests and buyback costs ([14] studied similar problems in the *offline* setting.)

Advertising auctions and bid optimization problems have been modeled as knapsack problems (both online and offline) in work by Aggarwal and Hartline [1], Borgs et al. [7] (which treats the problem of slot selection), and Rusmevichientong and Williamson [24] (which treats keyword selection). Chakrabarty et al. [9] modeled the bidding problem using online knapsack and gave an online algorithm whose competitive ratio is logarithmic in the ratio of maximum to minimum value density. The same problem was studied, under a random-ordering assumption, by Babaioff et al. [5]. Ad auctions with non-strategic budget-constrained bidders have also been modeled as an online b-matching problem, the so-called AdWords problem. Randomized algorithms achieving the optimal competitive ratio of $e/(e-1)$ for this problem were given by Mehta et al. [21] and by Buchbinder et al. [8].

In this work we introduce a buyback option in order to overcome the $\Omega(n)$ lower bound for (randomized) online algorithms facing adversarial input. Another approach for overcoming this lower bound is to assume that the requests come in a *random order*. The classical secretary problem [13] studies how to select online an element with maximum value in a randomly ordered sequence, succeeding with probability $1/e$. Extensions to various problems that involve accepting a set of requests — including knapsack and matroid domains — have been studied in [2, 5, 4, 12, 17].

The problem of online knapsack (without removal) was introduced and studied by [18, 19]. Our results for the online buyback knapsack problem relate to papers that study online knapsack problems with the possibility of removing elements that were already accepted to the knapsack. All prior work assumes that such removal has no cost ($f = 0$ in our formulation). [15] presents tight competitive algorithms for online *unweighted* knapsack. Bi-criteria results for the case in which elements can be *fractionally* accepted are presented in [22], while bi-criteria results for the case of integral acceptance are presented in [16].

The idea of a buyback is related to the idea of "opportunistic cancellations" which was presented in [6]. That paper shows that if ones allow cancellation of some deals with a fixed penalty this results in higher efficiency. The paper makes some distributional assumptions, while we work in the framework of worst case competitive analysis. [25] proposes a leveled commitment contracting protocol that allows self-interested agents to efficiently accommodate future events by having the possibility of unilaterally decommitting from a contract by paying some penalty.

Organization. The paper is organized as follows. In Section 2 we present the deterministic algorithm for a single item, as a warmup. Section 3 presents our formal general model for considering constrained online optimization with buyback. In Section 4 we consider knapsack set systems and present results for deterministic and randomized algorithms. Section 5 presents our new results for the single item case: a matching lower bound for deterministic algorithms, a randomized algorithm that beats that lower bound, and finally, an algorithm for heterogenous buyback factors.

2. WARMUP: THE SINGLE ITEM DETERMINISTIC ALGORITHM

Consider n requests from customers coming online in arbitrary order. There is a fixed buyback factor $f > 0$. Customer e pays v_e if she receives the item, and if we buy back

[2] In [11] there is a lower bound defined implicitly by a recurrence, and the authors conjecture that this lower bound matches the upper bound achieved by their algorithm. Our lower bound theorem confirms this conjecture.

the item from her later, we must refund this payment and pay an additional penalty of $f \cdot v_e$ in compensation. We next present a $\left(1 + 2f + 2\sqrt{f(1+f)}\right)$-competitive deterministic algorithm. The algorithm first appeared in [3, 10, 11] and we repeat its analysis here in order to introduce ideas and techniques that will be used throughout later sections.

2.1 The Deterministic Algorithm

For the single item case the buyback algorithm is simple. The algorithm assumes we are given a parameter $r > 1$ whose value may depend on the buyback factor f. We will later see that the optimal competitive ratio is achieved when $r = 1 + f + \sqrt{f(1+f)}$. The algorithm operates as follows.

- Accept the first request.
- Consider the rest of the requests in order of arrival. If the value of the arriving request is more than r times the value of currently accepted request, buy back the item and accept the arriving request.

Observe that setting $r = 1 + f$ in the above algorithm corresponds to the very natural idea of accepting a new bid whenever it is profitable to do so. However, when $r = 1 + f$ the algorithm has an unbounded competitive ratio. This is easily seen, for example, by considering a bid sequence defined recursively by $v_1 = 1$, $v_{i+1} = (1 + f)v_i + \varepsilon$ for some arbitrarily small $\varepsilon > 0$. For every $n > 0$, the algorithm's profit after $n + 1$ bids is only $1 + \varepsilon n$, whereas the optimum profit is greater than $(1 + f)^n$.

THEOREM 1 ([3, 10, 11]). *The single item buyback algorithm has a competitive ratio of $\frac{r(r-1)}{r-1-f}$. For a fixed buyback factor $f > 0$, this function is minimized by setting $r = 1 + f + \sqrt{f(1+f)}$, resulting in the competitive ratio $1 + 2f + 2\sqrt{f(1+f)}$.*

PROOF. We first prove that the competitive ratio is $\frac{r(r-1)}{r-1-f}$. Let m be the request with highest value, i.e., $OPT = v_m$. Let w be the final request selected. We claim $v_w \geq v_m/r$. If $w = m$ then this is trivially true, as $r > 1$. If on the other hand $w \neq m$, then at the time m was considered and rejected, the item was allocated to request u satisfying $v_m \leq r \cdot v_u$. As the winner's value monotonically increases, at the end it holds that $v_m \leq r \cdot v_w$ and the claim follows.

We next bound the total buyback payment, B. Denote the requests that are selected but subsequently bought back by $\ell(1), \ell(2), \ldots, \ell(j)$, numbered in *decreasing* order of arrival time. For notational convenience we will also define $\ell(0) = w$. For $i = 1, 2, \ldots, j$, the fact that the algorithm chose to buy back the item from $\ell(i)$ at the arrival time of $\ell(i-1)$ implies that $v_{\ell(i-1)} > r \cdot v_{\ell(i)}$. By induction on i we now obtain $v_{\ell(i)} < r^{-i} \cdot v_w$. Now,

$$B = f \cdot \sum_{i=1}^{j} v_{\ell(i)} \leq f v_w \cdot \sum_{i=1}^{j} r^{-i} < f v_w \cdot \frac{1}{r-1}.$$

Thus, the algorithm's profit $v_w - B$ satisfies

$$v_w - B > v_w \left(1 - \frac{f}{r-1}\right) \geq \frac{1}{r}\left(1 - \frac{f}{r-1}\right) v_m,$$

and the competitive ratio $\frac{r(r-1)}{r-1-f}$ follows by rearranging terms. It is now an exercise in calculus to verify that the optimal value of r and the resulting competitive ratio are as stated in the theorem. □

3. THE GENERAL MODEL

We consider a set $N = \{1, 2, ..., n\}$ of requests coming online, where request k from some customer arrives at time step k. Each request $k \in N$ has a value $v_k \geq 0$ associated with it. We assume that there is a set system $\mathcal{S}$ (a family of subsets of N) which represents the *feasible sets* of requests. For example, in the *knapsack domain* the elements of N are requests for non-negative quantities $(s_i)_{i=1}^n$ of a divisible good, and there is a finite supply C. A subset $S \subseteq N$ is feasible (i.e., belongs to $\mathcal{S}$) if and only if $\sum_{i \in S} s_i \leq C$. In a *matroid domain*, the elements of $\mathcal{S}$ constitute the independent sets in a matroid structure[3] with ground set N. Prototypical examples of matroid domains are the single-item domain (in which the only feasible sets are singletons), the k-item domain (in which feasible sets are those whose cardinality is at most k) and transversal matroid domains (in which there is a set G of goods for sale, request i desires one element from a subset $G_i \subseteq G$, and a set of requests $S \subseteq N$ is feasible if and only if there exists a way to assign a distinct element $g_i \in G_i$ to each request $i \in S$). The value of a feasible set S is denoted by $v(S) = \sum_{i \in S} v_i$, and we will denote the maximum value of a feasible set by OPT.

The algorithm A is responsible for maintaining a sequence of feasible sets, one at each time step. The feasible set at time step k is denoted by $S^{(k)}$. The only request that may be accepted at time step k is request k, i.e. $S^{(k)} \subseteq S^{(k-1)} \cup \{k\}$. If $k \in S^{(k)} \setminus S^{(n)}$ then we say that request k is *revoked*, *canceled*, or *bought back*. Note that a request, once canceled, can never be accepted again. Also note that our assumption that $S^{(k)}$ is feasible at every time step k (and not only at the final time step) is without loss of generality for matroids: if the algorithm holds an infeasible set at any time k, it is never optimal to retain elements that do not belong to the maximum-weight basis of this infeasible set. Thus we can assume that whenever the set of accepted requests becomes infeasible, the algorithm immediately cancels requests not in the maximum-weight basis to restore feasibility in the same time step. In contrast, our assumption that $S^{(k)}$ must always be feasible is *not* without loss of generality for knapsack domains.

The algorithm gets a net payoff of v_i for each element i in its final feasible set $F = S^{(n)}$, and it loses $f \cdot v_i$ for each canceled request, where $f > 0$ is a fixed *buyback factor* that is part of the problem specification. Denoting the set of canceled requests by $R = \left(\cup_{i=1}^n S^{(i)}\right) \setminus S^{(n)}$, the payoff is thus defined to be $\text{Payoff}(A) = v(F) - f \cdot v(R)$. We will sometimes denote the total buyback cost by $B = f \cdot v(R)$.

We require A to be an *online* algorithm, in the sense that it must choose the set $S^{(k)}$ based only on the following information: the values v_i for $1 \leq i \leq k$, the feasibility of any subsets $T \subseteq \{1, 2, \ldots, k\}$ (we assume the algorithm may query an oracle about the feasibility of any $T \subseteq \{1, 2, \ldots, k\}$ at time step k), and the algorithm's own random bits if it is a randomized algorithm. Our goal is to design algorithms that have good competitive ratio with respect to the optimal offline solution. The competitive ratio of algorithm A is thus

[3] A *matroid* $(\mathcal{U}, \mathcal{I})$ is constructed from a ground set $\mathcal{U} \neq \emptyset$ and a nonempty family $\mathcal{I}$ of subsets of $\mathcal{U}$, called the *independent* subsets of $\mathcal{U}$, such that if $B \in \mathcal{I}$ and $A \subseteq B$ then $A \in \mathcal{I}$ ($\mathcal{I}$ is hereditary). Additionally, if $A, B \in \mathcal{I}$ and $|A| < |B|$, then there is some element $x \in B \setminus A$ such that $A \cup \{x\} \in \mathcal{I}$ (exchange property).

$v(OPT)/\text{Payoff}(A)$. If A is randomized, the denominator is interpreted to mean the expected payoff. We will always assume oblivious adversaries, so the numerator is never a random variable.

4. AD CAMPAIGNS (KNAPSACK)

The online knapsack buyback problem exemplifies a very relevant aspect of display advertising: advertisers may want to either run all of their campaign or none of it. When there is a limited number of impressions this constraint yields a knapsack like problem. Unlike in the single item case (and in the matroid setting), even with a buyback factor of $f = 0$, there is no online buyback algorithm achieves the optimal offline performance (proof to follow). We begin by presenting a deterministic algorithm for the knapsack buyback problem. We then present a lower bound for deterministic algorithms. Finally, we discuss a randomized algorithm and a lower bound for such algorithms.

4.1 A Deterministic Algorithm

In this section we present a deterministic algorithm for the knapsack problem with buyback. Our first result is a bicriterion result. We show that if the largest element is of size at most γ times the knapsack capacity, where $0 < \gamma < 1$, then we get a competitive ratio of $1 + 2f + 2\sqrt{f(1+f)}$ with respect to the optimum solution for a knapsack whose capacity is scaled down by a factor of $(1 - 2\gamma)$. Then we derive a simple corollary about the competitive ratio when $\gamma < 1/2$.

Conventions. We consider a set $U = \{1, \ldots, n\}$ of requests presented online. Request $k \in U$ has size s_k, value v_k, and density (a.k.a., bang-per-buck) v_k/s_k (we sometimes identify the request with an item of the appropriate size and value). We will assume without loss of generality that the requests are totally ordered by density (e.g., by using request indices to break ties among requests of equal density). Let γ be the ratio of the largest request size to the capacity of the knapsack. Unless otherwise stated, we normalize the capacity of the knapsack to $C = 1$; therefore all requests have size at most γ. For a set $S \subseteq U$ let $\text{OPT}_C(S)$ be the value of the optimal offline fractional knapsack solution with capacity C on requests S. Consider the greedy algorithm that sorts requests in S by density and accepts the densest requests until the next object exceeds the knapsack capacity. Let $\text{dens}_C(S)$ denote the density of this request, or $\text{dens}_C(S) = 0$ if the total size of requests in S does not exceed C. Let $\text{greedy}_C(S)$ denote the requests accepted, and let $\text{reject}_C(S)$ denote the remaining requests (i.e., $\text{reject}_C(S) = S \setminus \text{greedy}_C(S)$).

Algorithm. Our knapsack algorithm admits requests following the greedy fractional offline algorithm on a knapsack with restricted capacity $C = 1 - 2\gamma$, but penalizes requests which are not yet in the knapsack by a multiplicative factor of $r > 1$. When the knapsack's capacity is exceeded, it buys back items greedily, starting from the ones with the lowest bang-per-buck, until the remaining items fit the capacity constraint once again.

Lemma 2. *When elements U are presented to the Knapsack Buyback Algorithm, the set of requests F selected satisfies $\text{OPT}_{1-2\gamma}(F) \geq \text{OPT}_{1-2\gamma}(U)/r$.*

Algorithm 1 Knapsack Buyback Algorithm

```
1: Given: parameters r > 1, 0 < γ < 1/2.
2: Initialize S^(0) = ∅.
3: for all requests k do
4:   if the density of request k is at least r · dens_{1−2γ}(S^(k−1)) then
5:     /* (Integrally) greedily buyback cheapest requests so as to fit request k, i.e., buyback all requests of reject_{1−s_k}(S^(k−1)) to make room for request k. */
6:     Add request k to knapsack.
7:     Set S^(k) ← greedy_1(S^(k−1) ∪ {k}).
8:     Buy back all requests of S^(k) \ S^(k−1)
9:   else
10:    S^(k) ← S^(k−1).
11:  end if
12: end for
13: Output F ← S^(n).
```

Proof. Let $U' = \cup_{k=1}^{n} S^{(k)} \subseteq U$ be the set of all requests that are ever admitted to the knapsack in Step 6, including the ones that get bought back in Step 7. Let S' be the largest prefix of U' that fits completely within the knapsack (i.e., $S' = \text{greedy}_1(U')$).

First, $S' = F$. To see this, note that the algorithm never buys back an element $j \in U'$ unless it has already bought back all elements of smaller density, and it never buys back an element which currently fits in the knapsack. Thus elements of S', once added into the knapsack, will never be bought back, implying that $S' \subseteq F$. The reverse inclusion follows from the fact that the most dense element $j \in F \setminus S'$ would have to satisfy (i) $S' \cup \{j\}$ fits the capacity constraint, (ii) j has higher density than any element of $U' \setminus S'$. If such an element existed, it would belong to $\text{greedy}_1(U')$, contradicting the definition of S'.

Consider the requests F^* (resp. U^*) that are selected by the optimal (offline) fractional knapsack solution with capacity $C = 1 - 2\gamma$ on F (resp. U). Each of these sets possibly contains one fractional request. For this analysis, replace the fractional request by an integral one with same density, whose size is scaled down so that the sum of all request sizes in F^* (resp. U^*) is exactly $1 - 2\gamma$. In other words, if F^* (resp. U^*) contains a fractional amount β of a request with size s and value v, then for this analysis we will instead treat the fractional request as an integral request of size βs and value βv. Thus, according to this convention, $\text{OPT}_{1-2\gamma}(U) = \sum_{k \in U^*} v_k$ and $\text{OPT}_{1-2\gamma}(F) = \sum_{k \in F^*} v_k$.

We now show that the total value in $F^* \setminus U^*$ is at least $1/r$ times that of $U^* \setminus F^*$. Using the fact that $F = S' = \text{greedy}_1(U')$, and that the largest request size is γ, we know that F^* is the optimal fractional knapsack solution with capacity $C = 1 - 2\gamma$ on U'. Thus, F^* contains all requests from U^* that are ever admitted to the knapsack, so the requests in $U^* \setminus F^*$ were not admitted to the knapsack and thus have density less than $r \cdot \text{dens}_{1-2\gamma}(F^*)$. Of course, the total capacity consumed by requests in $F^* \setminus U^*$ and $U^* \setminus F^*$ is identical. Thus the value of requests in $F^* \setminus U^*$ is at least $1/r$ times the value of the requests in $U^* \setminus F^*$. Thus the value of all requests in F^* is at least $1/r$ times the value of requests in U^*. This proves the lemma. □

Lemma 3. *The total amount spent on buyback is at most $\frac{f}{r-1}\,\text{OPT}_{1-2\gamma}(F)$*

PROOF. We use an accounting scheme where when a request is admitted to the knapsack, we charge it with the fraction of any requests that henceforth exceed the capacity $1-\gamma$. By the time such a request is (integrally) bought back by the algorithm because some fraction of it exceeds the capacity 1 we will have already charged the entirety of its buyback cost to admitted requests.

To make the analysis simpler we initially fill the knapsack up to capacity $1-\gamma$ with requests of value zero (and buyback cost zero). Now, if request k of size s_k is admitted to the knapsack, the request receives a charge equal to the buyback cost of s_k fractional units of requests with the least density. This maintains the invariant that the capacity of requests whose buyback is uncharged remains constant at $1-\gamma$. When a fractional buyback of request j is charged to another request k, we also charge k for the sum of all the charges that j ever received, scaled by a factor equal to the fraction of request j that was bought back.

Notice that if request k is admitted to the knapsack then its density is at least $r \cdot \mathrm{dens}_{1-2\gamma}(S^{(k-1)})$. Any requests with fractional buyback charged to k have density at most $\mathrm{dens}_{1-2\gamma}(S^{(k-1)})$. Thus, the new buyback cost charged to item k is at most $\frac{v_k}{r}$. Request k inherits buyback charges previously assessed to these requests. The total buyback charged to k is thus at most

$$v_k f \sum_{i=1}^{\infty} \frac{1}{r^i} = \frac{v_k}{r-1}.$$

When the algorithm ends, the final (fractional) knapsack from F with capacity $1-2\gamma$ has charged buyback of all requests (integrally) bought back by the algorithm to requests in F (as discussed, we have actually overcharged some requests that have not been bought back yet). The total cost of these buybacks is at most $\frac{f}{r-1}\,\mathrm{OPT}_{1-2\gamma}(F)$. □

THEOREM 4. *The Knapsack Buyback Algorithm with parameter r has net payoff (value less buyback cost) of at least*

$$\frac{r-1-f}{r(r-1)}\,\mathrm{OPT}_{1-2\gamma}(U).$$

For a fixed f the best way to pick r is to set $r = 1+f+\sqrt{f(1+f)}$. For such an r the net payoff is at least

$$\frac{1}{1+2f+2\sqrt{f(1+f)}}\,\mathrm{OPT}_{1-2\gamma}(U)$$

PROOF. By Lemma 2 it holds that

$$\mathrm{OPT}_{1-2\gamma}(F) \geq \mathrm{OPT}_{1-2\gamma}(U)/r.$$

By Lemma 3 the buyback cost B is at most $\frac{f}{r-1}\,\mathrm{OPT}_{1-2\gamma}(F)$ and as $F \subseteq U$ it holds that $\mathrm{OPT}_{1-2\gamma}(F) \leq \mathrm{OPT}_{1-2\gamma}(U)$ thus $B \leq \frac{f}{r-1}\,\mathrm{OPT}_{1-2\gamma}(U)$. We conclude that for the Knapsack Buyback Algorithm A it holds that

$$\begin{aligned} \mathit{Payoff}(A) &\geq \mathrm{OPT}_{1-2\gamma}(F) - B \\ &\geq \frac{\mathrm{OPT}_{1-2\gamma}(U)}{r} - \frac{f}{r-1}\,\mathrm{OPT}_{1-2\gamma}(U) \\ &= \frac{r-1-f}{r(r-1)}\,\mathrm{OPT}_{1-2\gamma}(U) \end{aligned}$$

Now the optimal choice of r and the resulting competitive ratio follow from elementary calculus. □

Note that the above result can be viewed as a bi-criterion result: the approximation is with respect to an optimal offline fractional solution for a knapsack of size $1-2\gamma$. With an additional loss of factor $1/(1-2\gamma)$ in the approximation ratio, we can clearly get a bound with respect to the optimal solution for the original knapsack size for the case that $\gamma < 1/2$. This is because $\mathrm{OPT}_{1-2\gamma}(U) \geq (1-2\gamma)\,\mathrm{OPT}_1(U)$, which follows from the definition of $\mathrm{OPT}_C(U)$ as the optimal **fractional** solution.

COROLLARY 5. *Assume $\gamma < 1/2$. The Knapsack Buyback Algorithm with $r = 1+f+\sqrt{f(1+f)}$ has net payoff (value less buyback cost) of at least*

$$\frac{1-2\gamma}{1+2f+2\sqrt{f(1+f)}}\,\mathrm{OPT}_1(U)$$

4.2 Randomized Knapsack Algorithms

In the preceding section, we assumed that the sizes of elements were bounded above by a γ fraction of the knapsack capacity. This restriction, for some $\gamma < 1$, is necessary in order to obtain a bounded competitive ratio using a deterministic algorithm.[4] In fact, the following easy theorem holds. (The proof is omitted for space reasons, and will appear in the full version of the paper.)

THEOREM 6. *For any f when $\gamma = 1$ no deterministic buyback algorithm for knapsack problems has a bounded competitive ratio that is a function of f.*

We can overcome this negative result and achieve a competitive algorithm that works for any γ using randomization. Consider the following randomized algorithm A': with probability 1/3 we run the Knapsack Buyback Algorithm (Algorithm 1) and with probability 2/3 we run the buyback algorithm for a single item (Section 2.1).

THEOREM 7. *For any γ the randomized algorithm A' has an expected net payoff (value less buyback cost) of at least*

$$\frac{1}{3(1+2f+2\sqrt{f(1+f)})}\,\mathrm{OPT}_1(U).$$

PROOF. Define $O_{1/2} = \{u \in \mathrm{OPT}_1(U) \text{ s.t. } s_u \geq 1/2\}$ to be the set of requests in $\mathrm{OPT}_1(U)$ with size at least 1/2. Let $O = \mathrm{OPT}_1(U) \setminus O_{1/2}$. Let $\delta(f) = 1+2f+2\sqrt{f(1+f)}$. Note that when A' runs the Knapsack Buyback Algorithm it gets at least $O/\delta(f)$ and when A' runs the single item algorithm (from Section 2.1) it gets at least $O_{1/2}/(2\delta(f))$ (the lost of factor 2 is due to the fact that $|O_{1/2}| \leq 2$ and the algorithm is competitive with the respect to the best request of $O_{1/2}$). We conclude the the expected payoff of A' is at least

$$\frac{1}{3} \cdot \frac{O}{\delta(f)} + \frac{2}{3} \cdot \frac{O_{1/2}}{2\delta(f)} = \frac{\mathrm{OPT}_1(U)}{3\delta(f)}$$

□

Note that our bi-criteria result shows that when f approaches 0, the Knapsack Buyback Algorithm (Algorithm 1) is close to being perfectly competitive (the competitive ratio goes to 1) with respect to the optimal knapsack solution with capacity $1-2\gamma$ (not 1). On the other hand the above randomized

[4]For γ in the range $[1/2, 1)$, our paper does not resolve the question of whether a bounded competitive ratio can be achieved.

algorithm has competitive ratio of at least 3, even when f is 0. This raises the following question: can a randomized knapsack buyback algorithm have a competitive ratio (with respect to the optimal algorithm on the full size of the knapsack) that approaches 1 when f goes to 0? We next show that the answer for this question is no, and losing some constant in the competitive ratio is inevitable.

THEOREM 8. *Any randomized buyback algorithm for knapsack problems has a competitive ratio of at least* $5/4$. *This is true for any* $f \geq 0$, *even* $f = 0$.

PROOF. We present two inputs such that for at least one of these inputs any randomized buyback algorithm has competitive ratio of at least $5/4$. In both cases the capacity C is 1. The first input has $s_1 = 1, v_1 = 1$ and $s_2 = 1/2, v_2 = 1/2$. The second input is the same as the first with an additional request $s_3 = 1/2, v_3 = 1$. Let p be the probability that the algorithm picks the second request (clearly when faced with the second request the algorithm cannot distinguish the two inputs). The optimal offline algorithm gets a value of 1 on the first input (by picking the first request) and a value of $3/2$ on the second input (by picking the second and third requests). The payoff of the algorithm on the first input is at most $(1-p)\cdot 1 + p \cdot 1/2 = 1 - p/2$, while its payoff on the second input is at most $(1-p)\cdot 1 + p \cdot 3/2 = 1 + p/2$. Thus the competitive ratio in the first case is at least $1/(1-p/2)$ and while in the second case it is at least $(3/2)/(1+p/2)$. It is easy to verify that the optimal way for the algorithm to pick p is to set $p = 2/5$ (in order to minimize the maximal of the two ratios). In this case the ratio is at least $5/4$. Thus it is at least $5/4$ for any p the algorithm picks. □

5. RESULTS FOR THE SINGLE ITEM CASE

In this section we first present a lower bound for deterministic algorithms in the single-item case, and we then show how to design a randomized algorithm that outperforms this lower bound. Finally, we present an extension to a case in which requests have varying buyback factors.

5.1 A Deterministic Lower Bound

In this section we prove that the competitive ratio $1 + 2f + 2\sqrt{f(1+f)}$ obtained by the algorithm in Section 2 is optimal for deterministic algorithms.

THEOREM 9. *For all* $\beta < 1+2f+2\sqrt{f(1+f)}$, *there does not exist a* β*-competitive deterministic online buyback algorithm.*

PROOF. Let us call a finite non-decreasing sequence of numbers $y_1 \leq y_2 \leq \cdots \leq y_n$ an *all-sell sequence* if the algorithm sells to every buyer (after buying back from the preceding buyer) when presented with the input sequence $y_1, y_2, \ldots, y_n$. Denote the set of all such finite sequences by $\mathcal{AS}$. Define an infinite sequence $x_1, x_2, \ldots$ recursively, by stipulating that $x_1 = 1$ and that for $n > 1$,

$$x_n = \inf\{x \geq x_{n-1} | x_1, x_2, \ldots, x_{n-1}, x \in \mathcal{AS}\}. \qquad (1)$$

Note that the set on the right side of (1) is non-empty, unless the algorithm has an unbounded competitive ratio.

When presented with the sequence $x_1, \ldots, x_n, (1-\varepsilon)x_{n+1}$, the algorithm sells to every costumer except the last one, and it buys back from every costumer except the last two. Thus its profit is $x_n - f\sum_{i=1}^{n-1} x_i$ — which is positive unless the algorithm has an unbounded competitive ratio — while the optimum profit is $(1-\varepsilon)x_{n+1}$. As this must hold for every $\varepsilon > 0$ and for every n, the β-competitiveness of the algorithm would imply

$$\forall n \quad x_{n+1} \leq \beta\left(x_n - f\sum_{i=1}^{n-1} x_i\right). \qquad (2)$$

A non-decreasing sequence of positive real numbers can never satisfy (2) when $\beta < 1+2f+2\sqrt{f(1+f)}$, as we prove below in Lemma 10. This completes the proof of Theorem 9. □

LEMMA 10. *For every infinite non-decreasing sequence of real numbers* $0 < x_1 \leq x_2 \leq \ldots$, *if* $\beta < 1+2f+2\sqrt{f(1+f)}$ *then there exists some* n *such that* (2) *is violated.*

PROOF. The proof is by contradiction. Assume that there is a nonempty set S of sequences $x_1 \leq x_2 \leq \ldots$ of positive reals satisfying

$$x_{n+1} \leq \beta x_n - \beta f \sum_{i=1}^{n-1} x_i \qquad (3)$$

for all $n \geq 1$. For any such sequence $\mathbf{x} \in S$, let $n(\mathbf{x})$ be the least n such that the inequality (3) is strict, or $n(\mathbf{x}) = \infty$ if there is no such n. We claim that $n(\mathbf{x})$ takes unboundedly large values as $\mathbf{x}$ ranges over S. Indeed, assume to the contrary that $\mathbf{x} \in S$ is a sequence such that

$$n(\mathbf{x}) = N = \max_{\mathbf{w}\in S} n(\mathbf{w}).$$

Let

$$\lambda = \frac{x_{N+1}}{\beta x_N - \beta f \sum_{i<N} x_i},$$

which is less than 1 by assumption. Then the sequence $\mathbf{x}' = \lambda x_1, \lambda x_2, \ldots, \lambda x_N, x_{N+1}, x_{N+2}, \ldots$ belongs to S, and it satisfies $n(\mathbf{x}') > N$, contradicting our choice of N.

Having thus established that $n(\mathbf{x})$ takes unboundedly large values as $\mathbf{x}$ ranges over the elements of S, we may conclude that the sequence defined by

$$y_1 = 1, \qquad y_{n+1} = \beta y_n - \beta f \sum_{i=1}^{n-1} y_i \qquad \text{for } n \geq 1$$

is non-decreasing. (To prove that $y_{m+1} \geq y_m$ for every m, let $\mathbf{x} \in S$ be a sequence such that $n(\mathbf{x}) > m$ and observe by induction that $y_i = x_i/x_1$ for $i = 1, 2, \ldots, m+1$. Since every sequence $\mathbf{x} \in S$ is non-decreasing, we obtain the inequality $y_{m+1} \geq y_m$ as claimed.) Now define $z_n = \sum_{i=1}^{n} y_i$, and observe that the recursion defining y_{n+1} implies that

$$z_{n+1} = (1+\beta)z_n - \beta(1+f)z_{n-1}.$$

This is a linear recursion, whose general solution is $z_n = as^n + bt^n$ where a, b are arbitrary constants and s, t are the roots of the quadratic equation

$$u^2 - (1+\beta)u + \beta(1+f) = 0. \qquad (4)$$

The discriminant of this quadratic is $(1+\beta)^2 - 4\beta(1+f)$, which is strictly negative given our hypothesis that $1 < \beta < 1 + 2f + 2\sqrt{f(1+f)}$. Hence the two roots s, t are complex conjugates with nonzero imaginary part. We claim also that a, b are complex conjugates. To see this, note that

$$1 = z_1 = a + b$$
$$1 + \beta = z_2 = as + b\bar{s}$$

which implies that the linear system

$$w_1 + w_2 = 1$$
$$s w_1 + \bar{s} w_2 = 1 + \beta$$

is solved by $(w_1, w_2) = (a, b)$ and also by $(w_1, w_2) = (\bar{b}, \bar{a})$. Since the linear system is nonsingular, it has a unique solution. Hence $(a, b) = (\bar{b}, \bar{a})$, confirming our claim that a, b are complex conjugates.

At this point we have established that there are complex numbers a, s such that $z_n = as^n + \bar{a}\bar{s}^n = 2\Re(as^n)$ for all n, and s has nonzero imaginary part. Write $a = qe^{i\phi}$, $s = re^{i\theta}$ where $q, r > 0$ and θ is not an integer multiple of π. Interchanging a, s with $\bar{a}, \bar{s}$ if necessary, we may assume without loss of generality that $0 < \theta < \pi$. Now we have $as^n = qr^n e^{i(\phi + n\theta)}$ which has negative real part if $(2m + \frac{1}{2})\pi < \phi + n\theta < (2m + \frac{3}{2})\pi$ for some integer m. In particular, letting m be the least integer such that $(2m + \frac{1}{2})\pi > \phi$ and recalling that $0 < \theta < \pi$, we find that there must be some $n > 0$ such that $(2m + \frac{1}{2})\pi < \phi + n\theta < (2m + \frac{3}{2})\pi$, implying that $\Re(as^n) < 0$. This contradicts the fact that $z_n > 0$ for every n. □

5.2 Randomized Algorithms

In this section, we show that there exist randomized algorithms whose competitive ratio is superior to the competitive ratio $\beta = 1 + 2f + 2\sqrt{f(1+f)}$ that is optimal for deterministic algorithms. This is true for both the single item case and for matroid domains. Recall that matroid set systems generalize many relevant allocation problems. The uniform matroid of rank k represents feasible sets of winners of a k-unit allocation problem. The transversal matroid represents feasible sets of winners when there is a matching constraint. Transversal matroids exemplify an important aspect of the display advertising problem: that certain advertisers are only interested in certain impressions (e.g., based on demographic, time of day, web page, etc.).

THEOREM 11. *For $0 < f < 1$ there exist randomized buyback algorithms for both the single item case and for matroid domains which have a competitive ratio of*

$$\frac{1}{1 - f^2}\left(1 + f + \sqrt{2f(1+f)}\right).$$

This is less than $1 + 2f + 2\sqrt{f(1+f)}$ when $f < \frac{1}{2}$.

For example, when $f = 0.1$ the randomized algorithm is 1.59-competitive whereas no deterministic algorithms is better than 1.86-competitive. Here we describe the algorithm for the single item case and its extension to matroid domains. The analysis of the algorithm is deferred to the full version of this paper for space reasons.

Intuitively, it is clear why randomization should allow for a superior competitive ratio: it protects the algorithm against input instances like the one constructed in Section 5.1, where the highest bid is $(1 - \varepsilon)$ times the algorithm's threshold value at the time the bid is received. If the algorithm's threshold value at each point in time is allowed to be a random variable, then we can ensure that it is unlikely that any particular bid will be just slightly below the threshold.

Thus, our randomized algorithm will be similar to the one analyzed in Section 2.1, except that the multiplier r will be chosen randomly (and independently of earlier choices) each time a bid is accepted. Also, in order to simplify the analysis of the algorithm, it will be convenient to apply this multiplicative factor of r to the value of the most recent *threshold* rather than the value of the most recently accepted *bid*. If, after sampling r and adjusting the threshold accordingly, it is still less than the value of the most recently accepted bid, then we again multiply the threshold by another random factor r' sampled from the same distribution, repeating as many times as necessary until the threshold becomes greater than the most recently accepted bid. The sequence of thresholds (including the ones that were skipped) thus defines a random increasing sequence of positive numbers $x_1, x_2, \ldots$, whose ratios are independent and identically distributed. Taking logarithms, we obtain a sequence $z_1, z_2, \ldots$ of real numbers whose differences are independent and identically distributed, i.e. a *renewal process* [20, 23]. Appendix A contains a review of the facts from renewal theory that are relevant to an analysis of our algorithm, including the definition of *stationary renewal processes* which plays a key role in our algorithm. Throughout the rest of this section, $\ldots, z_{-1}, z_0, z_1, \ldots$ will denote a stationary renewal process whose interarrival times have finite expected value μ and cumulative distribution function $F(x)$ satisfying $F(0) = 0$. For a real number t, let $z^+(t), z^-(t)$ denote the sample points of the process $(z_i)_{i\in\mathbb{Z}}$ that occur immediately after and before t, respectively. Thus,

$$z^+(t) = \min\{z_i \mid z_i \geq t,\ i \in \mathbb{Z}\}$$
$$z^-(t) = \max\{z_i \mid z_i < t,\ i \in \mathbb{Z}\}.$$

Our algorithm maintains a threshold value u, initialized to zero. At initialization time it also samples a stationary renewal process $\ldots, z_{-1}, z_0, z_1, \ldots$ with interarrival distribution F.[5] When it receives a bid whose value v is greater than u, it checks whether there is a sample point z_i such that $u < e^{z_i} < v$. If so, it accepts the bid and increases its threshold by setting $u = e^{z^-(\ln(v))}$, i.e. the largest value in the set $\{e^{z_i}\}_{i\in\mathbb{Z}}$ that is less than v. In Figure 2 we present a generalization of this algorithm to matroid domains; see Footnote 3 for definitions and notations relating to matroids.

To analyze the algorithm (against an oblivious adversary) let w be the final bid accepted by the algorithm, and let x be the sum of all bids accepted by the algorithm, including w. The total payoff is therefore $w - f(x - w) = (1 + f)w - fx$. More generally, in the matroid setting let Q denote the set of all elements picked during an execution of Algorithm 2 (including those which are later discarded) and let R denote the set of all elements that the algorithm picks but does not discard. The total payoff is therefore $(1 + f)v(R) - fv(Q)$. In the full version of the paper, we prove a lower bound on $v(R)$ and an upper bound on $v(Q)$ in terms of parameters ν, η defined by

$$\nu = \frac{1}{\mu}\int_0^\infty e^{-y}G(y)\,dy, \qquad \eta = \sup_{x>0}\left\{F_0(x)/\left(1 - e^{-x}\right)\right\}.$$

If OPT denotes the value of the maximum-weight matroid basis (e.g., the highest bid in the single-item case) then we prove in the full version of the paper that $\mathbf{E}[v(R)] \geq \nu \cdot \mathrm{OPT}$

[5] Of course, the actual algorithm doesn't do an infinite amount of work at initialization time, it lazily evaluates the numbers z_i only when necessary. In particular, any execution of the algorithm requires computing only $O(\log(v_n/v_1))$ of these numbers in expectation.

Algorithm 2 Randomized Matroid Buyback Algorithm

Given: A random increasing sequence $\{z_i\}_{i\in\mathbb{Z}}$ of real numbers tending to ∞.
Initialize $S = \emptyset$.
for all elements e, in order of arrival, **do**
  $z_e^+ = \min\{z_i \mid e^{z_i} \geq v_e, i \in \mathbb{Z}\}$.
  if $S \cup \{e\} \in \mathcal{I}$ **then**
    Sell to e.
  else
    Let e' be the element of smallest value such that $S \cup \{e\} \setminus \{e'\} \in \mathcal{I}$.
    if $z_e^+ > z_{e'}^+$ **then**
      Sell to e.
      $S = S \cup \{e\} \setminus \{e'\}$.
      Pay back $f \cdot v_{e'}$ to the buyer e'.
    end if
  end if
end for

and $\mathbf{E}[v(Q)] \leq \eta \cdot \mathrm{OPT}$, from which it immediately follows that the algorithm's payoff is at least $[(1+f)\nu - f\eta] \cdot \mathrm{OPT}$, i.e. its competitive ratio is no greater than $[(1+f)\nu - f\eta]^{-1}$.

The problem of designing competitive randomized algorithms thus reduces to the problem of designing non-negative non-decreasing functions $G(y)$ on the positive reals, to maximize $(1+f)\nu - f\eta$ where ν, η are defined as above. This infinite-dimensional optimization problem seems difficult; we do not know a description of the optimal solution. In the full version of the paper we show that a competitive ratio of $\frac{1}{1-f^2}\left(1 + f + \sqrt{2f(1+f)}\right)$ can be achieved using a *truncated Poisson process*, i.e. a stationary renewal process whose interarrival times are exponentially distributed up to a cutoff. (In other words, for some cutoff parameter $a > 0$, the interarrival times have cumulative distribution function $F(y) = 1 - e^{-y} \cdot \mathbf{1}_{y \leq a}$.)

5.3 Varying the Buyback Factor

We can extend the deterministic algorithm to the case that requests have some heterogeneity in their buyback factor. Assume that request i has value v_i and is requiring a buyback factor of (at least) f_i. Previously, to maintain the approximation with buyback factor f the algorithm accepted a request if it has value at least r times the previously accepted value. Now the idea is to use some of the extra value received on top of the minimal value required for acceptance in order to increase the buyback factor of the request, while maintaining the approximation.

Consider a sequence of requests of the form (v_i, f_i). Given a buyback factor f our algorithm from Section 2.1 can be viewed as an algorithm that rejects any request with $f_i > f$, and is $1 + 2f + 2\sqrt{f(1+f)}$-competitive with respect to the highest value request out of all other requests. We next present an extension of the algorithm that achieves the same competitive ratio with respect to a *superset* of the above requests; that superset includes some requests with $f_i > f$, as long as v_i is large enough. We note that our lower bound of Section 5.1 shows that there is no algorithm that is $1+2f+2\sqrt{f(1+f)}$-competitive with respect to *all* possible sequences of requests of the form (v_i, f_i).

We now consider the following extension of the algorithm of Section 2.1 which we call the *Flexible Buyback Factor Algorithm.* Fix $f > 0$ and $r > 1$. In the original algorithm, if the current winner has value v_w (and it is the m-th winner) and the current request has value v_i (and an implicit buyback factor $f_i = f$), then the request is accepted if $v_i \geq g(r, f, v_w, f_i)$ for $g(r, f, v_w, f_i) = r \cdot v_w$. In the new algorithm we redefine the function g to be

$$g(r, f, v_w, f_i) = \frac{f \cdot v_w \cdot \sum_{l=0}^{m} r^{-l}}{f \cdot \sum_{l=0}^{m+1} r^{-l} - max\{f, f_i\}}$$

If the denominator is negative we define g to be infinity (the request is always rejected). Thus for a request to be considered it must be the case that $f_i \leq f \cdot \sum_{l=0}^{m+1} r^{-l} < f \cdot \frac{r}{r-1}$. Note that for $f_i = f$ the new algorithm is the same as the original one.

We next define the set of requests R that the Flexible Buyback Factor Algorithm is competitive against. Assume that request (v_i, f_i) is considered when the current winner has value z. Then the request belongs to R if $f_i \leq f$ or $v_i \geq g(r, f, z, f_i)$, or $v_i < z$.

Note that R contains all requests whose buyback factor is less than or equal to f, thus the following result generalizes the result of Section 2.1.

THEOREM 12. *The Flexible Buyback Factor Algorithm is* $1+2f+2\sqrt{f(1+f)}$ *competitive with respect to the maximum value request in* R.

We defer the proof to the full version of the paper. The proof of Theorem 12 shows how one can use an increase in the request value over the minimal needed for acceptance towards accepting that request even with somewhat higher buyback factor than f. Finally, we note that alternatively it is possible to use the same increase in value to make it harder to bump out the request. That is, it is possible to keep the same buyback factor f and increase r (and the threshold) for future requests that attempt to bump out the current one.

6. REFERENCES

[1] Aggarwal, G., and Hartline, J. D. Knapsack auctions. In *SODA* (2006), pp. 1083–1092.

[2] Babaioff, M., Dinitz, M., Gupta, A., Immorlica, N., and Talwar, K. Secretary problems: weights and discounts. In *SODA* (2009), pp. 1245–1254.

[3] Babaioff, M., Hartline, J. D., and Kleinberg, R. Online algorithms with buyback. In *Proc. 4th Workshop on Ad Auctions* (2008).

[4] Babaioff, M., Immorlica, N., Kempe, D., and Kleinberg, R. A knapsack secretary problem with applications. In *APPROX-RANDOM* (2007), pp. 16–28.

[5] Babaioff, M., Immorlica, N., and Kleinberg, R. Matroids, secretary problems, and online mechanisms. In *SODA* (2007), pp. 434–443.

[6] Biyalogorsky, E., Carmon, Z., Fruchter, G. E., and Gerstner, E. Research note: Overselling with opportunistic cancellations. *Marketing Science 18*, 4 (1999), 605–610.

[7] Borgs, C., Chayes, J., Etesami, O., Immorlica, N., Jain, K., and Mahdian, M. Dynamics of bid optimization in online advertisement auctions. In *WWW* (2007).

[8] Buchbinder, N., Jain, K., and Naor, J. S. Online primal-dual algorithms for maximizing ad-auctions revenue. In *ESA* (2007), pp. 253–264.

[9] Chakrabarty, D., Zhou, Y., and Lukose, R. Budget constrained bidding in keyword auctions and online knapsack problems. In *WWW2007, Workshop on Sponsored Search Auctions* (2007).

[10] Constantin, F., Feldman, J., Muthukrishnan, S., and Pál, M. An online mechanism for ad slot reservations with cancellations. In *Proc. 4th Workshop on Ad Auctions* (2008).

[11] Constantin, F., Feldman, J., Muthukrishnan, S., and Pál, M. An online mechanism for ad slot reservations with cancellations. In *Proc. 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (2009), pp. 1265–1274.

[12] Dimitrov, N. B., and Plaxton, C. G. Competitive weighted matching in transversal matroids. In *ICALP (1)* (2008), pp. 397–408.

[13] Dynkin, E. B. Optimal choice of the stopping moment of a Markov process. *Dokl. Akad. Nauk SSSR 150* (1963), 238–240.

[14] Feige, U., Immorlica, N., Mirrokni, V. S., and Nazerzadeh, H. A combinatorial allocation mechanism with penalties for banner advertising. In *WWW* (2008), pp. 169–178.

[15] Iwama, K., and Taketomi, S. Removable online knapsack problems. In *ICALP* (2002), pp. 293–305.

[16] Iwama, K., and Zhang, G. Optimal resource augmentations for online knapsack. In *APPROX-RANDOM* (2007), pp. 180–188.

[17] Kleinberg, R. A multiple-choice secretary algorithm with applications to online auctions. In *SODA* (New York, 2005), ACM, pp. 630–631 (electronic).

[18] Lueker, G. S. Average-case analysis of off-line and on-line knapsack problems. In *SODA* (1995), pp. 179–188.

[19] Marchetti-Spaccamela, A., and Vercellis, C. Stochastic on-line knapsack problems. *Math. Program. 68*, 1 (1995), 73–104.

[20] Medhi, J. *Stochastic Models in Queueing Theory, Second Edition.* Academic Press, October 2002.

[21] Mehta, A., Saberi, A., Vazirani, U., and Vazirani, V. AdWords and generalized online matching. *J. ACM 54*, 5 (2007).

[22] Noga, J., and Sarbua, V. An online partially fractional knapsack problem. In *ISPAN '05: Proceedings of the 8th International Symposium on Parallel Architectures,Algorithms and Networks* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 108–112.

[23] Resnick, S. *Adventures in stochastic processes.* Birkhäuser, 1992.

[24] Rusmevichientong, P., and Williamson, D. An adaptive algorithm for selecting profitable keywords for search-based advertising services. In *EC* (2006).

[25] Sandholm, T., and Lesser, V. R. Advantages of a leveled commitment contracting protocol. In *AAAI/IAAI, Vol. 1* (1996), pp. 126–133.

APPENDIX

A. REVIEW OF RENEWAL THEORY

In this section we review some basic facts from renewal theory. A brief treatment of this material can be found in Section 1.7 of [20]; for a more thorough presentation see Chapter 3 of [23].

Let $x_1, x_2, \ldots$ be an infinite sequence of independent, identically distributed random variables each having cumulative distribution function F satisfying $F(0) = 0$, and let x_0 be another non-negative random variable, independent of $\{x_n\}_{n\geq 1}$, but possibly having a different distribution. The partial sums $z_k = x_0 + x_1 + \cdots + x_k$ constitute a stochastic process $\{z_n\}_{n\geq 0}$ called a *renewal process*. The process is called *delayed* if $\Pr(x_0 = 0) > 0$, otherwise it is called *pure*.

In our randomized algorithm, we are interested in using a renewal process that differs from the preceding definition in two respects. First, the points $\{z_i\}$ are indexed by the set of all integers, negative as well as positive. Second, and more importantly, the process is *stationary*, meaning that the distribution of the random set $\{z_i\}_{i\in\mathbb{Z}}$ is translation-invariant: for every fixed $t \in \mathbb{R}$, the random set $\{z_i + t\}_{i\in\mathbb{Z}}$ has the same distribution. We will always assume that the points of the set $\{z_i\}_{i\in\mathbb{Z}}$ are numbered so that $z_{-1} < 0 \leq z_0$. Note that this numbering convention doesn't interfere with the property of translation-invariance, since that property only asserts that $\{z_i\}_{i\in\mathbb{Z}}$ and $\{z_i + t\}_{i\in\mathbb{Z}}$ have the same distribution *as sets*, not as integer-indexed sequences.

A notable feature of stationary renewal processes is the so-called *inspection paradox*: the length of the random interval $[z_{-1}, z_0]$ that contains 0 has a different distribution from the lengths of the other intervals $[z_{i-1}, z_i]$, $i \neq 0$: it is biased to have a greater length in expectation. This reflects the fact that long intervals are more likely than short ones to contain the point 0. In fact, $z_0 - z_{-1}$ is sampled from the *size-biased* distribution that reweights intervals in proportion to their length. If $F(x)$ denotes the cumulative distribution function of the interarrival time $z_i - z_{i-1}$ ($i \neq 0$), and μ denotes the expected interarrival time, then $z_0 - z_{-1}$ has cumulative distribution function $H(x) = \frac{1}{\mu}\int_0^x (F(x) - F(y))\,dy$. (In the case when F has a density function $f(x)$, then H is the distribution with density $xf(x)$, matching our earlier informal description of the size-biased distribution.)

Conditional on the length of the interval $[z_{-1}, z_0]$, the distance of the point 0 from the right endpoint is uniformly distributed. In other words, z_0 has the same distribution as the product rs, where r is a uniformly random sample from $[0, 1]$ and s is sampled from the size-biased distribution, $H(x)$, defined above. As it happens, this means that z_0 has cumulative distribution function $F_0(x) = \int_0^x (1 - F(y))\,dy$. Since the distance from 0 to the left endpoint z_{-1} is also uniformly distributed, this means that the random $-z_{-1}$ also has distribution F_0.

For any t, define $z^-(t), z^+(t)$ to be the sample points of $\{z_i\}_{i\in\mathbb{Z}}$ that occur immediately before and after t; i.e.,

$$z^+(t) = \min\{z_i \mid z_i \geq t,\ i \in \mathbb{Z}\}$$
$$z^-(t) = \max\{z_i \mid z_i < t,\ i \in \mathbb{Z}\}.$$

We have $z^+(0) = z_0$, $z^-(0) = -z_{-1}$, so both $z^+(0)$ and $z^-(0)$ have cumulative distribution function F_0. By translation-invariance, the same holds for the distribution of $z^+(t) - t$ and $t - z^-(t)$, for every real number t.