Oracle-Efficient Online Learning and Auction Design

Miroslav Dudík*, Nika Haghtalab†, Haipeng Luo‡, Robert E. Schapire*, Vasilis Syrgkanis§, and Jennifer Wortman Vaughan*
*Microsoft Research, New York, USA, Email: {mdudik, schapire, jenn}@microsoft.com
†Carnegie Mellon University, Pittsburgh, USA, Email: nika@cmu.edu
‡University of Southern California, Los Angeles, USA, Email: haipengl@usc.edu
§Microsoft Research, Cambridge, USA, Email: vasy@microsoft.com

***Abstract*—We consider the design of computationally efficient online learning algorithms in an adversarial setting in which the learner has access to an offline optimization oracle. We present an algorithm called Generalized Follow-the-Perturbed-Leader and provide conditions under which it is oracle-efficient while achieving vanishing regret. Our results make significant progress on an open problem raised by Hazan and Koren [1], who showed that oracle-efficient algorithms do not exist in full generality and asked whether one can identify conditions under which oracle-efficient online learning may be possible. Our auction-design framework considers an auctioneer learning an optimal auction for a sequence of adversarially selected valuations with the goal of achieving revenue that is almost as good as the optimal auction in hindsight, among a class of auctions. We give oracle-efficient learning results for: (1) VCG auctions with bidder-specific reserves in single-parameter settings, (2) envy-free item-pricing auctions in multi-item settings, and (3) the level auctions of Morgenstern and Roughgarden [2] for single-item settings. The last result leads to an approximation of the overall optimal Myerson auction when bidders' valuations are drawn according to a fast-mixing Markov process, extending prior work that only gave such guarantees for the i.i.d. setting.**

We also derive various extensions, including: (1) oracle-efficient algorithms for the contextual learning setting in which the learner has access to side information (such as bidder demographics), (2) learning with approximate oracles such as those based on Maximal-in-Range algorithms, and (3) no-regret bidding algorithms in simultaneous auctions, which resolve an open problem of Daskalakis and Syrgkanis [3].

***Keywords*-online learning; auction design; revenue maximization; Follow-the-Perturbed-Leader**

I. Introduction

Online learning plays a major role in the adaptive optimization of computer systems, from the design of online marketplaces [4]–[7] to the optimization of routing schemes in communication networks [8]. The environments in these applications are constantly evolving, requiring continued adaptation of these systems. Online learning algorithms have been designed to robustly address this challenge, with performance guarantees that hold even when the environment is adversarial. However, the information-theoretically optimal learning algorithms that work with arbitrary payoff functions are computationally inefficient when the learner's action space is exponential in the natural problem representation [9]. For certain action spaces and environments, efficient online learning algorithms can be designed by reducing the online learning problem to an optimization problem [8], [10]–[12]. However, these approaches do not easily extend to the complex and highly non-linear problems faced by real learning systems, such as the learning systems used in online market design. In this paper, we address the problem of efficient online learning with an exponentially large action space under arbitrary learner objectives.

This goal is not achievable without some assumptions on the problem structure. Since an online optimization problem is at least as hard as the corresponding offline optimization problem [3], [13], a minimal assumption is the existence of an algorithm that returns a near-optimal solution to the offline problem. We assume that our learner has access to such an offline algorithm, which we call *an offline optimization oracle*. This oracle, for any (weighted) history of choices by the environment, returns an action of the learner that (approximately) maximizes the learner's reward. We seek to design *oracle-efficient learners*, that is, learners that run in polynomial time, with each oracle call counting $O(1)$.

An oracle-efficient learning algorithm can be viewed as a reduction from the online to the offline problem, providing conditions under which the online problem is not only as hard, but also as easy as the offline problem, and thereby offering computational equivalence between online and offline optimization. Apart from theoretical significance, reductions from online to offline optimization are also practically important. For example, if one has already developed and implemented a Bayesian optimization procedure which optimizes against a static stochastic environment, then our algorithm offers a black-box transformation of that procedure into an adaptive optimization algorithm with provable learning guarantees in non-stationary, non-stochastic environments. Even if the existing optimization system does not run in worst-case polynomial time, but is rather a well-performing fast heuristic, a reduction to offline optimization is capable of leveraging any expert domain knowledge that went into designing the heuristic, as well as any further improvements of the heuristic or even discovery of polynomial-time solutions.

Recent work of Hazan and Koren [1] shows that oracle-efficient learning in adversarial environments is not achievable in general, while leaving open the problem of identifying the properties under which oracle-efficient online learning may be possible [14]. We introduce a generic algorithm called *Generalized Follow-the-Perturbed-Leader* (Generalized FTPL) and derive sufficient conditions under which this algorithm yields oracle-efficient online learning. Our results are enabled by providing a new way of adding *regularization* so as to *stabilize* optimization algorithms in general optimization settings. The latter could be of independent interest beyond online learning. Our approach unifies and extends previous approaches to oracle-efficient learning, including the Follow-the-Perturbed Leader (FTPL) approach introduced by Kalai and Vempala [10] for linear objective functions, and its generalizations to submodular objective functions [12], adversarial contextual learning [15], and learning in simultaneous second-price auctions [3]. Furthermore, our sufficient conditions are related to the notion of a universal identification set of Goldman et al. [16] and oracle-efficient online optimization techniques of Daskalakis and Syrgkanis [3].

The second main contribution of our work is to introduce a new framework for the problem of adaptive auction design for revenue maximization and to demonstrate the power of Generalized FTPL through several applications in this framework. Traditional auction theory assumes that the valuations of the bidders are drawn from a population distribution which is known, thereby leading to a Bayesian optimization problem. The knowledge of the distribution by the seller is a strong assumption. Recent work in algorithmic mechanism design [2], [17]–[19] relaxes this assumption by solely assuming access to a set of samples from the distribution. In this work, we drop any distributional assumptions and introduce the adversarial learning framework of *online auction design*. On each round, a learner adaptively designs an auction rule for the allocation of a set of resources to a fresh set of bidders from a population.[1] The goal of the learner is to achieve average revenue at least as large as the revenue of the best auction from some target class. Unlike the standard approach to auction design, initiated by the seminal work of Myerson [20], our approach is devoid of any assumptions about a prior distribution on the valuations of the bidders for the resources at sale. Instead, similar to an agnostic approach in learning theory, we incorporate prior knowledge in the form of a target class of auction schemes that we want to compete with. This is especially appropriate when the auctioneer is restricted to using a particular design of auctions with power to make only a few design choices, such as deciding the reserve prices in a second-price auction. A special case of our framework is considered in the recent work of Roughgarden and Wang [7]. They study online learning of the class of single-item second-price auctions with bidder-specific reserves, and give an algorithm with performance that approaches a constant factor of the optimal revenue in hindsight. We go well beyond this specific setting and show that our Generalized FTPL can be used to optimize over several standard classes of auctions including VCG auctions with bidder-specific reserves and the level auctions of Morgenstern and Roughgarden [2], achieving low additive regret to the best auction in the class.

In the remainder of this section, we describe our main results and several extensions and applications in more detail, including (1) learning with side information (i.e., contextual learning); (2) learning with constant-factor approximate oracles (e.g., using Maximal-in-Range algorithms [21]); (3) regret bounds with respect to stronger benchmarks for the case in which the environment is not completely adversarial, but follows a fast-mixing Markov process. Most of the proofs and also formal statements of many of our results are deferred to the full version of this paper [22].

Our work contributes to two major research agendas: the design of efficient and oracle-efficient online learning algorithms [1], [3], [10]–[12], [23]–[26], and auction design using machine learning tools [2], [4], [6], [17], [18], [27]. The related work from both areas is described in more detail in the full version [22].

A. Oracle-Efficient Learning with Generalized FTPL

We consider the following online learning problem. On each round $t = 1, \ldots, T$, a learner chooses an action x_t from a finite set $\mathcal{X}$, and an adversary chooses an action y_t from a set $\mathcal{Y}$. The learner then observes y_t and receives a payoff $f(x_t, y_t) \in [0, 1]$, where the function f is fixed and known to the learner. The goal of the learner is to obtain low expected regret with respect to the best action in hindsight, i.e., to minimize

$$\textsc{Regret} := \mathbb{E}\left[\max_{x \in \mathcal{X}} \sum_{t=1}^{T} f(x, y_t) - \sum_{t=1}^{T} f(x_t, y_t)\right],$$

where the expectation is over the randomness of the learner.[2] We desire algorithms, called *no-regret algorithms*, for which this regret is sublinear in the time horizon T.

Our algorithm takes its name from the seminal Follow-the-Perturbed-Leader (FTPL) algorithm of Kalai and Vempala [10]. FTPL achieves low regret, $O(\sqrt{T \log |\mathcal{X}|})$, by independently perturbing the historical payoff of each of the learner's actions and choosing on each round the action with the highest perturbed payoff. However, this approach is

[1]Equivalently, the set of bidders on each round can be the same as long as they are myopic and optimize their utility separately in each round. Using our extension to contextual learning, this approach can also be applied when the learner's choice of auction is allowed to depend on features of the arriving set of bidders, such as demographic information.

[2]To simplify exposition, we assume that the adversary is oblivious, i.e., that the sequence $y_1, \ldots, y_T$ is chosen in advance. Our results generalize to adaptive adversaries using standard techniques [3], [28].

inefficient when the action space is exponential in the natural representation of the learning problem, because it requires creating $|\mathcal{X}|$ independent random variables.[3] Moreover, because of the form of the perturbation, the optimization of the perturbed payoffs cannot be performed by the offline optimization oracle for the same problem. We overcome both of these challenges by, first, generalizing FTPL to work with perturbations that can be compactly represented and are thus not necessarily independent across different actions (*sharing randomness*), and, second, by implementing such perturbations via synthetic histories of adversary actions (an approach introduced by Daskalakis and Syrgkanis [3]), thus creating offline problems of the same form as the online problem (*implementing randomness*).

Sharing Randomness: Our Generalized FTPL begins by drawing a random vector $\boldsymbol{\alpha} \in \mathbb{R}^N$ of dimension N, with components α_j drawn independently from a dispersed distribution D. The payoff of each of the learner's actions is perturbed by a linear combination of these independent variables, as prescribed by a *perturbation translation matrix* $\boldsymbol{\Gamma}$ of size $|\mathcal{X}| \times N$, with entries in $[0,1]$. Let $\boldsymbol{\Gamma}_x$ denote the row of $\boldsymbol{\Gamma}$ corresponding to x. On each round t, the algorithm outputs an action x_t that (approximately) maximizes the perturbed historical performance. In other words, x_t is chosen such that for all $x \in \mathcal{X}$,

$$\sum_{\tau=1}^{t-1} f(x_t, y_\tau) + \boldsymbol{\alpha} \cdot \boldsymbol{\Gamma}_{x_t} \geq \sum_{\tau=1}^{t-1} f(x, y_\tau) + \boldsymbol{\alpha} \cdot \boldsymbol{\Gamma}_x - \epsilon$$

for some fixed optimization accuracy $\epsilon \geq 0$. This procedure is fully described in Algorithm 1 of Section II.

We show that Generalized FTPL is no-regret as long as ϵ is sufficiently small and the translation matrix $\boldsymbol{\Gamma}$ satisfies an *admissibility* condition. This condition requires the rows of $\boldsymbol{\Gamma}$ to be (sufficiently) distinct so that each action's perturbation uses a different weighted combination of the N-dimensional noise. To the best of our knowledge, the approach of using an arbitrary matrix to induce shared randomness among actions of the learner is novel. The formal no-regret result is in Theorem II.5. The informal statement is the following:

Informal Theorem 1. *A translation matrix is δ-admissible if any two rows of the matrix are distinct and the minimum non-zero difference between any two values within a column is at least δ. Generalized FTPL with a δ-admissible matrix $\boldsymbol{\Gamma}$ and an appropriate distribution D achieves regret $O(N\sqrt{T}/\delta + \epsilon T)$.*

A technical challenge here is to show that the randomness induced by $\boldsymbol{\Gamma}$ on the set of actions $\mathcal{X}$ *stabilizes* the algorithm, i.e., the probability that $x_t \neq x_{t+1}$ is small. We use the admissibility of $\boldsymbol{\Gamma}$ to guide us through the analysis of stability. In particular, we consider how each column of $\boldsymbol{\Gamma}$ partitions actions of $\mathcal{X}$ to a few subsets (at most $1 + \delta^{-1}$) based on their corresponding entries in that column. Since the matrix rows are distinct, the algorithm is stable as a whole if, for each column, the partition to which the algorithm's chosen action belongs remains the same between consecutive rounds with probability close to 1. This allows us to decompose the stability analysis of the algorithm as a whole to the analysis of stability across partitions of each column. At the column level, stability of the partition between two rounds follows by showing that a switch between partitions happens only if the perturbation α_j corresponding to that column falls into a small sub-interval of the support of the distribution D, from which it is sampled. The latter probability is small if D is sufficiently dispersed. This final argument is similar in nature to the reason why perturbations lead to stability in the original FTPL [10].

Implementing Randomness: To ensure oracle-efficient learning, we additionally need the property that the induced action-level perturbations can be simulated by a (short) synthetic history of adversary actions. This allows us to avoid working with $\boldsymbol{\Gamma}$ directly, or even explicitly writing it down. This requirement is captured by our *implementability* condition, which states that each column of the translation matrix corresponds to a scaled version of the expected reward of the learner on some distribution of adversary actions. The formal statement is in Theorem II.9. The informal statement is the following:

Informal Theorem 2. *A translation matrix is implementable if each column corresponds to a scaled version of the expected reward of the learner against some finitely supported distribution of actions of the adversary. Generalized FTPL with an implementable translation matrix can be implemented with one oracle call per round and runs in time polynomial in N, T, and the size of the support of the distributions implementing the translation matrix. Oracle calls count $O(1)$ in the running time.*

The use of synthetic histories in online optimization was first explored by Daskalakis and Syrgkanis [3], who sample histories of length $\mathrm{poly}(|\mathcal{Y}|)$ from a fixed distribution. Our implementability property uses matrix $\boldsymbol{\Gamma}$ to obtain problem-specific distributions that stabilize online optimization with shorter histories.

For some learning problems, it is easier to first construct an implementable translation matrix and argue about its admissibility; for others, it is easier to construct an admissible matrix and argue about its implementability. We pursue both strategies in various applications, demonstrating the versatility of our conditions.

Our theorems yield the following simple sufficient condition for oracle-efficient no-regret learning (see Theorems II.5 and II.9 for more general statements):

[3] If payoffs are linear in some low-dimensional representation of $\mathcal{X}$ then the number of variables needed is equal to this dimension. But for non-linear payoffs, $|\mathcal{X}|$ variables are required.

> If there exist N adversary actions such that any two actions of the learner yield different rewards on at least one of these N actions, then Generalized FTPL with an appropriate translation matrix has regret $O(N\sqrt{T}/\delta)$ and its oracle-based runtime is $\mathrm{poly}(N, T)$ where δ is the smallest difference between distinct rewards obtainable on any one of the N adversary actions.

The aforementioned results establish a reduction from online optimization to offline optimization. Recall that in the oracle-based runtime, each oracle call counts $O(1)$. When the offline optimization problem can be solved in polynomial time, these results imply that the online optimization problem can also be solved in (fully) polynomial time. The formal statement is in Corollary II.10.

B. *Main Application: Online Auction Design*

In many applications of auction theory, including electronic marketplaces, a seller repeatedly sells an item or a set of items to a population of buyers, with a few arriving for each auction. In such cases, the seller can optimize his auction design in an online manner, using historical data consisting of observed bids. We consider a setting in which the seller would like to use this historical data to select an auction from a fixed target class. For example, a seller in a sponsored-search auction might be limited by practical constraints to consider only second-price auctions with bidder-specific reserves. The seller can optimize the revenue by using the historical data for each bidder to set these reserves. Similarly, a seller on eBay may be restricted to set a single reserve price for each item. Here, the seller can optimize the revenue by using historical data from auctions for similar goods to set the reserves for new items. In both cases, the goal is to leverage the historical data to pick an auction on each round in such a way that the seller's overall revenue compares favorably with the optimal auction from the target class.

More formally, on round $t = 1, \ldots, T$, a tuple of n bidders arrives with a vector of n bids or, equivalently, a vector of valuations (since we only consider truthful auctions), denoted $\mathbf{v}_t \in \mathcal{V}^n$. We allow these valuations to be arbitrary, e.g., chosen by an adversary. Prior to observing the bids, the auctioneer commits to an auction a_t from a class of truthful auctions $\mathcal{A}$. The goal of the auctioneer is to achieve a revenue that, in hindsight, is very close to the revenue that would have been achieved by the best fixed auction in class $\mathcal{A}$ if that auction were used on all rounds. In other words, the auctioneer aims to minimize the expected regret

$$\mathbb{E}\left[\max_{a\in\mathcal{A}} \sum_{t=1}^{T} \mathrm{Rev}(a, \mathbf{v}_t) - \sum_{t=1}^{T} \mathrm{Rev}(a_t, \mathbf{v}_t)\right],$$

where $\mathrm{Rev}(a, \mathbf{v})$ is the revenue of auction a on bid profile $\mathbf{v}$ and the expectation is over the actions of the auctioneer.

This problem can easily be cast in our oracle-efficient online learning framework. The learner's action space is the set of target auctions $\mathcal{A}$, while the adversary's action space is the set of bid or valuation vectors $\mathcal{V}^n$. The offline oracle is a revenue maximization oracle which computes an (approximately) optimal auction within the class $\mathcal{A}$ for any given set of valuation vectors. Using the Generalized FTPL with appropriate matrices $\mathbf{\Gamma}$, we provide the first oracle-efficient no-regret algorithms for three commonly studied auction classes:

- Vickrey-Clarkes-Groves (VCG) auctions with bidder-specific reserve prices in single-dimensional matroid settings, which are known to achieve half the revenue of the optimal auction in i.i.d. settings under some conditions [29];
- envy-free item-pricing mechanisms in combinatorial markets with unlimited supply, often studied in the static Bayesian setting [5], [30];
- single-item level auctions, introduced by Morgenstern and Roughgarden [2], who show that these auctions approximate, to an arbitrary accuracy, the Myerson auction [20], which is known to be optimal for the Bayesian independent-private-value setting.

The crux of our approach is in designing admissible and implementable matrices. For the first two mentioned classes, VCG auctions with bidder-specific reserves and envy-free item-pricing auctions, we show how to implement an (obviously admissible) matrix $\mathbf{\Gamma}$, where each row corresponds, respectively, to the concatenated binary representation of bidder reserves or item prices. We show that, surprisingly, any perturbation that is a linear function of this binary representation can be simulated by a distribution of bidder valuations. For the third class, level auctions, our challenge is to show that a clearly implementable matrix $\mathbf{\Gamma}$, with each column implemented by a single bid profile, is also admissible.

Table I summarizes the regret of our oracle-efficient algorithms and their computational efficiency, assuming oracle calls take $O(1)$ computation. All variants perform a single oracle call per round, so T oracle calls in total. The runtimes demonstrate an efficient reduction from the online problem to the offline problem.

C. *Extensions and Additional Applications*

We next overview several extensions and additional applications that appear in the full version [22]. Table II provides a summary.

Markovian Adversaries and Competing with the Optimal Auction: Morgenstern and Roughgarden [2] show that level auctions can provide an arbitrarily accurate approximation to the overall optimal Myerson auction in the Bayesian single-item auction setting if the values of the bidders are drawn from independent distributions and i.i.d. over time. Therefore, if the environment in an online setting picks

Table I
REGRET BOUNDS AND ORACLE-BASED RUNTIME FOR THE AUCTION CLASSES CONSIDERED IN THIS WORK, FOR n BIDDERS AND TIME HORIZON T. ALL OUR ALGORITHMS PERFORM A SINGLE ORACLE CALL PER ROUND.

Auction Class	Regret	Oracle-Based Runtime	Section
VCG with bidder-specific reserves, s-unit	$O(ns\sqrt{T}\log T)$	$O(T^2 + nT^{3/2}\log T)$	III-A
envy free k-item pricing with infinite supply and unit-demand or single-minded bidders	$O\big(nk\sqrt{T}\log(kT)\big)$	$O\big(T^2 + k^2T^{3/2}\log(kT)\big)$	full version [22]
level auction with discretization level m	$O(nm^2\sqrt{T})$	$O(T^2 + nmT)$	III-B

bidder valuations from independent distributions, standard online-to-batch reductions imply that the revenue of Generalized FTPL with the class of level auctions is close to the overall optimal (i.e., not just best-in-class) single-shot auction. We generalize this reasoning and show the same strong optimality guarantee when the valuations of bidders on each round are drawn from a fast-mixing Markov process that is independent across bidders but Markovian over rounds. For this setting, our results give an oracle-efficient algorithm with regret $O(n^{3/5}T^{9/10})$ to the overall optimal auction, rather than just best-in-class. This is the first result on competing with the Myerson optimal auction for non-i.i.d. distributions, as all prior work [2], [17]–[19] assumes i.i.d. samples.

Contextual Learning: In this setting, on each round t the learner observes a context σ_t before choosing an action. For example, in online auction design, the context might represent demographic information about the set of bidders. The goal of the learner is to compete with the best *policy* in some fixed class, where each policy is a mapping from a context σ_t to an action. We propose a contextual extension of the translation matrix $\mathbf{\Gamma}$. Generalized FTPL can be applied using this extended translation matrix and provides sublinear regret bounds for both the case in which there is a small "separator" of the policy class and the transductive setting in which the set of all possible contexts is known ahead of time. Our results extend and generalize the results of Syrgkanis et al. [15] from contextual combinatorial optimization to any learning setting that admits an implementable and admissible translation matrix.

The contextual extension is particularly useful in online auction design, because it allows the learner to use any side information available about the bidders before they place their bids to guide the choice of auction. While the number of bidders might be too large to learn about them individually, the learner can utilize the side information to design a common treatment for bidders that are similar, that is, to *generalize across a population*.

Our performance guarantees for adaptive auction design, similar to much prior work, rely on the assumption that the bidders are either myopic or are different on each round. One criticism of this assumption is that such adaptive mechanisms might be manipulated by strategic bidders who distort their bids so as to gain in the future. The contextual learning algorithms mitigate this risk by pooling similar bidders, which reduces the probability that the exact same bidder will be overly influential in the choices of the algorithm.

Approximate Oracles and Approximate Regret: For some problems there might not exist a sufficiently fast (e.g., polynomial-time or FPTAS) offline oracle with small additive error as required for Generalized FTPL. To make our results more applicable in practice, we extend them to handle oracles that are only required to return an action with performance that is within a constant multiplicative factor, $C \le 1$, of that of the optimal action in the class. We consider two examples of such oracles: Relaxation-based Approximations [5] and Maximal-in-Range (MIR) algorithms [21]. Our results hold in both cases with a modified version of regret, called C*-regret*, in which the online algorithm competes with C times the payoff of the optimal action in hindsight.

Additional Applications: Finally, we provide further applications of our work in the area of online combinatorial optimization with MIR approximate oracles, and in the area of no-regret learning for bid optimization in simultaneous second-price auctions.

- In the first application, we give a polynomial-time learning algorithm for online welfare maximization in multi-unit auctions that achieves $1/2$-regret, by invoking the polynomial-time MIR approximation algorithm of Dobzinski and Nisan [31] as an offline oracle.
- In the second application, we solve an open problem raised in the recent work of Daskalakis and Syrgkanis [3], who offered efficient learning algorithms only for the weaker benchmark of no-envy learning, rather than no-regret learning, in simultaneous second-price auctions, and left open the question of oracle-efficient no-regret learning. We show that no-regret learning in simultaneous item auctions is efficiently achievable, assuming access to an optimal bidding oracle against a known distribution of opponents bids (equivalently, against a distribution of item prices).

II. GENERALIZED FTPL AND ORACLE-EFFICIENT ONLINE LEARNING

In this section, we introduce the Generalized Follow-the-Perturbed-Leader (Generalized FTPL) algorithm and de-

Table II
ADDITIONAL RESULTS CONSIDERED IN THE FULL VERSION OF THE PAPER AND THEIR SIGNIFICANCE. HERE m IS THE DISCRETIZATION LEVEL OF THE PROBLEMS, n IS THE NUMBER OF BIDDERS, AND T IS THE TIME HORIZON.

Problem Class	Regret	Notes
Markovian, single item	$O(n^{3/5}T^{9/10})$	competes with Myerson's optimal auction
contextual online auction[a]	$O(\sqrt{T})$ or $O(T^{3/4})$	allows side information about bidders
welfare maximization, s-unit[b]	1/2-regret: $O(n^6\sqrt{T})$	fully polynomial-time algorithm
bidding in SiSPAs, k items	$O(km\sqrt{T})$	solves an open problem [3]

[a] The two regret bounds omit dependence on other parameters and are for the small separator setting and the transductive setting, respectively.
[b] The regime of interest in this problem is $s \gg n$.

scribe the conditions under which it efficiently reduces online learning to offline optimization.

As described in Section I-A, we consider the following online learning problem. On each round $t = 1, \ldots, T$, a learner chooses an action x_t from a finite set $\mathcal{X}$, and an adversary chooses an action y_t from a set $\mathcal{Y}$, which is not necessarily finite. The learner then observes y_t and receives a payoff $f(x_t, y_t) \in [0,1]$, where the function f is fixed and known to the learner. The goal of the learner is to obtain low expected regret with respect to the best action in hindsight, i.e., to minimize

$$\textsc{Regret} := \mathbb{E}\left[\max_{x\in\mathcal{X}} \sum_{t=1}^{T} f(x, y_t) - \sum_{t=1}^{T} f(x_t, y_t)\right],$$

where the expectation is over the randomness of the learner. An online algorithm is called a *no-regret algorithm* if its regret is sublinear in T, which means that its per-round regret goes to 0 as $T \to \infty$.

As discussed in Section I-A, our algorithm achieves sublinear regret by optimizing over a perturbed objective in each round. Unlike prior work [10], which creates an independent perturbation for every action, we create shared randomness among actions in $\mathcal{X}$. We first draw a random vector $\boldsymbol{\alpha} \in \mathbb{R}^N$ of size N, with components α_j drawn independently from a dispersed distribution D, and then perturb the payoff of each of the learner's actions by a linear combination of these independent variables, as prescribed by a *perturbation translation matrix* $\boldsymbol{\Gamma}$ of size $|\mathcal{X}| \times N$, with entries in $[0,1]$. The rows of $\boldsymbol{\Gamma}$, denoted $\boldsymbol{\Gamma}_x$, describe the linear combination for each action x. That is, on each round t, the payoff of each learner action $x \in \mathcal{X}$ is perturbed by $\boldsymbol{\alpha} \cdot \boldsymbol{\Gamma}_x$, and our Generalized FTPL algorithm outputs an action x that approximately maximizes $\sum_{\tau=1}^{t-1} f(x, y_\tau) + \boldsymbol{\alpha} \cdot \boldsymbol{\Gamma}_x$. This procedure is fully described in Algorithm 1. (For non-oblivious adversaries, a fresh random vector $\boldsymbol{\alpha}$ is drawn in each round.)

In the remainder of this section, we analyze the properties of matrix $\boldsymbol{\Gamma}$ that guarantee that Generalized FTPL is no-regret and that its perturbations can be efficiently transformed into synthetic history. These properties give rise to efficient reductions of online learning to offline optimization.

Algorithm 1: Generalized FTPL

Input: matrix $\boldsymbol{\Gamma} \in [0,1]^{|\mathcal{X}|\times N}$, distribution D over $\mathbb{R}$, and optimization accuracy $\epsilon \geq 0$.
Draw $\alpha_j \sim D$ independently for $j = 1, \ldots, N$.
for $t = 1, \ldots, T$ **do**
Choose any x_t such that for all $x \in \mathcal{X}$,

$$\sum_{\tau=1}^{t-1} f(x_t, y_\tau) + \boldsymbol{\alpha} \cdot \boldsymbol{\Gamma}_{x_t} \geq \sum_{\tau=1}^{t-1} f(x, y_\tau) + \boldsymbol{\alpha} \cdot \boldsymbol{\Gamma}_x - \epsilon.$$

Observe y_t and receive payoff $f(x_t, y_t)$.
end for

A. Regret Analysis

To analyze Generalized FTPL, we first bound its regret by the sum of a *stability* term, a *perturbation* term, and an *error* term in the following lemma. While this approach is standard [10], we include a proof in the full version of the paper for completeness.

Lemma II.1 (ϵ-FTPL Lemma)**.** *For Generalized FTPL,*

$$\textsc{Regret} \leq \mathbb{E}\left[\sum_{t=1}^{T} f(x_{t+1}, y_t) - f(x_t, y_t)\right] + \mathbb{E}\big[\boldsymbol{\alpha} \cdot (\boldsymbol{\Gamma}_{x_1} - \boldsymbol{\Gamma}_{x^*})\big] + \epsilon(T+1)$$

where $x^* = \arg\max_{x\in\mathcal{X}} \sum_{t=1}^{T} f(x, y_t)$.

In this lemma, the first term measures the stability of the algorithm, i.e., how often the action changes from round to round. The second term measures the strength of the perturbation, that is, how much the perturbation amount differs between the best action and the initial action. The third term measures the aggregated approximation error in choosing x_t that only approximately optimizes $\sum_{\tau=1}^{t-1} f(x, y_\tau) + \boldsymbol{\alpha} \cdot \boldsymbol{\Gamma}_x$.

To bound the stability term, we require that the matrix $\boldsymbol{\Gamma}$ be *admissible* and the distribution D be *dispersed* in the following sense.

Definition II.2 (δ-Admissible Translation Matrix)**.** *A translation matrix* $\boldsymbol{\Gamma}$ *is* admissible *if its rows are distinct. It is* δ-admissible *if it is admissible and distinct elements within* *each column differ by at least δ.*

Definition II.3 ((ρ, L)-Dispersed Distribution)**.** *A distribution D on the real line is (ρ, L)-dispersed if for any interval of length L, the probability measure placed by D on this interval is at most ρ.*

In the next lemma, we bound the stability term in Lemma II.1 by showing that with high probability, for all rounds t, we have $x_{t+1} = x_t$. At a high level, since all rows of an admissible matrix $\mathbf{\Gamma}$ are distinct, it suffices to show that the probability that $\mathbf{\Gamma}_{x_{t+1}} \neq \mathbf{\Gamma}_{x_t}$ is small. We prove this for each coordinate $\Gamma_{x_{t+1}j}$ separately, by showing that it is only possible to have $\Gamma_{x_{t+1}j} \neq \Gamma_{x_t j}$ when the random variable α_j falls in a small interval, which happens with only small probability for a sufficiently dispersed distribution D.[4]

Lemma II.4. *Consider Generalized FTPL with a δ-admissible matrix $\mathbf{\Gamma}$ with N columns and a $\big(\rho,\, (1{+}2\epsilon)\delta^{-1}\big)$-dispersed distribution D. Then,*

$$\mathbb{E}\left[\sum_{t=1}^{T} f(x_{t+1}, y_t) - f(x_t, y_t)\right] \le 2TN\rho(1 + \delta^{-1}).$$

Proof: Fix any $t \le T$. The bulk of the proof establishes that, with high probability, $\mathbf{\Gamma}_{x_{t+1}} = \mathbf{\Gamma}_{x_t}$, which by admissibility implies that $x_{t+1} = x_t$ and therefore $f(x_{t+1}, y_t) - f(x_t, y_t) = 0$.

Fix any $j \le N$. We first show that $\Gamma_{x_{t+1}j} = \Gamma_{x_t j}$ with high probability. Let V denote the set of values that appear in the j^{th} column of $\mathbf{\Gamma}$. By δ-admissibility, $|V| \le 1 + \delta^{-1}$. For any value $v \in V$, let x^v be any action that maximizes the perturbed cumulative payoff among those whose $\mathbf{\Gamma}$ entry in the j^{th} column equals v:

$$\begin{aligned} x^v &\in \operatorname*{arg\,max}_{x \in \mathcal{X}:\, \Gamma_{xj} = v} \left[\sum_{\tau=1}^{t-1} f(x, y_\tau) + \boldsymbol{\alpha} \cdot \mathbf{\Gamma}_x\right] \\ &\in \operatorname*{arg\,max}_{x \in \mathcal{X}:\, \Gamma_{xj} = v} \left[\sum_{\tau=1}^{t-1} f(x, y_\tau) + \boldsymbol{\alpha} \cdot \mathbf{\Gamma}_x - \alpha_j v\right]. \end{aligned}$$

For any $v, v' \in V$, define

$$\begin{aligned} \Delta_{vv'} = &\left(\sum_{\tau=1}^{t-1} f(x^v, y_\tau) + \boldsymbol{\alpha} \cdot \mathbf{\Gamma}_{x^v} - \alpha_j v\right) \\ &- \left(\sum_{\tau=1}^{t-1} f(x^{v'}, y_\tau) + \boldsymbol{\alpha} \cdot \mathbf{\Gamma}_{x^{v'}} - \alpha_j v'\right). \end{aligned}$$

Note that x^v and $\Delta_{vv'}$ are independent of α_j, as we removed the payoff perturbation corresponding to α_j.

If $\Gamma_{x_t j} = v$, then by the ϵ-optimality of x_t on the perturbed cumulative payoff, we have $\alpha_j(v' - v) - \epsilon \le \Delta_{vv'}$ for all $v' \in V$. Suppose $\Gamma_{x_{t+1}j} = v' \neq v$. Then by the optimality of $x^{v'}$ and the ϵ-optimality of x_{t+1},

$$\begin{aligned} &\sum_{\tau=1}^{t-1} f(x^{v'}, y_\tau) + f(x_{t+1}, y_t) + \boldsymbol{\alpha} \cdot \mathbf{\Gamma}_{x^{v'}} \\ &\ge \sum_{\tau=1}^{t-1} f(x_{t+1}, y_\tau) + f(x_{t+1}, y_t) + \boldsymbol{\alpha} \cdot \mathbf{\Gamma}_{x_{t+1}} \\ &\ge \sum_{\tau=1}^{t-1} f(x^v, y_\tau) + f(x^v, y_t) + \boldsymbol{\alpha} \cdot \mathbf{\Gamma}_{x^v} - \epsilon. \end{aligned}$$

Rearranging, we obtain for this same v' that

$$\begin{aligned} \Delta_{vv'} &\le \alpha_j(v' - v) + f(x_{t+1}, y_t) - f(x^v, y_t) + \epsilon \\ &\le \alpha_j(v' - v) + 1 + \epsilon. \end{aligned}$$

If $v' > v$, then $\alpha_j \ge \frac{\Delta_{vv'} - 1 - \epsilon}{v' - v} \ge \min_{\hat{v} \in V,\, \hat{v} > v} \frac{\Delta_{v\hat{v}} - 1 - \epsilon}{\hat{v} - v}$, and so $\alpha_j(\overline{v} - v) + 1 + \epsilon \ge \Delta_{v\overline{v}}$ where $\overline{v}$ is the value of $\hat{v}$ minimizing the expression on the right. Thus, in this case we have $-\epsilon \le \Delta_{v\overline{v}} - \alpha_j(\overline{v} - v) \le 1 + \epsilon$. Similarly, if $v' < v$, we have $-\epsilon \le \Delta_{v\underline{v}} - \alpha_j(\underline{v} - v) \le 1 + \epsilon$, where $\underline{v} = \arg\max_{\hat{v} \in V,\, \hat{v} < v} \frac{\Delta_{v\hat{v}} - 1 - \epsilon}{\hat{v} - v}$. So, we have

$$\begin{aligned} &\Pr\big[\Gamma_{x_{t+1}j} \neq \Gamma_{x_t j} \;\big|\; \alpha_k, k \neq j\big] \\ &\le \Pr\Big[\exists v \in V: -\epsilon \le \Delta_{v\overline{v}} - \alpha_j(\overline{v} - v) \le 1{+}\epsilon \;\text{ or } \\ &\qquad\qquad -\epsilon \le \Delta_{v\underline{v}} - \alpha_j(\underline{v} - v) \le 1{+}\epsilon \;\Big|\; \alpha_k, k \neq j\Big] \\ &\le \sum_{v \in V} \left(\Pr\left[\alpha_j \in \left[\tfrac{\Delta_{v\overline{v}} - 1 - \epsilon}{\overline{v} - v}, \tfrac{\Delta_{v\overline{v}} + \epsilon}{\overline{v} - v}\right] \;\middle|\; \alpha_k, k \neq j\right] \right. \\ &\qquad \left. + \Pr\left[\alpha_j \in \left[\tfrac{-\Delta_{v\underline{v}} - \epsilon}{v - \underline{v}}, \tfrac{-\Delta_{v\underline{v}} + 1 + \epsilon}{v - \underline{v}}\right] \;\middle|\; \alpha_k, k \neq j\right]\right) \\ &\le 2|V|\rho \le 2\rho(1 + \delta^{-1}). \end{aligned}$$

The first inequality on the last line follows from the fact that $\overline{v} - v \ge \delta$ and $v - \underline{v} \ge \delta$, the fact that D is $\big(\rho, \frac{1+2\epsilon}{\delta}\big)$-dispersed, and a union bound. The final inequality follows because $|V| \le 1 + \delta^{-1}$ by δ-admissibility.

Since this bound does not depend on the values of α_k for $k \neq j$, we can remove the conditioning and bound $\Pr[\Gamma_{x_{t+1}j} \neq \Gamma_{x_t j}] \le 2\rho(1 + \delta^{-1})$. Taking a union bound over all $j \le N$, we then have that, by admissibility, $\Pr[x_{t+1} \neq x_t] = \Pr\big[\mathbf{\Gamma}_{x_{t+1}} \neq \mathbf{\Gamma}_{x_t}\big] \le 2N\rho(1 + \delta^{-1})$, which implies the result. ■

To bound the regret, it remains to bound the perturbation term in Lemma II.1. This bound is specific to the distribution D. Many distribution families, including (discrete and continuous) uniform, Gaussian, Laplacian, and exponential can lead to a sublinear regret when the variance is set appropriately. Here we present a concrete regret analysis for a uniform distribution:

Theorem II.5. *Let $\mathbf{\Gamma}$ be a δ-admissible matrix with N columns and let D be the uniform distribution on $[0, 1/\eta]$ for $\eta = \delta/\sqrt{2T(1 + 2\epsilon)(1 + \delta)}$. Then, the regret of Generalized*

[4]The proof of Lemma II.4 implies a slightly tighter bound of $2TN\kappa\rho$, where κ is the maximum number of distinct elements in any column of $\mathbf{\Gamma}$. Note that δ-admissibility implies that $\kappa \le 1 + \delta^{-1}$.

FTPL can be bounded as

$$\textsc{Regret} \le \frac{N\sqrt{T}}{\delta} \cdot 2\sqrt{2(1+2\epsilon)(1+\delta)} \;+\; \epsilon(T+1) \ .$$

The proof follows from Lemmas II.1 and II.4, by bounding the perturbation term by $\|\boldsymbol{\alpha}\|_1 \le N/\eta$, then setting $\rho = \eta(1+2\epsilon)\delta^{-1}$, and finally using the value of η from the theorem, which minimizes the sum of the stability and perturbation terms, $2TN\eta(1+2\epsilon)\delta^{-1}(1+\delta^{-1}) + N/\eta$.

B. Oracle-Efficient Online Learning

We now define the offline oracle and oracle-efficient online learning more formally. Our oracles are defined for real-weighted datasets, but can be easily implemented by integer-weighted oracles (see full version [22]). Since many natural offline oracles are iterative optimization algorithms, which are only guaranteed to return an approximate solution in finite time, our definition assumes that the oracle takes the desired precision ϵ as an input. For ease of exposition, we assume that all numerical computations, even those involving real numbers, take $O(1)$ time (see full version [22] for a discussion).

Definition II.6 (Offline Oracle)**.** *An* offline oracle $\textsc{Opt}$ *is any algorithm that receives as input a weighted set of adversary actions $S = \{(w_\ell, y_\ell)\}_{\ell\in\mathcal{L}}$ with $w_\ell \in \mathbb{R}^+$, $y_\ell \in \mathcal{Y}$ and a desired precision ϵ, and returns an action $\hat{x} = \textsc{Opt}(S,\epsilon)$ such that*

$$\sum_{(w,y)\in S} wf(\hat{x},y) \ge \max_{x\in\mathcal{X}} \sum_{(w,y)\in S} wf(x,y) - \epsilon.$$

Definition II.7 (Oracle Efficiency)**.** *We say that an online algorithm is* oracle-efficient *with per-round complexity $g(\ldots)$ if its per-round running time is $O(g(\ldots))$ with oracle calls counting $O(1)$. Here $g(\ldots)$ denotes the fact that g may be a function of problem-specific parameters, including T.*

We next define a property of a translation matrix $\mathbf{\Gamma}$ which allows us to transform the perturbed objective into a dataset, thus achieving oracle-efficiency of Generalized FTPL:

Definition II.8. *A matrix $\mathbf{\Gamma}$ is* implementable *with complexity M if for each $j \in [N]$ there exists a weighted dataset S_j, with $|S_j| \le M$, such that for all $x, x' \in \mathcal{X}$:*

$$\Gamma_{xj} - \Gamma_{x'j} = \sum_{(w,y)\in S_j} w\big(f(x,y) - f(x',y)\big).$$

In this case, we say that weighted datasets S_j, $j \in [N]$, implement $\mathbf{\Gamma}$ with complexity M.

One simple but useful example of implementability is when each column j of $\mathbf{\Gamma}$ specifies the payoffs of learner's actions under a particular adversary action $y_j \in \mathcal{Y}$, i.e., $\Gamma_{xj} = f(x, y_j)$. In this case, $S_j = \{(1, y_j)\}$. Using an implementable $\mathbf{\Gamma}$ gives rise to an oracle-efficient variant of the Generalized FTPL, provided in Algorithm 2, in which

Algorithm 2: Oracle-Based Generalized FTPL

Input: datasets S_j implementing $\mathbf{\Gamma} \in [0,1]^{|\mathcal{X}|\times N}$, distribution D over $\mathbb{R}^+$, an offline oracle $\textsc{Opt}$.
Draw $\alpha_j \sim D$ independently for $j = 1, \ldots, N$.
for $t = 1, \ldots, T$ **do**
  Set $S = \big\{(1, y_1), \ldots, (1, y_{t-1})\big\} \cup \bigcup_{j\le N} \big\{(\alpha_j w, y) : (w,y) \in S_j\big\}$.
  Play $x_t = \textsc{Opt}\big(S, 1/\sqrt{T}\big)$.
  Observe y_t and receive payoff $f(x_t, y_t)$.
end for

we explicitly set $\epsilon = 1/\sqrt{T}$. Theorem II.9 shows that the output of this algorithm is equivalent to the output of Generalized FTPL and therefore the same regret guarantees hold. Note the assumption that the perturbations α_j are non-negative. The algorithm can be extended to negative perturbations when both $\mathbf{\Gamma}$ and $-\mathbf{\Gamma}$ are implementable. (See the full version for details.)

Theorem II.9. *If $\mathbf{\Gamma}$ is implementable with complexity M, then Algorithm 2 is an oracle-efficient implementation of Algorithm 1 with $\epsilon = 1/\sqrt{T}$ and has per-round complexity $O\big(T + NM\big)$.*

As an immediate corollary, the existence of a polynomial-time offline oracle implies the existence of polynomial-time online learner with regret $O(\sqrt{T})$ whenever we have access to an implementable and admissible matrix.

Corollary II.10. *Assume that $\mathbf{\Gamma} \in [0,1]^{|\mathcal{X}|\times N}$ is δ-admissible and implementable with complexity M, and that there exists an approximate offline oracle $\textsc{Opt}(\cdot, 1/\sqrt{T})$ that runs in time $\mathrm{poly}(N, M, T)$. Then Algorithm 2 with distribution D as defined in Theorem II.5 runs in time $\mathrm{poly}(N, M, T)$ and achieves regret $O(N\sqrt{T}/\delta)$.*

III. Online Auction Design

In this section, we apply the general techniques developed in Section II to obtain oracle-efficient no-regret algorithms for several common auction classes.

Consider a mechanism-design setting in which a seller wants to allocate $k \ge 1$ heterogeneous resources to a set of n bidders. The allocation to a bidder i is a subset of $\{1, \ldots, k\}$, which we represent as a vector in $\{0,1\}^k$, and the seller has some feasibility constraints on the allocations across bidders. Each bidder $i \in [n]$ has a combinatorial valuation function $v_i \in \mathcal{V}$, where $\mathcal{V} \subseteq \big(\{0,1\}^k \to [0,1]\big)$. We use $\mathbf{v} \in \mathcal{V}^n$ to denote the vector of valuation functions across all bidders. A special case of the setting is that of multi-item auctions for k heterogeneous items, where each resource is an item and the feasibility constraint simply states that no item is allocated to more than one bidder. Another special case is that of *single-parameter (service-based) environments*, which we describe in more detail in Section III-A.

An auction a takes as input a *bid profile* consisting of reported valuations for each bidder, and returns both the allocation for each bidder i and the price that he is charged. In this work, we only consider *truthful auctions*, where each bidder maximizes his utility by reporting his true valuation, irrespective of what other bidders report. We therefore make the assumption that each bidder reports v_i as their bid and refer to $\mathbf{v}$ not only as the valuation profile, but also as the bid profile throughout the rest of this section. The allocation that the bidder i receives is denoted $\mathbf{q}_i(\mathbf{v}) \in \{0,1\}^k$ and the price that he is charged is $p_i(\mathbf{v})$; we allow sets $\mathbf{q}_i(\mathbf{v})$ to overlap across bidders, and drop the argument $\mathbf{v}$ when it is clear from the context. We consider bidders with quasilinear utilities: the utility of bidder i is $v_i(\mathbf{q}_i(\mathbf{v})) - p_i(\mathbf{v})$. For an auction a with price function $\mathbf{p}(\cdot)$, we denote by $\mathrm{Rev}(a, \mathbf{v})$ the *revenue of the auction* for bid profile $\mathbf{v}$, i.e., $\mathrm{Rev}(a, \mathbf{v}) = \sum_{i \in [n]} p_i(\mathbf{v})$.

Fixing a class of truthful auctions $\mathcal{A}$ and a set of possible valuations $\mathcal{V}$, we consider the problem in which on each round $t = 1, \dots, T$, a learner chooses an auction $a_t \in \mathcal{A}$ while an adversary chooses a bid profile $\mathbf{v}_t \in \mathcal{V}^n$. The learner then observes $\mathbf{v}_t$ and receives revenue $\mathrm{Rev}(a_t, \mathbf{v}_t)$. The goal of the learner is to obtain low expected regret with respect to the best auction from $\mathcal{A}$ in hindsight. That is, we would like to guarantee that

$$\begin{aligned}\textsc{Regret} &:= \mathbb{E}\left[\max_{a \in \mathcal{A}} \sum_{t=1}^{T} \mathrm{Rev}(a, \mathbf{v}_t) - \sum_{t=1}^{T} \mathrm{Rev}(a_t, \mathbf{v}_t)\right] \\ &\le o(T)\mathrm{poly}(n, k).\end{aligned}$$

We require our online algorithm to be oracle-efficient, assuming access to an ϵ-optimal offline optimization oracle that takes as input a weighted set of bid profiles, $S = \{(w_\ell, \mathbf{v}_\ell)\}_{\ell \in \mathcal{L}}$, and returns an auction that achieves an approximately optimal revenue on S, i.e., a revenue at least $\max_{a \in \mathcal{A}} \sum_{(w,\mathbf{v}) \in S} w\mathrm{Rev}(a, \mathbf{v}) - \epsilon$. Throughout the section, we assume that there exists such an oracle for $\epsilon = 1/\sqrt{T}$, as needed in Algorithm 2.

Using the language of oracle-based online learning developed in Section II, the learner's action set $\mathcal{X}$ corresponds to the set of auctions $\mathcal{A}$, the adversary's action set $\mathcal{Y}$ corresponds to the set of bid profiles $\mathcal{V}^n$, the payoff of the learner f corresponds to the revenue generated by the auction, Rev, and we assume access to an offline optimization oracle $\textsc{Opt}$.

For several of the auction classes we consider, such as multi-item or multi-unit auctions, the revenue of an auction on a bid profile is in range $[0, R]$ for $R > 1$. In order to use the results of Section II, we implicitly re-scale all the revenue functions by dividing them by R before applying Theorem II.5. Note that, since $\mathbf{\Gamma}$ does not change, the admissibility condition keeps the regret of the normalized problem at $O(N\sqrt{T}/\delta)$, according to Theorem II.5. We then scale up to get a regret bound that is R times the regret for the normalized problem, i.e., $O(RN\sqrt{T}/\delta)$. This re-scaling does not increase the runtime, because the complexity of implementing $\mathbf{\Gamma}$ is unchanged, only the weights appearing in sets S_j are scaled up by a factor of R.

We now derive results for two auction classes: VCG auctions with bidder-specific reserves and level auctions. Each auction class is formally defined in its respective subsection. The full version of the paper also analyzes envy-free item-pricing auctions.

A. VCG with Bidder-Specific Reserves

In this section, we consider a standard class of auctions, VCG auctions with bidder-specific reserve prices, which we define more formally below and denote by $\mathcal{I}$. These auctions are known to approximately maximize the revenue when bidder valuations are drawn from independent (but not necessarily identical) distributions [29]. Recently, Roughgarden and Wang [7] considered online learning for this class and provided a computationally efficient algorithm whose total revenue is at least $1/2$ of the best revenue among auctions in $\mathcal{I}$, minus a term that is $o(T)$. We apply the techniques from Section II to generate an oracle-efficient online algorithm with low *additive* regret with respect to the optimal auction in the class $\mathcal{I}$, without any loss in multiplicative factors.

We go beyond single-item auctions and consider general *single-parameter* environments. In these environments, each bidder has one piece of private valuation for receiving a *service*, i.e., being included in the set of winning bidders. We allow for some combinations of bidders to be *served* simultaneously, and let $\mathcal{S} \subseteq 2^{[n]}$ be the family of feasible sets, i.e., sets of bidders that can be served simultaneously; with some abuse of notation we write $\mathbf{q} \in \mathcal{S}$, to mean that the set represented by the binary allocation vector $\mathbf{q}$ is in $\mathcal{S}$. We assume that any bidder is allowed to be the sole bidder served, i.e., that $\{i\} \in \mathcal{S}$ for all i, and that it is also allowed that no bidder be served, i.e., $\emptyset \in \mathcal{S}$.[5] Examples of such environments include single-item single-unit auctions (for which $\mathcal{S}$ contains only singletons and the empty set), single-item s-unit auctions (for which $\mathcal{S}$ contains any subset of size at most s), and combinatorial auctions with single-minded bidders. In the last case, we begin with some set of original items, define the service as receiving the desired bundle of items, and let $\mathcal{S}$ contain any subset of bidders seeking disjoint sets of items.

In a basic VCG auction, an allocation $\mathbf{q}^* \in \mathcal{S}$ is chosen to maximize social welfare, that is, maximize $\sum_{i=1}^n v_i q_i^*$, where we slightly simplify notation and use $v_i \in [0,1]$ to denote the valuation of bidder i for being served. Each bidder who is served is then charged the externality he imposes on others, $p_i(\mathbf{v}) = \max_{\mathbf{q} \in \mathcal{S}} \sum_{i' \neq i} v_{i'} q_{i'} - \sum_{i' \neq i} v_{i'} q_{i'}^*$, which can be shown to equal the minimum bid at which he would

[5] A more common and stronger assumption used in previous work [7], [29] is that $\mathcal{S}$ is a downward-closed matroid.

be served. Such auctions are known to be truthful. The most common example is the second-price auction for the single-item single-unit case in which the bidder with the highest bid receives the item and pays the second highest bid. VCG auctions with reserves, which maintain the property of truthfulness, are defined as follows.

Definition III.1 (VCG auctions with bidder-specific reserves)**.** *A VCG auction with bidder-specific reserves is specified by a vector* $\mathbf{r}$ *of reserve prices for each bidder. As a first step, all bidders whose bids are below their reserves (that is, bidders* i *for which* $v_i < r_i$*) are removed from the auction. If no bidders remain, no item is allocated. Otherwise, the basic VCG auction is run on the remaining bidders to determine the allocation. Each bidder who is served is charged the larger of his reserve and his VCG payment.*

Fixing the set $\mathcal{S}$ of feasible allocations, we denote by $\mathcal{I}$ the class of all VCG auctions with bidder-specific reserves. With a slight abuse of notation we write $\mathbf{r} \in \mathcal{I}$ to denote the auction with reserve prices $\mathbf{r}$. To apply the results from Section II, which require a finite action set for the learner, we limit attention to the finite set of auctions $\mathcal{I}_m \subseteq \mathcal{I}$ consisting of those auctions in which the reserve price for each bidder is a strictly positive integer multiple of $1/m$, i.e., those where $r_i \in \{1/m, \ldots, m/m\}$ for all i. We will show for some common choices of $\mathcal{S}$ that the best auction in this class yields almost as high a revenue as the best auction in $\mathcal{I}$.

We next show how to design a matrix $\mathbf{\Gamma}$ for $\mathcal{I}_m$ that is admissible and implementable. As a warmup, suppose we use the $|\mathcal{I}_m| \times n$ matrix $\mathbf{\Gamma}$ with entries $\Gamma_{\mathbf{r},i} = \mathrm{Rev}(\mathbf{r}, \mathbf{e}_i)$. That is, the i^{th} column of $\mathbf{\Gamma}$ corresponds to the revenue of each auction on a bid profile in which bidder i has valuation 1 and all others have valuation 0. By definition, $\mathbf{\Gamma}$ is implementable with complexity 1 using $S_j = \{(1, \mathbf{e}_j)\}$ for each $j \in [n]$. Moreover, $\mathrm{Rev}(\mathbf{r}, \mathbf{e}_i) = r_i$ so any two rows of $\mathbf{\Gamma}$ are different and $\mathbf{\Gamma}$ is thus $1/m$-admissible. By Theorems II.5 and II.9, we obtain an oracle-efficient implementation of the Generalized FTPL with regret $O(nm/\sqrt{T})$.

To improve this regret bound and obtain a regret that is polynomial in $\log m$ rather than m, we carefully construct another translation matrix that is implementable using a more complex dataset of adversarial actions. The translation matrix we design is quite intuitive. The row corresponding to an auction $\mathbf{r}$ contains a binary representation of its reserve prices. In this case, proving admissibility of the matrix is simple. The challenge is to show that this translation matrix is implementable using a dataset of adversarial actions.

Construction of $\mathbf{\Gamma}$: Let $\mathbf{\Gamma}^{\text{VCG}}$ be an $|\mathcal{I}_m| \times (n\lceil \log m \rceil)$ binary matrix, where the i^{th} collection of $\lceil \log m \rceil$ columns contains the binary encodings of the auctions' reserve prices for bidder i. More formally, for any $i \leq n$ and a bit position $\beta \leq \lceil \log m \rceil$, let $j = (i-1)\lceil \log m \rceil + \beta$ and set $\mathbf{\Gamma}^{\text{VCG}}_{\mathbf{r},j}$ to

	Binary encoding				
Auction $\mathbf{r}$	r_1		r_2		
$(1/3, 1/3)$	0	1	0	1	
$(1/3, 2/3)$	0	1	1	0	$\Delta = -1$
$(1/3, 3/3)$	0	1	1	1	
$(2/3, 1/3)$	1	0	0	1	
$(2/3, 2/3)$	1	0	1	0	$\Delta = 0$
$(2/3, 3/3)$	1	0	1	1	
$(3/3, 1/3)$	1	1	0	1	$\Delta' = 1$
$(3/3, 2/3)$	1	1	1	0	$\Delta' = -1$
$(3/3, 3/3)$	1	1	1	1	

Figure 1. $\mathbf{\Gamma}^{\text{VCG}}$ for $n = 2$ bidders and discretization $m = 3$.

be the β^{th} bit of mr_i.

Lemma III.2. $\mathbf{\Gamma}^{\text{VCG}}$ *is* 1*-admissible and implementable with complexity* m*.*

We defer the proof of this lemma to the full version of the paper. Here, we illustrate the main ideas through a simple example.

Example III.3. *Consider* $\mathbf{\Gamma}^{\text{VCG}}$ *for* $n = 2$ *bidders and* $m = 3$ *discretization levels, as demonstrated in Figure 1. As an example, we show how one can go about implementing columns* 1 *and* 4 *of* $\mathbf{\Gamma}^{\text{VCG}}$*.*

Consider the first column of $\mathbf{\Gamma}^{\text{VCG}}$*. It corresponds to the most significant bit of* r_1*. To implement this column, we need to find a weighted set of bid profiles that generate revenues with the same differences as those between the column entries. We consider bid profiles* $\mathbf{v}_h = (h/3, 0)$ *for* $h \in \{1, 2, 3\}$*, with the revenue* $\mathrm{Rev}(\mathbf{r}, \mathbf{v}_h) = r_1 \mathbf{1}_{(h/3 \geq r_1)}$*. To obtain the weights* w_h *for each* $\mathbf{v}_h$ *it is necessary (and sufficient) to match differences between entries corresponding to reserve prices with* $r_1 = \frac{1}{3}$ *vs* $\frac{2}{3}$*, and* $r_1 = \frac{2}{3}$ *vs* $\frac{3}{3}$ *(denoted by* Δ *in Figure 1), corresponding to the following equations:*

$$\frac{1}{3}(w_1 + w_2 + w_3) - \frac{2}{3}(w_2 + w_3) = -1,$$
$$\frac{2}{3}(w_2 + w_3) - \frac{3}{3}(w_3) = 0,$$

where the left-hand sides are the differences in the revenues and the right-hand sides are the differences Δ *between the corresponding column entries. Note that the weighted set* $S_1 = \{(3, \mathbf{v}_1), (2, \mathbf{v}_2), (4, \mathbf{v}_3)\}$ *satisfies these equations and implements the first column. Similarly, for implementing the fourth column, we consider bid profiles* $\mathbf{v}'_h = (0, h/3)$ *for* $h \in \{1, 2, 3\}$ *and equations dictated by the differences* Δ'*. One can verify that* $S_4 = \{(6, \mathbf{v}'_1), (0, \mathbf{v}'_2), (3, \mathbf{v}'_3)\}$ *implements this column.*

More generally, the proof of Lemma III.2 shows that $\mathbf{\Gamma}^{\text{VCG}}$ *is implementable by showing that any differences in values in one column that solely depend on a single bidder's reserve price lead to a feasible system of linear equations.*

The next theorem follows immediately from Lemma III.2, Theorems II.5 and II.9, and the fact that the maximum revenue is at most R. Note that R is bounded by the number of bidders that can be served simultaneously, which is at most n.

Theorem III.4. *Consider the online auction design problem for the class of VCG auctions with bidder-specific reserves, $\mathcal{I}_m$. Let $R = \max_{\mathbf{r},\mathbf{v}} \mathrm{Rev}(\mathbf{r},\mathbf{v})$ and let D be the uniform distribution as described in Theorem II.5. Then, the Oracle-Based Generalized FTPL algorithm with D and datasets that implement $\mathbf{\Gamma}^{\mathrm{VCG}}$ is oracle-efficient with per-round complexity $O(T + nm\log m)$ and has regret*

$$\mathbb{E}\left[\max_{\mathbf{r}\in\mathcal{I}_m}\sum_{t=1}^{T}\mathrm{Rev}(\mathbf{r},\mathbf{v}_t) - \sum_{t=1}^{T}\mathrm{Rev}(\mathbf{r}_t,\mathbf{v}_t)\right] \le O(nR\sqrt{T}\log m).$$

Now we return to the infinite class $\mathcal{I}$ of all VCG auctions with reserve prices $r_i \in [0,1]$. We show that $\mathcal{I}_m$ is a finite "cover" for this class when the family of feasible sets $\mathcal{S}$ consists of all subsets of size at most s, corresponding to single-item single-unit auctions (when $s = 1$) or more general single-item s-unit auctions. In such auctions, the items are allocated to the s highest bids that are above their reserve. We assume that the ties are resolved in favor of bidders with a lower index. We prove in the full version of the paper that for these auctions, the optimal revenue of $\mathcal{I}_m$ compared with that of $\mathcal{I}$ can decrease by at most $2s/m$ at each round. That is,

$$\max_{\mathbf{r}\in\mathcal{I}}\sum_{t=1}^{T}\mathrm{Rev}(\mathbf{r},\mathbf{v}_t) - \max_{\mathbf{r}\in\mathcal{I}_m}\sum_{t=1}^{T}\mathrm{Rev}(\mathbf{r},\mathbf{v}_t) \le \frac{2Ts}{m}.$$

Setting $m = \sqrt{T}$ and using Theorem III.4, we obtain the following result for the class of auctions $\mathcal{I}$.

Theorem III.5. *Consider the online auction design problem for the class of VCG auctions with bidder-specific reserves, $\mathcal{I}$, in s-unit auctions. Let D be the uniform distribution as described in Theorem II.5. Then the Oracle-Based Generalized FTPL algorithm with D and datasets that implement $\mathbf{\Gamma}^{\mathrm{VCG}}$ with $m = \sqrt{T}$ is oracle-efficient with per-round complexity $O(T + n\sqrt{T}\log T)$ and has regret*

$$\mathbb{E}\left[\max_{\mathbf{r}\in\mathcal{I}}\sum_{t=1}^{T}\mathrm{Rev}(\mathbf{r},\mathbf{v}_t) - \sum_{t=1}^{T}\mathrm{Rev}(\mathbf{r}_t,\mathbf{v}_t)\right] \le O(ns\sqrt{T}\log T).$$

B. Level Auctions

We next consider the class of *level auctions* introduced by Morgenstern and Roughgarden [2], who show that these auctions can achieve $(1-\epsilon)$-approximate revenue maximization if the valuations of the bidders are drawn independently (but not necessarily identically) from a distribution, thus approximating Myerson's optimal auction [20]. We derive oracle-efficient no-regret algorithms for this auction class.

The s-level auctions realize a single-item single-unit allocation as follows:

Definition III.6. *Given $s \ge 2$, an s-level auction $\boldsymbol{\theta}$ is defined by s distinct thresholds for each bidder i, $0 \le \theta_0^i < \cdots < \theta_{s-1}^i \le 1$. For any bid profile $\mathbf{v}$, we let $b_i^{\boldsymbol{\theta}}(v_i)$ denote the index b of the largest threshold $\theta_b^i \le v_i$, or -1 if $v_i < \theta_0^i$. If $v_i < \theta_0^i$ for all i, the item is not allocated. Otherwise, the item goes to the bidder with the largest index $b_i^{\boldsymbol{\theta}}(v_i)$, breaking ties in favor of bidders with smaller i. The winner pays the price equal to the minimum bid that he could have submitted and still won the item.*

When it is clear from the context, we omit $\boldsymbol{\theta}$ in $b_i^{\boldsymbol{\theta}}(v_i)$ and write just $b_i(v_i)$. We consider a class of auctions $\mathcal{S}_{s,m}$, consisting of s-level auctions with thresholds in the set $\{0, 1/m, \ldots, m/m\}$.

To construct an admissible and implementable $\mathbf{\Gamma}$ for $\mathcal{S}_{s,m}$, we begin with a matrix that is clearly implementable, with each column implemented by a single bid profile, and then show its admissibility.

We consider the bid profiles in which the only non-zero bids are $v_n = \ell/m$ for some $0 \le \ell \le m$, and $v_i = 1$ for a single bidder $i < n$. Note that bidder i wins the item in any such profile and pays θ_b^i corresponding to $b = \max\{0, b_n(v_n)\}$. We define a matrix $\mathbf{\Gamma}$ with one column for every bid profile of this form and an additional column for the bid profile $\mathbf{e}_n$, with the entries in each row consisting of the revenue of the corresponding auction on the given bid profile. Clearly, $\mathbf{\Gamma}$ is implementable. As for admissibility, take $\boldsymbol{\theta} \in \mathcal{S}_{s,m}$ and the corresponding row $\mathbf{\Gamma}_{\boldsymbol{\theta}}$. Note that as $v_n = \ell/m$ increases for $\ell = 0, \ldots, m$, there is an increase in $b_n(\ell/m) = -1, 0, \ldots, s-1$, possibly skipping the initial -1. As the level $b_n(v_n)$ increases, the auction revenue attains the values $\theta_0^i, \theta_1^i, \ldots, \theta_{s-1}^i$, changing exactly at those points where v_n crosses thresholds $\theta_1^n, \ldots, \theta_{s-1}^n$. Since any two consecutive thresholds of $\boldsymbol{\theta}$ are different, the thresholds of θ_b^i for $b \ge 0$ and θ_b^n for $b \ge 1$ can be reconstructed by analyzing the revenue of the auction and the values of v_n at which the revenue changes. The remaining threshold θ_0^n is equal to the revenue of the bid profile $\mathbf{v} = \mathbf{e}_n$. Since all of the parameters of the auction can be recovered from the entries in the row $\mathbf{\Gamma}_{\boldsymbol{\theta}}$, this shows that any two rows of $\mathbf{\Gamma}$ are different and $\mathbf{\Gamma}$ is $1/m$-admissible. This reasoning is summarized in the following construction and the corresponding lemma. See Figure 2 for more intuition.

Construction of $\mathbf{\Gamma}$: For $i \in \{1, \ldots, n-1\}$ and $\ell \in \{0, \ldots, m\}$, let $\mathbf{v}^{i,\ell} = \mathbf{e}_i + (\ell/m)\mathbf{e}_n$. Let $V = \{\mathbf{v}^{i,\ell}\}_{i,\ell} \cup \{\mathbf{e}_n\}$. Let $\mathbf{\Gamma}^{\mathrm{SL}}$ be the matrix of size $|\mathcal{S}_{s,m}| \times |V|$ with entries indexed by $(\boldsymbol{\theta}, \mathbf{v}) \in \mathcal{S}_{s,m} \times V$, such that $\Gamma^{\mathrm{SL}}_{\boldsymbol{\theta},\mathbf{v}} = \mathrm{Rev}(\boldsymbol{\theta}, \mathbf{v})$.

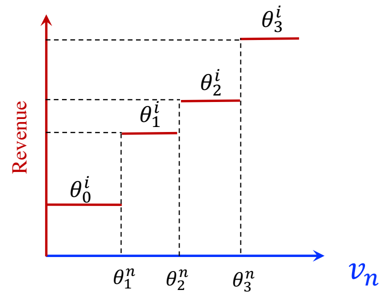

Figure 2. Demonstration of how $\boldsymbol{\theta}$ can be reconstructed by its revenue on the bid profiles in $V = \{\mathbf{v}^{i,\ell}\}_{i,\ell} \cup \{\mathbf{e}_n\}$. In this figure, the horizontal axis is the value of v_n and the vertical axis is the revenue of the auction when $v_i = 1$ and all other valuations are 0. As demonstrated, as the value v_n increases from 0 to 1, the revenue of the auction forms a step function with values $\theta_0^i, \theta_1^i, \ldots, \theta_{s-1}^i$, where the jumps in the values happen when v_n takes values $\theta_1^n, \theta_2^n, \ldots, \theta_{s-1}^n$.

Lemma III.7. *Γ^{SL} is $1/m$-admissible and implementable with complexity 1.*

Our next theorem is an immediate consequence of Lemma III.7, Theorems II.5 and II.9, and the fact that the revenue of the mechanism in each round is at most 1.

Theorem III.8. *Consider the online auction design problem for the class $\mathcal{S}_{s,m}$ of s-level auctions. Let D be the uniform distribution as described in Theorem II.5. Then the Oracle-Based Generalized FTPL algorithm with D and datasets that implement Γ^{SL} is oracle-efficient with per-round complexity $O(T + nm)$ and has regret*

$$\mathbb{E}\left[\max_{\boldsymbol{\theta} \in \mathcal{S}_{s,m}} \sum_{t=1}^{T} \mathrm{Rev}(\boldsymbol{\theta}, \mathbf{v}_t) - \sum_{t=1}^{T} \mathrm{Rev}(\boldsymbol{\theta}_t, \mathbf{v}_t)\right] \leq O(nm^2\sqrt{T}).$$

IV. Conclusion

We introduced a general-purpose no-regret algorithm for the online adversarial setting and gave sufficient conditions under which it is oracle-efficient. We hope our work serves as a stepping stone towards deeper understanding of such algorithms.

References

[1] E. Hazan and T. Koren, "The computational power of optimization in online learning," in *STOC*, 2016.
[2] J. H. Morgenstern and T. Roughgarden, "On the pseudo-dimension of nearly optimal auctions," in *NIPS*, 2015.
[3] C. Daskalakis and V. Syrgkanis, "Learning in auctions: Regret is hard, envy is easy," in *FOCS*, 2016.
[4] A. Blum and J. D. Hartline, "Near-optimal online auctions," in *SODA*, 2005.
[5] M.-F. Balcan and A. Blum, "Approximation algorithms and online mechanisms for item pricing," in *EC*, 2006.
[6] N. Cesa-Bianchi, C. Gentile, and Y. Mansour, "Regret minimization for reserve prices in second-price auctions," in *SODA*, 2013.
[7] T. Roughgarden and J. R. Wang, "Minimizing regret with multiple reserves," in *EC*, 2016.
[8] B. Awerbuch and R. Kleinberg, "Online linear optimization and adaptive routing," *JCSS*, vol. 74, no. 1, pp. 97–114, Feb. 2008.
[9] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *JCSS*, vol. 55, no. 1, pp. 119–139, 1997.
[10] A. Kalai and S. Vempala, "Efficient algorithms for online decision problems," *JCSS*, vol. 71, no. 3, pp. 291–307, 2005.
[11] S. Kakade and A. T. Kalai, "From batch to transductive online learning," in *NIPS*, 2005.
[12] E. Hazan and S. Kale, "Online submodular minimization," *JMLR*, vol. 13, no. Oct, pp. 2903–2922, 2012.
[13] N. Cesa-Bianchi, A. Conconi, and C. Gentile, "On the generalization ability of on-line learning algorithms," *IEEE Trans. on Information Theory*, vol. 50, no. 9, pp. 2050–2057, 2004.
[14] E. Hazan and T. Koren, "Learning in games with best-response oracles (talk)," 2016. [Online]. Available: https://simons.berkeley.edu/sites/default/files/docs/5595/koren.pdf
[15] V. Syrgkanis, A. Krishnamurthy, and R. E. Schapire, "Efficient algorithms for adversarial contextual learning," in *ICML*, 2016.
[16] S. A. Goldman, M. J. Kearns, and R. E. Schapire, "Exact identification of read-once formulas using fixed points of amplification functions," *SICOMP*, vol. 22, no. 4, pp. 705–726, 1993.
[17] R. Cole and T. Roughgarden, "The sample complexity of revenue maximization," in *STOC*, 2014.
[18] N. R. Devanur, Z. Huang, and C.-A. Psomas, "The sample complexity of auctions with side information," in *STOC*, 2016.
[19] T. Roughgarden and O. Schrijvers, "Ironing in the dark," in *EC*, 2016.
[20] R. B. Myerson, "Optimal auction design," *Mathematics of operations research*, vol. 6, no. 1, pp. 58–73, 1981.
[21] N. Nisan and A. Ronen, "Computationally feasible VCG mechanisms," *JAIR*, vol. 29, no. 1, pp. 19–47, May 2007.
[22] M. Dudík, N. Haghtalab, H. Luo, R. E. Schapire, V. Syrgkanis, and J. W. Vaughan, "Oracle-efficient learning and auction design," *arXiv preprint arXiv:1611.01688*, 2016.
[23] A. Agarwal, D. Hsu, S. Kale, J. Langford, L. Li, and R. Schapire, "Taming the monster: A fast and simple algorithm for contextual bandits," in *ICML*, 2014.
[24] M. Dudik, D. Hsu, S. Kale, N. Karampatziakis, J. Langford, L. Reyzin, and T. Zhang, "Efficient optimal learning for contextual bandits," in *UAI*, 2011.
[25] A. Rakhlin and K. Sridharan, "Bistro: An efficient relaxation-based method for contextual bandits," in *ICML*, 2016.
[26] V. Syrgkanis, H. Luo, A. Krishnamurthy, and R. E. Schapire, "Improved regret bounds for oracle-based adversarial contextual bandits," in *NIPS*, 2016.
[27] R. Kleinberg and T. Leighton, "The value of knowing a demand curve: Bounds on regret for online posted-price auctions," in *FOCS*, 2003.
[28] M. Hutter and J. Poland, "Adaptive online prediction by following the perturbed leader," *JMLR*, vol. 6, pp. 639–660, 2005.
[29] J. D. Hartline and T. Roughgarden, "Simple versus optimal mechanisms," in *EC*, 2009.
[30] V. Guruswami, J. D. Hartline, A. R. Karlin, D. Kempe, C. Kenyon, and F. McSherry, "On profit-maximizing envy-free pricing," in *SODA*, 2005.
[31] S. Dobzinski and N. Nisan, "Mechanisms for multi-unit auctions," in *EC*, 2007.