Mechanism Design via Machine Learning

Maria-Florina Balcan* **Avrim Blum*** **Jason D. Hartline†**
Yishay Mansour‡

May 2005
CMU-CS-05-143

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

* School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. {ninamf,avrim}@cs.cmu.edu.
† Microsoft Research, Mountain View, CA. hartline@microsoft.com.
‡ School of Computer Science, Tel-Aviv University. mansour@cs.tau.ac.il.

Research supported in part by NSF grants CCR-0105488, CCR-0122581, and IIS-0121678, as well as by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778, by a grant no. 1079/04 from the Israel Science Foundation, by a grant from BSF and an IBM faculty award. This publication reflects only the authors' views.

Keywords: Mechanism Design, Machine Learning, Sample Complexity, Profit Maximization, Unlimited Supply, Digital Good Auction, Attribute Auctions, Combinatorial Auctions.

Abstract

We use techniques from sample-complexity in machine learning to reduce problems of incentive-compatible mechanism design to standard algorithmic questions, for a broad class of revenue-maximizing pricing problems. Our reductions imply that for these problems, given an optimal (or β-approximation) algorithm for the standard algorithmic problem, we can convert it into a $(1+\epsilon)$-approximation (or $\beta(1+\epsilon)$-approximation) for the incentive-compatible mechanism design problem, so long as the number of bidders is sufficiently large as a function of an appropriate measure of complexity of the comparison class of solutions. We apply these results to the problem of auctioning a digital good, to the *attribute auction* problem which includes a wide variety of discriminatory pricing problems, and to the problem of item-pricing in unlimited-supply combinatorial auctions. From a machine learning perspective, these settings present several challenges: in particular, the loss function is discontinuous and asymmetric, and the range of bidders' valuations may be large.

1 Introduction

In recent years there has been substantial work on problems of algorithmic mechanism design. These problems typically take a form similar to classic algorithm design (or approximation-algorithm) questions, except that the inputs are each given by different agents who have their own interest in the outcome of the computation. Thus, the algorithms produced must be *incentive-compatible* — meaning that it is in each agent's best interest to report its true value — which greatly complicates the algorithm design problem.

We consider the design of revenue-maximizing pricing mechanisms in such a game theoretic setting where the consumers (a.k.a., agents or bidders) may choose to falsely report their preferences if it might benefit them. For example, we might be aiming to sell a digital good to consumers using a scheme that charges different prices depending on public attributes of bidders such as their geographical location, and wish to do so in a way that makes as much profit as we can. Our goal will be to produce incentive-compatible mechanisms that achieve revenue close to the optimal revenue possible from pricing functions in a given class had incentive-compatibility not been an issue. That is, we want to reduce the problem of incentive-compatible mechanism design in this setting to the standard algorithmic problem of optimizing over a given class of functions.

Our main contribution in this work is to use sample-complexity techniques in machine learning theory (see [2, 8, 25, 30]) to perform this type of reduction. When the number of agents is sufficiently large as a function of the complexity of the pricing functions being compared to, this reduction loses only a $(1 + \epsilon)$-factor in solution quality; that is, an algorithm (or β-approximation) for the standard algorithmic problem can be converted to a $(1 + \epsilon)$-approximation (or $\beta(1 + \epsilon)$-approximation) for the incentive-compatible mechanism design problem. We do this in a fairly general setting that includes the following as special cases:

Auction of digital goods to indistinguishable bidders. In this problem, studied in [21, 14], we have a digital good (a good of unlimited supply with zero marginal cost) and n bidders, where each bidder i has some valuation v_i between 1 and h. Our goal is to sell our good so as to make profit comparable to the best single price: the price p maximizing $p \times |\{i : v_i \geq p\}|$.

For this problem, Goldberg et al. [21] give a simple auction based on random sampling and show that it gives near 6-approximation so long as the optimal revenue is large compared to h.[1] We analyze a slight variant and show (Theorem 6) that it is a $(1 + \epsilon)$-approximation so long as the optimal revenue is large compared to $\frac{h}{\epsilon^2} \log(1/\epsilon)$.

Attribute Auctions. In many generalizations of the digital-good auction, the bidders are not a priori indistinguishable; instead, publicly known information about bidders may allow differential treatment. For example, the motion picture industry uses region encodings so that they can charge different prices for DVDs sold in different markets. In such a setting, we might hope to obtain more profit than is possible from a single sale price.

This introduces the natural question of how to use the distinguishing features of consumers to price-discriminate to the maximum benefit of the seller. We consider the following abstraction of these situations. In an *attribute auction*, the bidders are not indistinguishable but instead have a set of publicly-known *attributes* and the goal is to achieve revenue comparable to the best pricing function over these attributes from some available class $\mathcal{G}$ of pricing functions. For example, [6] considers the

[1]This problem has also been considered in a framework where the auction's performance is compared to the profit obtained from the optimal sale price that results in a sale of *at least two* items [14]. In this context the best known auction is $13/4$-competitive [24].

special case of 1-dimensional attributes and a comparison class $\mathcal{G}$ of piece-wise constant functions that divide the attribute space into contiguous regions (a.k.a., markets) and charge a single price in each.[2] Other natural classes $\mathcal{G}$ include linear or piece-wise linear functions over attributes. We give bounds for this setting more generally, including a generalization of the class of functions considered in [6] to higher dimensions.

Attribute-auctions are a fairly general setting that can model a number of problems including *multicast pricing* [14]. In this problem, each bidder resides at some node of a tree, and in order to sell its service to some bidder, the service-provider must have purchased all edges on the path from the root to that vertex. If we view each bidder's location as its public attribute, then this is a form of attribute-auction but with the additional complication that each proposed solution has some associated cost as well. [14] gives a 4-approximation to this problem, under the assumption that the optimal solution has revenue at least 4 times its cost and that there is sufficient competition at each node. Our reduction implies that if the optimal solution is even better: has revenue $O(1/\epsilon)$ times its cost and furthermore the average number of bidders at any node is $\tilde{O}(h/\epsilon^2)$, then we get a $(1+\epsilon)$-approximation. Moreover, using a natural form of structural risk minimization (SRM), we can achieve performance comparable to the best "simple" tree even in settings where the results of [14] do not hold.[3]

Item-pricing in combinatorial auctions. This problem is a different generalization of the first problem above, and studied in [16, 29]. The setting here is we have m different items, each in unlimited supply (like a supermarket), and bidders have valuations on *subsets* of items. Our goal is to achieve revenue nearly as large as the best sale that uses item prices (assigns a separate price to each item), a natural comparison class. Our results imply that $\tilde{O}(hm^2/\epsilon^2)$ bidders are sufficient to achieve revenue close to the optimal item-pricing (assuming the algorithmic problem can be solved for the given bidders), no matter how complicated those bidders' valuations are. In the unit-demand case, when each bidder wants at most one item (such as in pricing different versions of the same software or pricing airline tickets), our bounds give a $(1+\epsilon)$-approximation when the optimal revenue is large compared to $\tilde{O}(hm/\epsilon^2)$ which improves by roughly a factor of m over the results of [16].

A special case of this setting is the problem of auctioning the right to traverse paths in a network. In the case that the network is a tree and each user wants to reach the root (like drivers commuting into a city), then [29] give an exact algorithm for the algorithmic problem. Our reduction then yields a $(1+\epsilon)$-approximation so long as the number of bidders is sufficiently large.

The basic reduction we apply to solve these auction problems is as follows. Given an algorithm $\mathcal{A}$ (exact or approximate) for the non-incentive-compatible pricing problem (finding the optimal pricing function in class $\mathcal{G}$ for a given set of bidders) and given a set of bidders S, we will split bidders randomly into two sets S_1 and S_2, run the algorithm separately on each set (perhaps adding an additional penalty term to the objective to penalize solutions that are too "complex" according to some measure), and then apply the solution found on S_1 to S_2 and the solution found on S_2 to S_1. Sample-complexity techniques from machine learning theory can then give a guarantee on the quality of the results if the number of bidders is sufficiently large compared to (an appropriate measure of) the complexity of the class of possible solutions. From an economics perspective, this can be viewed as replacing the assumption that bidders come from a known distribution with the use of learning, over a random subsample S_i of an arbitrary set of bidders S, to get

[2]This is natural when attribute values are correlated with a willingness to pay.

[3]For example, consider an n-leaf tree of depth 1 where each leaf contains one bidder with value 1 and one with value h. Then the nodes themselves do not have sufficient competition for the results of [14] to hold, but by applying SRM our method can view the entire set as one market and achieve revenue nearly nh.

enough information about the set to apply to S_{2-i}. From a learning perspective, however, the mechanism-design setting presents a number of technical challenges: in particular, the loss function is discontinuous and asymmetric, and the range of bid values may be large.

In addition to the generic reduction, we also give specific analyses for several of the above problems, using their structure to yield better bounds on the number of bidders needed to achieve a desired approximation factor.

The form of the solutions: The reader will notice that in converting an algorithm (or approximation algorithm) for finding the best pricing function in $\mathcal{G}$ into an incentive-compatible mechanism, we produce a mechanism that does not belong to the class $\mathcal{G}$ itself. For example, even in the simplest case of auctioning a digital good to indistinguishable bidders, we compare performance to the best single sales price, and yet the auction itself does not in fact offer each bidder the same price (all bidders in S_1 get the same price, and all bidders in S_2 get the same price, but those two prices may be different). In fact, Goldberg and Hartline [17] show that this sort of behavior is necessary: it is not possible for an incentive-compatible auction to approximately maximize profit and offer all the bidders the same price.

In the context of market analysis, one can interpret our bounds (on the number of bidders needed for the basic mechanism described above to work well) as bounds on the number of customers one would need to query in order to get enough information about the market to produce a nearly-optimal pricing function in class $\mathcal{G}$.

Related work: Several papers [6, 7] have applied machine learning techniques to mechanism design in the context of online auctions. The online setting is more difficult than the "batch" setting we consider, but the flip-side is that as a result, that work only applies to quite simple mechanism design settings where the class $\mathcal{G}$ of comparison functions has small size and can be easily listed.

Structure of this paper: We begin by defining our general setting (Section 2) and giving our generic reductions (Section 3). We then proceed to give a tighter analysis for the basic auction of a digital good (Section 4) and describe in Section 5 how the complexity measures of Section 3 can be instantiated for the case of attribute auctions. We consider item-pricing in combinatorial auctions in Section 6 and the multicast pricing problem in Section 7. We give our conclusions and some open research directions in Section 8.

2 Definitions

We will be considering mechanism design problems of the following general type. We have a set S of n bidders, and we assume that each bidder i has some private information $priv_i$ (like how much they are willing to pay for a digital good), as well as public information pub_i (such as their location in a network). The game itself will be defined by an abstract space of legal *offers* (like an offer to sell a good at \$17) together with a mapping ρ that defines how much profit a given offer yields from a given bidder. For example, in the case of auctioning a digital good, $\rho(\text{"offer \$17"}, priv_i) = 17$ if $priv_i \geq 17$ and 0 otherwise. We can think of ρ as defining the assumption about how bidders behave as a function of their private values. The standard assumption in incentive compatible mechanism design is that bidders prefer the outcome that maximizes their *utility*, defined as the difference between their *valuation* for the outcome (as specified by their preferences) and the payment they are required to make. We will assume that ρ is defined to model this behavior; that is, for any *fixed* offer, a bidder's utility is maximized when plugging his true private information into ρ. We now introduce the notion of a comparison class of pricing functions.

Definition 1 *A* comparison class, $\mathcal{G}$, *of pricing functions is a set of functions* g *that map the public information of a bidder to an offer. The* profit *of a function* g *is* $\sum_i \rho(g(pub_i), priv_i)$. *Note that we are implicitly considering only* unlimited supply *mechanism design problems, because the profit from bidder* i *does not depend on whether* g *received profit from other bidders.*

Given a comparison class, $\mathcal{G}$, the *algorithm design* problem is: given both the public and private information in S, find the $g \in \mathcal{G}$ of highest total profit $\mathrm{OPT}_{\mathcal{G}}$. Some of the problems we consider will also have costs for various functions g: for instance, in multicast pricing, a comparison function g consists of both a tree and a proposed price at each node, and its cost is the cost of the tree. In this case, we should think of ρ as a **revenue** function, and the algorithm design problem will be to find the g of highest revenue minus cost. In our reductions, we may also want to perform "structural risk minimization", which adds additional fake penalties to different functions g based on some measure of their complexity, in which case we will need to assume we have an algorithm that optimizes revenue minus penalty.

We now need to define what we mean by an incentive compatible mechanism. An incentive-compatible mechanism is a function that takes in the public information of all the bidders, plus the private information of all bidders *except* the given bidder i and outputs an offer offer_i. The profit of this mechanism is then $\sum_i \rho(\mathit{offer}_i, priv_i)$. Our goal will be to design such a mechanism whose total profit is nearly as large as the profit of the best function in comparison class $\mathcal{G}$. Note that typically our mechanisms will not actually belong to $\mathcal{G}$, such as offering one price to some subset of bidders and another price to another even if our class $\mathcal{G}$ is the set of all single price functions.

One final point at this level of generality: we will assume that we are given an upper bound h on the value of ρ; that is, no individual bidder can influence profit by more than h. This term will come into our sample-complexity bounds.

2.1 Examples

Auction of digital goods to indistinguishable bidders. As described in the introduction, in this setting the bidders have no public information (equivalently, all the bidders have the *same* public information *pub*) and the private information of bidder i is exactly its valuation v_i for the digital good, which is a real number between 1 and h. Here, a natural comparison class $\mathcal{G} = \{g_p\}$ is the class of all functions that offer a single price p, and ρ is a function defined by $\rho(p, priv_i) = p$ if $p \leq priv_i$ and $\rho(p, priv_i) = 0$ otherwise.

Attribute Auctions. This is the same as the setting above except now each bidder i is associated a public **attribute** $pub_i \in \mathcal{X}$ where $\mathcal{X}$ is the **attribute space**. We view $\mathcal{X}$ as an abstract space, but one can envision it as $\mathbb{R}^d$, for example. $\mathcal{G}$ is then a class of pricing functions from $\mathcal{X}$ to $\mathbb{R}_+$, such as all linear functions or all functions that partition $\mathcal{X}$ into k markets (say based on distance to k cluster centers) and offer a different price in each. The mapping ρ is a function from $\mathbb{R}_+ \times [1, h]$ to $[0, h]$ defined (as in the case of indistinguishable bidders) by $\rho(p, priv_i) = p$ if $p \leq priv_i$ and $\rho(p, priv_i) = 0$ otherwise. We will give analysis for several interesting classes of comparison functions in Section 5.

Combinatorial Auctions. Here we have a set J of m distinct items, each in unlimited supply. Each consumer has a private valuation $v_i(s)$ for each bundle $s \subseteq J$ of items, which measures how much receiving bundle s would be worth to the consumer i. The private information of bidder i can be described by a vector of all its valuations on subsets of J (for simplicity, we assume bidders are indistinguishable, i.e., no public information). A natural class of comparison functions $\mathcal{G}$ (studied in [29]) is the class of functions that assign a separate price to each item[4], such that the price of a bundle is just the sum of the prices of the items in it (called item-pricing). The mapping ρ is then defined by assuming bidders will buy the bundle (if any) with largest positive gap between its value to them and its total cost.[5]

3 Generic Reductions

We are interested in reducing incentive-compatible mechanism design to the standard algorithm design problem. Our reductions will be based on random sampling. Let $\mathcal{A}$ be an algorithm for the (non incentive-compatible) problem of optimizing over $\mathcal{G}$. The simplest mechanism that we consider, which we call $\text{RSOPF}_{(\mathcal{G},\mathcal{A})}$ (Random Sampling Optimal Pricing Function), is the following generalization of the random sampling digital-goods auction from [21]:

1. Randomly split the bidders into two groups S_1 and S_2, flipping a fair coin for each bidder.
2. Run $\mathcal{A}$ to determine the best (or approximately best) function $g_1 \in \mathcal{G}$ over S_1, and similarly the best (or approximately best) $g_2 \in \mathcal{G}$ over S_2.
3. Finally, apply g_1 to all bidders in S_2 and g_2 to all bidders in S_1.

We will also consider various more refined versions of $\text{RSOPF}_{(\mathcal{G},\mathcal{A})}$, that discretize $\mathcal{G}$ or perform some type of *structural risk minimization* (in which case we will need to assume $\mathcal{A}$ can optimize over the modifications made to $\mathcal{G}$).

3.1 The Basic Analysis

In order to simplify notation, for a given setting (defined by ρ and $\mathcal{G}$), for a pricing function g and bidder i define $g(i)$ to be the profit made by g on i; i.e., $g(i) = \rho(g(pub_i), priv_i)$. Similarly, for a set of bidders $S' \subseteq S$, let $g(S') = \sum_{i \in S'} g(i)$. So, $\text{OPT}_{\mathcal{G}} = \max_{g \in \mathcal{G}} g(S)$. If $g_1(i) = g_2(i)$ for all $i \in S$ then they are equivalent from the point of view of the auction; we will use $|\mathcal{G}|$ to denote the number of *different* such functions in $\mathcal{G}$.[6]

The following lemma is key to our analysis. Note that using Hoeffding bounds would produce an h^2 term in the exponent; by applying McDiarmid's inequality instead we only need a factor of $O(h)$.

Lemma 1 *Consider a pricing function g and a profit level p. If we randomly partition S into S_1 and S_2, then the probability that $|g(S_1) - g(S_2)| \geq \epsilon \max[g(S), p]$ is at most $2e^{-\epsilon^2 p/(2h)}$.*

Proof: Let $Y_1, \ldots, Y_n$ be i.i.d random variables that define the partition of S into S_1 and S_2: that is, Y_i is 1 with probability $1/2$ and Y_i is 2 with probability $1/2$. Let $t(y_1, ..., y_n) = \sum_{i: y_i = 1} g(i)$. So, as a random

[4]So, in this setting $\mathcal{G}$ is the class of the form $\{g | g : \{pub\} \to [1, h]^m\}$.

[5]Formally, for any pricing function p over bundles, $\rho(p, v_i) = p(s^*)$ where $s^* = \text{argmax}_{s \subseteq S}[v_i(s) - p(s)]$, and we require for purpose of individual rationality that $p(\emptyset) = v_i(\emptyset) = 0$.

[6]Note that in our mechanism, when choosing a function in $\mathcal{G}$ to apply to S_2, the auction will only be looking at values $g(i)$ for $i \in S_1$, and vice-versa. Thus the mechanism will not really "know" if g_1 and g_2 are equivalent over S when making its selection. Nonetheless, this definition of $|\mathcal{G}|$ is useful for analysis.

variable, $g(S_1) = t(Y_1, ..., Y_n)$ and clearly $\mathbf{E}[t(Y_1, ..., Y_n)] = g(S)/2$. Assume first that $g(S) \geq p$. From the McDiarmid concentration inequality (see Appendix A), plugging $c_i = g(i)$ in Theorem 15, we get:

$$\mathbf{Pr}\left\{\left|g(S_1) - \frac{g(S)}{2}\right| \geq \frac{\epsilon}{2}g(S)\right\} \leq 2e^{-\left[\frac{\epsilon^2 g(S)^2}{2\sum_{i=1}^{n} g(i)^2}\right]}.$$

Since $\sum_{i=1}^{n} g(i)^2 \leq \max_i\{g(i)\} \sum_{i=1}^{n} g(i)$, we obtain:

$$\mathbf{Pr}\left\{\left|g(S_1) - \frac{g(S)}{2}\right| \geq \frac{\epsilon}{2}g(S)\right\} \leq 2e^{-\left[\frac{\epsilon^2 g(S)}{2h}\right]}.$$

Moreover, since $g(S_1) + g(S_2) = g(S)$ and $g(S) \geq p$, we get that $\mathbf{Pr}\{|g(S_1) - g(S_2)| \geq \epsilon g(S)\} \leq 2e^{-\epsilon^2 p/(2h)}$. Consider now that $g(S) < p$. Again, using the McDiarmid inequality we have

$$\mathbf{Pr}\{|g(S_1) - g(S_2)| \geq \epsilon p\} \leq 2e^{-\left[\frac{\epsilon^2 p^2}{2\sum_{i=1}^{n} g(i)^2}\right]}.$$

Since $\sum_{i=1}^{n} g(i)^2 \leq hg(S) \leq ph$ we obtain again that $\mathbf{Pr}\{|g(S_1) - g(S_2)| \geq \epsilon n\} \leq 2e^{-\epsilon^2 p/(2h)}$, which gives us the desired bound. ■

Notice that Lemma 1 implies that:

Corollary 1 *Suppose we randomly partition S into S_1 and S_2. With probability at least $1 - \delta$, we obtain that for all functions g in $\mathcal{G}$ such that $g(S) \geq \frac{2h}{\epsilon^2}[\ln(2|\mathcal{G}|/\delta)]$ we have $|g(S_1) - g(S_2)| \leq \epsilon g(S)$.*

Proof: Follows from Lemma 1 by plugging in $p = g(S)$ and then using the union bound over all $g \in \mathcal{G}$. ■

We can now give our simplest generic reduction, based on just the number of functions in $\mathcal{G}$. Note that in many settings (see Sections 3.3.3, 4, and 5.2) we will be able to get stronger guarantees by a more refined analysis.

Theorem 1 *Given comparison class $\mathcal{G}$ and a β-approximation algorithm $\mathcal{A}$ for optimizing over $\mathcal{G}$, then so long as $\mathrm{OPT}_{\mathcal{G}} \geq \beta\frac{18h}{\epsilon^2}\ln(2|\mathcal{G}|/\delta)$, then with probability at least $1 - \delta$, the profit of RSOPF$_{(\mathcal{G},\mathcal{A})}$ is at least $(1-\epsilon)\,\mathrm{OPT}_{\mathcal{G}}/\beta$.*

Proof: Let g_1 be the function in $\mathcal{G}$ produced by $\mathcal{A}$ over S_1 and g_2 be the function in $\mathcal{G}$ produced by $\mathcal{A}$ over S_2. Let g_{OPT} be the optimal function in $\mathcal{G}$ over S; so $g_{\mathrm{OPT}}(S) = \mathrm{OPT}_{\mathcal{G}}$. Since the optimal function over S_1 is at least as good as g_{OPT} on S_1 (and likewise for S_2), the fact that $\mathcal{A}$ is a β-approximation implies that $g_1(S_1) \geq g_{\mathrm{OPT}}(S_1)/\beta$ and $g_2(S_2) \geq g_{\mathrm{OPT}}(S_2)/\beta$.

Let $p = \frac{18h}{\epsilon^2}\ln(2|\mathcal{G}|/\delta)$. Using Lemma 1 (applying the union bound over all $g \in \mathcal{G}$), we have that with probability $1 - \delta$, every $g \in \mathcal{G}$ satisfies $|g(S_1) - g(S_2)| \leq \frac{\epsilon}{3}\max[g(S), p]$. In particular, $g_1(S_2) \geq g_1(S_1) - \frac{\epsilon}{3}\max[g_1(S), p]$, and $g_2(S_1) \geq g_2(S_2) - \frac{\epsilon}{3}\max[g_2(S), p]$.

Since $\mathrm{OPT}_{\mathcal{G}} \geq \beta p$, summing the above two inequalities and performing a case-analysis we get that the profit of RSOPF$_{(\mathcal{G},\mathcal{A})}$, namely the sum $g_1(S_2) + g_2(S_1)$, is at least $(1-\epsilon)\,\mathrm{OPT}_{\mathcal{G}}/\beta$. More specifically, assume first that $g_1(S) \geq p$ and $g_2(S) \geq p$. This implies that $g_1(S_2) \geq g_1(S_1) - \frac{\epsilon}{3}g_1(S)$ and $g_2(S_1) \geq$

$g_2(S_2) - \frac{\epsilon}{3} g_2(S)$, and therefore $(1+\frac{\epsilon}{3})g_1(S_2) \geq (1-\frac{\epsilon}{3})g_1(S_1)$ and $(1+\frac{\epsilon}{3})g_2(S_1) \geq (1-\frac{\epsilon}{3})g_2(S_2)$. So, the profit of RSOPF$_{(\mathcal{G},\mathcal{A})}$ in this case is at least $\frac{1-\epsilon/3}{1+\epsilon/3}(g_1(S_1)+g_2(S_2)) \geq \frac{1-\epsilon/3}{1+\epsilon/3} \mathrm{OPT}_{\mathcal{G}}/\beta \geq (1-\epsilon)\,\mathrm{OPT}_{\mathcal{G}}/\beta$. If both $g_1(S) < p$ and $g_2(S) < p$, then $g_1(S_2) \geq g_1(S_1) - \frac{\epsilon}{3}p$ and $g_2(S_1) \geq g_2(S_2) - \frac{\epsilon}{3}p$, and so the profit of RSOPF$_{(\mathcal{G},\mathcal{A})}$ in this case is at least $\mathrm{OPT}_{\mathcal{G}}/\beta - \frac{2}{3}\epsilon p$ which is at least $(1-\epsilon)\,\mathrm{OPT}_{\mathcal{G}}/\beta$ by our assumption that $\mathrm{OPT}_{\mathcal{G}} \geq \beta p$. Finally, assume without loss of generality that $g_1(S) \geq p$ and $g_2(S) < p$. This implies that $g_1(S_2) \geq g_1(S_1) - \frac{\epsilon}{3}g_1(S)$ and $g_2(S_1) \geq g_2(S_2) - \frac{\epsilon}{3}p$. The former inequality implies that $(1+\frac{\epsilon}{3})g_1(S_2) \geq (1-\frac{\epsilon}{3})g_1(S_1)$, and so $g_1(S_2) \geq (1-2\epsilon/3)g_1(S_1)$, and the latter inequality implies that $g_2(S_1) \geq g_2(S_2) - \frac{\epsilon}{3}\mathrm{OPT}_{\mathcal{G}}/\beta$. Together we have that $g_1(S_2)+g_2(S_1) \geq (1-2\epsilon/3)g_{\mathrm{OPT}}(S_1)/\beta + g_{\mathrm{OPT}}(S_2)/\beta - \frac{\epsilon}{3}\mathrm{OPT}_{\mathcal{G}}/\beta \geq (1-\epsilon)\,\mathrm{OPT}_{\mathcal{G}}/\beta$. ■

Notice that Theorem 1 implies that:

Corollary 2 *Given comparison class $\mathcal{G}$ and a β-approximation algorithm $\mathcal{A}$ for optimizing over $\mathcal{G}$, then so long as $\mathrm{OPT}_{\mathcal{G}} \geq \beta n$ and the number of bidders n satisfies*

$$n \geq \frac{18h}{\epsilon^2}\ln(2|\mathcal{G}|/\delta),$$

then with probability at least $1-\delta$, the profit of RSOPF$_{(\mathcal{G},\mathcal{A})}$ is at least $(1-\epsilon)\,\mathrm{OPT}_{\mathcal{G}}/\beta$.

For example, in the digital-good auction with the comparison-class of prices discretized to powers of $1+\epsilon$ we have $\mathrm{OPT}_{\mathcal{G}} \geq n$ (since each bidder's valuation is at least 1), $\beta = 1$ (since the algorithmic problem is easy), and $|\mathcal{G}| = O(\log_{1+\epsilon} h)$. So, Corollary 2 says that $O(\frac{h}{\epsilon^2}\log\log_{1+\epsilon} h)$ bidders are sufficient to perform nearly as well as optimal. In Section 4 we give even better bounds for this case.

3.2 Structural Risk Minimization

In many natural cases, $\mathcal{G}$ consists of functions at different "levels of complexity" k, such as partitioning bidders into k markets for different values of k. One natural approach to such a setting is to perform *structural risk minimization* (SRM): that is, to assign a penalty term to functions based on their complexity and then to run a version of RSOPF$_{(\mathcal{G},\mathcal{A})}$ in which $\mathcal{A}$ optimizes profit minus penalty. Specifically, let $\bar{\mathcal{G}}$ be a series of pricing function classes $\mathcal{G}_1 \subseteq \mathcal{G}_2 \subseteq \ldots$, and let pen be a penalty function defined over these classes. We then define the procedure RSOPF-SRM$_{(\bar{\mathcal{G}},\mathsf{pen})}$ as follows:

1. Randomly partition the bidders into two sets, S_1 and S_2, flipping fair coin for each bidder.
2. Compute g_1 to maximize $\max_k \max_{g\in\mathcal{G}_k}[g(S_1) - \mathsf{pen}(\mathcal{G}_k)]$ and similarly compute g_2 from S_2.
3. Use price function g_1 for bidders in S_2 and g_2 for bidders in S_1.

We can now derive a guarantee for the RSOPF-SRM$_{(\bar{\mathcal{G}},\mathsf{pen})}$ mechanism as follows:

Theorem 2 *Assuming that we have a β-approximation algorithm for solving the optimization problem required by RSOPF-SRM$_{(\bar{\mathcal{G}},\mathsf{pen})}$ then for any given value of n, ϵ, and δ, with probability at least $1-\delta$, the revenue of RSOPF-SRM$_{(\bar{\mathcal{G}},\mathsf{pen})}$ for $\mathsf{pen}(\mathcal{G}_k) = \frac{8h}{\epsilon^2}\ln(8k^2|\mathcal{G}_k|/\delta)$ is*

$$\max_k\left(\frac{1}{\beta}\left[(1-\epsilon)\,\mathrm{OPT}_k - \widetilde{\mathsf{pen}}(\mathcal{G}_k)\right]\right),$$

where $\widetilde{\mathsf{pen}}(\mathcal{G}_k) = 2\mathsf{pen}(\mathcal{G}_k)$.

Proof: Using Corollary 1 and a union bound over the values $\delta_k = \delta/(4k^2)$, we obtain that with probability at least $1-\delta$, simultaneously for all k and for all functions g in $\mathcal{G}_k$ such that $g(S) \geq \frac{8h}{\epsilon^2}\ln(8k^2|\mathcal{G}_k|/\delta)\mathsf{pen}(\mathcal{G}_k)$, we have $|g(S_1) - g(S_2)| \leq \frac{\epsilon}{2}g(S)$. Let k^* be the optimal index, namely let k^* be the index such that $(1-\epsilon)\,\mathrm{OPT}_{k^*} - \widetilde{\mathsf{pen}}(\mathcal{G}_{k^*}) = \max_k((1-\epsilon)\,\mathrm{OPT}_k - \widetilde{\mathsf{pen}}(\mathcal{G}_k))$, and let k_i be the index of the best function (according to our criterion) over S_i , for $i = 1, 2$. By our assumption that g_1 and g_2 were chosen by a β-approximation algorithm, we have $g_i(S_i) - \mathsf{pen}(\mathcal{G}_{k_i}) \geq \frac{1}{\beta}\left(g_{\mathrm{OPT}_{k^*}}(S_i) - \mathsf{pen}(\mathcal{G}_{k^*})\right)$, for $i = 1, 2$.

We will argue next that $g_1(S_2) \geq \frac{1}{\beta}\frac{1-\epsilon/2}{1+\epsilon/2}\left(g_{\mathrm{OPT}_{k^*}}(S_1) - \mathsf{pen}(\mathcal{G}_{k^*})\right)$. First, if $g_1(S_1) < \mathsf{pen}(\mathcal{G}_{k_1})$, then the conclusion is clear since we have $0 > g_1(S_1) - \mathsf{pen}(\mathcal{G}_{k_1}) \geq g_{\mathrm{OPT}_{k^*}}(S_1) - \mathsf{pen}(\mathcal{G}_{k^*})$. If $g_1(S_1) \geq \mathsf{pen}(\mathcal{G}_{k_1})$, then as argued above we have $|g_1(S_1) - g_1(S_2)| \leq \frac{\epsilon}{2}g_1(S)$ and so $g_1(S_2) \geq \frac{1-\epsilon/2}{1+\epsilon/2}g_1(S_1) \geq \frac{1}{\beta}\frac{1-\epsilon/2}{1+\epsilon/2}\left(g_{\mathrm{OPT}_{k^*}}(S_1) - \mathsf{pen}(\mathcal{G}_{k^*})\right)$. Similarly, we can prove that $g_2(S_1) \geq \frac{1}{\beta}\frac{1-\epsilon/2}{1+\epsilon/2}\left(g_{\mathrm{OPT}_{k^*}}(S_2) - \mathsf{pen}(\mathcal{G}_{k^*})\right)$. All these together imply that the profit of RSOPF-SRM$_{(\bar{\mathcal{G}},\mathsf{pen})}$, namely $g_1(S_2) + g_2(S_1)$, is at least

$$\frac{1}{\beta}\frac{1-\epsilon/2}{1+\epsilon/2}\left(g_{\mathrm{OPT}_{k^*}}(S) - 2\mathsf{pen}(\mathcal{G}_{k^*})\right) \geq \frac{1}{\beta}\left((1-\epsilon)\,\mathrm{OPT}_{k^*} - \widetilde{\mathsf{pen}}(\mathcal{G}_{k^*})\right),$$

which implies the desired result. ■

Clearly, when $\beta = 1$ (i.e. we have an optimal algorithm for the underlying algorithmic problem), we get the following result.

Corollary 3 *Assuming that we have an exact algorithm for solving the optimization problem required by RSOPF-SRM*$_{(\bar{\mathcal{G}},\mathsf{pen})}$ *then for any given value of n, ϵ, and δ, with probability at least $1-\delta$, the revenue of RSOPF-SRM*$_{(\bar{\mathcal{G}},\mathsf{pen})}$ *for* $\mathsf{pen}(\mathcal{G}_k) = \frac{8h}{\epsilon^2}\ln(8k^2|\mathcal{G}_k|/\delta)$ *is*

$$\max_k((1-\epsilon)\,\mathrm{OPT}_k - \widetilde{\mathsf{pen}}(\mathcal{G}_k)),$$

where $\widetilde{\mathsf{pen}}(\mathcal{G}_k) = 2\mathsf{pen}(\mathcal{G}_k)$.

3.3 Improving the Bounds

The results above say, in essence, that if we have enough bidders so that the optimal profit is large compared to $\frac{h}{\epsilon^2}\log(|\mathcal{G}|)$, then our mechanism will perform nearly as well as the best function in $\mathcal{G}$. In these bounds, one should think of $\log(|\mathcal{G}|)$ as a measure of the complexity of class $\mathcal{G}$ — for instance, it can be thought of as the number of bits needed to describe a typical function in that class. However, in many cases one can achieve a better bound, by adapting techniques developed for analyzing generalization performance in machine learning theory. In this section, we discuss a number of such methods that can produce better bounds. These include both *analysis* techniques (such as using appropriate forms of *covering numbers*), where we do not change the mechanism but instead provide a stronger guarantee, and *design* techniques (like *discretizing*), where we modify the mechanism to produce a better bound.

3.3.1 Discretizing

In many cases, we can greatly reduce $|\mathcal{G}|$ without much affecting $\mathrm{OPT}_{\mathcal{G}}$ by performing some type of discretization. For instance, for auctioning a digital good, there are infinitely many single-price functions but only $\log_{1+\epsilon} h \approx \frac{1}{\epsilon}\ln h$ prices at powers of $(1+\epsilon)$. Also, since rounding down the optimal price to the nearest power of $1+\epsilon$ can reduce revenue for this auction by at most a factor of $1+\epsilon$, the optimal function in the discretized class must be close to the optimal function in the original class. More generally, if we can find a smaller class $\mathcal{G}'$ such that $\mathrm{OPT}_{\mathcal{G}'}$ is guaranteed to be close to $\mathrm{OPT}_{\mathcal{G}}$, then we can instruct our algorithm $\mathcal{A}$ to optimize over $\mathcal{G}'$ and get better bounds. In Section 6 we discuss an interesting discretization for the case of *combinatorial auctions*.

3.3.2 Counting Possible Outputs

Suppose we can argue that our algorithm $\mathcal{A}$, run on a subset of S, will only ever output pricing functions from a restricted set $\mathcal{G}_{\mathcal{A}} \subset \mathcal{G}$. For example, if $\mathcal{A}$ picks the optimal single price over its input for the problem of auctioning a digital good, then this price must be one of the bids, so $|\mathcal{G}_{\mathcal{A}}| \leq n$. Then, we can simply replace $|\mathcal{G}|$ with $|\mathcal{G}_{\mathcal{A}}|$ (or $|\mathcal{G}_{\mathcal{A}}|+1$ if the optimal function is not in $\mathcal{G}_{\mathcal{A}}$) in all the above arguments. Formally, we can say that:

Theorem 3 *Suppose our algorithm $\mathcal{A}$, run on a subset of S, can only output pricing functions from a restricted set $\mathcal{G}_{\mathcal{A}} \subset \mathcal{G}$. Then all the bounds in sections 3.1 and 3.2 hold with $|\mathcal{G}|$ replaced by $|\mathcal{G}_{\mathcal{A}}|$.*

3.3.3 Using Covering Numbers

The main idea of these arguments is the following. Suppose $\mathcal{G}$ has the property that there exists a much smaller class $\mathcal{G}'$ that "covers" it, with respect to the given set of bidders S. Then one can show that if all functions in $\mathcal{G}'$ perform similarly on S_1 as they do on S_2, then this will be true for all functions in $\mathcal{G}$ as well. These kind of arguments are quite often used in Machine Learning (see for instance [2, 9, 12, 30]), but the main challenge is to define the right notion of "covers" for our mechanism design setting to get good and meaningful bounds.

We present in the following two notions of covers that are especially suited for our setting. We start with the weaker, but more intuitive notion of an L_∞ multiplicative γ-cover, and then discuss the less intuitive, but stronger notion of L_1 multiplicative γ-cover. Specifically, we define these covers as follows:

Definition 2 *$\mathcal{G}'$ is an L_∞ multiplicative γ-cover of $\mathcal{G}$ with respect to S if, for every $g \in \mathcal{G}$, there exists $g' \in \mathcal{G}'$ such that g' extracts the same revenue as g does from every bidder, up to a $1+\gamma$ factor; that is, $|g(i) - g'(i)| \leq \gamma g(i)$ for all i.*

Definition 3 *$\mathcal{G}'$ is an L_1 multiplicative γ-cover of $\mathcal{G}$ with respect to S if for every $g \in \mathcal{G}$ there exists $g' \in \mathcal{G}'$ such that $\sum_{i \in S} |g(i) - g'(i)| \leq \gamma \sum_{i \in S} g(i)$.*

Note that any L_∞ cover is also a L_1 cover. We begin by proving the following structural lemma regarding the L_∞ multiplicative γ-covers.

Lemma 2 *Let $\mathcal{G}'$ be an L_∞ multiplicative γ-cover of $\mathcal{G}$ with respect to S. If for every $g' \in \mathcal{G}'$ we have $|g'(S_1) - g'(S_2)| \leq \epsilon' \max[g'(S), p]$, then we also have $|g(S_1) - g(S_2)| \leq (\epsilon'(1+\gamma)+\gamma)\max[g(S), p]$ for every $g \in \mathcal{G}$.*

Proof: Clearly, $|g(S_1) - g(S_2)| \leq |g(S_1) - g'(S_1)| + |g'(S_1) - g'(S_2)| + |g'(S_2) - g(S_2)|$, and using the definition of an L_∞ multiplicative γ-cover we get $|g(S_1) - g(S_2)| \leq \gamma g(S_1) + |g'(S_1) - g'(S_2)| + \gamma g(S_2)$. Finally, using the assumption that $|g'(S_1) - g'(S_2)| \leq \epsilon' \max[g'(S), p]$ for every $g' \in \mathcal{G}'$, we get the desired result, namely, $|g(S_1) - g(S_2)| \leq (\epsilon'(1+\gamma)+\gamma)\max[g(S), p]$, for every $g \in \mathcal{G}$. ■

Using Lemma 2, we can now get the following bound:

Theorem 4 *Given comparison class $\mathcal{G}$ and a β-approximation algorithm $\mathcal{A}$ for optimizing over $\mathcal{G}$, then so long as* $\mathrm{OPT}_{\mathcal{G}} \geq \beta \frac{72h}{\epsilon^2} \ln(2|\mathcal{G}'|/\delta)$ *for some $\frac{\epsilon}{12}$-cover $\mathcal{G}'$ of $\mathcal{G}$ with respect to S, then with probability at least $1-\delta$, the profit of RSOPF$_{(\mathcal{G},\mathcal{A})}$ is at least* $(1-\epsilon)\,\mathrm{OPT}_{\mathcal{G}}/\beta$.

Proof Sketch: Let $p = \frac{72h}{\epsilon^2} \ln(2|\mathcal{G}'|/\delta)$. By Lemma 1, applying the union bound, we have that with probability $1-\delta$, every $g' \in \mathcal{G}'$ satisfies $|g'(S_1) - g'(S_2)| \leq \frac{\epsilon}{6} \max[g'(S), p]$. Using Lemma 2 with ϵ' set to $\frac{\epsilon}{6}$ and γ set to $\frac{\epsilon}{12}$ we obtain that with probability $1-\delta$, every $g \in \mathcal{G}$ satisfies $|g(S_1) - g(S_2)| \leq \frac{\epsilon}{3} \max[g(S), p]$. Finally, proceeding as in the proof of Theorem 1 we obtain the desired result. ■

Notice that Theorem 4 implies that:

Corollary 4 *Given comparison class $\mathcal{G}$ and a β-approximation algorithm $\mathcal{A}$ for optimizing over $\mathcal{G}$, then so long as $\mathrm{OPT}_{\mathcal{G}} \geq \beta n$ and the number of bidders satisfies*

$$n \geq \frac{72h}{\epsilon^2} \ln(2|\mathcal{G}'|/\delta)$$

for some $\frac{\epsilon}{12}$-cover $\mathcal{G}'$ of $\mathcal{G}$ with respect to S, then with probability at least $1-\delta$, the profit of RSOPF$_{(\mathcal{G},\mathcal{A})}$ is at least $(1-\epsilon)\,\mathrm{OPT}_{\mathcal{G}}/\beta$.

For example, for the digital-good auction, the set of prices at powers of $1+\epsilon$ together with the set of bidders' valuations $\{priv_i | i \in S\}$ is an L_∞ multiplicative ϵ-cover of the set of all single-price functions. This means that even if $\mathcal{A}$ chooses the best price *without* discretizing, then (using $\beta = 1$ and the fact that $\mathrm{OPT}_{\mathcal{G}} \geq n$ since all valuations are assumed to be at least 1) we get that $O(\frac{h}{\epsilon^2} \log(\frac{h}{\delta\epsilon}))$ bidders are sufficient for the mechanism to be within an ϵ factor of optimal.

We will now consider the L_1 multiplicative γ-covers, and we will start by proving the following structural lemma characterizing these L_1 covers.

Lemma 3 *If $\sum_{i\in S} |g(i) - g'(i)| \leq \gamma \sum_{i\in S} g(i)$ and $g'(S_1) \geq g'(S_2) - \epsilon \max[g'(S), p]$, then $g(S_1) \geq g(S_2) - \epsilon \max[g'(S), p] - \gamma g(S)$.*

Proof: Let $\vec{\Delta}_{g_1 g_2}(S) = \sum_{i\in S} \max(g_1(i) - g_2(i), 0)$ and consider $\Delta_{gg'}(S) = \vec{\Delta}_{gg'}(S) + \vec{\Delta}_{g'g}(S)$. Clearly, for any $S' \subseteq S$ we have $\vec{\Delta}_{gg'}(S) \geq \vec{\Delta}_{gg'}(S')$ and likewise $\Delta_{gg'}(S) \geq \Delta_{gg'}(S')$. Also, for any subset $S' \subseteq S$ we have $g(S') - g'(S') \leq \vec{\Delta}_{gg'}(S)$. Now, from $g'(S_1) \geq g'(S_2) - \epsilon \max[g'(S), p]$ we obtain that $g(S_1) + \vec{\Delta}_{g'g}(S) \geq g'(S_2) - \epsilon \max[g'(S), p] \geq g(S_2) - \vec{\Delta}_{gg'}(S) - \epsilon \max[g'(S), p]$. Therefore we have $g(S_1) \geq g(S_2) - \Delta_{gg'}(S) - \epsilon \max[g'(S), p]$, which finally implies that $g(S_1) \geq g(S_2) - \epsilon \max[g'(S), p] - \gamma g(S)$. ■

Using Lemma 3, we can now get the following bound:

Theorem 5 *Given comparison class $\mathcal{G}$ and a β-approximation algorithm $\mathcal{A}$ for optimizing over $\mathcal{G}$, then so long as $\mathrm{OPT}_{\mathcal{G}} \geq n$ and the number of bidders n satisfies*

$$n \geq \frac{8h}{\epsilon^2} \ln(2\left|\mathcal{G}'\right|/\delta),$$

for some γ-cover $\mathcal{G}'$ of $\mathcal{G}$ with respect to S such that $\mathcal{G}' \subseteq \mathcal{G}$, then with probability at least $1-\delta$, the profit of RSOPF$_{(\mathcal{G},\mathcal{A})}$ is at least $(1/\beta - \epsilon - 2\gamma)\,\mathrm{OPT}_{\mathcal{G}}$.

Proof: Let g_1 be the function in $\mathcal{G}$ produced by $\mathcal{A}$ over S_1 and g_2 be the function in $\mathcal{G}$ produced by $\mathcal{A}$ over S_2. Let g_{OPT} (resp. g'_{OPT}) be the optimal function in $\mathcal{G}$ (resp. $\mathcal{G}'$) over S. Of course, $\mathcal{G}' \subseteq \mathcal{G}$ implies that $g'_{\mathrm{OPT}}(S) \leq g_{\mathrm{OPT}}(S) = \mathrm{OPT}_{\mathcal{G}}$. Since the optimal function over S_1 is at least as good as g_{OPT} on S_1 (and likewise for S_2), the fact that $\mathcal{A}$ is a β-approximation implies that $g_1(S_1) \geq g_{\mathrm{OPT}}(S_1)/\beta$ and $g_2(S_2) \geq g_{\mathrm{OPT}}(S_2)/\beta$.

By Lemma 1 (using $p = n$) and plugging in our bound on n and applying the union bound, with probability at least $1-\delta$, every $g' \in \mathcal{G}'$ satisfies $|g'(S_1) - g'(S_2)| \leq \frac{\epsilon}{2} \max\left[g'(S), n\right]$. Since $\mathcal{G}'$ is a γ-cover of $\mathcal{G}$, this combined with Lemma 3 implies that all $g \in \mathcal{G}$ satisfy $g(S_2) \geq g(S_1) - (\frac{\epsilon}{2} + \gamma)\max[\mathrm{OPT}_{\mathcal{G}}, n]$. In particular, $g_1(S_2) \geq g_1(S_1) - (\frac{\epsilon}{2} + \gamma)\max[\mathrm{OPT}_{\mathcal{G}}, n]$, and $g_2(S_1) \geq g_2(S_2) - (\frac{\epsilon}{2} + \gamma)\max[\mathrm{OPT}_{\mathcal{G}}, n]$.

Since $\mathrm{OPT}_{\mathcal{G}} \geq n$, summing the above two inequalities and performing a simple case-analysis we get that the profit of $\mathrm{RSOPF}_{(\mathcal{G},\mathcal{A})}$, namely $g_1(S_2) + g_2(S_1)$, is at least $(1/\beta - \epsilon - 2\gamma)\,\mathrm{OPT}_{\mathcal{G}}$. ■

We will demonstrate the utility of L_1 multiplicative covers in Section 4 by showing the existence of L_1 covers of size $o(n)$ for the digital good auction; note this is not possible for L_∞ multiplicative covers. It is worth noting that a straightforward application of analogous ϵ-cover results in learning theory [2] (which would require an additive, rather than multiplicative gap of ϵ for every bidder) would add an extra factor of h into our sample-size bounds.

4 Auctioning Digital Goods to Indistinguishable Bidders

We now consider applying the results in Section 3 to the problem of auctioning a digital good to indistinguishable bidders. Here a natural class of comparison functions $\mathcal{G}$ is the set of all constant-price functions (see for instance [20]). Clearly in this case, it is trivial to solve the underlying algorithm problem optimally: given a set of bidders, just output the constant price that maximizes the price times the number of bidders with bids at least as high as the price. Also, it is easy to see that the optimal price output will be one of the bid values. Thus, applying Theorem 3 with the bound on $|\mathcal{G}_{\mathcal{A}}| = n$, we get an approximately optimal auction with an additive loss $O(h \log n)$.

We can obtain better results using γ-cover arguments and Theorem 5 as follows. Let $b_1, \ldots, b_n$ be the bids of the n bidders sorted from highest to lowest. Define $\mathcal{G}'$ as $\{b_i \,:\, j \in \mathbb{Z} \wedge i = \lfloor (1+\gamma')^j \rfloor \wedge i \in \{1, \ldots, n\}\} \cup \{(1+\gamma')^i \,:\, i \in \{1, \ldots, \log_{1+\gamma'} h\}\}$. Consider $g \in \mathcal{G}$ and find the $g' \in \mathcal{G}'$ that offers the largest price less than the offer price of g. First, all the winners in S on g also win in g'. Second, the offer price of g' is within a factor of $1 + \gamma'$ of the offer price of g'. Third, g' has at most a factor of $1 + \gamma'$ more winners than g. The first two facts above imply that $\vec{\Delta}_{gg'}(S) \leq \gamma' g(S)$. The third fact implies that $\vec{\Delta}_{g'g}(S) \leq \gamma' g(S)$. Thus, $\Delta_{gg'} \leq 2\gamma' g(S)$ and therefore, $\mathcal{G}'$ is a $2\gamma'$-cover of $\mathcal{G}$. Since $|\mathcal{G}'|$ is $O(\log hn)$, the additive loss of $\mathrm{RSOPF}_{(\mathcal{G},\mathcal{A})}$ is $O(h \log\log nh)$.[7]

We can also apply the discretization technique by defining $\mathcal{G}'$ to be the set of all constant-price functions whose price $p \in [1, h]$ is a power of $(1 + \epsilon/2)$: if we can get revenue at least $(1 - \epsilon/2)$ times the optimal in this class, we will be within $(1 - \epsilon)$ of the optimal fixed price overall. Applying Corollary 2 ($\mathcal{A}$ can trivially find the best function in $\mathcal{G}'$ by simply trying all of them), with probability $1 - \delta$ we get at least $1 - \epsilon$ times the optimal fixed price so long as the number of bidders n is at least $\frac{72h}{\epsilon^2}\ln(\frac{4\ln h}{\epsilon\delta}) = O(h \log\log h)$. We now present a more refined analysis, which gives us even better guarantees.

[7] It is interesting to contrast these results with that of [21] which showed that RSOPF over the set of constant-price functions is near 6-competitive with the promise that $n \gg h$. A much more complicated analysis of RSOPF in a slightly different competitive framework is given in [20].

Theorem 6 *Let $\mathcal{G}$ be the class of constant price functions, discretized at powers of $(1+\frac{\epsilon}{2})$, and let $\delta < 1/2$. Then with probability $1-\delta$, RSOPF$_{(\mathcal{G},\mathcal{A})}$ obtains profit at least*

$$\mathrm{OPT}_{\mathcal{G}} - 8\sqrt{h\,\mathrm{OPT}_{\mathcal{G}}\log(2/(\epsilon\delta))}.$$

So, this implies that for $\mathrm{OPT}_{\mathcal{G}} \geq (\frac{16}{\epsilon})^2 h\log(2/(\epsilon\delta))$ we get profit at least $(1-\epsilon/2)\,\mathrm{OPT}_{\mathcal{G}}$, which is at least $(1-\epsilon)$ times the optimal non-discretized fixed price. So, even in the worst-case that the optimal single-price solution is at price 1 (so $\mathrm{OPT}_{\mathcal{G}} = n$) we get an $O(\log\log h)$ improvement over the generic bound, but if $\mathrm{OPT}_{\mathcal{G}}$ extracts substantially more profit on average per bidder, we can get an improvement of up to $O(h\log\log h)$.

To prove Theorem 6, let us for convenience define α to be the discretization parameter (which was $\epsilon/2$ above) and assume h is a power of $(1+\alpha)$. For comparison function g_v offering price v, let n_v denote the number of winners (bidders whose value is at least v), and let $r_v = v \cdot n_v$ denote the profit of g_v on S. Denote by $\hat{r}_v$ the observed revenue of g_v on S_1 (and so $\hat{r}_v = v \cdot \hat{n}_v$, where $\hat{n}_v$ is the number of winners in S_1 for g_v). So, we have $\mathbf{E}[\hat{r}_v] = \frac{r_v}{2}$. We now begin with the following lemma.

Lemma 4 *Let $\epsilon < 1, \delta < 1/2$. With probability at least $1-\delta$ we have that, for every $g_v \in \mathcal{G}$ the observed revenue on S_1 satisfies:*

$$\left|\hat{r}_v - \frac{r_v}{2}\right| \leq \max\left(\frac{h\log(1/(\alpha\delta))}{\epsilon}, \epsilon r_v\right).$$

Proof: First for a given price v let $a_{n,v}$ be $|\hat{n}_v - \frac{n_v}{2}|$. To prove our lemma we will use the consequence of Chernoff bound we present in Appendix A (see Theorem 16). For any v and $j \geq 1$ we consider $n' = \frac{(1+\alpha)^j\log(1/(\alpha\delta))}{\epsilon^2}$, and so we get $\mathbf{Pr}\left\{a_{n,v} \geq \epsilon\max\left(n_v, \frac{(1+\alpha)^j\log(1/(\alpha\delta))}{\epsilon^2}\right)\right\} \leq 2e^{-2(1+\alpha)^j\log(1/(\alpha\delta))}$. This further implies that we have $a_{n,v} \geq \epsilon\max\left(n_v, \frac{(1+\alpha)^j\log(1/(\alpha\delta))}{\epsilon^2}\right)$ with probability at most $2(\alpha\delta)^{2(1+\alpha)^j}$. Therefore for $v = h/(1+\alpha)^j$ we have $\mathbf{Pr}\left\{\left|\hat{r}_v - \frac{r_v}{2}\right| \geq \max\left(\frac{h\log(1/(\alpha\delta))}{\epsilon}, \epsilon r_v\right)\right\} \leq 2(\alpha\delta)^{2(1+\alpha)^j}$, and so the probability that there exists a $g_v \in \mathcal{G}$ such that $\left|\hat{r}_v - \frac{r_v}{2}\right| \geq \max\left(\frac{h}{\epsilon}, \epsilon r_v\right)$ is at most $2\sum_j(\alpha\delta)^{2(1+\alpha)^j} \leq 2\sum_{j'}\frac{1}{\alpha}(\alpha\delta)^{2\cdot 2^{j'}} \leq \delta$. This implies that with high probability, at least $1-\delta$, we have that simultaneously, for every $g_v \in \mathcal{G}$ the observed revenue on S_1 satisfies:

$$\left|\hat{r}_v - \frac{r_v}{2}\right| \leq \max\left(\frac{h\log(1/(\alpha\delta))}{\epsilon}, \epsilon r_v\right),$$

as desired. ■

Proof of Theorem 6: Assume now that it is the case that for every $g_v \in \mathcal{G}$ we have $\left|\hat{r}_v - \frac{r_v}{2}\right| \leq \max\left(\frac{H}{\epsilon}, \epsilon r_v\right)$, where $H = h\log(2/(\alpha\delta))$. Let v^* be the optimal price level among prices in $\mathcal{G}$, and let $\tilde{v}^*$ be the price that looks best on S_1. Obviously, our gain on S_2 is $r_{\tilde{v}^*} - \hat{r}_{\tilde{v}^*}$. We have $\hat{r}_{v^*} \geq \frac{r_v^*}{2} - \frac{H}{\epsilon} - \epsilon r_{v^*} r_{v^*}(1-2\epsilon)/2 - \frac{H}{\epsilon}$, $\hat{r}_{\tilde{v}^*} \geq \hat{r}_{v^*}$ and $\hat{r}_{\tilde{v}^*} \leq \frac{r_{\tilde{v}^*}}{2} + \frac{H}{\epsilon} + \epsilon r_{\tilde{v}^*} \leq \frac{r_{\tilde{v}^*}}{2} + \frac{H}{\epsilon} + \epsilon r_{v^*}$, and therefore $r_{\tilde{v}^*} - \hat{r}_{\tilde{v}^*} \geq \hat{r}_{\tilde{v}^*} - \frac{H}{\epsilon} - \epsilon r_{v^*}$, which finally implies that $r_{\tilde{v}^*} - \hat{r}_{\tilde{v}^*} \geq r_{v^*}\left(\frac{1}{2} - 2\epsilon\right) - 2\frac{H}{\epsilon}$. This implies that with probability at least $1-\delta/2$ our gain on S_2 is at least $r_{v^*}\left(\frac{1}{2} - 2\epsilon\right) - 2\frac{H}{\epsilon}$, and similarly our gain on S_1 is at least $r_{v^*}\left(\frac{1}{2} - 2\epsilon\right) - 2\frac{H}{\epsilon}$. Therefore, with probability $1-\delta$, our revenue is $\mathrm{OPT}_{\mathcal{G}}(1-4\epsilon) - 4\frac{h\log(1/(\alpha\delta))}{\epsilon}$. Optimizing the bound we set $\epsilon = \sqrt{h\log(1/(\alpha\delta))/OPT_{\mathcal{G}}}$ and get a revenue of $OPT_{\mathcal{G}} - 8\sqrt{h\;OPT_{\mathcal{G}}\;\log(1/(\alpha\delta))}$, which completes the proof. ■

5 Attribute Auctions

We now consider applying the results in Section 3 to Attribute Auctions. We begin by instantiating the results in Section 3 for market pricing auctions, and show how can we can use standard combinatorial dimensions in Learning Theory (e.g. the Vapnik-Chervonenkis (VC) dimension: see Appendix B and [2, 12, 25, 30] for a more complete treatment) in order to bound the induced complexity of a comparison class of functions. We then give an analysis for general pricing functions over the attribute space that uses the notion of covers to avoid discretization. In the Appendix C we also show how we can also obtain bounds for the case of partial information.

5.1 Market Pricing

For attribute auctions, one natural class of comparison functions are those that partition bidders into *markets* in some simple way and then offer a single sale price in each market. For example, suppose we define $\mathcal{G}_k$ to be the set of functions that choose k bidders $b_1, \ldots, b_k$, use these as cluster centers to partition S into k markets based on distance to the nearest center in attribute space, and then offer a single price in each market. In that case, if we discretize prices to powers of $(1+\epsilon)$, then clearly the number of functions in $\mathcal{G}_k$ is at most $n^k(\log_{1+\epsilon} h)^k$, so Corollary 2 implies that so long as $n \geq \frac{18h}{\epsilon^2}\left[\ln\left(\frac{2}{\delta}\right) + k\ln n + k\ln\left(\log_{1+\epsilon} h\right)\right]$ and we can solve the algorithmic problem, then with probability at least $1-\delta$, we can get profit at least $(1-\epsilon)\,\mathrm{OPT}_{\mathcal{G}_k}$.

However, we can also consider other ways of defining markets as follows. Let C be any class of subsets of $\mathcal{X}$, which we will call *feasible markets*. For k a positive integer, we consider $F_{k+1}(C)$ to be the set of all pricing functions of the following form: pick k disjoint subsets s_1,...,s_k from C, and $k+1$ prices p_0,...,p_k discretized to powers of $1+\epsilon$. Assign price p_i to bidders in s_i, and price p_0 to bidders not in any of s_1,...,s_k. For example, if $\mathcal{X} = \mathbb{R}^d$ a natural C might be the set of axis-parallel rectangles in $\mathbb{R}^d$. The specific case of $d = 1$ was studied in [6].

We can apply the results in Section 3 by using the machinery of VC-dimension (see [2, 8, 25, 30]) to count the number of distinct such functions over any given set of bidders S. In particular, let $D = VCdim(C)$ be the VC-dimension of C and assume $D < \infty$. Define $C[S]$ to be the number of distinct subsets of S induced by C. Then, from Sauer's Lemma (see Appendix B) $C[S] \leq \left(\frac{en}{D}\right)^D$, and therefore the number of different pricing functions in $F_k(C)$ over S is at most $\left(\log_{1+\epsilon} h\right)^k \left(\frac{en}{D}\right)^{kD}$. Thus applying Corollary 2 here we get:

Corollary 5 *Given a β-approximation algorithm $\mathcal{A}$ for optimizing over $\mathcal{G} = F_k(C)$, then so long as $\mathrm{OPT}_{\mathcal{G}} \geq \beta n$ and the number of bidders n satisfies*

$$n \geq \frac{18h}{\epsilon^2}\left[\ln\left(\frac{2}{\delta}\right) + k\ln\left(\frac{1}{\epsilon}\ln h\right) + kD\ln\left(\frac{ne}{D}\right)\right],$$

then with probability at least $1-\delta$, the profit of RSOPF$_{\mathcal{G},\mathcal{A}}$ is at least $(1-\epsilon)\,\mathrm{OPT}_{\mathcal{G}}/\beta$.

The above lemma has "n" on both sides of the inequality. Simple algebra yields:

Corollary 6 *Given a β-approximation algorithm $\mathcal{A}$ for optimizing over $\mathcal{G} = F_k(C)$, then so long as $\mathrm{OPT}_{\mathcal{G}} \geq \beta n$ and the number of bidders n satisfies*

$$n \geq \frac{36h}{\epsilon^2}\left[\ln\left(\frac{2}{\delta}\right) + k\ln\left(\frac{1}{\epsilon}\ln h\right) + kD\ln\left(\frac{36kh}{\epsilon^2}\right)\right],$$

then with probability at least $1-\delta$, *the profit of RSOPF*$_{\mathcal{G},\mathcal{A}}$ *is at least* $(1-\epsilon)\,\mathrm{OPT}_{\mathcal{G}}/\beta$.

Proof: Since $\ln a \le ab - \ln b - 1$ for all $a, b > 0$, we have:

$$\frac{18kDh}{\epsilon^2}\ln n \le \frac{18kDh}{\epsilon^2}\left[\frac{\epsilon^2}{36kDh}n + \ln\left(\frac{36kDh}{\epsilon^2}\right) - 1\right] = \frac{n}{2} + \frac{18kDh}{\epsilon^2}\ln\left(\frac{36kDh}{e\epsilon^2}\right).$$

Therefore, it suffices to have:

$$n \ge \frac{n}{2} + \frac{18h}{\epsilon^2}\left[\ln\left(\frac{2}{\delta}\right) + k\ln L + kD\ln\left(\frac{36kh}{\epsilon^2}\right)\right],$$

so $n \ge \frac{36h}{\epsilon^2}\left[\ln\left(\frac{2}{\delta}\right) + k\ln L + kD\ln\left(\frac{36kh}{\epsilon^2}\right)\right]$ suffices. ∎

For certain classes C we can get better bounds. In the following, denote by C_k the concept class of unions of at most k sets from C, and let L be $\lceil \log_{1+\alpha} h\rceil$. If C is the class of intervals on the line, then the VC-dimension of C_k is $2k$, and so the number of different pricing functions in $F_k(C)$ over S is at most $L^k\left(\frac{en}{2k}\right)^{2k}$; also, if C is the class of all axis parallel rectangles in d dimensions, then the VC-dimension of C_k is $O(kd)$ [15]. In these cases we can remove the $\log k$ term in our bounds, which is nice because it means we can interpret our results (e.g., Corollary 6) as charging OPT a penalty for each market it creates. However, we do not know how to remove this $\log k$ term in general, since in general the VC-dimension of C_k can be as large as $2Dk\log(2Dk)$ (see [4, 13]).

Corollary 6 gives a guarantee in the revenue of RSOPF$_{F_k(C),\mathcal{A}}$ so long as we have enough bidders n. In the following, for $k \ge 0$ let $\mathrm{OPT}_k = \mathrm{OPT}_{F_k(C)}$. We can also use Theorem 1 and Corollary 2 to show a bound that holds for all n, but with an additive loss term (we assume for simplicity here that $\beta = 1$):

Theorem 7 *For any given value of* n, k, ϵ, *and* δ, *with probability at least* $1-\delta$, *the revenue of RSOPF*$_{F_k(C),\mathcal{A}}$ *is*

$$(1-\epsilon)\,\mathrm{OPT}_k - h\cdot r_F(k, D, h, \epsilon, \delta),$$

where $r_F(k, D, h, \epsilon, \delta) = O\left(\frac{kD}{\epsilon^2}\ln\left(\frac{kDh}{\epsilon\delta}\right)\right)$.

Proof Sketch: We will prove the bound with the "$(1-\epsilon)$" term replaced by $\min\left(\frac{(1-\epsilon')^2}{1+\epsilon'}, 1-2\epsilon'\right)$, which then implies our desired result using $\epsilon' = \epsilon/3$. If $n \ge \frac{36h}{\epsilon'^2}\left[\ln\left(\frac{2}{\delta}\right) + k\ln\left(\frac{1}{\epsilon'}\ln h\right) + kD\ln\left(\frac{36kh}{\epsilon'^2}\right)\right]$, then the desired statement follows directly from Corollary 6. Otherwise, consider first the case when we have $\mathrm{OPT}_k \ge \frac{4h}{\epsilon'^2(1-\epsilon')}\left[\ln\left(\frac{2}{\delta}\right) + k\ln L + kD\ln\left(\frac{ne}{D}\right)\right]$. Let g_i be the optimal pricing function in $F_k(C)$ over S_i, for $i = 1, 2$, and let g_{OPT} be the optimal pricing function in $F_k(C)$ over S (therefore we have $g_i(S_i) \ge g_{\mathrm{OPT}}(S_i)$). From Corollary 1, we have $g_{\mathrm{OPT}}(S_i) \ge \frac{2h}{\epsilon'^2}\left[\ln\left(\frac{2}{\delta}\right) + k\ln L + kD\ln\left(\frac{ne}{D}\right)\right]$, for $i = 1, 2$. This implies that $g_i(S_i) \ge \frac{2h}{\epsilon'^2}\left[\ln\left(\frac{2}{\delta}\right) + k\ln L + kD\ln\left(\frac{ne}{D}\right)\right]$. Using again Corollary 1, we obtain that $g_i(S_j) \ge \frac{1-\epsilon'}{1+\epsilon'}g_i(S_i)$ for $j \ne i$, which then implies the desired result. To complete the proof just notice that if both $\mathrm{OPT}_k \le \frac{4h}{\epsilon'^2(1-\epsilon')}\left[\ln\left(\frac{2}{\delta}\right) + k\ln L + kD\ln\left(\frac{ne}{D}\right)\right]$ and $n \le \frac{4h}{\epsilon'^2}\left[\ln\left(\frac{2}{\delta}\right) + k\ln\left(\frac{2}{\epsilon'}\ln h\right) + kD\ln\left(\frac{4kh}{\epsilon'^2}\right)\right]$, then we easily get the desired statement. ∎

Finally, as in Theorem 2 we can extend our results to use Structural Risk Minimization, where we want the algorithm to optimize over k, by viewing the additive loss term as a penalty function.

Theorem 8 *Let $\bar{\mathcal{G}}$ be the sequence of pricing function classes $F_1(C), F_2(C), \ldots, F_n(C)$, and let $\mathsf{pen}(F_k(C))$ be the additive-loss term below. Then for any value of n, ϵ and δ with probability $1-\delta$ the revenue of RSOPF-SRM$_{\bar{\mathcal{G}},\mathsf{pen}}$ is*

$$\max_k \left((1-\epsilon)\,\mathrm{OPT}_k - h \cdot r'_F(k, D, h, \epsilon, \delta)\right),$$

where $r'_F(k, D, h, \epsilon, \delta) = O\left(\frac{kD}{\epsilon^2}\ln\left(\frac{kDh}{\epsilon\delta}\right)\right)$.

To illustrate the relevance of Theorem 7, notice that even for the special case of pricing using interval functions (the case of $d = 1$ studied in [6]), the following lower bound holds.

Theorem 9 *For the case that C is the class of intervals on the line, there is no incentive compatible mechanism whose expected revenue is at least $\frac{3}{4}\,\mathrm{OPT}_k - o(kh)$.*

Proof: Consider $kh/2$ bidders with *distinct* attributes[8], $h/2$ each of whom independently has a $1/h$ probability of having valuation h and a $1 - 1/h$ probability of having valuation 1. Then, any incentive-compatible mechanism has expected profit at most $kh/2$ because for any given bidder and any given proposed price, the expected profit (over randomization in the bidder's valuation) is at most 1. However, there is at least a 50% chance we will have at least $k/2$ bidders of valuation h, and in that case OPT_k can give $k/2 - 1$ of those bidders a price of h and the rest a price of 1 for an expected profit of $(k/2-1)h + (kh/2 - k/2 + 1)1 = kh - h - k/2 + 1$. On the other hand even if that does not occur, we always have $\mathrm{OPT}_k \geq kh/2$. So, the expected profit of OPT_k is at least $3kh/4 - h/2 - k/4$. Thus no incentive-compatible mechanism can have profit at least $\frac{3}{4}\,\mathrm{OPT}_k - o(kh)$. ■

A similar lower bound holds for most base classes; note also for the case of intervals on the line, an auction in [6] essentially matches this lower bound.

5.2 General Pricing Functions over the Attribute Space

In this section we generalize the results in Section 5.1 in two ways: to general classes of pricing functions (not just piecewise-constant functions defined over markets) and by removing the need for discretization by using covering arguments (that we discussed in Section 3.3.3). For example, we might want to consider a comparison class of linear functions over the attributes, or quadratic functions, or perhaps functions that divide the space into markets and are linear (rather than constant) in each market.

Assume in the following that $\mathcal{X} \subseteq \mathbb{R}^d$, and let $\mathcal{G}$ be a fixed class of pricing functions over the attribute space $\mathcal{X}$. Let $\mathcal{G}_d$ be the class of decision surfaces (in $\mathbb{R}^{d+1}$) induced by $\mathcal{G}$: that is, to each $g \in \mathcal{G}$ we associate the set of all $(x, v) \in \mathcal{X} \times [1, h]$ such that $g(x) \leq v$. Also, let us denote by D the VC-dimension of class $\mathcal{G}_d$ (i.e., $D = VCdim(\mathcal{G}_d)$), and let's assume that $D < \infty$. Then using Corollary 4 we can show that:

Theorem 10 *Given comparison class $\mathcal{G}$ and a β-approximation algorithm $\mathcal{A}$ for optimizing over $\mathcal{G}$, then so long as $\mathrm{OPT}_{\mathcal{G}} \geq \beta n$ and the number of bidders n satisfies*

$$n \geq \frac{72h}{\epsilon^2}\left[\ln\left(\frac{2}{\delta}\right) + D\ln\left[\frac{ne}{D}\left(\frac{12}{\epsilon}\ln h + 1\right)\right]\right]$$

then with probability at least $1-\delta$, the profit of RSOPF$_{(\mathcal{G},\mathcal{A})}$ is at least $(1-\epsilon)\,\mathrm{OPT}_{\mathcal{G}}/\beta$.

[8] Assume for instance that bidder i has attribute $pub_i = i$.

Proof Sketch: Let $\alpha = \frac{\epsilon}{12}$. For each bidder (x, v) we conceptually introduce $O(\frac{1}{\alpha} \ln h)$ "phantom bidders" having the same attribute value x and bid values $1, (1+\alpha), (1+\alpha)^2, \cdots, h$. Let S^* be the set S together with the set of all phantom bidders; let $n^* = |S^*|$. Let $Split$ be the set of possible splittings of S^* with surfaces from $\mathcal{G}_d$. We clearly have $|Split| \leq \mathcal{G}_d[n^*]$. For each element $s \in Split$ consider a representative function in $\mathcal{G}$ that induces splitting s in terms of its winning bidders, and let $Split_{\mathcal{G}}$ be the set of these representative functions. Now notice that $Split_{\mathcal{G}}$ is actually an L_∞ multiplicative α-cover for $\mathcal{G}$ with respect to S, since for every function in $\mathcal{G}$ there is a function in $Split_{\mathcal{G}}$ that extracts nearly the same profit from every bidder in the L_∞ multiplicative sense; i.e. for every function in $g \in \mathcal{G}$, there exists $g' \in Split_{\mathcal{G}}$ such that for every $(x, v) \in S$, we have both $g'((x, v)) \leq (1 + \alpha)g((x, v))$ and $g((x, v)) \leq (1 + \alpha)g'((x, v))$.

From Sauer's lemma we know $|Split_{\mathcal{G}}| \leq \left(\frac{n^* e}{D}\right)^D$, and applying Corollary 4, we finally get the desired statement. ■

Finally, using simple algebra (to remove the "n" on the RHS) we obtain:

Theorem 11 *Given comparison class $\mathcal{G}$ and a β-approximation algorithm $\mathcal{A}$ for optimizing over $\mathcal{G}$, then so long as $\mathrm{OPT}_{\mathcal{G}} \geq \beta n$ and the number of bidders n satisfies*

$$n \geq \frac{154h}{\epsilon^2}\left[\ln\left(\frac{2}{\delta}\right) + D\ln\left(\frac{154h}{\epsilon^2}\left(\frac{12}{\epsilon}\ln h + 1\right)\right)\right],$$

then with probability at least $1-\delta$, the profit of RSOPF$_{(\mathcal{G},\mathcal{A})}$ is at least $(1-\epsilon)\,\mathrm{OPT}_{\mathcal{G}}/\beta$.

The above theorem is the analog of Corollary 2. Using it and Theorem 4, we can then derive (in the same way as we did for Theorem 7) a bound that holds for all n (i.e. the analogue of Theorem 7). We can further extend the results here to get bounds for the corresponding SRM auction (as we did for Theorem 8).

6 Combinatorial Auctions

Combinatorial auctions have received much attention in recent years because of the difficulty of merging the complexity issue of computing an optimal outcome with the game-theoretic issue of incentive compatibility. To date almost exclusively the focus has been on socially optimal combinatorial auctions.[9] Deviating from this literature, we look at the goal of profit maximization of the seller in the case where the items for sale are available in unlimited supply. We consider the general version of the combinatorial auction problem as well as the special cases of *unit-demand* bidders (each who desires only singleton bundles) and *single-minded* bidders (each of whom has a single desired bundle).

It is interesting to restrict our attention to the case of item-pricing, where the auctioneer intuitively is attempting to set a price for each of the distinct items and bidders then choose their favorite bundle given these prices. Item-pricing is without loss of generality for the unit-demand case, and the general bundle-pricing can be realized with an auction with $m' = 2^m$ "items", one for each of possible bundle of the original m items.[10]

[9] A notable exception is the recent work of Likhodedov and Sandholm [27] which gives both a randomized auction that is a $O(\log h)$-approximation in worst case and a deterministic auction that is an $O(\log h)$ *average case* approximation to the optimal revenue not only in the unlimited supply case that we consider here, but also in the important limited supply special case where the bidders have *additive valuations*. They also present a number of simulations that show the usefulness of their techniques.

[10] We make the assumption that all desired bundles contain at most one of each item. This assumption can be easily relaxed and our results applied given any bound on the number of copies of each item that are desired by any one consumer. Of course this reduction produces an exponential blowup in the number of items.

For combinatorial auctions, the size of the class of all possible item-pricings, $|\mathcal{G}|$, is infinite. Following the guidelines established in Section 3.3 we look at obtaining bounds for a discretized set of item prices, $\mathcal{G}'$ (see Section 6.1), and bounds obtained from counting possible outcomes in $\mathcal{G}_\mathcal{A}$ (see Section 6.2). A summary of our results is given in Table 1.

	general	unit-demand	single-minded
$\lvert\mathcal{G}'\rvert$	$O(\log^m_{1+\epsilon^2} \frac{nm}{\epsilon})$	$O(\log^m_{1+\epsilon^2} \frac{n}{\epsilon})$	$O(\log^m_{1+\epsilon} \frac{nm}{\epsilon})$
$\lvert\mathcal{G}_\mathcal{A}\rvert$	$n^m 2^{2m^2}$	$n^m(m+1)^{2m}$	$(n+m)^m$

Table 1: Size of comparison classes for combinatorial auctions.

We can apply Theorem 1 and Corollary 2 to the sizes of the complexity classes in Table 1 to get good bounds on the profit of random sampling auctions for combinatorial item pricing. In particular, using Corollary 2 we get that $\tilde{O}(hm^2/\epsilon^2)$ bidders are sufficient to achieve revenue close to the optimum item-pricing in the general case, and $\tilde{O}(hm/\epsilon^2)$ bidders are sufficient for the unit-demand case. Also, by using Theorem 1 instead of Corollary 2 we can replace the condition on the number of bidders with a condition on $\mathrm{OPT}_\mathcal{G}$, which is factor of m improvement on the bound given by [16].

6.1 Bounds via Discretization

We can obtain good performance bounds if we are willing to optimize over a small class of discretized item-pricings (see Section 3.3.1). In particular, if we can find a small class $\mathcal{G}'$ with the property that $\mathrm{OPT}_{\mathcal{G}'}$ is guaranteed to be close to $\mathrm{OPT}_\mathcal{G}$, we can argue that $\mathrm{RSOPF}_{(\mathcal{G}',\mathcal{A})}$ performs well compared to $\mathrm{OPT}_\mathcal{G}$ using bounds on the size of $|\mathcal{G}'|$. Prior to this work, [23] shows how to construct discretized classes $\mathcal{G}'$ of price vectors with $\mathrm{OPT}_{\mathcal{G}'} \geq \frac{1}{1+\epsilon}\mathrm{OPT}_\mathcal{G}$ and that are of sizes $O(m^m \log^m_{1+\epsilon} \frac{n}{\epsilon})$ for the unit-demand case and $O(\log^m_{1+\epsilon} \frac{nm}{\epsilon})$ for the single-minded case. Nisan [28] gives the basic argument necessary to generalize these results to obtain the result in Theorem 12 which applies to combinatorial auctions in general. We note in passing that Theorem 12 allows for generalization and improvement of the computational results of [23]. The discretization results we obtain are summarized in the first row of Table 1.

We state and prove now the main result of this section.

Theorem 12 *Let k be the size of the maximum desired bundle. Let $\mathbf{p}'$ be the optimal discretized price vector that uses item prices equal to 0 or powers of $(1+\epsilon)$ in the range $[h\epsilon/nk, h]$ and let $\mathbf{p}^*$ be the optimal price vector. Then we have:*

$$\mathbf{p}'(S) \geq (1 - 2\sqrt{\epsilon})\mathbf{p}^*(S).$$

Proof: Consider $\delta = \sqrt{\epsilon}$. For the optimal price vector $\mathbf{p}^*$ with item j priced at p^*_j (i.e. $\mathbf{p}^*(S) = \mathrm{OPT}_\mathcal{G}$), consider a price vector $\mathbf{p}$ with p_j in $[(1-\delta)p^*_j, (1-\delta+\delta^2)p^*_j]$ if $p^*_j \geq h\delta^2/nk$ and 0 otherwise. Note that such a price vector $\mathbf{p}$ lies in the set of price vectors that have item prices equal to 0 or powers of $(1+\epsilon)$ in the range $[h\epsilon/nk, h]$. We show now that $\mathbf{p}(S) \geq (1-2\sqrt{\epsilon})\mathbf{p}^*(S)$ holds, which clearly implies the desired result.

Let J be a multi-set of items and $\mathrm{Profit}(J) = \sum_{j \in J} p^*_j$ be the payment necessary to purchase bundle J under pricing $\mathbf{p}^*$. Define $R_j = p^*_j - p_j$. Thus we have:

$$(\delta - \delta^2)p^*_j \leq R_j \leq \delta p^*_j + \delta^2 h/nk.$$

This implies that for any multiset J with $|J| \leq k$, we have the following upper and lower bounds:

$$\sum_{j \in J} R_j \geq (\delta - \delta^2)\text{Profit}(J)\ , \tag{1}$$

$$\sum_{j \in J} R_j \leq \delta\text{Profit}(J) + h\delta^2/n. \tag{2}$$

Let J_i^* and J_i be the bundles that bidder i prefers under pricing $\mathbf{p}^*$ and $\mathbf{p}$, respectively. Consider bidder i who switches from bundle J_i^* to bundle J_i when the item prices are decreased from $\mathbf{p}^*$ to $\mathbf{p}$. This implies that:

$$\sum_{j \in J_i^*} R_j \leq \sum_{j \in J_i} R_j.$$

Combining this with equations (1) and (2) and canceling a common factor of δ we see that:

$$(1 - \delta)\text{Profit}(J_i^*) \leq \text{Profit}(J_i) + h\delta/n.$$

Summing over all bidders i, we see that the total profit under our new pricing $\mathbf{p}$ is at least $(1-\delta)\,\text{OPT}_{\mathcal{G}} - h\delta$. Since $\text{OPT}_{\mathcal{G}} \geq h$, we finally obtain that the profit under $\mathbf{p}$ is at least $(1 - 2\delta)\,\text{OPT}_{\mathcal{G}}$.[11] ■

Note that we can now apply Theorem 12 by letting $\mathcal{G}'$ be the class of item prices equal to 0 or powers of $(1 + \epsilon)$ in the range $[h\epsilon/nk, h]$ (where k bounds the maximum size of a bundle). Using for instance Corollary 2 we obtain the following guarantee:

Corollary 7 *Given a β-approximation algorithm $\mathcal{A}$ optimizing over $\mathcal{G}'$, then so long as $\text{OPT}_{\mathcal{G}'} \geq \beta n$ and the number of bidders n satisfies*

$$n \geq \frac{18h}{\epsilon^2}\left(m \ln(\log_{1+\epsilon^2} nk) + \ln\left(\frac{2}{\delta}\right)\right),$$

then with probability at least $1 - \delta$, the profit of RSOPF$_{\mathcal{G}',\mathcal{A}}$ is at least $(1 - 3\epsilon)\,\text{OPT}_{\mathcal{G}}/\beta$.

6.2 Bounds via Counting

We now show how to use the technique of counting possible outcomes (See Section 3.3.2) to get a bound on the performance of the random sampling auction with an algorithm $\mathcal{A}$ for item-pricing. This approach calls for bounding $|\mathcal{G}_{\mathcal{A}}|$, the number of different pricing schemes RSOPF$_{(\mathcal{G},\mathcal{A})}$ can possibly output. Our results for this approach are summarized in the second row of Table 1.

Recall that bidder i's utility for a bundle J given pricing $\mathbf{p}$ is $u_i = v_i(J) - \sum_{j \in J} p_j$ (this is specified by ρ). We now make the following claim about the regions of the space of possible pricings, $\mathbb{R}_+^m$, in which bidder i's most desired bundle is fixed.

Claim 1 *A bidder's valuation function over subset of items, $v_i(J)$, partitions the space of item-pricings into convex regions based on the bundle J allocated to the bidder.*

[11] Notice that we are effectively assuming that $h = \max_{i \in S} \max_{s \subseteq S} v_i(s)$.

Proof: Suppose the allocation to a particular bidder for $\mathbf{p}$ and $\mathbf{p}'$ are the same, J. Then for any other bundle J' we have:

$$v_i(J) - \sum_{j \in J} p_j \geq v_i(J') - \sum_{j \in J'} p_j$$

and

$$v_i(J) - \sum_{j \in J} p'_j \geq v_i(J') - \sum_{j \in J'} p'_j.$$

If we now consider any price vector $\alpha\mathbf{p} + (1-\alpha)\mathbf{p}'$, for $\alpha \in [0,1]$, these imply:

$$v_i(J) - \sum_{j \in J} (\alpha p_j + (1-\alpha)p'_j) \geq v_i(J') - \sum_{j \in J'} (\alpha p_j + (1-\alpha)p'_j).$$

This clearly implies that this agent prefers allocation J on any convex combination of $\mathbf{p}$ and $\mathbf{p}'$. Hence the region of prices for which the agent prefers bundle J is convex. ■

The above claim shows that we can divide the space of pricings into convex regions based on an agents most desirable bundle. Consider fixing an outcome, i.e., the bundles $J_1, \ldots, J_n$, obtained by the n agents. This outcome arises for pricings that are in the intersection over agents i, of set of pricings where agent i obtains bundle J_i, which is clearly also a convex region. Since different outcomes partition the space of possible pricings, these convex regions are polytopes joined by hyperplanes.

Definition 4 *For agents S, let Verts$_S$ denote the set of vertices of the polytopes that partition the space of prices by the allocation produced.*

Claim 2 *For $S' \subseteq S$ we have Verts$_{S'} \subseteq$ Verts$_S$.*

Proof: We show the claim for $S' = S \setminus \{i\}$ and without loss of generality fix $i = 1$. The full claim then follows by induction.

The space of prices is partitioned into polytopes by the valuations of the $n-1$ agents $S' = \{2, \ldots, n\}$. Consider a particular allocation the the $n-1$ agents S': $J_2, \ldots, J_n$. This polytope is partitioned into polytopes by the valuation of agent 1 based on the bundle J_1 that agent 1 receives (i.e., by intersecting the polytope for J_1 with the polytope for $J_2, \ldots, J_n$). The vertices of these polytopes include all vertices of the original polytope for $J_2, \ldots, J_n$ and new vertices created when further partitioning this polytope by the allocation to agent 1. As this holds for all $J_2, \ldots, J_n$, it implies that the vertices of the polytopes for all allocations to the n agents, Verts$_S$, is a superset of the vertices of the polytopes for all allocations to the $n-1$ agents in S', Verts$_{S'}$. Induction gives the claim. ■

Now we consider optimal pricings. Note that when fixing an allocation $J_1, \ldots, J_n$ we are looking for an optimal price point within the polytope that gives this allocation. Our objective function for this optimization is linear. Let n_j be the number of copies of item j allocated by the allocation. The algorithms payoff for prices $\mathbf{p} = (p_1, \ldots, p_m)$ is $\sum_j p_j n_j$. Thus, all optimal pricings of this allocation lie on facets of the polytope and in particular there is an optimal pricing that is at a vertex of the polytope. Over the space of all possible allocations, all optimal pricings are on facets of the allocation defining polytopes and there exists an optimal pricing that is at a vertex of one of the polytopes.

Lemma 5 *Given an algorithm $\mathcal{A}$ that always outputs a vertex of the polytope then $\mathcal{G}_\mathcal{A} \subseteq$ Verts$_S$.*

Proof: This follows from the fact that $\text{RSOPF}_{(\mathcal{G},\mathcal{A})}$ runs $\mathcal{A}$ on a subset S' of S which has $\text{Verts}_{S'} \subset \text{Verts}_S$. $\mathcal{A}$ must pick a price vector from $\text{Verts}_{S'}$. By Claim 2 this price vector must also be in Verts_S. This gives the lemma. ■

We now discuss getting a bound on Verts_S for n agents, m distinct items, and various types of preferences.

Theorem 13 *We have the following upper bounds on $|\text{Verts}_S|$:*

1. $(n+m)^m$ *for single-minded preferences.*
2. $n^m(m+1)^{2m}$ *for unit-demand preferences.*
3. $n^m 2^{2m^2}$ *for arbitrary preferences.*

Proof: We consider how many possible bundles, M, an agent might obtain as a function of the pricing. An agent with single-minded preferences will always obtain one of $M_s = 2$ bundles: either they obtain their desired bundle or they receive nothing (the empty bundle). An agent with unit-demand preferences receives one of the m items or nothing for a total of $M_u = m+1$ possible bundles. An agent with general preferences receives one of the $M_g = 2^m$ possible bundles.[12]

We now bound the number of hyperplanes necessary to partition the pricing space into M convex regions (e.g., that specify which bundle the agent receives). For convex regions, each pair of regions can meet in at most one hyperplane. Thus, the total number of hyperplanes necessary to partition the pricing space into regions is at most $\binom{M}{2}$. Of course we wish to restrict our pricings to be non-negative, so we must add m additional hyperplanes at $p_j = 0$ for all j.

For all n agents, we simply intersect the regions of all agents. This does not add any new hyperplanes. Furthermore, we only need to count the m hyperplanes that restrict to non-negative pricings once. Thus, the total number of hyperplanes necessary for specifying the regions of allocation for n agents with M convex regions each, is $K = n\binom{M}{2} + m$. Thus, $K_s = n + m$, $K_u \leq n\binom{m+1}{2} + m \leq n(m+1)^2$, and $K_g \leq n\binom{2^m}{2} + m \leq n2^{2m}$ (for $m \geq 2$).

Of course, K hyperplanes in m dimensional space intersect in at most $\binom{K}{m} \leq K^m$ vertices. Not all of these intersections are vertices of polytopes defining out allocation, still K^m is an upper bound on the size of Verts_S. Plugging this in gives us the desired bounds of $(n+m)^m$, $n^m(m+1)^{2m}$, and $n^m 2^{2m^2}$ respectively for single-minded, unit-demand, and general preferences. ■

We note that are above arguments apply to approximation algorithms that always output a price corresponding to the vertex of a polytope as well. Though we do not consider this direction here, it is entirely possible that it is not computationally difficult to post-process the solution of an algorithm that is not a vertex of a polytope to get a solution that is on a vertex of a polytope. This would further motivate the analysis above. If for some reason, restricting to algorithms that return vertices is undesirable, it is possible to use cover arguments on the set of vertices we obtain when we add additional hyperplanes corresponding to the discretization of the preceding section.

[12] Here we make the assumption that desired bundles are simple sets. If they are actually multi-sets with bounded multiplicity k, then the agent could receive one of at most $M_g = (k+1)^m$ bundles.

6.3 Combinatorial Auctions: Lower Bounds

We show in the following an interesting lower bound for combinatorial auctions.[13] Notice that our upper bounds and this lower bound are quite close.

Theorem 14 *For agents with unit-demand, single-minded, or general preferences, there is no randomized incentive compatible mechanism whose revenue is* $\Omega(\mathrm{OPT} - o(mh))$.

Proof: Consider the following probability distribution over valuations of agents preferences. Assume we have $n = mh/2$ agents in total, and $h/2$ agents desire item j only, $j \in \{1, \cdots m\}$. [14] Each of these agents has valuation h with probability $1/h$ and valuation 1 with probability $1 - 1/h$.

Notice now any incentive-compatible mechanism has expected profit at most n. To see this, note that for each bidder, any proposed price has expected profit (over the randomization in the selection of his valuation) of at most 1. Moreover, the expected profit of $\mathrm{OPT}_{\mathcal{G}}$ is at least $n + mh/8$. For each item j, there is at least a $1/4$ chance that some bidder has valuation h. For those items, $\mathrm{OPT}_{\mathcal{G}}$ gets at least a profit of h. For the rest, $\mathrm{OPT}_{\mathcal{G}}$ gets a profit of $h/2$. So, overall, $\mathrm{OPT}_{\mathcal{G}}$ gets an expected profit of at least $mh/4 + (3/4)h/2 = n + mh/8$. All these together imply the desired result. ■

6.4 Algorithms for Item-pricing

Given standard complexity assumptions, most item-pricing problems are not polynomial time solvable, even for simple special cases. We review these results here. We restrict our attention to the unlimited supply special case, though some of the work we mention also considers limited supply item-pricing. Algorithmic pricing problems in this form were first posed by Guruswami et al. [29] though item-pricing for unit-demand consumers with several alternative payment rules (i.e., non-standard functions ρ mapping offers to payments) were independently considered by Aggarwal et al. [1].

For consumers with single-minded preferences, [29] gives a simple logarithmic approximation algorithm. Demaine et al. [11] show that this algorithm is essentially the best possible by showing the problem to be hard to approximate better than a logarithmic factor.[15] Both Briest and Krysta [10] and Grigoriev et al. [22] proved that optimal pricing is NP-hard for the special case known as "the highway problem" where there is a linear order on the items and all desired bundles are for sets of consecutive items (actually this hardness result follows for the more specific case where the desired bundles for any two agents, S_i and $S_{i'}$, satisfy one of the following: $S_i \subset S_{i'}$, $S_{i'} \subset S_i$, or $S_i \cup S_{i'} = \emptyset$). In the case when the cardinality of the desired bundles are bounded by k, independently Briest and Krysta [10] and Balcan and Blum [3] provided approximation algorithms with good guarantees. Specifically, Briest and Krysta [10] provided an $O(k^2)$-approximation algorithm, while Balcan and Blum [3] provided an $O(k)$-approximation algorithm.[16] Finally, when the number of distinct items for sale, m, is constant, Hartline and Koltun [23] show that it is possible to improve on the trivial $O(n^m)$ algorithm by giving a near-linear time approximation scheme. Their approximation algorithm is actually an exact algorithm for the problem of optimizing over a discretized set of item prices $\mathcal{G}'$ which is directly applicable to our auction $\mathrm{RSOPF}_{(\mathcal{G}',\mathcal{A})}$, discussed above.

For consumers with unit-demand preferences, [29] (and [1] essentially) give a trivial logarithmic approximation algorithm and show that the optimization problem is APX-hard (meaning that standard complexity

[13] This proof follows the standard approach for lower bounds for revenue maximizing auctions that was first given by Goldberg et al. in [19].

[14] Notice that these preferences are both unit-demand and single-minded.

[15] Technically, the lower bound is logarithmic in m, whereas the upper bound is $O(\log m + \log n)$.

[16] Moreover, Balcan and Blum [3] showed how to adapt their algorithms to the *online* setting.

assumptions imply that there does not exist a polynomial time approximation scheme (PTAS) for the problem). Again, Hartline and Koltun show how to improve on the trivial $O(n^m)$ algorithm in the case where the number of distinct items for sale, m, is constant. They give a near-linear time approximation scheme that is based on considering a discretized set of item prices; however, discretization of Nisan [28] discussed above gives a significant improvement on their algorithm and also generalizes it to be applicable to the problem of item-pricing for consumers with general combinatorial preferences.

7 Multicast Pricing

In the multicast pricing problem, each bidder resides at some node of a tree, and in order to sell its service to some bidder, the service-provider must have purchased all edges on the path from the root to that vertex. Given a set of edge costs, our goal as service-provider is to determine a subtree together with prices at nodes of this tree that achieves highest revenue minus cost. A 4-approximation to this problem, under the assumption that the optimal solution has revenue at least 4 times its cost and that there is sufficient competition at each node is given in [14].

Using our generic results we can say that so long as the optimal solution has revenue at least $1/\epsilon$ times its cost, and we have on average $\tilde{O}(h/\epsilon^2)$ bidders at each node (using Theorem 1) or at least $\tilde{O}(h/\epsilon^2)$ revenue at each node (using Corollary 1) then we get a $(1 + O(\epsilon))$-approximation.

Briefly, to apply the generic results, we define our algorithm $\mathcal{A}$ so that it finds the revenue-maximizing tree but *only over* the subset of trees whose revenue on the given subset of bidders is at least $(2 + \epsilon)/\epsilon$ times its cost. By Corollary 1, with high probability the optimal tree has this property over both S_1 and S_2, and so the revenue achieved by $\mathcal{A}$ is nearly that of the optimal tree, and by design the cost of the tree produced by $\mathcal{A}$ is only an $O(\epsilon)$ factor of revenue.

We can also apply structural-risk-minimization in the case that the total number of bidders is not sufficient for the entire class of trees. In particular, one interesting case is the comparison-class of functions that choose some subtree and add fake "markups" between 0 and nh to the edges of that subtree, and then perform cost-sharing on the result (also add a "super-root" with a single zero-cost edge into the root). If we define $\mathcal{G}_k$ to be the set of such functions whose subtree has k edges, then $|\mathcal{G}_k| \leq (n \log_{1+\epsilon}(nh))^k$. We can then perform SRM using Theorem 2. An interesting special case to consider is a simple depth-1 multicast tree whose edges have cost 0 and with two bidders at each leaf: one with value 1 and one with value h. In this case, there is not sufficient competition at the leaves for the results of [14], but we can extract $\Omega(nh)$ using $\mathcal{G}_1$.

8 Conclusions and Discussion

In this work we have made the connection between machine learning and mechanism design explicit. In doing so, we obtain a *unified* approach to considering a variety of profit maximizing mechanism design problems including many that have been previously considered in the literature.

Some of our techniques give suggestions for the *design* of mechanisms and others for their *analysis*. In terms of design, these include the use of discretization to produce smaller function classes, and the use of structural-risk-minimization to choose an appropriate level of complexity of the mechanism for a given set of bidders. In terms of analysis, these include both the use of basic sample-complexity arguments, and the notion of multiplicative covers for better bounding the true complexity of a given set of functions.[17]

[17] It is worth noting that using covering numbers is a common technique in deriving sample complexity bounds in Machine

Our bounds on random sampling auctions for digital goods [21] not only show how the auction profit approaches the optimal profit, but also weaken the required assumptions by a constant factor. Similarly for random sampling auctions for multiple digital goods [16] our unified analysis gives a bound that approaches the optimal profit with assumptions weakened by a factor of more than m, the number of distinct items. This multiple digital good auction problem is a special case of the a more general unlimited supply combinatorial auction problem for which we obtain the first positive worst-case results by showing that it is possible to approximate the optimal profit with an incentive-compatible mechanism. Furthermore, unlike the case for combinatorial auctions for social welfare maximization, our incentive-compatible mechanisms can be based on approximation algorithms instead of exact ones.

We have also explored the attribute auction problem proposed in [6], a special case of general profit maximizing mechanism design, in a very general setting: the attribute values can be multi-dimensional and the target pricing functions considered can be arbitrarily complex. We bound the performance of random sampling auctions as a function of the complexity of the target pricing functions. Our attribute auction results can be used for more general problems such as multicast pricing, where there is a cost to be paid by the mechanism that is a function of its outcome.

Our random sampling auctions assume the existence of exact or approximate pricing algorithms. Solutions to these pricing problem have been proposed for several of our settings. In particular, optimal item-pricings for combinatorial auctions in the single-minded and unit-demand special cases have been considered in [3, 10, 23, 29]. On the other hand for attribute auctions, many of the clustering and market-segmenting pricing algorithms have yet to be considered at all.

Probably the most important direction for future work is in relaxing the assumption that the items for sale are available in unlimited supply. In the random sampling framework, we propose the following mechanism: randomly partition the bidders into two sets, evenly divide the items among the two sets, compute the optimal *envy-free*[18] pricing function for the two partitions, and applying the pricing function to the opposite partition. Of course, a pricing function g that is envy-free for S_1 may not necessarily be envy-free for S_2. There are several approaches that may work here. First, we could artificially deplete the supply by a constant factor and ask for an pricing function that is envy-free for the depleted supply. Then it may be possible to argue that it is envy-free for both S_1 and S_2 with high probability. Another option would be to take the bidders of S_1 in an arbitrary (or random) order and allow them to take an item if they desire one. When we run out of items, stop. The remaining bidders get none, whether they want one or not. It is easy to see that the technique outlined above results in an incentive compatible mechanism. Is it also close to optimal?

It is possible to further generalize the feasibility constraints imposed by limited supply to arrive at the general single-parameter agent auction problem (See e.g., [18] for a precise definition). This abstract problem can be viewed as auctioning a service to a number of agents where the service provider must pay a cost that is a function of the agents served. In its full generality, this cost function could be arbitrary. Note that the multicast pricing problem is a special case of this problem where the cost function is defined by a tree. The possibly asymmetric cost function can be viewed as endowing the agents with public attributes, or the agents could have additional attributes. A very interesting direction for future research is in determining for what classes of cost functions the general problem of profit maximization in this setting can be solved.

The final direction of investigation we propose is that of generalizing the special purpose bounds we obtain for digital good auctions (Section 4) to our general unlimited supply setting (Section 3). Recall

Learning and this was our source of inspiration. However, it turned out that the *right* notion of cover for our mechanism design setting is a very specific one and quite different from what one would normally consider in Machine Learning.

[18] To generalize envy-freedom [29] to attribute auctions, declare a price function $g \in \mathcal{G}$ envy-free for bidders S if there are enough items such that all bidders that have strictly positive utility for an item under g can simultaneously be sold one.

that in for digital goods and indistinguishable bidders we were able to employ a telescoping argument to reduce the additive loss term to $O(h)$ which is optimal up to a constant factor. This takes advantage of the property of single-price pricing functions: that the payoff for any given bidder is upper-bounded by the offer price. This allows us to use non-uniform bounds on the payoffs of the different pricing functions and these non-uniform bounds telescope. Can some form of this telescoping be generalized to attribute auctions, combinatorial auctions, or our general bounds? It would be also interesting to see if one can use some of the very recent techniques and ideas used in the context of Learning Theory and Empirical Processes (see e.g. [9, 5, 26]) to get better bounds for our mechanism design setting. In particular, it would be interesting to investigate data dependent bounding techniques in this setting.

References

[1] G. Aggarwal, T. Feder, R. Motwani, and A. Zhu. Algorithms for multi-product pricing. In *Proceedings of the International Colloquium on Automata, Languages, and Programming*, pages 72–83, 2004.

[2] M. Anthony and P. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

[3] M.-F. Balcan and A. Blum. Approximation Algorithms for Item Pricing. Technical Report. CMU-CS-05-176, 2005.

[4] P. Bartlett and W. Maass. Vapnik Chervonenkis Dimension of Neural Nets. In *The Handbook of Brain Theory and Neural Networks*. MIT Press, 2003.

[5] P. Bartlett and S. Mendelson. Rademacher and Gaussian Complexities Risk Bounds and Structural Results. *Journal of Machine Learning Research*, 54(3):463–482, 2002.

[6] A. Blum and J. Hartline. Near-Optimal Online Auctions. In *Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms*, pages 1156 – 1163, 2005.

[7] A. Blum, V. Kumar, A. Rudra, and F. Wu. Online Learning in Online Auctions. In *Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms*, pages 137 – 146, 2003.

[8] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis Dimension. *Journal of the ACM*, 36:929– 965, 1989.

[9] O. Bousquet, S. Boucheron, and G. Lugosi. Theory of Classification: A Survey of Recent Advances. *ESAIM: Probability and Statistics*, 2005.

[10] P. Briest and P. Krysta. Single-Minded Unlimited Supply Pricing on Sparse Instances. In *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms*, 2006.

[11] E. Demaine, U. Feige, M.T. Hajiaghayi, and M. Salavatipour. Combination Can Be Hard: Approximability of the Unique Coverage Problem . In *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms*, 2006.

[12] L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1996.

[13] L. Devroye and G. Lugosi. *Combinatorial Methods in Density Estimation*. Springer-Verlag, 2001.

[14] A. Fiat, A. Goldberg, J. Hartline, and A. Karlin. Competitive Generalized Auctions. In *Proceedings 34th ACM Symposium on the Theory of Computing*, pages 72 – 81, 2002.

[15] P. Fische and S. Kwek. Minimizing Disagreement for Geometric Regions Using Dynamic Programming, with Applications to Machine Learning and Computer Graphics. 1996.

[16] A. Goldberg and J. Hartline. Competitive Auctions for Multiple Digital Goods. In *Proceedings of the 9th Annual European Symposium on Algorithms*, pages 416 – 427, 2001.

[17] A. Goldberg and J. Hartline. Competitiveness via Consensus. In *Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms*, pages 215 – 222, 2003.

[18] A. Goldberg and J. Hartline. Collusion-Resistant Mechanisms for Single-Parameter Agents. In *Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms*, pages 620 – 629, 2005.

[19] A. Goldberg, J. Hartline, A. Karlin, and M. Saks. A Lower Bound on the Competitive Ratio of Truthful Auctions. In *Proceedings 21st Symposium on Theoretical Aspects of Computer Science*, pages 644–655, 2004.

[20] A. Goldberg, J. Hartline, A. Karlin, M. Saks, and A. Wright. Competitive Auctions and Digital Goods. *Games and Economic Behavior*, 2002. Submitted for publication. An earlier version available as InterTrust Technical Report STAR-TR-99.09.01.

[21] A. Goldberg, J. Hartline, and A. Wright. Competitive Auctions and Digital Goods. In *Proceeding of the 12th ACM-SIAM Symposium on Discrete Algorithms*, pages 735–744, 2001.

[22] A. Grigoriev, J. van Loon, R. Sitters, and M. Uetz. How to Sell a Graph: Guideliness for Graph Retailers. Meteor Research Memorandum RM/06/001, Maastricht University, 2005.

[23] J. Hartline and V. Koltun. Near-Optimal Pricing in Near-Linear Time. In *Proceedings of the 9th Workshop on Algorithms and Data Structures*, pages 422–431, 2005.

[24] J. Hartline and R. McGrew. From Optimal Limited to Unlimited Supply Auctions. In *Proceedings of the 6th ACM Conference on Electronic Commerce*, pages 175 – 182, 2005.

[25] M. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

[26] V. Koltchinskii. Rademacher Penalties and Structural Risk Minimization. *IEEE Transactions of Information Theory*, 54(3):1902–1914, 2001.

[27] A. Likhodedov and T. Sandholm. Approximating Revenue-Maximizing Combinatorial Auctions. In *The Twentieth National Conference on Artificial Intelligence (AAAI)*, pages 267–274, 2005.

[28] N. Nissan. Personal communication, 2005.

[29] V. Guruswami, J. Hartline, A. Karlin, D. Kempe, C. Kenyon, F. McSherry. On Profit-Maximizing Envy-Free Pricing. In *Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms*, pages 1164 – 1173, 2005.

[30] V. Vapnik. *Statistical Learning Theory*. Springer-Verlag, 1998.

A Concentration Inequalities

Here is the McDiarmid inequality (see [12]) we use in our proofs:

Theorem 15 *Let $Y_1, ..., Y_n$ be independent random variables taking values in some set A, and assume that $t : A \rightarrow R$ satisfies:*

$$\sup_{y_1,...,y_n \in A, \overline{y}_i \in A} |t(y_1, ..., y_n) - t(y_1, ..., y_{i-1}, \overline{y}_i, y_{i+1}, y_n)| \leq c_i,$$

for all i, $1 \leq i \leq n$. Then for all $\gamma > 0$ we have:

$$\mathbf{Pr}\left\{|t(Y_1, ..., Y_n) - \mathbf{E}[t(Y_1, ..., Y_n)]| \geq \gamma\right\} \leq 2e^{\left[-\frac{2\gamma^2}{\sum_{i=1}^{n} c_i^2}\right]}$$

Here is also a consequence of the Chernoff bound that we used in Lemma 4.

Theorem 16 *Let $X_1, ..., X_n$ be independent Poisson trials such that, for $1 \leq i \leq n$, $\mathbf{Pr}[X_i = 1] = 1/2$ and let $X = \sum\limits_{i=1}^{n} X_i$. Then any n' we have:*

$$\mathbf{Pr}\left\{\left|X - \frac{n}{2}\right| \geq \epsilon \max\{n, n'\}\right\} \leq 2e^{\left[-2n'\epsilon^2\right]}$$

B VC Dimension and Its Properties

We briefly describe here the notion of VCdimension and some of its properties; for a more complete treatment see [2, 8, 25, 30].

We will first introduce some notation. Let C be a class of binary functions from X to $\{0, 1\}$. For any $S \subseteq X$, let us denote by $C[S]$ the set of all dichotomies on S realized by C; i.e. if S=$\{x_1, \cdots, x_m\}$, then $C[S] \subseteq \{0, 1\}^m$ and $C[S] = \{(c(x_1), \cdots, c(x_m)); c \in C\}$. Also, for any positive integer m, let $C[m]$ be the maximum number of ways to split m points from X using concepts in C; that means $C[m] = \max\{|C[S]|; |S| = m, S \subseteq X\}$. We say that $S = \{x_1, \cdots, x_m\}$ is *shattered* by C if every dichotomy of S has a representative in C (i.e. $|C[S]| = 2^m$). We can now define the notion of VC dimension as follows:

Definition 5 *The VC dimension of C is defined to be the size of the largest set S which is shattered by C; i.e. $VCdim(C) = \max\{|S|; S \subseteq X, \text{ S shattered by } C\}$.*

Then Sauer's lemma states that:

Theorem 17 *For any class C with finite $VCdim(C) = D$, we have $C[m] \leq \sum\limits_{i=0}^{D} \binom{m}{i}$, for all positive integers m.*

This further implies that:

Corollary 8 *For any class C with finite $VCdim(C) = D$, we have $C[m] = 2^m$ if $m \leq D$ and $C[m] \leq \left(\frac{em}{D}\right)^D$ if $m > D$.*

C Attribute Auctions: Partial Information

We analyze here Attribute Auctions in a Partial Information setting. In the following we assume that the bidders *do not reveal* their private value v_i, but the only observed signal is whether bidder i buys the item at a certain offer price or not. [19]

At a high level, the strategies we consider are of the following form. The auctioneer will divide the set of bidders into two groups, S_1 and S_2. He will use the bidders in S_1 to "learn" the distribution of values, by offering randomly different prices. After this, according to the values observed in S_1, he will decide on a specific pricing function, and use it on the bidders in S_2.

C.1 Constant Pricing

For clarity, we start with the simple case of a single market. Namely, the pricing functions are constant and from the set V, where V is the set of all prices of the form $(1+\alpha)^j$. Denote by $L = |V| = \lceil \log_{1+\alpha} h \rceil$.

We will consider two algorithms. Both split the set S randomly into S_1 and S_2. Let n_v be the number of winners at value v in S and let $r_v = v \cdot n_v$ denote $\text{profit}_{g_v}(S)$ for constant function $g_v(x) = v$. Also denote by $n_{v,i}$ be the number of winners at value v in S_i, for $i = 1, 2$ and let $r^* = r_{v^*} = \max_{v \in V} r_v$.

We describe now the first algorithm, `PI-uniform`. Let $C_1(\epsilon) = \frac{6}{\epsilon^2}$ and $C_2(\epsilon) = \frac{3}{\epsilon^2}(1+\epsilon)$. Algorithm `PI-uniform` first offers to each bidder in S_1 a price chosen at random from V. Specifically, for each $i \in S_1$, `PI-uniform` selects a random price p_i uniformly from V and offers bidder i the price p_i. Let m_v be the number of bidders in S_1 for which $p_i = v$. Let $\hat{n}_v$ be the subset of those m_v bidders for which $v_i \geq v$, namely the number of bidders i that bought when offered price $p_i = v$. A price p is called *considered* if $\hat{n}_p \geq C_2(\epsilon)A$, where $A = \log\left(\frac{2L}{\delta}\right)$. Let U be the set of considered prices and let $\bar{p} = \arg\max_{p \in U}\{\hat{n}_p p\}$. Finally, `PI-uniform` offers each bidder in S_2 the price $\bar{p}$ and its revenue on S_2 is $n_{\bar{p},2}\bar{p}$.

From the definition of `PI-uniform` we have that $E[\hat{n}_v] = \frac{1}{2L} n_v$, and $E[n_{v,i}] = \frac{n_v}{2}$, for $i = 1, 2$. Using again Chernoff bound we can prove that:

Lemma 6 *With probability at least $1-\delta$, for any $v \in V$, we have:*
(1) if $n_v \geq C_1(\epsilon)LA$ then we have $\frac{L\hat{n}_v}{n_v} \in \left[\frac{1}{2}(1-\epsilon), \frac{1}{2}(1+\epsilon)\right]$ and $\frac{n_{v,2}}{n_v} \in \left[\frac{1}{2}(1-\epsilon), \frac{1}{2}(1+\epsilon)\right]$.
(2) if $n_v < C_1(\epsilon)LA$ then we have $\hat{n}_v < C_2(\epsilon)A$.

Using Lemma 6, we can now derive the performance of `PI-uniform`.

Theorem 18 *For any set of bidders S, with probability at least $1-\delta$ the revenue of* `PI-uniform` *is at least $\min\{\frac{r^*}{2}(1-\epsilon), r^* - h \cdot d(\epsilon, \delta)\}$, where $d(\epsilon, \delta) = O\left(\frac{1}{\epsilon^2} L \log\left(\frac{2L}{\delta}\right)\right)$.*

Proof: We will prove a bound of $\min\{\frac{1}{2}\frac{(1-\epsilon')^2}{1+\epsilon'}r^*, r^* - \frac{2}{1-\epsilon'}C_2(\epsilon')hL\log\left(\frac{2L}{\delta}\right)\}$, which obviously implies the desired result. Let p^* be the optimal fixed price. If $n_{p^*} < \frac{2}{1-\epsilon'}C_2(\epsilon')LA$, then the theorem holds. Otherwise we have $\hat{n}_{p^*} \geq C_2(\epsilon')A$, and therefore the price p^* is considered and $\hat{n}_{p^*} \geq \frac{1-\epsilon'}{2} \cdot \frac{1}{2L} n_{p^*}$. For the selected price $\bar{p}$ we have that $\bar{p}\hat{n}_{\bar{p}} \geq p^* \hat{n}_{p^*}$; also since price $\bar{p}$ was *considered*, we have that $\hat{n}_{\bar{p}} \geq C_2(\epsilon')A$, and therefore $n_{\bar{p}} \geq C_1(\epsilon')LA$. This implies that $\hat{n}_{\bar{p}} \leq \frac{1}{2L}(1+\epsilon')n_{\bar{p}}$ and $n_{\bar{p},2} \geq n_{\bar{p}}\frac{1-\epsilon'}{2}$. This implies that $g_{\bar{p}}(S_2) = \bar{p}n_{\bar{p},2} \geq \frac{1-\epsilon'}{2}\bar{p}n_{\bar{p}} \geq \frac{1-\epsilon'}{2}\frac{2L}{1+\epsilon'}\bar{p}\hat{n}_{\bar{p}} \geq \frac{1-\epsilon'}{2}\frac{2L}{1-\epsilon'}p^*\hat{n}_{p^*} \geq \frac{1}{2}\frac{(1-\epsilon')^2}{1+\epsilon'}p^* n_{p^*} = \frac{1}{2}\frac{(1-\epsilon')^2}{1+\epsilon'}r^*$ which completes the proof. ■

[19] Remember, we consider the function ρ defined as follows: if bidder i is offered the item at price p, then he buys it iff $p \leq v_i$, and in the case when he buys the item the auctioneer's revenue is p.

The main objective of the second algorithm is to lower the penalty in the case the optimal revenue depends on a few bidders. The main idea is to sample more the higher prices. Let us assume for convenience that V is a power of $1+\alpha$, and let $V = \left\{\frac{h}{(1+\alpha)^i} | 0 \le i \le \log_{1+\alpha} h\right\}$. Let $C_3(\epsilon) = \frac{3}{\epsilon^2}\frac{1+\alpha}{\alpha}$ and let $C_4(\epsilon) = \frac{3}{\epsilon^2}$. The second algorithm `PI-expo`, for each $i \in S_1$ selects a random price $p_i = \frac{h}{(1+\alpha)^i}$ with probability $\frac{\alpha}{1+\alpha}\frac{1}{(1+\alpha)^i}$, and offers bidder i the price p_i. Let U be the set of prices $\{p_j | \hat{n}_{p_j} \ge C_4(\epsilon)A\}$. Algorithm `PI-expo` selects a price $\bar{p} \in U$ that maximizes $(1+\alpha)^i p\hat{n}_p$, where $p_i = \frac{h}{(1+\alpha)^i}$.

Clearly, for $v = \frac{h}{(1+\alpha)^i}$, using the price sampling of `PI-expo`, we have that $E[\hat{n}_v] = \frac{\alpha}{1+\alpha}\frac{n_v}{(1+\alpha)^i}$, and also $E[n_{v,i}] = n_v/2$, for $i = 1, 2$. Using Chernoff bound we can prove that:

Lemma 7 *With probability $1-\delta$ we have the following:*

(1) for any $v = \frac{h}{(1+\alpha)^i}$, if $n_v \ge C_3(\epsilon)(1+\alpha)^i A$, then we have $(1+\alpha)^i \frac{\hat{n}_v}{n_v} \in \left[\frac{\alpha}{1+\alpha}(1-\epsilon), \frac{\alpha}{1+\alpha}(1+\epsilon)\right]$ and $\frac{n_{v,2}}{n_v} \in \left[\frac{1}{2}(1-\epsilon), \frac{1}{2}(1-\epsilon)\right]$.

(2) for any $v = \frac{h}{(1+\alpha)^i}$, if $n_v < C_3(\epsilon)(1+\alpha)^i A$ we have $\hat{n}_v < C_4(\epsilon)A$.

Using Lemma 7, we can now derive the performance of the `PI-uniform` algorithm.

Theorem 19 *For any set of bidders S, with probability at least $1-\delta$ the revenue of* `PI-expo` *is at least* $\min\{r^*(\frac{1}{2}-\epsilon), r^* - \frac{1+\alpha}{\alpha}\frac{1}{1-\epsilon}C_4(\epsilon)h\log\frac{2L}{\delta}\}$.

Proof: Let $p^* = \frac{h}{(1+\alpha)^j}$ be the optimal fixed price. We analyze two cases depending on n_{p^*}. If $n_{p^*} < \frac{1+\alpha}{\alpha}\frac{1}{1-\epsilon}C_4(\epsilon)(1+\alpha)^j A$, then clearly the theorem holds. Consider now the case when $\hat{n}_{p^*} \ge C_4(\epsilon)A$. In this case, the price p^* is considered and also $\hat{n}_{p^*} \ge \frac{\alpha}{1+\alpha}(1-\epsilon)\frac{1}{(1+\alpha)^j}n_{p^*}$. Let $\bar{p} = \frac{h}{(1+\alpha)^i}$ be the selected price; then we clealry have $(1+\alpha)^i\bar{p}\hat{n}_{\bar{p}} \ge (1+\alpha)^j p^*\hat{n}_{p^*}$. Since $\bar{p} = \frac{h}{(1+\alpha)^i}$ was considered we also have that $\hat{n}_{\bar{p}} \ge C_4(\epsilon)A$, and therefore $n_{\bar{p}} \ge C_3(\epsilon)(1+\alpha)^i A$. This implies that $\hat{n}_{\bar{p}} \ge \frac{\alpha}{1+\alpha}(1-\epsilon)\frac{1}{(1+\alpha)^j}n_{\bar{p}}$ and $n_{\bar{p},2} \ge \frac{1-\epsilon}{2}n_{\bar{p}}$. All these imply that we have $g_{\bar{p}}(S_2) = \bar{p}n_{\bar{p},2} \ge \frac{1-\epsilon}{2}\bar{p}n_{\bar{p}} \ge \frac{1-\epsilon}{2}\frac{1+\alpha}{\alpha}\frac{1}{1-\epsilon}(1+\alpha)^i\bar{p}\hat{n}_{\bar{p}} \ge \frac{1-\epsilon}{2}\frac{1+\alpha}{\alpha}\frac{1}{1-\epsilon}(1+\alpha)^j\bar{p}\hat{n}_{\bar{p}}p^*\hat{n}_{p^*} \ge \frac{1-\epsilon}{2}p^*n_{p^*} = r^*\frac{1-\epsilon}{2}$, which completes the proof. ■

Also notice that (in both algorithms `PI-uniform` and `PI-expo`), by using an *uneven* partition of the bidders (just putting an ϵ fraction in S_1) we can get almost $(1-\epsilon)r^*$ (but in this case we will loose an extra $1/\epsilon$ in our additive term).

C.2 General Pricing Schemes

We now extend the result to a family of pricing function $\mathcal{G}$ (we analyze here for simplicity the discretized version of $\mathcal{G}$ to the powers of $1+\alpha$). In the following we denote by x_i the attribute of bidder i (that means $x_i = pub_i$). We perform the extension for the uniform price sampling. The algorithm `PI-`$\mathcal{G}$ does the following. Similar to `PI-uniform` splits the bidders randomly to S_1 and S_2 and offers each bidder in S_1 the price p_i uniform from V (as before, we consider $L = |V|$). After this initial step, `PI-`$\mathcal{G}$ computes, for each function $g \in \mathcal{G}$, an estimate $\hat{g}(S_1)$ in the following way. Let $S_{1,g}$ be the set of bidders in S_1 for which $g(x_i) = p_i$, namely the bidders for which the offered price equals the price suggested by g. The estimate $\hat{g}(S_1)$ is the revenue of g on $S_{1,g}$. The algorithm `PI-`$\mathcal{G}$ selects the function g^1 that maximizes $\hat{g}(S_1)$ and uses g^1 for pricing in S_2. Namely, for each $i \in S_2$ algorithm `PI-`$\mathcal{G}$ offers a price $g^1(x_i)$. It's easy to see that for any pricing function g we have that $E[\hat{g}(S_1)] = \frac{g(S)}{2L}$ and $E[g(S_2)] = \frac{g(S)}{2}$. Considering $C_5(\epsilon =)\frac{2}{\epsilon^2}$, then it is possible to show that:

Lemma 8 *With probability at least* $1-\delta$*, for any pricing function* $g \in \mathcal{G}$ *we have that the following holds: if* $g(S) \geq C_5(\epsilon)L^2 h \log(|\mathcal{G}[S]|/\delta)$*, then* $L\hat{g}(S_1)/g(S) \in \left[\frac{1}{2}(1-\epsilon), \frac{1}{2}(1+\epsilon)\right]$ *and* $g(S_2)/g(S) \in \left[\frac{1}{2}(1-\epsilon), \frac{1}{2}(1+\epsilon)\right]$.

Using Lemma 8, we can finally derive the following theorem.

Theorem 20 *For any set of bidders* S*, with probability at least* $1-\delta$ *the revenue of* `PI-`$\mathcal{G}$ *is at least* $min\{\frac{r^*}{2}(1-\epsilon), r^* - \frac{2}{1-\epsilon}C_5(\epsilon)L^3 h \log(|\mathcal{G}[S]|/\delta\}$*, where* $r^* = \max_{g \in \mathcal{G}} g(S)$.

Proof: Let g^* be the optimal pricing function. If $r^* < \frac{2}{1-\epsilon}C_5(\epsilon)L^3 h \log(|\mathcal{G}[S]|/\delta)$, then the theorem holds. Otherwise $g^*(S) \geq \frac{2}{1-\epsilon}C_5(\epsilon)L^3 h \log(|\mathcal{G}[S]|/\delta)$, which implies that $\hat{g}(S_1) \geq C_5 L^2 h \log(|\mathcal{G}[S]|/\delta)$. For the selected function g^1 we have that $\hat{g^1}(S_1) \geq \hat{g^*}(S_1)$. Since $g(S) \geq \hat{g}(S_1)$ we have that $g^1(S) \geq \hat{g^1}(S_1) \geq C_5(\epsilon)|L|^2 h \log(|\mathcal{G}[S]|/\delta)$. Therefore $\hat{g^1}(S_1) \leq g^1(S)\frac{1+\epsilon}{2L}$ and $g^1(S_2) \geq \frac{1-\epsilon}{2}g(S)$. This implies that $g^1(S_2) \geq \frac{1+\epsilon}{2}g^1(S) \geq L\hat{g^1}(S_1) \geq L\hat{g^*}(S_1) \geq \frac{1-\epsilon}{2}g^*(S)$, which completes the proof. ■