Minimizing Regret with Multiple Reserves

TIM ROUGHGARDEN, Stanford University
JOSHUA R. WANG, Stanford University

We study the problem of computing and learning non-anonymous reserve prices to maximize revenue. We first define the MAXIMIZING MULTIPLE RESERVES (MMR) problem in single-parameter matroid environments, where the input is m valuation profiles $\mathbf{v}^1, \ldots, \mathbf{v}^m$, indexed by the same n bidders, and the goal is to compute the vector $\mathbf{r}$ of (non-anonymous) reserve prices that maximizes the total revenue obtained on these profiles by the VCG mechanism with reserves $\mathbf{r}$. We prove that the problem is APX-hard, even in the special case of single-item environments, and give a polynomial-time $\frac{1}{2}$-approximation algorithm for it in arbitrary matroid environments.

We then consider the online no-regret learning problem, and show how to exploit the special structure of the MMR problem to translate our offline approximation algorithm into an online learning algorithm that achieves asympototically time-averaged revenue at least $\frac{1}{2}$ times that of the best fixed reserve prices in hindsight. On the negative side, we show that, quite generally, computational hardness for the offline optimization problem translates to computational hardness for obtaining vanishing time-averaged regret. Thus our hardness result for the MMR problem implies that computationally efficient online learning requires approximation, even in the special case of single-item auction environments.

CCS Concepts: •**Theory of computation** → **Computational pricing and auctions;** *Approximation algorithms analysis; Online algorithms; Algorithmic mechanism design;*

Additional Key Words and Phrases: Second-Price Auction, Non-Anonymous Reserve Prices, No Alpha-Regret

ACM Reference Format:
Tim Roughgarden and Joshua R. Wang, 2016. Minimizing Regret with Multiple Reserves. *ACM Trans. Embedd. Comput. Syst.* 9, 4, Article 39 (March 2010), 17 pages.
DOI: 0000001.0000001

1. INTRODUCTION

A basic issue in the design and deployment of revenue-maximizing auctions is the determination of appropriate reserve prices. For example, consider the well-studied problem of selecting an anonymous reserve price in a second-price (Vickrey) auction for a single item — equivalently, an opening bid in an eBay auction.[1] There are several versions of the problem, depending on the informational assumptions made.

Baysian optimization. The standard economic approach is to assume a prior distribution over the valuations of the bidders participating in the auction, and to maximize expected revenue with respect to this distribution. If bidders' valuations

[1] In a second-price single-item auction with a reserve price r, the winner is the highest bidder that clears the reserve (if any), and the selling price is the maximum of r and the second-highest bid. Bidding truthfully is a dominant strategy for every bidder.

Author's addresses: Tim Roughgarden: Department of Computer Science, Stanford University, 474 Gates Building, 353 Serra Mall, Stanford, CA 94305. Email: tim@cs.stanford.edu. Joshua R. Wang: Department of Computer Science, Stanford University, 460 Gates Building, 353 Serra Mall, Stanford, CA 94305. Email: joshua.wang@cs.stanford.edu. This research was supported in part by NSF grants CCF-1215965 and CCF-1524062, and a Stanford Graduate Fellowship.

DOI: 0000001.0000001

are drawn i.i.d. from a distribution that satisfies some modest technical conditions ("regularity"), then setting the reserve price equal to the distribution's monopoly price yields an optimal auction [Myerson 1981]. Chawla et al. [2007], Hartline and Roughgarden [2009], and Alaei et al. [2015] provide guidance on how to set the reserve price when the i.i.d. or regularity assumptions are relaxed.

Offline optimization. Given m arbitrary valuation profiles $\mathbf{v}^1, \ldots, \mathbf{v}^m$ (each an n-vector, where n is the number of bidders), the goal here is to determine the reserve price of a second-price auction that maximizes the average auction revenue across these profiles. There is always an optimal reserve price equal to some bidder's valuation in one the m profiles, so there are at most mn different reserve prices that need to be tried. Thus the problem is easy to solve in polynomial time.

Batch learning. In this version, there is an unknown distribution F over bidders' valuations. A learning algorithm is given m i.i.d. samples from F, and must then output a reserve price. The goal is to output a reserve price that achieves expected revenue (w.r.t. fresh draws from F) close to that obtained by the best reserve price (for F), with high probability over the samples. This problem was studied implicitly by Dhangwatnotai et al. [2010] and explicitly by Medina and Mohri [2014] and Huang et al. [2015].

Online no-regret learning. In online learning, valuation profiles $\mathbf{v}^1, \ldots, \mathbf{v}^T$ arrive one at a time, and at time t a learning algorithm chooses a reserve price as a function of the previously seen profiles $\mathbf{v}^1, \ldots, \mathbf{v}^{t-1}$. The goal is to choose reserve prices over time so that the time-averaged revenue is almost as high as that achieved by the best fixed reserve price in hindsight. Versions of this problem were studied by Kleinberg and Leighton [2003], Blum and Hartline [2005], and Cesa-Bianchi et al. [2013].

There are well known connections between these different problems. For example, an efficient exact or approximate algorithm for the offline optimization problem implies, in "black-box" fashion, an efficient learning algorithm for the batch learning problem with the same approximation factor (namely the "empirical risk minimizer" (ERM), see e.g. [Anthony and Bartlett 1999]). An efficient exact algorithm for the offline problem can also be translated into an efficient online no-regret learning algorithm (see e.g. [Cesa-Bianchi and Lugosi 2006]).[2]

In this paper, we study the problem of computing or learning *non-anonymous reserve prices*, where a different reserve price r_i can be used for each bidder i. For example, in a single-item second-price auction with reserves $r_1, \ldots, r_n$, the winner is the highest bidder who clears her reserve (if any), and the selling price is either her reserve price or the second-highest bid of a bidder who cleared her own reserve price, whichever is larger.[3]

All four genres of problems are relevant and interesting with non-anonymous reserve prices; we study the latter three in this work.

Baysian optimization. Hartline and Roughgarden [2009], extending [Chawla et al. 2007], study the power of non-anonymous reserve prices in Bayesian settings,

[2] The positive results by Kleinberg and Leighton [2003] and Cesa-Bianchi et al. [2013] are interesting and non-trivial because they apply even when the price-setter receives only limited feedback from each auction.

[3] These are called "eager reserves" by Dhangwatnotai et al. [2010]. With "lazy reserves," the winner is the highest bidder, if she clears her reserve (and no one, otherwise), and the selling price is the maximum of her reserve and the second-highest bid overall. Note that whenever a sale occurs with lazy reserves, a sale would also occur with eager reserves (but not vice versa). Eager reserves are superior from both a welfare and revenue standpoint; see Paes Leme et al. [2016] for these and many other interesting comparisons.

where bidders' valuations are independently but not identically distributed. The main results show that setting each bidder's reserve to the monopoly price for her distribution yields a mechanism with near-optimal expected revenue in many scenarios.[4]

Offline optimization. Given m arbitrary valuation profiles $\mathbf{v}^1, \ldots, \mathbf{v}^m$, each indexed by the same set of bidders, the goal is now to compute one reserve price per bidder to maximize the average revenue of the second-price auction with reserves across these profiles. In contrast to the anonymous reserve price setting, this optimization problem is non-trivial: there are now m^n different price vectors that could be optimal. This problem has not been studied previously, except for the independent work by Paes Leme et al. [2016], who proved that the problem is NP-hard.[5]

Batch learning. The learning algorithm must now compute, based on samples, one reserve price per bidder. This problem was studied by Morgenstern and Roughgarden [2015]: their "1-level auctions" correspond exactly to second-price auctions with non-anonymous reserve prices.[6] Their work focuses on sample complexity bounds. Their learning algorithm uses an algorithm for offline optimization as a subroutine, but they do not address its computational efficiency.

Online no-regret learning. The online learning algorithm must now compute one reserve price per bidder in each time step, and perform almost as well as the best fixed non-anonymous reserve prices in hindsight. This problem has not been previously considered.

We study the problem of computing and learning multiple reserve prices primarily because of its basic nature, but we note that non-anonymous reserve prices are used in practice (often in a disguised form, for better optics). For example, in sponsored search auctions a "quality score" is associated with each advertiser, and different quality scores translate to different effective reserve prices for different advertisers (see e.g. [Athey and Nekipelov 2012]). Along the same lines, in the ongoing FCC Incentive Auction (for re-allocating spectrum licenses from TV broadcasters to mobile broadband companies), a particular formula is used to set different opening bids for different participants, depending on the market share and geographic location of the broadcaster (see e.g. [Cramton et al. 2015]).

1.1. Our Results

We first consider the offline optimization problem MAXIMIZING MULTIPLE RESERVES (MMR) in general matroid environments (see Section 2 for background). The input is m valuation profiles $\mathbf{v}^1, \ldots, \mathbf{v}^m$, indexed by the same n bidders. The goal is to compute the vector $\mathbf{r}$ of reserve prices that maximizes the total revenue obtained on these profiles by the Vickrey-Clarke-Groves mechanism with reserves $\mathbf{r}$. We prove that the problem is APX-hard (i.e., hard to approximate better than some fixed constant), even in the special case of single-item environments, and give a polynomial-time $\frac{1}{2}$-approximation algorithm for it in arbitrary matroid environments.[7]

[4] These results are extended to lazy monopoly reserves by Dhangwatnotai et al. [2010].

[5] The analogous problem with lazy reserve prices is easy [Paes Leme et al. 2016].

[6] The problem of batch learning a near-optimal auction that need not be reserve-price-based is studied by Cole and Roughgarden [2014] and Devanur et al. [2016].

[7] Both our approximation algorithm and analysis approach bear some resemblance to the lookahead auction of Ronen [2001] and its extension to matroids by Chawla et al. [2014]. (The uniform distribution over the valuation profiles in an offline instance can be thought of as a correlated valuation distribution.) We require

This immediately implies a polynomial-time approximate learning algorithm for the batch learning problem, which (with high probability) computes reserve prices leading to expected revenue (w.r.t. the unknown distribution F) at least $\frac{1}{2} - \epsilon$ times that obtained with the best reserve prices for F.

We then consider the online no-regret learning problem. The $\frac{1}{2}$-approximation algorithm for the offline problem does not automatically imply an approximately no-regret online learning algorithm.[8] We show how to exploit the special structure of the MMR problem to translate our offline approximation algorithm into an online learning algorithm that achieves time-averaged revenue at least $\frac{1}{2}$ times that of the best fixed reserve prices in hindsight, less an $o(1)$ error term (as $T \to \infty$). This positive result applies to arbitrary matroid environments. On the negative side, we show that, quite generally, computational hardness for the offline optimization problem translates to computational hardness for obtaining vanishing time-averaged regret.[9] (This is not obvious because an online algorithm might achieve high revenue with different reserves at different times, while never figuring out the best fixed reserves in hindsight.) This general translation may be of independent interest. In any case, our hardness result for the MMR problem carries over to the problem of online no-regret learning, even in the special case of single-item auction environments.

2. PRELIMINARIES

2.1. Matroid Environments

For our purposes, an environment is defined by a set E of bidders, and a non-empty collection $\mathcal{I} \subseteq 2^E$ of feasible sets of bidders, which are the subsets of bidders that can simultaneously "win." For example, in a k-unit auction with unit-demand bidders, $\mathcal{I}$ is all subsets of E that have size at most k. Each bidder has a private *valuation* for winning. Depending on the setting, bidders' valuations can be drawn from a (possibly unknown) distribution, or arbitrary.

We consider *matroid* environments, where (i) the set system $(E, \mathcal{I})$ is *downward-closed*, meaning that if $T \in \mathcal{I}$ and $S \subseteq T$, then $S \in \mathcal{I}$; and (ii) the *exchange property* holds, stating that whenever $S, T \in \mathcal{I}$ with $|T| < |S|$, there is some $i \in S \setminus T$ such that $T \cup \{i\} \in \mathcal{I}$. Examples of matroid environments include digital goods (where $\mathcal{I} = 2^E$), k-unit auctions (where $\mathcal{I}$ is all subsets of size at most k), and certain unit-demand matching markets (corresponding to transversal matroids).

2.2. The VCG Mechanism with Non-Anonymous Reserves

Name the bidders $E = \{1, 2, \ldots, n\}$. A (deterministic) mechanism $\mathcal{M}$ comprises an *allocation rule* $\mathbf{x}$ that maps every bid vector $\mathbf{b}$ to a characteristic vector of a feasible set (in $\{0,1\}^n$), and a *payment rule* $\mathbf{p}$ that maps every bid vector $\mathbf{b}$ to a non-negative payment vector in $[0, \infty)^n$. We assume that every bidder i aims to maximize its quasi-linear utility $u_i(\mathbf{b}) = v_i \cdot x_i(\mathbf{b}) - p_i(\mathbf{b})$, where v_i is its private valuation for winning. We call a mechanism $\mathcal{M}$ *truthful* if for every bidder i and fixed bids $\mathbf{b}_{-i}$ of the other bidders, bidder i maximizes its utility by setting its bid b_i to its private valuation v_i. Since we

a different argument because we restrict our solution to a fixed vector of reserve prices; the lookahead auction uses "conditional reserve prices." To see the difference, suppose there are two bidders and two valuation profiles, $(1, 2)$ and $(2, 4)$. The lookahead auction extracts the full welfare with respect to the uniform distribution over the two profiles (by always charging the second bidder twice the first bid), and no fixed set of reserve prices can compete.

[8] For *linear* optimization problems, there is such a black-box reduction [Kakade et al. 2009]. But in the MMR problem, the revenue is not a linear function of the reserves or of the valuations.

[9] Related results, based on somewhat different arguments, were developed independently by Daskalakis and Syrgkanis [2016], in the context of utility-maximization for a bidder (rather than revenue-maximization for a seller) in simultaneous second-price auctions.

only consider truthful mechanisms, in the rest of the paper we use valuations and bids interchangeably. The *efficiency* or *welfare* of the outcome of a mechanism is the sum of the winners' valuations, and the *revenue* is the sum of the winners' payments.

The *VCG mechanism* chooses the feasible set $S \in \mathcal{I}$ that maximizes the welfare $\sum_{i \in S} v_i$. Each winner in the mechanism pays the smallest bid at which she would continue to win; this results in a truthful mechanisms. In a single-item setting, this is just the second-price auction.

Let r_i be a *reserve price* for bidder i. The *VCG mechanism with reserves* $\mathbf{r}$ works as follows, given bids $\mathbf{v}$: (1) remove all bidders i with $v_i < r_i$; (2) run the VCG mechanism on the remaining bidders to determine the winners; (3) charge each winning bidder i the larger of r_i and its VCG payment in step (2). This is again a truthful mechanism (for any $\mathbf{r}$).

Matroid environments are special in that the VCG mechanism can be implemented using a greedy algorithm. Specifically, to choose the winners: sort the bidders in order of decreasing bids, and in one pass in this order, add a bidder to the winner set if and only if doing so preserves feasibility. It is well known and easy to prove that this algorithm computes the welfare-maximizing outcome.

This greedy algorithm also reveals what the payments are. Consider some winning bidder i. As a thought experiment, imagine re-running the greedy algorithm without i, and let $f(i)$ be the first winning bidder after which it would be infeasible to add w the winner set (if any). The VCG payment of i is the valuation of $f(i)$, or zero if $f(i)$ does not exist.

3. THE OFFLINE OPTIMIZATION PROBLEM: ALGORITHMS

The offline MMR problem is interesting in its own right, and algorithms for it also serve as a useful starting point for designing online MMR algorithms. We warm-up with the single-item case (Section 3.1) and then address the general matroid environment case (Section 3.2).

3.1. Warm-Up: The Single-Item Case

In the single-item special case of the MMR problem, the input is valuation profiles $\mathbf{v}^1, \ldots, \mathbf{v}^m$, and the goal is to choose (non-anonymous) reserve prices $\mathbf{r}$ to maximize the average revenue of a second-price auction with reserves $\mathbf{r}$ on these profiles. Without loss of generality, each reserve price r_i is equal to the valuation of bidder i in one of the m valuation profiles. (This still leaves a search space of m^n reserve price vectors.)

Observe the tension in setting the reserve price r_i for a bidder i: increasing it may increase revenue on profiles where i is a winner (provided r_i stays below v_i), but it may also decrease revenue if r_i passes v_i, even when i is only the *second-highest* bidder (v_i was previously setting the price of the winner). The following algorithm balances these considerations.

MMR Algorithm (Single-Item Case)

```
for each bidder $i = 1, 2, \ldots, n$ do
    let $S_i$ denote the profiles where $i$ has the highest valuation;
    let $v^j_{(2)}$ denote the second highest valuation in $\mathbf{v}^j$;
    let $r_i$ maximize $\sum_{j \in S_i} q_j(r_i)$, where $q_j(r_i)$ is $r_i - v^j_{(2)}$ if $r_i \in [v^j_{(2)}, v^j_i]$, and 0 otherwise
end
return either $\mathbf{r}$ or the all-zero vector, whichever generates more revenue;
```

In words, $q_j(r_i)$ represents the additional revenue obtained by the reserve price r_i from bidder i on day j, above and beyond the revenue already obtained by a reserve price of 0.

This algorithm can be implemented in polynomial time. Each reserve price r_i can be computed independently, and there are only m different relevant choices for each.

LEMMA 3.1. *The algorithm above is a $\frac{1}{2}$-approximation algorithm for the* MMR *problem in single-item environments.*

PROOF. Fix an input $\mathbf{v}^1, \ldots, \mathbf{v}^m$ and let $\mathbf{r}^*$ denote the optimal reserve prices. The revenue obtained from a valuation profile $\mathbf{v}^j$ with $j \in S_i$ is at most $\max\{v^j_{(2)}, r^*_i\}$. (If the item is sold to someone other than the highest bidder, then the selling price is at most $v^j_{(2)}$.) Overall, the optimal revenue can be upper bounded by

$$\sum_{i=1}^{n} \sum_{j \in S_i} \max\{v^j_{(2)}, r^*_i\} \leq \sum_{j=1}^{m} v^j_{(2)} + \sum_{i=1}^{n} \sum_{j \in S_i} q_j(r^*_i).$$

The all-zero reserves obtain revenue equal to the first term on the right-hand side. The reserves r computed by the algorithm above obtain, by construction, revenue that is at least the second term on the right-hand side (with r_i chosen to maximize the ith inner sum). The better of these two reserve price vectors earns revenue at least $\frac{1}{2}$ times the maximum possible. □

Example 3.2. Our analysis in Lemma 3.1 is tight: for every $\epsilon > 0$, there is an input where the algorithm achieves less than $(1/2 + \epsilon)$ times the optimal revenue. The bad input is as follows: let n be an integer to be chosen later, and let there be two bidders and n valuation profiles. The first bidder has valuation n on the first day, valuation 0 on the second day, and valuation $1 + 1/n$ on every other day. The second bidder has valuation 0 on the first day, valuation $1 + 1/n$ on the second day, and valuation 1 on every other day. For this input, the all-zero vector will achieve a revenue of $(n - 2)$. Also, r will be the vector $(n, 1 + 1/n)$, which achieves a revenue of $(n + 1 + 1/n)$. However, choosing $\mathbf{r}^\star$ to be $(n, 1)$ results in a revenue of $(2n - 1)$. When $n > \frac{5}{4}\epsilon^{-1} + \frac{1}{2}$, the algorithm's revenue is less than $(1/2 + \epsilon)$ times the optimal revenue.

3.2. An Offline MMR Algorithm

We now consider matroid environments. The new complication is that there are multiple winners, and we need to determine the relationship between the winners in the VCG mechanism (without reserves) and the winners in an optimal solution.

The generalization of our previous algorithm is as follows.

MMR Algorithm (Matroid Case)

```
for each bidder $i = 1, 2, \ldots, n$ do
    let $S_i$ denote the profiles where $i$ is a winner in the VCG mechanism (without reserves);
    for $j \in S_i$, let $p^j_i$ denote $i$'s payment in the VCG mechanism (without reserves) with the profile $\mathbf{v}^j$;
    let $r_i$ maximize $\sum_{j \in S_i} q^j(r_i)$, where $q^j(r_i)$ is $r_i - p^j_i$ if $r_i \in [p^j_i, v^j_i]$, and 0 otherwise
end
return either r or the all-zero vector, whichever leads to more revenue;
```

As in the single-item special case, it is straightforward to implement this algorithm so that it runs in polynomial time.

As a lead-up to the analysis, the next three propositions review well-known properties of matroids and the VCG mechanism (see e.g. [Talwar 2003]).

PROPOSITION 3.3 ([SCHRIJVER 2003], COROLLARY 39.12A). *Let W, W' be feasible sets of size k in a matroid M. Then there is a bijection $f : W' \setminus W \to W \setminus W'$ such that for every $i \in W' \setminus W$, the set $W \setminus \{f(i)\} \cup \{i\}$ is a feasible set in M.*

From the definition of VCG payments, we have:

PROPOSITION 3.4. *Suppose that the winners in the VCG mechanism with zero reserves are W, and that $i \in W$ but $j \notin W$, and that $W \setminus \{i\} \cup \{j\}$ is a feasible set. Then the VCG payment of i is at least the valuation of j.*

The following proposition explains what happens when a winner is removed from the auction:

PROPOSITION 3.5. *Suppose the current winners of a matroid auction are W, and we then remove some winner $w \in W$ from the auction. Then the new set of winners will be $W \setminus \{w\} \cup p(w)$, if $p(w)$ exists, or $W \setminus \{w\}$ otherwise.*

LEMMA 3.6 (REVENUE DECOMPOSITION LEMMA). *Consider a valuation profile* v *and reserve prices* $\mathrm{r}^\star$. *Let the winners of the VCG mechanism with profile* v *and no reserves be W, with p_i denoting the payment of i. Then, the revenue of* $\mathrm{r}^\star$ *on* v *is at most*

$$\sum_{i \in W} p_i + \sum_{i \in W} q(r_i^*),$$

where $q(r_i^*)$ is $r_i^* - p_i$ if $r_i \in [p_i, v_i]$, and 0 otherwise.

PROOF. The winning set W with zero reserves is a basis of the matroid of feasible subsets. The reserves $\mathrm{r}^\star$ result in some set of winners $W^\star$. Extend $W^\star$ to a basis using the exchange property (Section 2), and apply Proposition 3.3 to get a bijection from a superset of $W^\star$ to W. Ignoring elements not in $W^\star$, we get a injection f from $W^\star$ to W. Furthermore, we are guaranteed that for every bidder $i \in W^\star$, $W \setminus \{i\} \cup \{f(i)\}$ is feasible. Applying Proposition 3.4, this means that whenever $f(i) \neq i$, $p_{f(i)}$ is at least the valuation of i.

Next, consider a bidder $i \in W$. We claim that increasing the reserves for bidders other than i (from 0 to some positive amount) can never cause i to lose or to pay more than p_i. For the first statement, the only effect of reserve prices on the winning set is to remove bidders before invoking the greedy algorithm. By Proposition 3.5, removing a winner and re-running the greedy algorithm simply causes them to be replaced (i.e., no other bidders are removed as a consequence). Trivially, removing a nonwinner has no effect on the winning set. Thus if i's reserve is not changed, it remains a winner. Similarly, removing other bidders can only decrease the valuation of the losing bidder that sets i's price (recall Section 2).

These two facts imply that, for every winner $i \in W^\star$, we can use f to find a winner $f(i) \in W$ such that either (i) i's valuation, and hence contribution to the revenue of $\mathrm{r}^\star$, is at most $p_{f(i)}$ or (ii) $f(i) = i$ and i only pays more than p_i to the extent that $r_i^\star$ is larger than p_i (while also being at most v_i). The quanity in (ii) is precisely $q(r_i^*)$. Because f is an injection, we can sum over all $w \in W^\star$ to get the inequality in the lemma statement. □

The lemma easily implies that our algorithm is a $\frac{1}{2}$-approximation.

THEOREM 3.7. *The algorithm above is a $\frac{1}{2}$-approximation algorithm for the* MMR *problem in matroid environments.*

PROOF. Fix an input $\mathbf{v}^1, \ldots, \mathbf{v}^m$ and let $\mathbf{r}^*$ denote the optimal reserve prices. Let W_j denote the winners in the VCG mechanism with profile $\mathbf{v}^j$ and no reserves, and p_i^j the payment made by a bidder $i \in W_j$. Applying Lemma 3.6 to each profile $\mathbf{v}^j$ and summing over j, the total revenue obtained by $\mathbf{r}^*$ is at most

$$\sum_{j=1}^{m} \left(\sum_{i \in W_j} p_i^j + \sum_{i \in W_j} q^j(r_i^*) \right) = \sum_{j=1}^{m} \sum_{i \in W_j} p_i^j + \sum_{i=1}^{n} \sum_{j \in S_i} q^j(r_i^*),$$

where S_i and q^j are defined as in the algorithm description. The all-zero reserves obtain total revenue equal to the first term on the right-hand side. The reserves r computed by the algorithm above earn revenue that is at least the second term on the right-hand side (again, with r_i chosen to maximize the ith inner sum). The better of these two reserve price vectors earns revenue at least $\frac{1}{2}$ times the maximum possible. □

3.3. Consequences for Batch Learning

Recall the batch learning problem mentioned in the Introduction: valuation profiles $\mathbf{v}^1, \ldots, \mathbf{v}^m$ are sampled i.i.d. from an unknown distribution F and given as a batch to a learning algorithm. The responsibility of the learning algorithm is to output non-anonymous reserve prices r that earn expected revenue (on a new i.i.d. draw from F) within ϵ of that obtained by the optimal non-anonymous reserve prices (for F). Morgenstern and Roughgarden [2015] show that, information-theoretically, $m = \Omega(\frac{H^2}{\epsilon^2} n \log n)$ samples are enough to in principle compute such reserve prices in matroid environments (with high probability over the samples, and where H is a bound on the maximum valuation). This guarantee is realized by the "empirical risk minimizing (ERM)" algorithm, which simply solves the offline optimization problem (using the samples as input) and returns the result. Morgenstern and Roughgarden [2015] did not address computational complexity issues in their work. Since the offline optimization problem is NP-hard to approximate to within some constant (Theorem 5.2 below), the ERM algorithm cannot be implemented in polynomial time (unless $P = NP$).

On the positive side, the same machinery (based on "uniform convergence") used to establish the guarantee of the ERM algorithm applies to approximation algorithms for the offline optimization problem. Thus our $\frac{1}{2}$-approximation algorithm for the MMR problem (Theorem 3.7) can be used as a black box to efficiently compute, from $m = \Omega(\frac{H^2}{\epsilon^2} n \log n)$ i.i.d. samples from an unknown distribution F, reserve prices that obtain expected revenue at least half that of the maximum possible (with high probability, and less ϵ).

4. ONLINE LEARNING ALGORITHMS

4.1. Regret-Minimization

This section considers the online no-regret learning problem mentioned in the Introduction. Here, valuation profiles $\mathbf{v}^1, \ldots, \mathbf{v}^T$ arrive one at a time (indexed by the same set of n bidders in a matroid environment), and at time t a learning algorithm chooses reserve prices $\mathbf{r}^t$ as a function of the previously seen profiles $\mathbf{v}^1, \ldots, \mathbf{v}^{t-1}$. We use $R(\mathbf{r}^t, \mathbf{v}^t)$ to denote the revenue earned by the VCG mechanism with reserve prices $\mathbf{r}^t$ on the valuation profile $\mathbf{v}^t$. The goal is to choose reserve prices over time so that the time-averaged revenue is almost as high as that achieved by the best fixed reserve prices in hindsight. Formally, the *regret* of a sequence $\mathbf{r}^1, \ldots, \mathbf{r}^T$ of reserve prices with respect to a sequence of valuation profiles $\mathbf{v}^1, \ldots, \mathbf{v}^T$ is

$$\max_{\mathbf{r}} \frac{1}{T} \sum_{t=1}^{T} R(\mathbf{r}, \mathbf{v}^t) - \frac{1}{T} \sum_{t=1}^{T} R(\mathbf{r}^t, \mathbf{v}^t),$$

and the goal is drive this quantity toward 0 as quickly as possible (as T grows large). Since regret corresponds to additive error, we normalize valuations to lie in the range $[0, 1]$.

We prove in Section 6 that this notion of regret is too optimistic. Our main positive result is an online learning algorithm with vanishing *α-regret*, a quantity defined by [Kakade et al. 2009] to introduce approximation into no-regret guarantees:

$$\alpha \cdot \max_{\mathbf{r}} \frac{1}{T} \sum_{t=1}^{T} R(\mathbf{r}, \mathbf{v}^t) - \frac{1}{T} \sum_{t=1}^{T} R(\mathbf{r}^t, \mathbf{v}^t).$$

4.2. An Online MMR Algorithm

We now work toward an algorithm for the online setting with good α-regret. A natural idea is to apply the existing machinery for translating offline α-approximation algorithms into online learning algorithms with vanishing α-regret. However, these techniques do not immediately apply in our setting. First, the number of actions (corresponding to reserve price vectors) is exponential in n, so we cannot separately track the past performance of each reserve price vector. The well-known "follow-the-perturbed-leader (FTPL)" algorithm of [Kalai and Vempala 2003] (based on the idea of [Hannan 1957]) can be used to translate certain α-approximation algorithms to online no-α-regret algorithms for arbitrary action spaces (subsets of $\mathbb{R}^n$), provided the payoff in each time step is a linear function of the chosen action. In our setting, the corresponding function $R(\mathbf{r}^t, \mathbf{v}^t)$ is not linear in $\mathbf{r}^t$ or $\mathbf{v}^t$. Nonetheless, we show how to combine the main ideas of the FTPL algorithm and of our offline approximation algorithm in Section 3 to obtain an online learning algorithm with vanishing $\frac{1}{2}$-regret.[10]

Our learning algorithm, defined for an arbitrary matroid environment, is given on the next page. It is straightforward to implement this algorithm so that it runs in $O(n\sqrt{T})$ at each time step, and $O(nT^{3/2})$ time overall. A key point is that each reserve price r_i^t is determined independently for each bidder i, enabling us to keep track of only Kn quantities (as opposed to a quantity for each of the K^n reserve price vectors).[11]

The analysis challenge is to show that our online algorithm earns revenue comparable to each term of the Revenue Decomposition Lemma (Lemma 3.6), despite having to produce reserve prices before seeing the valuation profile for which they will be used.

[10] This use of the FTPL algorithm in online auction design is reminiscent of [Blum and Hartline 2005]; the problem studied their corresponds to a single-bidder setting in our work.

[11] The algorithm is described for the case where the time horizon T is known in advance. Standard doubling arguments extend the algorithm and analysis to the case of an a priori unknown time horizon.

MMR Algorithm (Online)

```
set $K = \sqrt{T}$;
set $\epsilon = \sqrt{\log K/T}$;
for each round $t = 1, 2, \ldots, T$ do
    for each bidder $i = 1, 2, \ldots, n$ do
        let $S_i$ denote the previous rounds where $i$ is a winner in the VCG mechanism
          (without reserves);
        for $j \in S_i$, let $p_i^j$ denote $i$'s payment in the VCG mechanism (without reserves) with
          the profile $\mathbf{v}^j$;
        for each reserve price $r = 1/K, 2/K, \ldots, 1$ do
            draw a random variable $X_{i,r}$ from the standard exponential distribution (each
              $x \geq 0$ has probability density $e^{-x}$);
            choose $Y_{i,r}$ to be $+\frac{1}{\epsilon}X_{i,r}$ or $-\frac{1}{\epsilon}X_{i,r}$ uniformly at random;
        end
        Choose $r_i^t \in \{1/K, 2/K, \ldots, 1\}$ to maximize $Y_{i,r} + \sum_{j \in S_i} q^j(r_i^t)$, where $q^j(r_i^t)$ is
          $r_i^t - p_i^j$ if $r_i^t \in [p_i^j, v_i^j]$, and 0 otherwise;
    end
    return either $\mathbf{r}^t$ or all the-zero vector, each with $\frac{1}{2}$ probability;
end
```

THEOREM 4.1. *The* $\frac{1}{2}$*-regret of the online learning algorithm above is* $O(n\sqrt{\log T/T})$.

PROOF. Fix an input sequence $\mathbf{v}^1, \ldots, \mathbf{v}^T$ and let $\mathbf{r}^\star$ denote the optimal reserve prices. Let W_t denote the winners in the VCG mechanism with profile $\mathbf{v}^t$ and no reserves, and p_i^t the payment made by a bidder $i \in W_t$. Lemma 3.6 again implies, after summing over t, that the total revenue obtained by $\mathbf{r}^\star$ is at most

$$\sum_{t=1}^{T}\left(\sum_{i \in W_t} p_i^t + \sum_{i \in W_t} q^t(r_i^*)\right) = \sum_{t=1}^{T}\sum_{i \in W_t} p_i^t + \sum_{i=1}^{n}\sum_{t \in S_i} q^t(r_i^*),$$

where S_i and q^t are defined as in the algorithm description. Choosing the all-zero reserve prices every round obtains total revenue equal to the first term on the right-hand side. We will prove that choosing the $\mathbf{r}^t$ computed by the algorithm every round obtains total expected revenue equal to the second term on the right-hand side, less an error of $O(n\sqrt{T\log T})$.

The algorithm limits itself to reserve prices from the set $\{1/K, 2/K, \ldots, 1\}$ rather than $[0, 1]$. Then, for each bidder i, it tries to find a reserve r_i that maximizes $\sum_{t \in S_i} q^t(r_i)$, with perturbations as in the FTPL algorithm. [Kalai and Vempala 2003] showed that in an online decision-making setting with K actions (a.k.a. "experts"), each incurring a cost in $[0, 1]$ each round, the expected cost of FTPL algorithm (with weights $Y_{i,r}$ rechosen every round as in our algorithm) is at most the cost of the best fixed expert plus $\epsilon T + \frac{O(\log K)}{\epsilon}$. Our algorithm treats each possible reserve as an action and considers $1 - q^t$ to be the cost of choosing an expert in round t. Note that our algorithm runs the FTPL algorithm every round for bidder i, but S_i only increases when i turned out to be a winner in the VCG mechanism without reserves. When we only consider rounds where i is a winner in the VCG mechanism without reserves, the guarantee becomes:

$$\mathbb{E}\left[\sum_{t\in S_i} q^t(r_i^t)\right] \geq \sum_{t\in S_i} q^t(r_i^\star) - \epsilon T - \frac{O(\log K)}{\epsilon} \geq \sum_{t\in S_i} q^t(r_i^\star) - O(\sqrt{T\log T}),$$

with second inequality following from our choice of ϵ. However, the $\mathbf{r}^\star$ in this equation is limited to the set $\{1/K, 2/K, \ldots, 1\}$. This restriction costs at most $\frac{T}{K} = \sqrt{T}$ revenue (we chose K to balance this term with the regret of FTPL), which folds into the $O(\sqrt{T\log T})$ term.

Summing over all bidders i, choosing $\mathbf{r}^t$ every round gives expected revenue at least

$$\sum_{i=1}^{n}\sum_{t\in S_i} q^t(r_i^\star) - O(n\sqrt{T\log T}).$$

We conclude that randomly returning either $\mathbf{r}^t$ or the all-zero vector at each round t yields expected revenue at least $\frac{1}{2}$ times the maximum revenue achieved by any fixed set of reserve prices, less an error of $O(n\sqrt{T\log T})$. Hence our algorithm has $O(n\sqrt{\log T/T})$ (time-averaged) $\frac{1}{2}$-regret, as claimed. □

5. OFFLINE LOWER BOUND

In this section, we prove that the MMR problem is NP-hard to approximate within a $(1+\epsilon)$ factor for some constant $\epsilon > 0$, even in the special case of single-item environments. Section 6 extends this hardness result to the online no-regret learning problem.

Our proof approach is to provide an L-reduction, in the sense of Papadimitriou and Yannakakis [1988], from DOMINATINGSET on graphs of bounded degree to the MMR problem.

Definition 5.1 (*[Papadimitriou and Yannakakis 1988]*). Suppose Π and Π' are two maximization problems. We say Π *L-reduces* to Π' if there are poly-time algorithms f, g and constants $\alpha, \beta > 0$ such that given an instance I of Π:

(1) Algorithm f produces an instance I' of Π' such that $OPT(I') \leq \alpha OPT(I)$.
(2) Given a solution of I' with cost c', algorithm g produces a solution of I with cost c with $|c - OPT(I)| \leq \beta|c' - OPT(I')|$.

This definition is designed so that L-reductions compose and if Π L-reduces to Π', then a $(1+\epsilon)$-approximation for Π' yields a $(1+\alpha\beta\epsilon)$-approximation for Π. The DOMINATINGSET problem on graphs of bounded degree is hard to approximate to within some constant, so our L-reduction proves the same hardness result for the MMR problem.

THEOREM 5.2. DOMINATINGSET *on graphs of bounded degree B L-reduces to the* MMR *problem in single-item environments with $\alpha = 3B+2$ and $\beta = 1$.*

PROOF. Suppose we have a Dominating Set-B instance $G = (V, E)$. Suppose it has n nodes and the smallest dominating set uses k nodes. We produce an MMR instance with n bidders and $2n$ valuation profiles. The profiles are divided into two groups: n blue profiles and n red profiles. For convenience, we allow valuations in $[0,2]$ rather than $[0,1]$ (by scaling, it doesn't matter). The valuation of bidder i in the j^{th} blue profile will be 1 if $i = j$ or $(i,j) \in E$ and 0 otherwise. The valuation of bidder i in the j^{th} red profile will be 2 if $i = j$ and 0 otherwise.

For every bidder, the only valuations they ever have are 0, 1, and 2. Hence the only possible reserve choices for each bidder are 1 or 2 (we can always round up to the nearest valuation). We associate a reserve price of 1 with choosing the node to be in the dominating set and a reserve price of 2 with not choosing the node. In a blue profile, a bidder can only clear a reserve of 1. In a red profile, the relevant bidder faces no competition, and hence a second-price auction with research only extracts the revenue from them by choosing the reserve of 2. The blue profiles encode the dominating set constraint, while the red profiles encode the objective to minimize nodes chosen.

Suppose there is a dominating set of size k. Choosing the associated reserve prices, as described above, achieves a revenue of $3n - k$: we achieve a revenue of 1 in the blue profiles since we cover them all by definition, a revenue of 1 from each red profile corresponding to one of the k chosen nodes, and a revenue of 2 from each of the other $n - k$ red profiles.

Since every node in the original problem covered at most $B + 1$ nodes, the original optimum was at least $n/(B + 1)$. Combining this bound with our argument above, the optimal revenue is at most $\frac{3B+2}{B+1}n$. Hence we can choose $\alpha = 3B + 2$.

Next, we show that if there are non-anonymous reserve prices that get within c of the new optimum, then we can produce a dominating set that gets within c of the old optimum (with a polynomial-time algorithm). Suppose we achieve a revenue of $3n - k$ for some k. We claim that we can assume that we achieve a revenue of 1 in every blue profile. For if we do not, then consider any bidder who has a valuation of 1 in that profile. Changing their reserve to 1 nets an additional 1 revenue in this profile, and loses at most 1 revenue (from their red profile). Hence in polynomial time, we can transform a solution to one that only has more revenue and achieves a revenue of 1 every blue profile. But then our association of nodes with reserve prices gives us a dominating set of size at most k. This completes the proof with $\beta = 1$. □

Tight approximation bounds are not known for bounded-degree dominating set and related problems, and as a result there is a gap between our lower bound in Theorem 5.2 and our $\frac{1}{2}$-approximation algorithm. Using the state-of-the-art result of [Chlebik and Chlebikova 2008] (with $B = 5$), we get that the MMR problem cannot be approximated better than a factor of $\frac{884}{885}$. This inapproximability result holds for generally for estimating the value of an optimal solution to the MMR problem; this will be useful in the next section.

COROLLARY 5.3. *It is NP-hard to distinguish between single-item* MMR *instances in which the maximum-possible revenue is at least a given parameter X, and those instances in which the maximum-possible revenue is at most $\frac{884}{885}X$, even when $X \geq \frac{m}{2}$.*

6. ONLINE LOWER BOUNDS

6.1. Online MMR Lower Bound

Our hardness result for offline optimization (Theorem 5.2) translates to an analogous hardness result for online learning. This is not immediately obvious because an online learning algorithm could conceivably achieve high revenue with different reserves at different times, while never figuring out the best fixed reserves in hindsight. Our randomized polynomial-time reduction is the following (where $\alpha \in (\frac{884}{885}, 1]$):

MMR Reduction (Offline to Online)

```
Input: valuation profiles $\mathbf{v}^1, \ldots, \mathbf{v}^m$ and target $X$;
let $A$ be the online algorithm and $f(T)$ its $\alpha$-regret after $T$ rounds;
choose the smallest $T$ such that $f(T)$ is at most $\frac{1}{4} \cdot (\alpha - \frac{884}{885})$;
for rounds $t = 1, 2, \ldots, T$ do
    ask $A$ for its next choice of reserves $\mathbf{r}^t$ and then give it a uniformly random valuation
      profile from $\{\mathbf{v}^1, \ldots, \mathbf{v}^m\}$;
end
return YES if at least one of the $\mathbf{r}^t$ results in strictly more than $\frac{884}{885}X$ total revenue on
$\{\mathbf{v}^1, \ldots, \mathbf{v}^m\}$, and NO otherwise;
```

We note that whenever the resulting offline algorithm says “yes” it must be correct — it finds a reserve price vector witnessing the lower bound on the optimal objective function value. Thus the algorithm has one-sided error (with false negatives only).

For the purposes of the following theorem, we say that an online learning algorithm has *no α-regret* if for every fixed constant $\epsilon > 0$, the number of rounds needed to drive the α-regret down to ϵ is bounded by a polynomial in the number of bidders n. For example, any regret bound of the form $poly(n)/T^{\delta}$ (for any fixed $\delta > 0$) satisfies this condition.

THEOREM 6.1. *For all constants $\alpha > \frac{884}{885}$, there is no polynomial-time algorithm for the online* MMR *problem with no α-regret, unless $NP \subseteq RP$.*

PROOF. We show that if the conjectured online algorithm exists, then the reduction above distinguishes between single-item MMR instances in which the maximum-possible revenue is at least a given parameter X, and those instances in which the maximum-possible revenue is at most $\frac{884}{885}X$, where X is at least $\frac{m}{2}$ (with constant one-sided error). By Corollary 5.3, this would imply $NP \subseteq RP$.

For the analysis, our proof begins with algorithm A, which has $f(T)$ α-regret, and gradually transforms it to the same algorithm produced by our reduction. Recall that the α-regret bound means that

$$\alpha \frac{1}{T} \sum_{t=1}^{T} R(\mathbf{r}^{\star}, \mathbf{v}^t) - \mathbb{E}\left[\frac{1}{T} \sum_{t=1}^{T} R(\mathbf{r}^t, \mathbf{v}^t)\right] \leq f(T),$$

for every sequence $\mathbf{v}^1, \ldots, \mathbf{v}^T$, where the expectation is over the coin flips of A.

Suppose we now change the setup as follows. For simplicity, assume that T is a multiple of m. Define the *m-periodic regret* as the regret if only rounds 1, $m+1$, $2m+1$, etc. contribute to the regret computation. (The best fixed auction in hindsight is still computed with respect to the valuation profiles at all rounds, not just this subset of rounds.)

Derive algorithm A' for the new setup from A as follows: in rounds 1, $m+1, \ldots,$ A' simulates A and also feeds it the new valuation profile. On other rounds A' behaves arbitrarily and does not feed anything into algorithm A. We know A has $f(T/m)$ α-regret with respect to the sequence it is given. But the best fixed auction for this subsequence is only better than the best fixed auction for the entire sequence. Hence A' has $f(T/m)$ m-periodic α regret with respect to the original sequence. That is:

$$\alpha \frac{m}{T} \sum_{t=1}^{T/m} R(\mathbf{r}^{mt-m+1}, \mathbf{v}^{mt-m+1}) - \mathbb{E}\left[\frac{m}{T} \sum_{t=1}^{T/m} R(\mathbf{r}^{\star}, \mathbf{v}^{mt-m+1})\right] \leq f(T/m).$$

Next, we define an online algorithm A'' that uses the same reserves $\mathbf{r}^t$ for m rounds in a row before changing, and bound its α-regret. At a round $1, m+1, \ldots$, A'' simulates A'. Whatever auction A' recommends, A'' plays this auction for the next m rounds. During these rounds, A'' receives m valuation profiles. After these rounds finish, A'' feeds these m valuation profiles into A', one at a time in a random order (uniform over all $m!$ possible orderings).

We now bound the α-regret of A''. Fix a sequence of valuations $\mathbf{v}^1, \ldots, \mathbf{v}^T$ that an adversary chooses for A'', with the best fixed reserves $\mathbf{r}^\star$. Notice that A'' will give these same valuations to A', but possibly out of order. However, reordering *does not affect the best fixed reserves*. Furthermore, the m-periodic α-regret guarantee of A' holds with respect to any input sequence, and in particular any reordering of our input sequence.

Consider a block of valuations, $\mathbf{v}^{(k-1)m+1}, \ldots, \mathbf{v}^{km}$ that A'' must choose the same reserves for. We can express the expected average difference between the revenue of A'' and α times the revenue of $\mathbf{r}^\star$ on the entire block as follows:

$$\alpha \frac{1}{m} \sum_{t=(k-1)m+1}^{km} R(\mathbf{r}^\star, \mathbf{v}^t) - \mathbb{E}\left[\frac{1}{m} \sum_{t=(k-1)m+1}^{km} R(\mathbf{r}^{(k-1)m+1}, \mathbf{v}^t)\right],$$

where $\mathbf{r}^{(k-1)m+1}$ is the auction recommended by A' at the beginning of the block and the expectation is over any internal coin flips of the algorithm (this quantity is independent of the ordering of the profiles in the block). We can express the expected difference between the revenue of A' and α times the revenue of $\mathbf{r}^\star$ on the very first valuation of this block as follows:

$$\alpha \sum_{t=(k-1)m+1}^{km} Pr[\mathbf{v}^t \text{ first}] \cdot R(\mathbf{r}^\star, \mathbf{v}^t) - \mathbb{E}\left[\sum_{t=(k-1)m+1}^{km} Pr[\mathbf{v}^t \text{ first}] R(\mathbf{r}^{(k-1)m+1}, \mathbf{v}^t)\right],$$

where the probabilities are over the random ordering and the expectation over the coin flips of the algorithm. Because the probability that any particular $\mathbf{v}^t$ comes first in this block is $\frac{1}{m}$, these two expressions are equal. This allows us to relate the m-periodic α-regret of A' with the α-regret of A''. Summing the first expression over all blocks and averaging by the number of blocks yields the α-regret of A''. Summing the second expression over all blocks and averaging by the number of blocks yields the m-periodic α-regret of A', averaged over all orderings. Since (m-periodic) regret is defined with respect to worst-case inputs, this means the α-regret of A'' is $f(T/m)$.

Unraveling the definitions of A'' and A', A'' really just samples one valuation profile from every block of m and runs A, which is the same as the algorithm produced by our reduction. The reduction algorithm above, in effect, uses A'' to approximate the offline problem, giving it T blocks, where each block is a copy of the m valuation profiles in its offline input, for a total of Tm rounds given to A'', which will simulate A for T rounds. By our assumption on the regret guarantee, T will be polynomial in n. By our choice of T, from the guarantee for A'', we know:

$$\alpha \frac{1}{m} \underbrace{\frac{1}{T} \sum_{t=1}^{mT} R(\mathbf{r}^\star, \mathbf{v}^t)}_{OPT} - \mathbb{E}\left[\frac{1}{mT} \sum_{t=1}^{mT} R(\mathbf{r}^t, \mathbf{v}^t)\right] \leq \tfrac{1}{4} \cdot (\alpha - \tfrac{884}{885}),$$

where $\mathbf{r}^t$ is the action played by A'' at time t and the expectation is over the coin flips of A'', and hence

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{m} R(\mathbf{r}^t, \mathbf{v}^j)\right] \geq \alpha OPT - \tfrac{m}{4}\cdot(\alpha - \tfrac{884}{885}).$$

When we have a YES instance, OPT is at least X. This implies OPT is also at least $\frac{m}{2}$. The second term on the right-hand side is then at most $\frac{1}{2}(\alpha - \frac{884}{885})OPT$, so the right-hand side is at least $\frac{\alpha+884/885}{2}OPT$.

However, no reserves $\mathbf{r}^t$ can get more total revenue on all valuation profiles than OPT. The left-hand side is the expected (over the randomness of A'') average (over the outputs of A'' to A') total revenue of A'' on all valuation profiles. The actual average total revenue lies between 0 and OPT, but we have shown its expectation is at least $\frac{\alpha+884/885}{2}OPT$. By Markov's inequality, this means there is a constant probability that the average total revenue is better than $\frac{884}{885}OPT$, and hence better than $\frac{884}{885}X$. In this event, there must be at least one $\mathbf{r}^t$ which results in strictly more than $\frac{884}{885}X$ total revenue on the valuation profiles. The constant success probability only depends on α, which was assumed to be a constant. (And since there is one-sided error, this correctness probability can be amplified to a constant arbitrarily close to 1 by repetitions.)

On the other hand, suppose we have a NO instance. This means that OPT is at most $\frac{884}{885}X$, and hence our algorithm cannot find reserves $\mathbf{r}^t$ which result in more than $\frac{884}{885}X$ total revenue. This completes the correctness analysis. □

6.2. General Offline to Online Inapproximability Reduction

The reduction of Theorem 6.1 can generalized to a wide class of online problems. Suppose we have an online problem where every round, the algorithm commits to an action and then receives a payoff vector with description length n which can be used to compute the payoff of any action in $poly(n)$ time. Payoffs lie in the range $[0, 1]$. The corresponding offline problem is to compute the best fixed action given m (succinct descriptions of) payoff vectors and a probability for each payoff vector occurring. Note that this is a different formulation than our offline problem, which corresponds to the uniform distribution over the m payoff vectors, and with the answer scaled by a factor of m. In this formulation, the answer to the problem also lies in the range $[0, 1]$. We show that if this offline problem cannot be approximated, then neither can the online problem:

THEOREM 6.2. *Suppose it is NP-hard to distinguish between offline instances in which the maximum payoff is at least a given parameter X, and those instances in which the maximum payoff is at most αX, even when $X \geq \beta$ for some constants $\alpha, \beta > 0$. Then unless $NP \subseteq RP$, for all constants $\alpha' > \alpha$ there is no polynomial-time algorithm for the online problem with no α'-regret.*

PROOF. The proof of Theorem 6.1 did not rely on any specific facts about the MMR problem. We can still transform any online algorithm A with α'-regret $f(T)$ into A' with m-periodic α'-regret $f(T/m)$. However, we need to be more detailed with our transformation from A' to A'' to account for the fact that we are no longer using a uniform distribution. We plan to again produce an A'' which must output the same action for m rounds in a row; these m rounds constitute a block. We augment the problem so that the input for each round t in a block comes with a probability p_t, and the probabilities for a round sum to 1. These probabilities indicate how much each round is weighted; the (time-averaged) α-regret of a weighted online algorithm over a block of valuations

$\mathbf{v}^{(k-1)m+1}, \ldots, \mathbf{v}^{km}$ is:

$$\alpha \sum_{t=(k-1)m+1}^{km} p_t R(\mathbf{r}^\star, \mathbf{v}^t) - \mathbb{E}\left[\sum_{t=(k-1)m+1}^{km} p_t R(\mathbf{r}^{(k-1)m+1}, \mathbf{v}^t)\right],$$

where $\mathbf{r}^{(k-1)m+1}$ is the action chosen for the entire block. Our A'' is as before, but instead of choosing an block ordering uniformly at random, we choose round t to be the first in the block with probability p_t. Again reordering does not affect the best fixed reserves. However, under this reordering and the definition of m-periodic regret for A', we get that the α'-regret of A'' is $f(T/m)$.

Choosing the smallest T (the number of blocks or equivalently, random samples) such that $f(T)$ is at most $\frac{1}{2}\beta(\alpha' - \alpha)$ requires T to only be $poly(n)$. This choice of T guarantees the expected average payoff of an output of A'' is at least $\alpha' \cdot OPT$ minus $\frac{1}{2}\beta m(\alpha' - \alpha)$. For YES instances, this shows the expected average total payoff is at least $\frac{\alpha'+\alpha}{2}X$, so with constant probability some action returned by A'' will have total payoff more than αX. For NO instances, all actions have total payoff at most αX. This RP algorithm would solve an NP-hard problem, contradicting our hypothesis. Hence no such online algorithm can exist. □

REFERENCES

S. Alaei, J. D. Hartline, R. Niazadeh, E. Pountourakis, and Y. Yuan. 2015. Optimal Auction vs. Anonymous Pricing. In *Proceedings of the 56th Annual Symposium on Foundations of Computer Science (FOCS)*. 1446–1463.

Martin Anthony and Peter L. Bartlett. 1999. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, NY, NY, USA.

S. Athey and D. Nekipelov. 2012. A Structural Model of Sponsored Search Advertising Auctions. (2012). Working paper.

Avrim Blum and Jason D Hartline. 2005. Near-optimal online auctions. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 1156–1163.

Nicolò Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. 2013. Regret Minimization for Reserve Prices in Second-Price Auctions.. In *SODA*. SIAM, NY, NY, USA., 1190–1204.

N. Cesa-Bianchi and G. Lugosi. 2006. *Prediction, Learning, and Games*. Cambridge University Press.

S. Chawla, H. Fu, and A. R. Karlin. 2014. Approximate revenue maximization in interdependent value settings. In *Proceedings of the 15th ACM Conference on Electronic Commerce (EC)*. 277–294.

Shuchi Chawla, Jason D. Hartline, and Robert Kleinberg. 2007. Algorithmic Pricing via Virtual Valuations. In *Proceedings of the 8th ACM Conference on Electronic Commerce (EC '07)*. 243–251.

Miroslav Chlebik and Janka Chlebikova. 2008. Approximation hardness of dominating set problems in bounded degree graphs. *Information and Computation* 206, 11 (2008), 1264–1275.

Richard Cole and Tim Roughgarden. 2014. The Sample Complexity of Revenue Maximization. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*. 243–252.

Peter Cramton, Hector Lopez, David Malec, and Pacharasut Sujarittanonta. 2015. Design of the Reverse Auction in the Broadcast Incentive Auction. (2015). An expert report in response to Comment Public Notice FCC 14 -191.

Constantinos Daskalakis and Vasilis Syrgkanis. 2016. Learning in Auctions: Regret is Hard, Envy is Easy. In *Proceedings of the 57th Annual Symposium on Foundations of Computer Science (FOCS)*.

Nikhil R. Devanur, Zhiyi Huang, and Christos-Alexandros Psomas. 2016. The Sample Complexity of Auctions with Side Information. (2016). To appear in *STOC '16*.

P. Dhangwatnotai, T. Roughgarden, and Q. Yan. 2010. Revenue Maximization with a Single Sample. In *Proceedings of the 11th ACM Conference on Electronic Commerce (EC)*. 129–138.

James Hannan. 1957. Approximation to Bayes risk in repeated play. *Contributions to the Theory of Games* 3, 97-139 (1957), 2.

J. D. Hartline and T. Roughgarden. 2009. Simple versus Optimal Mechanisms. In *Proceedings of the 10th ACM Conference on Electronic Commerce (EC)*. 225–234.

Zhiyi Huang, Yishay Mansour, and Tim Roughgarden. 2015. Making the Most of your Samples. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation (EC '15)*. ACM, New York, NY, USA, forthcoming.

Sham M. Kakade, Adam Tauman Kalai, and Katrina Ligett. 2009. Playing Games with Approximation Algorithms. *SIAM J. Comput.* 39, 3 (2009), 1088–1106.

Adam Kalai and Santosh Vempala. 2003. Efficient algorithms for online decision problems. In *Learning Theory and Kernel Machines*. Springer, 26–40.

Robert D. Kleinberg and Frank Thomson Leighton. 2003. The Value of Knowing a Demand Curve: Bounds on Regret for Online Posted-Price Auctions.. In *FOCS*. IEEE Computer Society, NY, NY, USA., 594–605.

Andres Munoz Medina and Mehryar Mohri. 2014. Learning Theory and Algorithms for revenue optimization in second price auctions with reserve. In *Proceedings of The 31st Intl. Conf. on Machine Learning*. 262–270.

Jamie H Morgenstern and Tim Roughgarden. 2015. On the Pseudo-Dimension of Nearly Optimal Auctions. In *Advances in Neural Information Processing Systems*. 136–144.

Roger B. Myerson. 1981. Optimal auction design. *Mathematics of operations research* 6, 1 (1981), 58–73.

R. Paes Leme, M. Pál, and S. Vassilvitskii. 2016. A Field Guide to Personalized Reserve Prices. (2016). To appear in *WWW '16*.

Christos Papadimitriou and Mihalis Yannakakis. 1988. Optimization, approximation, and complexity classes. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*. ACM, 229–234.

Amir Ronen. 2001. On approximating optimal auctions. In *Proceedings of the 3rd ACM conference on Electronic Commerce*. ACM, 11–17.

Alexander Schrijver. 2003. *Combinatorial optimization: polyhedra and efficiency*. Vol. 24. Springer Science & Business Media.

Kunal Talwar. 2003. The price of truth: Frugality in truthful mechanisms. In *STACS 2003*. Springer, 608–619.