2018 IEEE 59th Annual Symposium on Foundations of Computer Science

Dispersion for Data-Driven Algorithm Design, Online Learning, and Private Optimization

Maria-Florina Balcan, Travis Dick, and Ellen Vitercik
Computer Science Department
Carnegie Mellon University
Pittsburgh, USA
Email: {ninamf, tdick, vitercik}@cs.cmu.edu

***Abstract*—A crucial problem in modern data science is data-driven algorithm design, where the goal is to choose the best algorithm, or algorithm parameters, for a specific application domain. In practice, we often optimize over a parametric algorithm family, searching for parameters with high performance on a collection of typical problem instances. While effective in practice, these procedures generally have not come with provable guarantees. A recent line of work initiated by a seminal paper of Gupta and Roughgarden [1] analyzes application-specific algorithm selection from a theoretical perspective. We progress this research direction in several important settings. We provide upper and lower bounds on regret for algorithm selection in online settings, where problems arrive sequentially and we must choose parameters online. We also consider differentially private algorithm selection, where the goal is to find good parameters for a set of problems without divulging too much sensitive information contained therein.**

We analyze several important parameterized families of algorithms, including SDP-rounding schemes for problems formulated as integer quadratic programs as well as greedy techniques for several canonical subset selection problems. The cost function that measures an algorithm's performance is often a volatile piecewise Lipschitz function of its parameters, since a small change to the parameters can lead to a cascade of different decisions made by the algorithm. We present general techniques for optimizing the sum or average of piecewise Lipschitz functions when the underlying functions satisfy a sufficient and general condition called *dispersion*. Intuitively, a set of piecewise Lipschitz functions is dispersed if no small region contains many of the functions' discontinuities.

Using dispersion, we improve over the best-known online learning regret bounds for a variety problems, prove regret bounds for problems not previously studied, and provide matching regret lower bounds. In the private optimization setting, we show how to optimize performance while preserving privacy for several important problems, providing matching upper and lower bounds on performance loss due to privacy preservation. Though algorithm selection is our primary motivation, we believe the notion of dispersion may be of independent interest. Therefore, we present our results for the more general problem of optimizing piecewise Lipschitz functions. Finally, we uncover dispersion in domains beyond algorithm selection, namely, auction design and pricing, providing online and privacy guarantees for these problems as well.

***Keywords*-disperion; algorithm design; online optimization; differentially private optimization; piecewise Lipschitz;**

I. Introduction

Data-driven algorithm design, that is, choosing the best algorithm for a specific application, is a critical problem in modern data science and algorithm design. Rather than use off-the-shelf algorithms with only worst-case guarantees, a practitioner will often optimize over a family of parametrized algorithms, tuning the algorithm's parameters based on typical problems from his domain. Ideally, the resulting algorithm will have high performance on future problems, but these procedures have historically come with no guarantees. In a seminal work, Gupta and Roughgarden [1] study algorithm selection in a distributional learning setting. Modeling an application domain as a distribution over typical problems, they show that a bound on the intrinsic complexity of the algorithm family prescribes the number of samples sufficient to ensure that any algorithm's empirical and expected performance are close.

We advance the foundations of algorithm selection in several important directions: online and private algorithm selection. In the online setting, problem instances arrive one-by-one, perhaps adversarially. The goal is to select parameters for each instance in order to minimize *regret*, which is the difference between the cumulative performance of those parameters and the optimal parameters in hindsight. We also study private algorithm selection, where the goal is to find high-performing parameters over a set of problems without revealing sensitive information contained therein. Preserving privacy is crucial when problems depend on individuals' medical or purchase data, for example.

We analyze several important, infinite families of parameterized algorithms. These include greedy techniques for canonical subset selection problems such as the knapsack and maximum weight independent set problems. We also study SDP-rounding schemes for problems that can be formulated as integer quadratic programs, such as max-cut, max-2sat, and correlation clustering. In these cases, our goal is to optimize, online or privately, the utility function that measures an algorithm's performance as a function of its parameters, such as the value of the items added to the knapsack by a parameterized knapsack algorithm. The

2575-8454/18/$31.00 ©2018 IEEE
DOI 10.1109/FOCS.2018.00064

key challenge is the volatility of this function: a small tweak to the algorithm's parameters can cause a cascade of changes in the algorithm's behavior. For example, greedy algorithms typically build a solution by iteratively adding items that maximize a scoring rule. Prior work has proposed parameterizing these scoring rules and tuning the parameter to obtain the best performance for a given application [1]. Slightly adjusting the parameter can cause the algorithm to select items in a completely different order, potentially causing a sharp change in the quality of the selected items.

Despite this challenge, we show that in many cases, these utility functions are well-behaved in several respects and thus can be optimized online and privately. Specifically, these functions are piecewise Lipschitz and moreover, they satisfy a condition we call *dispersion*. Roughly speaking, a collection of piecewise Lipschitz functions is *dispersed* if no small region of space contains discontinuities for many of the functions. We provide general techniques for online and private optimization of the sum or average of dispersed piecewise Lipschitz functions. Taking advantage of dispersion in online learning, we improve over the best-known regret bounds for a variety problems, prove regret bounds for problems not previously studied, and provide matching regret lower bounds. In the privacy setting, we show how to optimize performance while preserving privacy for several important problems, giving matching upper and lower bounds on performance loss due to privacy.

Though our main motivation is algorithm selection, we expect dispersion is even more widely applicable, opening up an exciting research direction. For this reason, we present our main results more generally for optimizing piecewise Lipschitz functions. We also uncover dispersion in domains beyond algorithm selection, namely, auction design and pricing, so we prove online and privacy guarantees for these problems as well. Finally, we answer several open questions: Cohen-Addad and Kanade [2] asked how to optimize piecewise Lipschitz functions and Gupta and Roughgarden [1] asked which algorithm selection problems can be solved with no regret algorithms. As a bonus, we also show that dispersion implies generalization guarantees in the distributional setting. In this setting, the configuration procedure is given an iid sample of problem instances drawn from an unknown distribution $\mathcal{D}$, and the goal is to find the algorithm parameters with highest expected utility. By bounding the empirical Rademacher complexity, we show that the sample and expected utility for all algorithms in our class are close, implying that the optimal algorithm on the sample is approximately optimal in expectation.

A. Our contributions

In order to present our contributions, we briefly outline the notation we will use. Let $\mathcal{A}$ be an infinite set of algorithms parameterized by a set $\mathcal{C} \subseteq \mathbb{R}^d$. For example, $\mathcal{A}$ might be the set of knapsack greedy algorithms that add items to the knapsack in decreasing order of $v(i)/s(i)^\rho$, where $v(i)$ and $s(i)$ are the value and size of item i and ρ is a parameter. Next, let Π be a set of problem instances for $\mathcal{A}$, such as knapsack problem instances, and let $u : \Pi \times \mathcal{C} \to [0, H]$ be a utility function where $u(x, \boldsymbol{\rho})$ measures the performance of the algorithm with parameters $\boldsymbol{\rho}$ on problem instance $x \in \Pi$. For example, $u(x, \rho)$ could be the value of the items chosen by the knapsack algorithm with parameter ρ on input x.

We now summarize our main contributions. Since our results apply beyond application-specific algorithm selection, we describe them for the more general problem of optimizing piecewise Lipschitz functions.

Dispersion: Let $u_1, \dots, u_T$ be a set of functions mapping a set $\mathcal{C} \subseteq \mathbb{R}^d$ to $[0, H]$. For example, in the application-specific algorithm selection setting, given a collection of problem instances $x_1, \dots, x_T \in \Pi$ and a utility function $u : \Pi \times \mathcal{C} \to [0, H]$, each function $u_i(\cdot)$ might equal the function $u(x_i, \cdot)$, measuring an algorithm's performance on a fixed problem instance as a function of its parameters. Dispersion is a constraint on the functions $u_1, \dots, u_T$. We assume that for each function u_i, we can partition $\mathcal{C}$ into sets $\mathcal{C}_1, \dots, \mathcal{C}_K$ such that u_i is L-Lipschitz on each piece, but u_i may have discontinuities at the boundaries between pieces. In our applications, each set $\mathcal{C}_i$ is connected, but our general results hold for arbitrary sets. Informally, the functions $u_1, \dots, u_T$ are (w, k)-dispersed if every Euclidean ball of radius w contains discontinuities for at most k of those functions (see Section II for a formal definition). This guarantees that although each function u_i may have discontinuities, they do not concentrate in a small region of space. Dispersion is sufficient to prove strong learning generalization guarantees, online learning regret bounds, and private optimization bounds when optimizing the empirical utility $\frac{1}{T}\sum_{i=1}^{T} u_i$. In our applications, $w = T^{\alpha-1}$ and $k = \tilde{O}(T^\alpha)$ with high probability for any $1/2 \le \alpha \le 1$, ignoring problem-specific multiplicands.

Online learning: We prove that dispersion implies strong regret bounds in online learning, a fundamental area of machine learning [3]. In this setting, a sequence of functions $u_1, \dots, u_T$ arrive one-by-one. At time t, the learning algorithm chooses a parameter vector $\boldsymbol{\rho}_t$ and then either observes the function u_t in the full information setting or the scalar $u_t(\boldsymbol{\rho}_t)$ in the bandit setting. The goal is to minimize expected regret: $\mathbb{E}[\max_{\boldsymbol{\rho}\in\mathcal{C}} \sum u_t(\boldsymbol{\rho}) - u_t(\boldsymbol{\rho}_t)]$. Under full information, we show that the exponentially-weighted forecaster [3] has regret bounded by $\tilde{O}(H(\sqrt{Td}+k) + TLw)$. When $w = 1/\sqrt{T}$ and $k = \tilde{O}(\sqrt{T})$, this results in $\tilde{O}(\sqrt{T}(H\sqrt{d} + L))$ regret. We also prove a matching lower bound. This algorithm also preserves (ϵ, δ)-differential privacy with regret bounded by $\tilde{O}(H(\sqrt{Td}/\epsilon + k + \delta) + TLw)$. Finally, under bandit feedback, we show that a discretization-based algorithm achieves regret at most $\tilde{O}(H(\sqrt{dT(3R/w)^d} + k) + TLw)$. When $w = T^{-1/(d+2)}$ and $k = \tilde{O}(T^{(d+1)/(d+2)})$, this gives a bound

of $\tilde{O}(T^{(d+1)/(d+2)}(H\sqrt{d(3R)^d}+L))$, matching the dependence on T of a lower bound by Kleinberg et al. [4] for (globally) Lipschitz functions.

Online algorithm selection is generally not possible: Gupta and Roughgarden [1] give an algorithm selection problem for which no online algorithm can achieve sublinear regret. Therefore, additional structure is necessary to prove guarantees, which we characterize using dispersion.

Private batch optimization: We demonstrate that it is possible to optimize over a set of dispersed functions while preserving *differential privacy* [5]. In this setting, the goal is to find the parameter $\boldsymbol{\rho}$ that maximizes average utility on a set $\mathcal{S} = \{u_1, \ldots, u_T\}$ of functions $u_i : \mathcal{C} \to \mathbb{R}$ without divulging much information about any single function u_i. Providing privacy at the granularity of functions is suitable when each function encodes sensitive information about one or a small group of individuals and each individual's information is used to define only a small number of functions. For example, in the case of auction design and pricing problems, each function u_i is defined by a set of buyers' bids or valuations for a set of items. If a single buyer's information is only encoded by a single function, then we preserve her privacy by not revealing sensitive information about any one function u_i. This will be the case, for example, if the buyers do not repeatedly return to buy the same items day after day. This is a common assumption in online auction design and pricing [6, 7, 8, 9, 10, 11, 12] because it means the buyers will not be strategic, aiming to trick the algorithm into setting lower prices in the future.

Differential privacy requires that an algorithm is randomized and its output distribution is insensitive to changing a single point in the input data. Formally, two multisets $\mathcal{S}$ and $\mathcal{S}'$ of T functions are *neighboring*, denoted $\mathcal{S} \sim \mathcal{S}'$, if $|\mathcal{S} \Delta \mathcal{S}'| \leq 1$. A randomized algorithm $\mathcal{A}$ is (ϵ, δ)-*differentially private* if, for any neighboring multisets $\mathcal{S} \sim \mathcal{S}'$ and set $\mathcal{O}$ of outcomes, $\Pr(\mathcal{A}(\mathcal{S}) \in \mathcal{O}) \leq e^{\epsilon} \Pr(\mathcal{A}(\mathcal{S}') \in \mathcal{O}) + \delta$. In our setting, the algorithm's input is a set $\mathcal{S}$ of T functions, and the output is a point $\boldsymbol{\rho} \in \mathcal{C}$ that approximately maximizes the average of those functions. We show that the exponential mechanism [13] outputs $\hat{\boldsymbol{\rho}} \in \mathcal{C}$ such that with high probability $\frac{1}{T}\sum_{i=1}^{T} u_i(\hat{\boldsymbol{\rho}}) \geq \max_{\boldsymbol{\rho} \in \mathcal{C}} \frac{1}{T}\sum_{i=1}^{T} u_i(\boldsymbol{\rho}) - \tilde{O}(\frac{H}{T}(\frac{d}{\epsilon}+k)+Lw)$ while preserving $(\epsilon, 0)$-differential privacy. We also give a matching lower bound. Our private algorithms always preserve privacy, even when dispersion does not hold.

Computational efficiency: In our settings, the functions have additional structure that enables us to design efficient implementations of our algorithms: for one-dimensional problems, there is a closed-form expression for the integral of the piecewise Lipschitz functions on each piece and for multi-dimensional problems, the functions are piecewise concave. We leverage tools from high-dimensional geometry [14, 15] to efficiently implement the integration and sampling steps required by our algorithms. Our algorithms have running time linear in the number of pieces of the utility function and polynomial in all other parameters.

B. Dispersion in algorithm selection problems

Algorithm selection.: We study algorithm selection for integer quadratic programs (IQPs) of the form $\max_{\boldsymbol{z} \in \{\pm 1\}^n} \boldsymbol{z}^\top A \boldsymbol{z}$, where $A \in \mathbb{R}^{n \times n}$ for some n. Many classic NP-hard problems can be formulated as IQPs, including max-cut [16], max-2SAT [16], and correlation clustering [17]. Many IQP approximation algorithms are semidefinite programming (SDP) rounding schemes; they solve the SDP relaxation of the IQP and round the resulting vectors to binary values. We study two families of SDP rounding techniques: s-linear rounding [18] and outward rotation [19], which include the Goemans-Williamson algorithm [16] as a special case. Due to these algorithms' inherent randomization, finding an optimal rounding function over T problem instances with n variables amounts to optimizing the sum of $(1/T^{1-\alpha}, \tilde{O}(nT^{\alpha}))$-dispersed functions for $1/2 \leq \alpha < 1$. This holds even for adversarial (non-stochastic) instances, implying strong online learning guarantees.

We also study greedy algorithm selection for two canonical subset selection problems: the knapsack and maximum weight independent set (MWIS) problems. Greedy algorithms are typically defined by a scoring rule determining the order the algorithm adds elements to the solution set. For example, Gupta and Roughgarden [1] introduce a parameterized knapsack algorithm that adds items in decreasing order of $v(i)/s(i)^{\rho}$, where $v(i)$ and $s(i)$ are the value and size of item i. Under mild conditions — roughly, that the items' values are drawn from distributions with bounded density functions and that each item's size is independent from its value — we show that the utility functions induced by T knapsack instances with n items are $(1/T^{1-\alpha}, \tilde{O}(nT^{\alpha}))$-dispersed for any $1/2 \leq \alpha < 1$.

Pricing problems and auction design: Market designers use machine learning to design auctions and set prices [20, 21]. In the online setting, at each time step there is a set of goods for sale and a set of consumers who place bids for those goods. The goal is to set auction parameters, such as reserve prices, that are nearly as good as the best fixed parameters in hindsight. Here, "best" may be defined in terms of revenue or social welfare, for example. In the offline setting, the algorithm receives a set of bidder valuations sampled from an unknown distribution and aims to find parameters that are nearly optimal in expectation (e.g., [22, 23, 24, 25, 26, 27, 28, 29, 8, 30, 31, 32]). We analyze multi-item, multi-bidder second price auctions with reserves, as well as pricing problems, where the algorithm sets prices and buyers decide what to buy based on their utility functions. These classic mechanisms have been studied for decades in both economics and computer science. We note that data-driven mechanism design problems are effectively

algorithm design problems with incentive constraints: the input to a mechanism is the buyers' bids or valuations, and the output is an allocation of the goods and a description of the payments required of the buyers. For ease of exposition, we discuss algorithm and mechanism design separately.

C. Related work

Gupta and Roughgarden [1] and Balcan et al. [33] study algorithm selection in the distributional learning setting, where there is a distribution $\mathcal{D}$ over problem instances. A learning algorithm receives a set $\mathcal{S}$ of samples from $\mathcal{D}$. Those two works provide *uniform convergence guarantees*, which bound the difference between the average performance over $\mathcal{S}$ of any algorithm in a class $\mathcal{A}$ and its expected performance on $\mathcal{D}$. It is known that regret bounds imply generalization guarantees for various online-to-batch conversion algorithms [34], but in this work, we also show that dispersion can be used to explicitly provide uniform convergence guarantees via Rademacher complexity. Beyond this connection, our work is a significant departure from these works since we give guarantees for private algorithm selection and we give no regret algorithms, whereas Gupta and Roughgarden [1] only study online MWIS algorithm selection, proving their algorithm has small constant per-round regret.

Private empirical risk minimization (ERM): The goal of private ERM is to find the best machine learning model parameters based on private data. Techniques include objective and output perturbation [35], stochastic gradient descent, and the exponential mechanism [14]. These works focus on minimizing data-dependent convex functions, so parameters near the optimum also have high utility, which is not the case in our settings.

Private algorithm configuration: Kusner et al. [36] develop private Bayesian optimization techniques for tuning algorithm parameters. Their methods implicitly assume that the utility function is differentiable. Meanwhile, the class of functions we consider have discontinuities between pieces, and it is not enough to privately optimize on each piece, since the boundaries themselves are data-dependent.

Online optimization: Prior work on online algorithm selection focuses on significantly more restricted settings. Cohen-Addad and Kanade [2] study single-dimensional piecewise constant functions under a "smoothed adversary," where the adversary chooses a distribution per boundary from which that boundary is drawn. Thus, the boundaries are independent. Moreover, each distribution must have bounded density. Gupta and Roughgarden [1] study online MWIS greedy algorithm selection under a smoothed adversary, where the adversary chooses a distribution per vertex from which its weight is drawn. Thus, the vertex weights are independent and again, each distribution must have bounded density. In contrast, we allow for more correlations among the elements of each problem instance. Our analysis also applies to the substantially more general setting of optimizing piecewise Lipschitz functions. We show several new applications of our techniques in algorithm selection for SDP rounding schemes, price setting, and auction design, none of which were covered by prior work. Furthermore, we provide differential privacy results and generalization guarantees.

Neither Cohen-Addad and Kanade [2] nor Gupta and Roughgarden [1] develop a general theory of dispersion, but we can map their analysis into our setting. In essence, Cohen-Addad and Kanade [2], who provide the tighter analysis, show that if the functions the algorithm sees map from $[0,1]$ to $[0,1]$ and are $(w,1)$-dispersed, then the regret of their algorithm is bounded by $O(\sqrt{T\ln(1/w)})$. Under a smoothed adversary, the functions are $(w,1)$-dispersed for an appropriate choice of w. In this work, we show that using the more general notion of (w,k)-dispersion is essential to proving tight learning bounds for more powerful adversaries. We provide a sequence of piecewise constant functions $u_1,\ldots,u_T$ mapping $[0,1]$ to $[0,1]$ that are $(1/4,\sqrt{T})$-dispersed, which means that our regret bound is $O(\sqrt{T\log(1/w)}+k)=O(\sqrt{T})$. However, these functions are not $(w,1)$-disperse for any $w \geq 2^{-T}$, so the regret bound by Cohen-Addad and Kanade [2] is trivial, since $\sqrt{T\log(1/w)}$ with $w=2^{-T}$ equals T. Similarly, Weed et al. [37] and Feng et al. [38] use a notion similar to $(w,1)$-dispersion to prove learning guarantees for the specific problem of learning to bid, as do Rakhlin et al. [39] for learning threshold functions under a smoothed adversary.

Our online bandit results are related to those of Kleinberg [40] for the "continuum-armed bandit" problem. They consider bandit problems where the set of arms is the interval $[0,1]$ and each payout function is uniformly locally Lipschitz. We relax this requirement, allowing each payout function to be Lipschitz with a number of discontinuities. In exchange, we require that the overall sequence of payout functions is fairly nice, in the sense that their discontinuities do not tightly concentrate. The follow-up work on Multi-armed Bandits in Metric Spaces [4] considers the stochastic bandit problem where the space of arms is an arbitrary metric space and the mean payoff function is Lipschitz. They introduce the zooming algorithm, which has better regret bounds than the discretization approach of Kleinberg [40] when either the max-min covering dimension or the (payout-dependent) zooming dimension are smaller than the covering dimension. In contrast, we consider optimization over $\mathbb{R}^d$ under the ℓ_2 metric, where this algorithm does not give improved regret in the worst case.

Auction design and pricing: Several works [6, 7, 8, 9, 10, 11] present stylized online learning algorithms for revenue maximization under specific auction classes. In contrast, our online algorithms are highly general and apply to many optimization problems beyond auction design. Dudík et al. [12] also provide online algorithms for auction design.

They discretize each set of mechanisms they consider and prove their algorithms have low regret over the discretized set. When the bidders have simple valuations (unit-demand and single-parameter) minimizing regret over the discretized set amounts to minimizing regret over the entire mechanism class. In contrast, we study bidders with fully general valuations, as well as additive and unit-demand valuations.

A long line of work has studied *generalization guarantees* for auction design and pricing problems (e.g., [22, 23, 24, 25, 26, 27, 28, 29, 8, 30, 41, 31, 32]). These works study the distributional setting where there is an unknown distribution over buyers' values and the goal is to use samples from this distribution to design a mechanism with high expected revenue. Generalization guarantees bound the difference between a mechanism's empirical revenue over the set of samples and expected revenue over the distribution. For example, several of these works [25, 26, 30, 31, 32, 42, 43] use learning theoretic tools such as pseudo-dimension and Rademacher complexity to derive these generalization guarantees. In contrast, we study online and private mechanism design, which requires a distinct set of analysis tools beyond those used in the distributional setting.

Bubeck et al. [8] study auction design in both the online and distributional settings when there is a single item for sale. They take advantage of structure exhibited in this well-studied single-item setting, such as the precise form of the optimal single-item auction [44]. Meanwhile, our algorithms and guarantees apply to the more general problem of optimizing piecewise Lipschitz functions.

II. Dispersion condition

In this section we formally define (w, k)-dispersion using the same notation as in Section I-A. Recall that Π is a set of instances, $\mathcal{C} \subset \mathbb{R}^d$ is a parameter space, and u is an abstract utility function. Throughout this paper, we use the ℓ_2 distance and let $B(\boldsymbol{\rho}, r) = \{\boldsymbol{\rho}' \in \mathbb{R}^d : \|\boldsymbol{\rho} - \boldsymbol{\rho}'\|_2 \leq r\}$ denote a ball of radius r centered at $\boldsymbol{\rho}$.

Definition 1. Let $u_1, \ldots, u_T : \mathcal{C} \to [0, H]$ be a collection of functions where u_i is piecewise Lipschitz over a partition $\mathcal{P}_i$ of $\mathcal{C}$. We say that $\mathcal{P}_i$ splits a set A if A intersects with at least two sets in $\mathcal{P}_i$ (see Figure 1). The collection of functions is (w, k)-*dispersed* if every ball of radius w is split by at most k of the partitions $\mathcal{P}_1, \ldots, \mathcal{P}_T$. More generally, the functions are (w, k)-*dispersed at the maximizer* if there exists a point $\boldsymbol{\rho}^* \in \operatorname{argmax}_{\boldsymbol{\rho} \in \mathcal{C}} \sum_{i=1}^{T} u_i(\boldsymbol{\rho})$ such that the ball $B(\boldsymbol{\rho}^*, w)$ is split by at most k of the partitions $\mathcal{P}_1, \ldots, \mathcal{P}_T$.

Given $\mathcal{S} = \{x_1, \ldots, x_T\} \subseteq \Pi$ and a utility function $u : \Pi \times \mathcal{C} \to [0, H]$, we equivalently say that u is (w, k)-*dispersed for $\mathcal{S}$ (at the maximizer)* if $\{u(x_1, \cdot), \ldots, u(x_T, \cdot)\}$ is (w, k)-dispersed (at the maximizer).

We often show that the discontinuities of a piecewise Lipschitz function $u : \mathbb{R} \to \mathbb{R}$ are random variables with κ-*bounded distributions*. A density function $f : \mathbb{R} \to \mathbb{R}$ corresponds to a κ-bounded distribution if $\max\{f(x)\} \leq \kappa$.[1] To prove dispersion we will use the following probabilistic lemma, showing that samples from κ-bounded distributions do not tightly concentrate.

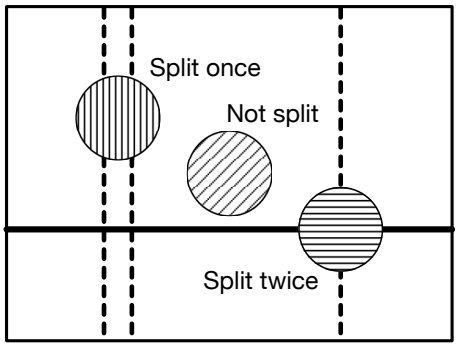

Figure 1: The dashed and solid lines correspond to two partitionings of the rectangle. Each of the displayed balls is either not split, split by one partition, or split by both.

Lemma 1. *Let $\mathcal{B} = \{\beta_1, \ldots, \beta_r\} \subset \mathbb{R}$ be a collection of samples where each β_i is drawn from a κ-bounded distribution with density function p_i. For any $\zeta \geq 0$, the following statements hold with probability at least $1 - \zeta$:*

1) *If the β_i are independent, then every interval of width w contains at most $k = O(rw\kappa + \sqrt{r \log(1/\zeta)})$ samples. In particular, for any $\alpha \geq 1/2$ we can take $w = 1/(\kappa r^{1-\alpha})$ and $k = O(r^\alpha \sqrt{\log(1/\zeta)})$.*
2) *If the samples can be partitioned into P buckets $\mathcal{B}_1, \ldots, \mathcal{B}_P$ such that each $\mathcal{B}_i$ contains independent samples and $|\mathcal{B}_i| \leq M$, then every interval of width w contains at most $k = O(PMw\kappa + \sqrt{M \log(P/\zeta)})$. In particular, for any $\alpha \geq 1/2$ we can take $w = 1/(\kappa M^{1-\alpha})$ and $k = O(PM^\alpha \sqrt{M \log(P/\zeta)})$.*

Proof sketch: If the β_i are independent, the expected number of samples in any interval of width w is at most $r\kappa w$. Since the VC-dimension of intervals is 2, it follows that with probability at least $1-\zeta$, no interval contains more than $r\kappa w + O(\sqrt{r \log(1/\zeta)})$ samples.

The second claim follows by applying this counting argument to each of the buckets $\mathcal{B}_i$ with failure probability $\zeta' = \zeta/P$ and taking the union bound over all buckets. With probability at least $1 - \zeta$, every interval of width w contains at most $M\kappa w + O(\sqrt{M \log(P/\zeta)})$ samples from each bucket, and at most $k = PM\kappa w + O(P\sqrt{M \log(P/\zeta)})$ samples in total from all P buckets. ∎

Lemma 1 allows us to provide dispersion guarantees for "smoothed adversaries" in online learning. Under this type of adversary, the discontinuity locations for each function u_i are random variables, due to the smoothness of the adversary. In our algorithm selection applications, the randomness

[1] For example, for all $\mu \in \mathbb{R}$, $\mathcal{N}(\mu, \sigma)$ is $\frac{1}{2\pi\sigma}$-bounded.

of discontinuities may be a byproduct of the randomness in the algorithm's inputs. For example, in the case of knapsack algorithm configuration, the item values and sizes may be drawn from distributions chosen by the adversary. This induces randomness in the discontinuity locations of the algorithm's cost function. We can thus apply Lemma 1 to guarantee dispersion.

We also use Lemma 1 to guarantee dispersion even when the adversary is not smoothed. Surprisingly, we show that dispersion holds for IQP algorithm configuration without *any* assumptions on the input instances. In this case, we exploit the fact that the algorithms are themselves randomized. This randomness implies that the discontinuities of the algorithm's cost function are random variables, and thus Lemma 1 implies dispersion.

III. Online optimization

In this setting, a sequence of functions $u_1, \dots, u_T$ arrive one-by-one. At time t, the learning algorithm chooses a vector $\boldsymbol{\rho}_t$ and then either observes the function $u_t(\cdot)$ in the full information setting or the value $u_t(\boldsymbol{\rho}_t)$ in the bandit setting. The goal is to minimize expected regret: $\mathbb{E}[\max_{\boldsymbol{\rho}\in\mathcal{C}} \sum_{t=1}^{T}(u_t(\boldsymbol{\rho}) - u_t(\boldsymbol{\rho}_t))]$. In our applications, the functions $u_1, \dots, u_T$ are random, either due to internal randomization in the algorithms we are configuring or from assumptions on the adversary[2]. We show that the functions are (w,k)-dispersed with probability $1-\zeta$ over the choice of $u_1, \dots, u_T$. The following regret bounds hold in expectation with an additional term of $HT\zeta$ bounding the effect of the rare event where the functions are not dispersed.

Full information. The *exponentially-weighted forecaster* algorithm samples the vectors $\boldsymbol{\rho}_t$ from the distribution $p_t(\boldsymbol{\rho}) \propto \exp(\lambda \sum_{s=1}^{t-1} u_s(\boldsymbol{\rho}))$. We prove the following regret bound. The full proof is in the full version of the paper [46].

Theorem 1. *Let $u_1, \dots, u_T : \mathcal{C} \to [0, H]$ be any sequence of piecewise L-Lipschitz functions that are (w,k)-dispersed at the maximizer $\boldsymbol{\rho}^*$. Suppose $\mathcal{C} \subset \mathbb{R}^d$ is contained in a ball of radius R and $B(\boldsymbol{\rho}^*, w) \subset \mathcal{C}$. The exponentially weighted forecaster with $\lambda = \sqrt{d\ln(R/w)/T}/H$ has expected regret bounded by $O(H(\sqrt{Td\log(R/w)} + k) + TLw)$.*

For all $t \in [T]$, suppose $\sum_{s=1}^{t} u_s$ is piecewise Lipschitz over at most K pieces. When $d = 1$ and $\exp(\sum_{s=1}^{t} u_s)$ can be integrated in constant time on each of its pieces, the running time is $O(TK)$. When $d > 1$ and $\sum_{s=1}^{t} u_s$ is piecewise concave over convex pieces, we provide an efficient approximate implementation. For parameters $\eta, \zeta \in (0,1)$ and $\lambda = \sqrt{(d\ln(R/w)+\eta)/T}/H$, this algorithm has expected regret bounded by $O(H(\sqrt{Td\ln(R/w)+\eta}+k+\zeta T)+LTw)$, and running time $\tilde{O}(T(K\cdot\mathrm{poly}(d, 1/\eta) + \mathrm{poly}(d, L, 1/\eta))$.

Proof sketch: Let U_t be the function $\sum_{i=1}^{t-1} u_i(\cdot)$ and let $W_t = \int_{\mathcal{C}} \exp(\lambda U_t(\boldsymbol{\rho}))\, d\boldsymbol{\rho}$. We use (w,k)-dispersion to lower bound W_{T+1}/W_1 in terms of the optimal parameter's total payout. Combining this with a standard upper bound on W_{T+1}/W_1 in terms of the learner's expected payout gives the regret bound. To lower bound W_{T+1}/W_1, let $\boldsymbol{\rho}^*$ be the optimal parameter and let $\mathrm{OPT} = U_{T+1}(\boldsymbol{\rho}^*)$. Also, let $\mathcal{B}^*$ be the ball of radius w around $\boldsymbol{\rho}^*$. From (w,k)-dispersion, we know that for all $\boldsymbol{\rho} \in \mathcal{B}^*$, $U_{T+1}(\boldsymbol{\rho}) \geq \mathrm{OPT} - Hk - LTw$. Therefore,

$$\begin{aligned} W_{T+1} &= \int_{\mathcal{C}} \exp(\lambda U_{T+1}(\boldsymbol{\rho}))\, d\boldsymbol{\rho} \geq \int_{\mathcal{B}^*} \exp(\lambda U_{T+1}(\boldsymbol{\rho}))\, d\boldsymbol{\rho} \\ &\geq \int_{\mathcal{B}^*} \exp(\lambda(\mathrm{OPT} - Hk - LTw)) d\boldsymbol{\rho} \\ &\geq \mathrm{Vol}(B(\boldsymbol{\rho}^*, w)) \exp(\lambda(\mathrm{OPT} - Hk - LTw)). \end{aligned}$$

Moreover, $W_1 = \int_{\mathcal{C}} \exp(\lambda U_1(\boldsymbol{\rho}))\, d\boldsymbol{\rho} \leq \mathrm{Vol}(B(\mathbf{0}, R))$. Therefore,

$$\frac{W_{T+1}}{W_1} \geq \frac{\mathrm{Vol}(B(\boldsymbol{\rho}^*, w))}{\mathrm{Vol}(B(\mathbf{0}, R))} \exp(\lambda(\mathrm{OPT} - Hk - LTw)).$$

The volume ratio is equal to $(w/R)^d$, since the volume of a ball of radius r in $\mathbb{R}^d$ is proportional to r^d. Therefore, $W_{T+1}/W_1 \geq (w/R)^d \exp(\lambda(\mathrm{OPT} - Hk - LTw))$. Combining the upper and lower bounds on $\frac{W_{T+1}}{W_1}$ gives the result.

Our efficient algorithm (see the full version [46]) approximately samples from p_t. Let $\mathcal{C}_1, \dots, \mathcal{C}_K$ be the partition of $\mathcal{C}$ over which $\sum u_t(\cdot)$ is piecewise concave. Our algorithm picks $\mathcal{C}_I$ with probability approximately proportional to $\int_{\mathcal{C}_I} p_t$ [15] and outputs a sample from the conditional distribution of p_t on $\mathcal{C}_I$ [14]. Crucially, we prove that the algorithm's output distribution is close to p_t, so every event concerning the outcome of the approximate algorithm occurs with about the same probability as it does under p_t. ∎

The requirement that $B(\boldsymbol{\rho}^*, w) \subset \mathcal{C}$ is for convenience. In the full version of the paper [46], we show how to transform the problem to satisfy this. Setting $\lambda = \sqrt{d/T}/H$, which does not require knowledge of w, has regret $O(H(\sqrt{Td}\log(R/w) + k) + TLw)$. Under alternative settings of λ, we show that our algorithms are (ϵ,δ)-differentially private with regret bounds of $\tilde{O}(H\sqrt{T}/\epsilon + Hk + LTw)$ in the single-dimensional setting and $\tilde{O}(H\sqrt{T}d/\epsilon + H(k+\delta) + LTw)$ in the d-dimensional setting (see the full version of the paper [46]).

Next, we prove a matching lower bound. The full proof is in the full version of the paper [46].

Theorem 2. *Suppose $T \geq 2d\log(4d)$. For any algorithm, there are piecewise constant functions $u_1, \dots, u_T$ mapping $[0,1]^d$ to $\{0,1\}$ such that if $D = \{(w,k) : \{u_1, \dots, u_T\}$ is (w,k)-dispersed at the maximizer$\}$, then $\mathbb{E}[\sum_{t=1}^{T} u_t(\boldsymbol{\rho}_t) -$*

[2]As we describe in Section I-C, prior research [45, 2] also makes assumptions on the adversary. For example, Cohen-Addad and Kanade [2] focus on adversaries that choose distributions with bounded densities from which the discontinuities of u_t are drawn. In the full version of the paper [46], we show that their smoothness assumption implies dispersion with high probability.

$u_t(\boldsymbol{\rho}^*)] = \Omega(\inf_{(w,k)\in D}\{\sqrt{Td\log(1/w)}+k\})$, *where* $\boldsymbol{\rho}^* \in \operatorname{argmax}_{\rho\in\mathcal{C}} \sum_{t=1}^{T} u_t(\rho)$.

Proof sketch: For each dimension, the adversary plays a sequence of axis-aligned halfspaces with thresholds that divide the set of optimal parameters in two. The adversary plays each halfspace $\Theta(\frac{T}{d\log d})$ times, randomly switching which side of the halfspace has a positive label, thus forcing regret of at least $\Omega(\sqrt{Td\log(d)})$. Since the adversary repeatedly divides the set of optimal parameters in two, the resulting set of optimal parameters is contained in an axis-aligned cube of side length $\Theta(1/d)$. The adversary then plays $\sqrt{T}$ copies of the indicator function of a ball of radius $(2d)^{-T}$ at the center of this cube. This ensures the functions are not $(w,0)$-dispersed at the maximizer for any $w \geq (2d)^{-T}$, and thus prior regret analyses [2] give a trivial bound of T. However, the functions are $(\Theta(1/d), \sqrt{T})$-dispersed, so the regret is $\Omega(\inf_{(w,k)\in D}\{\sqrt{Td\log(1/w)}+k\})$. ∎

Bandit feedback. We now study online optimization under bandit feedback.

Theorem 3. *Let $u_1, \ldots, u_T : \mathcal{C} \to [0,H]$ be any sequence of piecewise L-Lipschitz functions that are (w,k)-dispersed at the maximizer $\boldsymbol{\rho}^*$. Moreover, suppose that $\mathcal{C} \subset \mathbb{R}^d$ is contained in a ball of radius R and that $B(\boldsymbol{\rho}^*, w) \subset \mathcal{C}$. There is a bandit-feedback online optimization algorithm with regret $O(H\sqrt{Td(3R/w)^d\log(R/w)} + TLw + Hk)$. The per-round running time is $O((3R/w)^d)$.*

Proof: Let $\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_M$ be a w-net for $\mathcal{C}$. The main insight is that (w,k)-dispersion implies that the difference in utility between the best point in hindsight from the net and the best point in hindsight from $\mathcal{C}$ is at most $Hk+TLw$. Therefore, we only need to compete with the best point in the net. We use the Exp3 algorithm [47] to choose parameters $\hat{\boldsymbol{\rho}}_1, \ldots, \hat{\boldsymbol{\rho}}_T$ by playing the bandit with M arms, where on round t arm i has payout $u_t(\boldsymbol{\rho}_i)$. The expected regret of Exp3 is $\tilde{O}(H\sqrt{TM\log M})$ relative to our net. In the full version of the paper [46], we show $M \leq (3R/w)^d$, so the overall regret is $\tilde{O}(H\sqrt{Td(3R/w)^d\log(R/w)}+TLw+Hk)$ with respect to $\mathcal{C}$. ∎

If $w = T^{\frac{d+1}{d+2}-1} = \frac{1}{T^{1/(d+2)}}$ and $k = \tilde{O}(T^{\frac{d+1}{d+2}})$, Theorem 3 gives the optimal exponent on T. Specifically, the regret is $\tilde{O}\big(T^{(d+1)/(d+2)}(H\sqrt{d(3R)^d}+L)\big)$, and no algorithm can have regret $O(T^\gamma)$ for $\gamma < (d+1)/(d+2)$ for the special case of (globally) Lispchitz functions [4].

IV. Differentially private optimization

We show that the exponential mechanism, which is $(\epsilon,0)$-differentially private, has high utility when optimizing the mean of dispersed functions. In this setting, the algorithm is given a collection of functions $u_1, \ldots, u_T : \mathcal{C} \to [0,H]$, each of which depends on some sensitive information. In cases where each function u_i encodes sensitive information about one or a small group of individuals and each individual is present in a small number of functions, we can give meaningful privacy guarantees by providing differential privacy for each function in the collection. We say that two sets of T functions are neighboring if they differ on at most one function. Recall that the exponential mechanism outputs a sample from the distribution with density proportional to $f^{\epsilon}_{\exp}(\boldsymbol{\rho}) = \exp\big(\frac{\epsilon}{2\Delta T}\sum_{i=1}^{T} u_i(\boldsymbol{\rho})\big)$, where Δ is the sensitivity of the average utility. Since the functions u_i are bounded, the sensitivity of $\frac{1}{T}\sum_{i=1}^{T} u_i$ satisfies $\Delta \leq H/T$. The following theorem states our utility guarantee. The full proof is in the full version of the paper [46].

Theorem 4. *Let $u_1, \ldots, u_T : \mathcal{C} \to [0,H]$ be piecewise L-Lipschitz and (w,k)-dispersed at the maximizer $\boldsymbol{\rho}^*$, and suppose that $\mathcal{C} \subset \mathbb{R}^d$ is convex, contained in a ball of radius R, and $B(\boldsymbol{\rho}^*, w) \subset \mathcal{C}$. For any $\epsilon > 0$, with probability at least $1-\zeta$, the output $\hat{\boldsymbol{\rho}}$ of the exponential mechanism satisfies $\frac{1}{T}\sum_{i=1}^{T} u_i(\hat{\boldsymbol{\rho}}) \geq \frac{1}{T}\sum_{i=1}^{T} u_i(\boldsymbol{\rho}^*) - O(\frac{H}{T\epsilon}(d\log\frac{R}{w} + \log\frac{1}{\zeta}) + Lw + \frac{Hk}{T})$.*

When $d = 1$, this algorithm is efficient, provided $f^{\epsilon}_{\exp}$ can be efficiently integrated on each piece of $\sum_i u_i$. For $d > 1$ we also provide an efficient approximate sampling algorithm when $\sum_i u_i$ is piecewise concave defined on K convex pieces. This algorithm preserves (ϵ,δ)-differential privacy for $\epsilon > 0$, $\delta > 0$ with the same utility guarantee (with $\zeta = \delta$). The running time of this algorithm is $\tilde{O}(K \cdot \operatorname{poly}(d, 1/\epsilon) + \operatorname{poly}(d, L, 1/\epsilon))$.

Proof sketch: The exponential mechanism can fail to output a good parameter if there are drastically more bad parameters than good. The key insight is that due to dispersion, the set of good parameters is not too small. In particular, we know that every $\boldsymbol{\rho} \in B(\boldsymbol{\rho}^*, w)$ has $\frac{1}{T}\sum_i u_i(\boldsymbol{\rho}) \geq \frac{1}{T}\sum_i u_i(\boldsymbol{\rho}^*) - \frac{Hk}{T} - Lw$ because at most k of the functions u_i for have discontinuities in $B(\boldsymbol{\rho}^*, w)$ and the rest are L-Lipschitz.

In a bit more detail, for a constant c fixed later on, the probability that a sample from $\mu_{\exp}$ lands in $E = \{\boldsymbol{\rho} : \frac{1}{T}\sum_i u_i(\boldsymbol{\rho}) \leq c\}$ is F/Z, where $F = \int_E f_{\exp}$ and $Z = \int_{\mathcal{C}} f_{\exp}$. We know that $F \leq \exp\big(\frac{T\epsilon c}{2H}\big)\operatorname{Vol}(E) \leq \exp\big(\frac{T\epsilon c}{2H}\big)\operatorname{Vol}\big(B(0,R)\big)$, where the second inequality follows from the fact that a ball of radius R contains the entire space $\mathcal{C}$. To lower bound Z, we use the fact that at most k of the functions $u_1, \ldots, u_T$ have discontinuities in the ball $B(\boldsymbol{\rho}^*, w)$ and the rest of the functions are L-Lipschitz. It follows that for any $\boldsymbol{\rho} \in B(\boldsymbol{\rho}^*, w)$, we have $\frac{1}{T}\sum_i u_i(\boldsymbol{\rho}) \geq \frac{1}{T}\sum_i u_i(\boldsymbol{\rho}^*) - \frac{Hk}{|S|} - Lw$. This is because each of the k functions with boundaries can affect the average utility by at most $H/|T|$ and otherwise $\frac{1}{T}\sum_i u_i(\cdot)$ is L-Lipschitz. Since $B(\boldsymbol{\rho}^*, w) \subset \mathcal{C}$, this gives $Z \geq \exp\big(\frac{T\epsilon}{2H}(\frac{1}{T}\sum_i u_i(\boldsymbol{\rho}^*)) - \frac{Hk}{T} - Lw)\big)\operatorname{Vol}\big(B(\boldsymbol{\rho}^*, w)\big)$.

Putting the bounds together, we have that $F/Z \leq \exp\big(\frac{T\epsilon}{2H}(c - \frac{1}{T}\sum_i u_i(\boldsymbol{\rho}^*) + \frac{Hk}{T} + Lw\big) \cdot \frac{\operatorname{Vol}(B(0,R))}{\operatorname{Vol}(B(\boldsymbol{\rho}^*, w))}$. The

volume ratio is equal to $(R/w)^d$, since the volume of a ball of radius r in $\mathbb{R}^d$ is proportional to r^d. Setting this bound to ζ and solving for c gives the result.

Our efficient implementation (see the full version [46]) relies on the same tools as our approximate implementation of the exponentially weighted forecaster. The main step is proving the distribution of $\hat{\boldsymbol{\rho}}$ is close to the distribution with density $f_{\exp}$. ∎

In the full version of the paper [46], we give also give a discretization-based computationally inefficient algorithm in d dimensions that satisfies $(\epsilon, 0)$-differential privacy.

We can tune the value of w to make the dependence on L logarithmic: if $T \geq \frac{2Hd}{w\epsilon L}$, then with probability $1-\zeta$, $\frac{1}{T}\sum_i u_i(\hat{\boldsymbol{\rho}}) \geq \frac{1}{T}\sum_i u_i(\boldsymbol{\rho}^*) - O\big(\frac{Hd}{T\epsilon}\log\frac{L\epsilon RT}{Hd} + \frac{Hk}{T} + \frac{H\log(1/\zeta)}{T\epsilon}\big)$ (see the full version of the paper [46].).

Finally, we provide a matching lower bound. See the full version of the paper [46] for the full proofs when $d = 1$ and $d \geq 2$. In those theorems, we generalize Theorem 5 to obtain lower bounds that hold with probability $1-\zeta$ for any ζ. When $d = 1$, the worst-case functions correspond to MWIS instances.

Theorem 5. *Suppose either $d = 1$ and $\epsilon > 0$, or $d \geq 2$ and $0 < \epsilon \leq \frac{3d-4}{5}$. For any ϵ-differentially private algorithm $\mathcal{A}$ and any $T \geq 2$, there is a multi-set $\mathcal{S} = \{u_1, \ldots, u_T\}$ of piecewise constant functions mapping $\{\boldsymbol{\rho} \in \mathbb{R}^d : \|\boldsymbol{\rho}\|_2 \leq 2\sqrt{d}\}$ to $[0,1]$ such that if $D = \{(w,k) : \mathcal{S}$ is (w,k)-dispersed at the maximizer$\}$, then with probability at least $\frac{9}{10}$, $\frac{1}{T}\sum_{i=1}^T u_i(\hat{\boldsymbol{\rho}}) \leq \max_{\boldsymbol{\rho}} \frac{1}{T}\sum_{i=1}^T u_i(\boldsymbol{\rho}) - \Omega\left(\inf_{(w,k)\in D}\left\{\frac{d}{T\epsilon}\log\frac{R}{w} + \frac{k}{T}\right\}\right)$, where $R = 2\sqrt{d}$.*

Proof sketch: We construct $t \geq 2^{4d/5}$ multi-sets $\mathcal{S}_1, \ldots, \mathcal{S}_t$, each with T piecewise constant functions. For every pair $\mathcal{S}_i$ and $\mathcal{S}_j$, $|\mathcal{S}_i \Delta \mathcal{S}_j|$ is small but the set $I_{\mathcal{S}_i}$ of parameters maximizing $\sum_{u\in\mathcal{S}_i} u(\boldsymbol{\rho})$ is disjoint from $I_{\mathcal{S}_j}$. Therefore, for every pair $\mathcal{S}_i$ and $\mathcal{S}_j$, the distributions $\mathcal{A}(\mathcal{S}_i)$ and $\mathcal{A}(\mathcal{S}_j)$ are similar, and since $I_{\mathcal{S}_1}, \ldots, I_{\mathcal{S}_t}$ are disjoint, this means that for some $\mathcal{S}_i$, with probability $\frac{9}{10}$, the output of $\mathcal{A}(\mathcal{S}_i) \notin I_{\mathcal{S}_i}$. The key challenge is constructing the sets $\mathcal{S}_i$ so that the suboptimality of any point not in $I_{\mathcal{S}_i}$ is $\frac{d}{T\epsilon}\log\frac{R}{w} + \frac{k}{T}$, where w and k are dispersion parameters for $\mathcal{S}_i$. We construct $\mathcal{S}_i$ so that this suboptimality is $\Theta(\frac{d}{T\epsilon})$, which gives the desired result if $w = \Theta(R)$ and $k = \Theta(\frac{d}{\epsilon})$. To achieve these conditions, we carefully fill each $\mathcal{S}_i$ with indicator functions of balls centered on a selection of vertices from the d-dimensional hypercube. ∎

V. Dispersion in application-specific algorithm selection

We now analyze dispersion for a range of algorithm configuration problems. In the private setting, the algorithm receives samples $\mathcal{S} \sim \mathcal{D}^T$, where $\mathcal{D}$ is an arbitrary distribution over problem instances Π. The goal is to privately find a value $\hat{\boldsymbol{\rho}}$ that nearly maximizes $\sum_{x\in\mathcal{S}} u(x, \boldsymbol{\rho})$. In our applications, prior work [30, 1, 33] shows that $\hat{\boldsymbol{\rho}}$ nearly maximizes $\mathbb{E}_{x\sim\mathcal{D}}[u(x,\boldsymbol{\rho})]$. In the online setting, the goal is to find a value $\boldsymbol{\rho}$ that is nearly optimal in hindsight over a stream $x_1, \ldots, x_T$ of instances, or equivalently, over a stream $u_1 = u(x_1, \cdot), \ldots, u_T = u(x_T, \cdot)$ of functions. Each x_t is drawn from a distribution $\mathcal{D}^{(t)}$, which may be adversarial. Thus in both settings, $\{x_1, \ldots, x_T\} \sim \mathcal{D}^{(1)} \times \cdots \times \mathcal{D}^{(T)}$, but in the private setting, $\mathcal{D}^{(1)} = \cdots = \mathcal{D}^{(T)}$.

Greedy algorithms. We study greedy algorithm configuration for two important problems: the maximum weight independent set (MWIS) and knapsack problems. In MWIS, there is a graph and a weight $w(v) \in \mathbb{R}_{\geq 0}$ for each vertex v. The goal is to find a set of non-adjacent vertices with maximum weight. The classic greedy algorithm repeatedly adds a vertex v which maximizes $w(v)/(1+\deg(v))$ to the independent set and deletes v and its neighbors from the graph. Gupta and Roughgarden [1] propose the greedy heuristic $w(v)/(1+\deg(v))^\rho$ where $\rho \in \mathcal{C} = [0, B]$ for some $B \in \mathbb{R}$. When $\rho = 1$, the approximation ratio is $1/D$, where D is the graph's maximum degree [48]. We represent a graph as a tuple $(\boldsymbol{w}, \boldsymbol{e}) \in \mathbb{R}^n \times \{0,1\}^{\binom{n}{2}}$, ordering the vertices $v_1, \ldots, v_n$ in a fixed but arbitrary way. The function $u(\boldsymbol{w}, \boldsymbol{e}, \cdot)$ maps a parameter ρ to the weight of the vertices in the set returned by the algorithm parameterized by ρ.

Theorem 6. *Suppose all vertex weights are in $(0,1]$ and for each $\mathcal{D}^{(i)}$, every pair of vertex weights has a κ-bounded joint distribution. For any $\boldsymbol{w}$ and $\boldsymbol{e}$, $u(\boldsymbol{w}, \boldsymbol{e}, \cdot)$ is piecewise 0-Lipschitz and for any $\alpha \geq 1/2$, with probability $1-\zeta$ over $\mathcal{S} \sim \bigtimes_{i=1}^T \mathcal{D}^{(i)}$, u is $(\frac{1}{T^{1-\alpha}\kappa\ln n}, O(n^4 T^\alpha \sqrt{\ln(n/\zeta)}))$-dispersed with respect to $\mathcal{S}$.*

Proof sketch: The utility $u(\boldsymbol{w}^{(t)}, \boldsymbol{e}^{(t)}, \rho)$ has a discontinuity when the ordering of two vertices under the greedy score swaps. Thus, the discontinuities have the form $(\ln(w_i^{(t)}) - \ln(w_j^{(t)}))/(\ln(d_1) - \ln(d_2))$ for all $t \in [T]$ and $i, j, d_1, d_2 \in [n]$, where $w_j^{(t)}$ is the weight of the j^{th} vertex of $(\boldsymbol{w}^{(t)}, \boldsymbol{e}^{(t)})$ [45]. We show that when pairs of vertex weights have κ-bounded joint distributions, then the discontinuities each have $(\kappa \ln n)$-bounded distributions. Let $\mathcal{B}_{i,j,d_1,d_2}$ be the set of discontinuities contributed by vertices i and j with degrees d_1 and d_2 across all instances in $\mathcal{S}$. The buckets $\mathcal{B}_{i,j,d_1,d_2}$ partition the discontinuities into n^4 sets of independent random variables. Therefore, applying Lemma 1 with $P = n^4$ and $M = T$ proves the claim. ∎

In the full version of the paper [46], we prove Theorem 6 and demonstrate that it implies strong optimization guarantees. The analysis for the knapsack problem is similar (see the full version of the paper [46]).

Integer quadratic programming (IQP) algorithms. We now apply our dispersion analysis to two popular IQP approximation algorithms: s-linear [18] and outward rotation rounding algorithms [19]. The goal is to maximize a function $\sum_{i,j\in[n]} a_{ij} z_i z_j$ over $\boldsymbol{z} \in \{\pm 1\}^n$, where the

matrix $A = (a_{ij})$ has non-negative diagonal entries. Both algorithms are generalizations of the Goemans-Williamson (GW) max-cut algorithm [16]. They first solve the SDP relaxation $\sum_{i,j\in[n]} a_{ij}\langle \boldsymbol{u}_i, \boldsymbol{u}_j\rangle$ subject to the constraint that $\|\boldsymbol{u}_i\| = 1$ for $i \in [n]$ and then round the vectors $\boldsymbol{u}_i$ to $\{\pm 1\}$. Under s-linear rounding, the algorithm samples a standard Gaussian $\boldsymbol{Z} \sim \mathcal{N}_n$ and sets $z_i = 1$ with probability $1/2 + \phi_s(\langle \boldsymbol{u}_i, \boldsymbol{Z}\rangle)/2$ and -1 otherwise, where $\phi_s(y) = -\mathbb{1}_{y<-s} + \frac{y}{s}\cdot \mathbb{1}_{-s\le y\le s} + \mathbb{1}_{y>s}$ and s is a parameter. The outward rotation algorithm first maps each $\boldsymbol{u}_i$ to $\boldsymbol{u}'_i \in \mathbb{R}^{2n}$ by $\boldsymbol{u}'_i = [\cos(\gamma)\boldsymbol{u}_i\,;\,\sin(\gamma)\boldsymbol{e}_i]$ and sets $z_i = \operatorname{sgn}(\langle \boldsymbol{u}'_i, \boldsymbol{Z}\rangle)$, where $\boldsymbol{e}_i$ is the i^{th} standard basis vector, $\boldsymbol{Z} \in \mathbb{R}^{2n}$ is a standard Gaussian, and $\gamma \in [0, \pi/2]$ is a parameter. Feige and Langberg [18] and Zwick [19] prove that these rounding functions provide a better worst-case approximation ratio on graphs with "light" max-cuts, where the max-cut does not constitute a large fraction of the edges.

Our utility u maps the algorithm parameter (either s or γ) to the objective value obtained. We exploit the randomness of these algorithms to guarantee dispersion. To facilitate this analysis, we imagine that the Gaussians $\boldsymbol{Z}$ are sampled ahead of time and included as part of the problem instance. For s-linear rounding, we write the utility as $u_{\text{slin}}(A, \boldsymbol{Z}, s) = \sum_{i=1}^n a_i^2 + \sum_{i\neq j} a_{ij}\phi_s(v_i)\phi_s(v_j)$, where $v_i = \langle \boldsymbol{u_i}, \boldsymbol{Z}\rangle$. For outward rotations, $u_{\text{owr}}(A, \boldsymbol{Z}, \gamma) = \sum_{i,j} a_{ij}\operatorname{sgn}(v'_i)\operatorname{sgn}(v'_j)$, where $v'_i = \langle \boldsymbol{u}'_i, \boldsymbol{Z}\rangle$.

First, we prove a dispersion guarantee for u_{owr}. The full proof is in the full version [46], where we also demonstrate the theorem's implications for our optimization settings.

Theorem 7. *For any matrix A and vector $\boldsymbol{Z}$, $u_{\text{owr}}(A, \boldsymbol{Z}, \cdot)$ is piecewise 0-Lipschitz. With probability $1-\zeta$ over $\boldsymbol{Z}^{(1)}, \dots, \boldsymbol{Z}^{(T)} \sim \mathcal{N}_{2n}$, for any $A^{(1)}, \dots, A^{(T)} \in \mathbb{R}^{n\times n}$ and any $\alpha \ge 1/2$, u_{owr} is $(T^{\alpha-1}, O(nT^\alpha\sqrt{\log(n/\zeta)}))$-dispersed with respect to $\mathcal{S} = \{(A^{(t)}, \boldsymbol{Z}^{(t)})\}_{t=1}^T$.*

Proof sketch: The discontinuities of $u_{\text{owr}}(A, \boldsymbol{Z}, \gamma)$ occur whenever $\langle \boldsymbol{u}'_i, \boldsymbol{Z}\rangle$ shifts from positive to negative for some $i \in [n]$. Between discontinuities, the function is constant. By definition of $\boldsymbol{u}'_i$, this happens when $\gamma = \tan^{-1}(-\langle \boldsymbol{u}_i, \boldsymbol{Z}[1,\dots,n]\rangle / Z[n+i])$, which comes from a $1/\pi$-bounded distribution. The next challenge is that the discontinuities are not independent: the n discontinuities from instance t depend on the same vector $\boldsymbol{Z}^{(t)}$. To overcome this, we let $\mathcal{B}_i$ denote the set of discontinuities contributed by vector $\boldsymbol{u}_i$ across all instances. The buckets $\mathcal{B}_i$ partition the set of discontinuities into $P = n$ sets, each containing at most T discontinuities. We then apply Lemma 1 with P and $M = T$ to prove the claim. ∎

Next, we prove the following guarantee for u_{slin}. The full proof is in the full version [46], where we also demonstrate the theorem's implications for our optimization settings.

Theorem 8. *With probability $1-\zeta$ over $\boldsymbol{Z}^{(1)}, \dots, \boldsymbol{Z}^{(T)} \sim \mathcal{N}_n$, for any matrices $A^{(1)}, \dots, A^{(T)}$ and any $\alpha \ge 1/2$, the function $\frac{1}{T}\sum_{i=1}^T u_{\text{slin}}(\boldsymbol{Z}^{(i)}, A^{(i)}, \cdot)$ is piecewise L-Lipschitz with $L = \tilde{O}(MT^3n^5/\zeta^3)$, where $M = \max|a_{ij}^{(t)}|$, and u_{slin} is $(T^{\alpha-1}, O(nT^\alpha\sqrt{\log(n/\zeta)}))$-dispersed with respect to $\mathcal{S} = \{(A^{(t)}, \boldsymbol{Z}^{(t)})\}_{t=1}^T$.*

Proof sketch: We show that over the randomness of $\boldsymbol{Z}^{(1)}, \dots, \boldsymbol{Z}^{(T)}$, u_{slin} is (w,k)-dispersed. By definition of ϕ_s, the discontinuities of $u_{\text{slin}}(A^{(t)}, \boldsymbol{Z}^{(t)}, \cdot)$ have the form $s = |\langle \boldsymbol{u}_i^{(t)}, \boldsymbol{Z}^{(t)}\rangle|$, where $\boldsymbol{u}_i^{(t)}$ is the i^{th} vector in the solution to SDP-relaxation of $A^{(t)}$. These random variables have density bounded by $\sqrt{2/\pi}$. Let $\mathcal{B}_i$ be the set of discontinuities contributed by $\boldsymbol{u}_i^{(1)}, \dots, \boldsymbol{u}_i^{(T)}$. The points within each $\mathcal{B}_i$ are independent. We apply Lemma 1 with $P = n$ and $M = T$ and arrive at our dispersion guarantee. Proving that the piecewise portions of u_{slin} are Lipschitz is complicated by the fact that they are quadratic in $1/s$, so the slope may go to $\pm\infty$ as s goes to 0. However, if s is smaller than the smallest boundary s_0, $\sum_{i=1}^T u_{\text{slin}}(\boldsymbol{Z}^{(i)}, A^{(i)}, \cdot)$ is constant because ϕ_s deterministically maps the variables to -1 or 1, as in the GW algorithm. We prove that s_0 is not too small using anti-concentration bounds. ∎

VI. Dispersion in pricing problems and auction design

In this section, we study n-bidder, m-item posted price mechanisms and second price auctions. We denote all n buyers' valuations for all 2^m bundles $b_1, \dots, b_{2^m} \subseteq [m]$ by $\boldsymbol{v} = (v_1(b_1), \dots, v_1(b_{2^m}), \dots, v_n(b_1), \dots, v_n(b_{2^m}))$. We study buyers with additive valuations $\big(v_j(b) = \sum_{i\in b} v_j(\{i\})\big)$ and unit-demand valuations $(v_j(b) = \max_{i\in b} v_j(\{i\}))$. We also study buyers with general valuations, where there is no assumption on v_j beyond the fact that it is nonnegative, monotone, and $v_j(\emptyset) = 0$.

Posted price mechanisms are defined by m prices $\rho_1, \dots, \rho_m$ and a fixed ordering over the buyers. In order, each buyer has the option of buying her utility-maximizing bundle among the remaining items. In other words, suppose it is buyer j's turn in the ordering and let I be the set of items that buyers before her in the ordering did not buy. Then she will buy the bundle $b \subseteq I$ that maximizes $v_j(b) - \sum_{i\in b}\rho_i$.

Second price item auctions with anonymous reserve prices are defined by m *reserve prices* $\rho_1, \dots, \rho_m$. The bidders submit bids for each of the items. For each item i, the highest bidder wins the item if her bid is above ρ_i and she pays the maximum of the second highest bid for item i and ρ_i. These auctions are only *strategy proof* for additive bidders, which means that buyers have no incentive to misreport their values. Therefore, we restrict our attention to this setting and assume the bids equal the values.

In this setting, Π is a set of valuation vectors $\boldsymbol{v}$ and as in Section V, each $\mathcal{D}^{(t)}$ is a distribution over Π. The following results hold whenever the utility function corresponds to *revenue* (the sum of the payments) or *social surplus* (the

sum of the buyers' values for their allocations). The full proof is in the full version of the paper [46].

Theorem 9. *Suppose that $u(\boldsymbol{v}, \boldsymbol{\rho})$ is the social welfare (respectively, revenue) of the posted price mechanism with prices $\boldsymbol{\rho}$ and buyers' values $\boldsymbol{v}$. In this case, $L = 0$ (respectively, $L = 1$). For any $\alpha \geq 1/2$, the following are each true with probability at least $1 - \zeta$ over $\mathcal{S} \sim \mathcal{D}^{(1)} \times \cdots \times \mathcal{D}^{(T)}$.*

1) *Suppose the buyers have additive valuations and for each $\mathcal{D}^{(t)}$, the item values have κ-bounded marginal distributions. Then u is $(1/(\kappa T^{1-\alpha})), O(nmT^{\alpha}\sqrt{\ln((nm)/\zeta)}))$-dispersed with respect to $\mathcal{S}$.*
2) *Suppose the buyers are unit-demand with $v_j(\{i\}) \in [0, W]$. Also, suppose that for each $\mathcal{D}^{(t)}$, each bidder j, and every pair of items i and i', $v_j(\{i\})$ and $v_j(\{i'\})$ have a κ-bounded joint distribution. Then u is $(O(1/(W\kappa T^{1-\alpha})), O(nm^2T^{\alpha}\sqrt{\ln((nm)/\zeta)}))$-dispersed with respect to $\mathcal{S}$.*
3) *Suppose the buyers have general valuations in $[0, W]$. Also, suppose that for each $\mathcal{D}^{(t)}$, each bidder j, and every pair of bundles b and b', $v_j(b)$ and $v_j(b')$ have a κ-bounded joint distribution. Then u is $(1/(W\kappa T^{1-\alpha}), O(n2^{2m}T^{\alpha}\sqrt{\ln((n2^m)/\zeta)}))$-dispersed with respect to $\mathcal{S}$.*

Proof sketch: We sketch the proof for additive buyers. Given a valuation vector $\boldsymbol{v}$, let $\mathcal{P}_{\boldsymbol{v}}$ be the partition of $\mathcal{C}$ over which $u(\boldsymbol{v}, \cdot)$ is Lipschitz. We prove that the boundaries of $\mathcal{P}_{\boldsymbol{v}}$ correspond to a set of hyperplanes. Since the buyers are additive, these hyperplanes are axis-aligned: buyer j will be willing to buy item i at a price ρ_i if and only if $v_j(\{i\}) \geq \rho_i$. Next, consider a set $\mathcal{S} = \left\{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(T)}\right\}$ of buyers' valuations and the hyperplanes corresponding to each partition $\mathcal{P}_{\boldsymbol{v}^{(i)}}$. The key insight is that these hyperplanes can be partitioned into $P = nm$ buckets consisting of parallel hyperplanes with offsets independently drawn from κ-bounded distributions. For additive buyers, these sets of hyperplanes have the form $\{v_j^{(1)}(\{i\}) = \rho_i, \ldots, v_j^{(T)}(\{i\}) = \rho_i\}$ for every item i and every bidder j. Using Lemma 1, we show that within each bucket, the offsets are (w, k)-dispersed, for $w = O(1/(\kappa T^{1-\alpha}))$ and $k = \tilde{O}(nmT^{\alpha})$. Since the hyperplanes within each set are parallel, and since their offsets are dispersed, for any ball $\mathcal{B}$ of radius w in $\mathcal{C}$, at most k hyperplanes from each set intersect $\mathcal{B}$. By a union bound, this implies that the u is (w, nmk)-dispersed with respect to $\mathcal{S}$. ∎

We use a similar technique to analyze second-price item auctions. The full proof is in the full version [46], where we also show that Theorem 9 and the following theorem imply strong optimization guarantees in our settings.

Theorem 10. *Suppose that $u(\boldsymbol{v}, \boldsymbol{\rho})$ is the social welfare (respectively, revenue) of the second-price auction with reserves $\boldsymbol{\rho}$ and bids $\boldsymbol{v}$. In this case, $L = 0$ (respectively, $L = 1$). Also, for each $\mathcal{D}^{(t)}$ and each item i, suppose the distribution over $\max_{j \in [n]} v_j(\{i\})$ is κ-bounded. For any $\alpha \geq 1/2$, with probability $1 - \zeta$ over $\mathcal{S} \sim \bigtimes_{t=1}^{T} \mathcal{D}^{(t)}$, u is $(O(1/(\kappa T^{1-\alpha})), O(mT^{\alpha}\sqrt{\ln(m/\zeta)}))$-dispersed with respect to $\mathcal{S}$.*

VII. Generalization guarantees for distributional learning

It is known that regret bounds imply generalization guarantees for various online-to-batch conversion algorithms [34], but we also show that dispersion can be used to explicitly provide *uniform convergence guarantees,* which bound the difference between any function's average value on a set of samples drawn from a distribution and its expected value. Our primary tool is *empirical Rademacher complexity* [49, 50], which is defined as follows. Let $\mathcal{F} = \{f_{\boldsymbol{\rho}} : \Pi \to [0, 1] : \boldsymbol{\rho} \in \mathcal{C}\}$, where $\mathcal{C} \subset \mathbb{R}^d$ is a parameter space and let $\mathcal{S} = \{x_1, \ldots, x_T\} \subseteq \Pi$. (We use this notation for the sake of generality beyond algorithm selection, but mapping to the notation from Section I-A, $f_{\boldsymbol{\rho}}(x) = u(x, \boldsymbol{\rho})$.) The empirical Rademacher complexity of $\mathcal{F}$ with respect to $\mathcal{S}$ is defined as $\hat{R}(\mathcal{F}, \mathcal{S}) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{f \in \mathcal{F}} \frac{1}{T}\sum_{i=1}^{T} \sigma_i f(x_i)\right]$, where $\sigma_i \sim U(\{-1, 1\})$. Classic results from learning theory [49, 50] guarantee that for any distribution $\mathcal{D}$ over Π, with probability $1 - \zeta$ over $\mathcal{S} = \{x_1, \ldots, x_T\} \sim \mathcal{D}^T$, for all $f_{\boldsymbol{\rho}} \in \mathcal{F}$, $\left|\frac{1}{T}\sum_{i=1}^{T} f_{\boldsymbol{\rho}}(x_i) - \mathbb{E}_{x \sim \mathcal{D}}[f_{\boldsymbol{\rho}}(x)]\right| = O(\hat{R}(\mathcal{F}, \mathcal{S}) + \sqrt{\log(1/\zeta)/T})$. Our bounds depend on the the dispersion parameters of functions belonging to the *dual* class $\mathcal{G}$. That is, let $\mathcal{G} = \{u_x : \mathcal{C} \to \mathbb{R} : x \in \Pi\}$ be the set of functions $u_x(\boldsymbol{\rho}) = f_{\boldsymbol{\rho}}(x)$ where x is fixed and $\boldsymbol{\rho}$ varies. We bound $\hat{R}(\mathcal{F}, \mathcal{S})$ in terms of the dispersion parameters satisfied by $u_{x_1}, \ldots, u_{x_T} \in \mathcal{G}$. Moreover, even if these functions are not well dispersed, we can always upper bound $\hat{R}(\mathcal{F}, \mathcal{S})$ in terms of the pseudo-dimension of $\mathcal{F}$, denoted by $\operatorname{Pdim}(\mathcal{F})$ (we review the definition in Appendix in the full version of the paper [46]). The full proof is in the full version [46].

Theorem 11. *Let $\mathcal{F} = \{f_{\boldsymbol{\rho}} : \Pi \to [0, 1] : \boldsymbol{\rho} \in \mathcal{C}\}$ be parameterized by $\mathcal{C} \subset \mathbb{R}^d$, where $\mathcal{C}$ lies in a ball of radius R. For any set $\mathcal{S} = \{x_1, \ldots, x_T\}$, suppose the functions $u_{x_i}(\boldsymbol{\rho}) = f_{\boldsymbol{\rho}}(x_i)$ for $i \in [T]$ are piecewise L-Lipschitz and (w, k)-dispersed. Then $\hat{R}(\mathcal{F}, \mathcal{S}) \leq O(\min\{\sqrt{d\log(R/w)/T} + Lw + k/T, \sqrt{\operatorname{Pdim}(\mathcal{F})/T}\})$.*

Proof sketch: The key idea is that when the functions $u_{x_1}, \ldots, u_{x_T}$ are (w, k)-dispersed, any pair of parameters $\boldsymbol{\rho}$ and $\boldsymbol{\rho}'$ with $\|\boldsymbol{\rho} - \boldsymbol{\rho}'\|_2 \leq w$ satisfy $|f_{\boldsymbol{\rho}}(x_i) - f_{\boldsymbol{\rho}'}(x_i)| = |u_{x_i}(\boldsymbol{\rho}) - u_{x_i}(\boldsymbol{\rho}')| \leq Lw$ for all but at most k of the elements in $\mathcal{S}$. Therefore, we can approximate the functions in $\mathcal{F}$ on the set $\mathcal{S}$ with a finite subset $\hat{\mathcal{F}}_w = \{f_{\hat{\boldsymbol{\rho}}} : \hat{\boldsymbol{\rho}} \in \hat{\mathcal{C}}_w\}$, where $\hat{\mathcal{C}}_w$ is a w-net for $\mathcal{C}$. Since $\hat{\mathcal{F}}_w$ is finite, its empirical Rademacher complexity is $O((\log|\hat{\mathcal{F}}_w|/T)^{1/2})$. We then argue that the empirical Rademacher complexity of $\mathcal{F}$ is not much larger, since all functions in $\mathcal{F}$ are approximated by some function in $\hat{\mathcal{F}}_w$. ∎

VIII. Conclusion

We study online and private optimization for application-specific algorithm selection. We introduce a general condition, dispersion, that allows us to provide strong guarantees for both of these settings. As we demonstrate, many problems in algorithm and auction design reduce to optimizing dispersed functions. In this way, we connect learning theory, differential privacy, online learning, bandits, high dimensional sampling, computational economics, and algorithm design. Our main motivation is algorithm selection, but we expect that dispersion is even more widely applicable, opening up an exciting research direction.

Acknowledgements

The authors would like to thank Yishay Mansour for valuable feedback and discussion. This work was supported in part by NSF grants CCF-1422910, CCF-1535967, IIS-1618714, an Amazon Research Award, a Microsoft Research Faculty Fellowship, a Google Research Award, a NSF Graduate Research Fellowship, and a Microsoft Research Women's Fellowship.

References

[1] R. Gupta and T. Roughgarden, "A PAC approach to application-specific algorithm selection," *SIAM Journal on Computing*, vol. 46, no. 3, pp. 992–1017, 2017.

[2] V. Cohen-Addad and V. Kanade, "Online Optimization of Smoothed Piecewise Constant Functions," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[3] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge university press, 2006.

[4] R. Kleinberg, A. Slivkins, and E. Upfal, "Multi-armed bandits in metric spaces," in *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, 2008.

[5] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the Theory of Cryptography Conference (TCC)*. Springer, 2006, pp. 265–284.

[6] A. Blum and J. D. Hartline, "Near-optimal online auctions," in *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*. Society for Industrial and Applied Mathematics, 2005, pp. 1156–1163.

[7] A. Blum, V. Kumar, A. Rudra, and F. Wu, "Online learning in online auctions," *Theoretical Computer Science*, vol. 324, no. 2-3, pp. 137–146, 2004.

[8] S. Bubeck, N. R. Devanur, Z. Huang, and R. Niazadeh, "Online auctions and multi-scale online learning," *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2017.

[9] N. Cesa-Bianchi, C. Gentile, and Y. Mansour, "Regret minimization for reserve prices in second-price auctions," *IEEE Transactions on Information Theory*, vol. 61, no. 1, pp. 549–564, 2015.

[10] R. Kleinberg and T. Leighton, "The value of knowing a demand curve: Bounds on regret for online posted-price auctions," in *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, 2003.

[11] T. Roughgarden and J. R. Wang, "Minimizing regret with multiple reserves," in *Proceedings of the ACM Conference on Economics and Computation (EC)*. ACM, 2016, pp. 601–616.

[12] M. Dudík, N. Haghtalab, H. Luo, R. E. Schapire, V. Syrgkanis, and J. W. Vaughan, "Oracle-efficient learning and auction design," *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, 2017.

[13] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, 2007, pp. 94–103.

[14] R. Bassily, A. Smith, and A. Thakurta, "Differentially private empirical risk minimization: Efficient algorithms and tight error bounds," in *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, 2014.

[15] L. Lovász and S. Vempala, "Fast algorithms for logconcave functions: Sampling, rounding, integration, and optimization," in *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, 2006.

[16] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM (JACM)*, vol. 42, no. 6, pp. 1115–1145, 1995.

[17] M. Charikar and A. Wirth, "Maximizing quadratic programs: extending Grothendieck's inequality," in *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, 2004.

[18] U. Feige and M. Langberg, "The RPR^2 rounding technique for semidefinite programs," *Journal of Algorithms*, vol. 60, no. 1, pp. 1–23, 2006.

[19] U. Zwick, "Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to max cut and other problems," in *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, 1999.

[20] H. Yee and B. Ifrach, "Aerosolve: Machine learning for humans," *Open Source*, 2015. [Online]. Available: http://nerds.airbnb.com/aerosolve/

[21] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers *et al.*, "Practical lessons from predicting clicks on ads at Facebook," in *Proceedings of the International Workshop on Data Mining for Online Advertising*, 2014.

[22] E. Elkind, "Designing and learning optimal finite support auctions," in *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007.

[23] R. Cole and T. Roughgarden, "The sample complexity of revenue maximization," in *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, 2014.

[24] Z. Huang, Y. Mansour, and T. Roughgarden, "Making the most of your samples," in *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2015.

[25] A. M. Medina and M. Mohri, "Learning theory and algorithms for revenue optimization in second price auctions with reserve," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2014.

[26] J. Morgenstern and T. Roughgarden, "On the pseudo-dimension of nearly optimal auctions," in *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

[27] T. Roughgarden and O. Schrijvers, "Ironing in the dark," in *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2016.

[28] N. R. Devanur, Z. Huang, and C.-A. Psomas, "The sample complexity of auctions with side information," in *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, 2016.

[29] Y. A. Gonczarowski and N. Nisan, "Efficient empirical revenue maximization in single-parameter auction environments," in *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, 2017, pp. 856–868.

[30] J. Morgenstern and T. Roughgarden, "Learning simple auctions," in *Proceedings of the Conference on Learning Theory (COLT)*, 2016.

[31] M.-F. Balcan, T. Sandholm, and E. Vitercik, "Sample complexity of automated mechanism design," in *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2016.

[32] M.-F. Balcan, T. Sandholm, and E. Vitercik, "A general theory of sample complexity for multi-item profit maximization," *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2018.

[33] M.-F. Balcan, V. Nagarajan, E. Vitercik, and C. White, "Learning-theoretic foundations of algorithm configuration for combinatorial partitioning problems," *Proceedings of the Conference on Learning Theory (COLT)*, 2017.

[34] N. Cesa-Bianchi, A. Conconi, and C. Gentile, "On the generalization ability of on-line learning algorithms," in *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2002, pp. 359–366.

[35] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *Journal of Machine Learning Research*, vol. 12, no. Mar, pp. 1069–1109, 2011.

[36] M. Kusner, J. Gardner, R. Garnett, and K. Weinberger, "Differentially private Bayesian optimization," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2015, pp. 918–927.

[37] J. Weed, V. Perchet, and P. Rigollet, "Online learning in repeated auctions," in *Proceedings of the Conference on Learning Theory (COLT)*, 2016, pp. 1562–1583.

[38] Z. Feng, C. Podimata, and V. Syrgkanis, "Learning to bid without knowing your value," *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2018.

[39] A. Rakhlin, K. Sridharan, and A. Tewari, "Online learning: Stochastic, constrained, and smoothed adversaries," in *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2011.

[40] R. Kleinberg, "Nearly tight bounds for the continuum-armed bandit problem," in *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2004.

[41] K. Goldner and A. R. Karlin, "A prior-independent revenue-maximizing auction for multiple additive bidders," in *Proceedings of the Conference on Web and Internet Economics (WINE)*, 2016.

[42] A. M. Medina and S. Vassilvitskii, "Revenue optimization with approximate bid predictions," *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2017.

[43] V. Syrgkanis, "A sample complexity measure with applications to learning optimal auctions," *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2017.

[44] R. Myerson, "Optimal auction design," *Mathematics of Operation Research*, vol. 6, pp. 58–73, 1981.

[45] R. Gupta and T. Roughgarden, "A PAC approach to application-specific algorithm selection," in *Proceedings of the ACM Conference on Innovations in Theoretical Computer Science (ITCS)*. ACM, 2016, pp. 123–134.

[46] M.-F. Balcan, T. Dick, and E. Vitercik, "Dispersion for data-driven algorithm design, online learning, and private optimization," *arXiv preprint arXiv:1711.03091*, 2018.

[47] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Shapire, "The nonstochastic multiarmed bandit problem," in *SIAM Journal on Computing*, 2003.

[48] S. Sakai, M. Togasaki, and K. Yamazaki, "A note on greedy algorithms for the maximum weighted independent set problem," *Discrete Applied Mathematics*, vol. 126, no. 2, pp. 313–322, 2003.

[49] V. Koltchinskii, "Rademacher penalties and structural risk minimization," *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 1902–1914, 2001.

[50] P. L. Bartlett and S. Mendelson, "Rademacher and gaussian complexities: Risk bounds and structural results," *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 463–482, 2002.