
Adaptive Contract Design for Crowdsourcing

Chien-Ju Ho*
UCLA

Aleksandrs Slivkins
Microsoft Research

Jennifer Wortman Vaughan
Microsoft Research and UCLA

Abstract

Crowdsourcing markets have emerged as a popular platform for matching available workers with tasks to complete. The payment for a particular task is typically set by the task’s requester, and may be adjusted based on the quality of the completed work, for example, through the use of “bonus” payments. In this paper, we study the requester’s problem of dynamically adjusting quality-contingent payments for tasks. We consider a multi-round version of the well-known *principal-agent* model, whereby in each round a worker makes a strategic choice of the effort level which is not directly observable by the requester. In particular, our formulation significantly generalizes the budget-free online task pricing problems studied in prior work.

We treat this problem as a multi-armed bandit problem, with each “arm” representing a potential contract. To cope with the large (and in fact, infinite) number of arms, we propose a new algorithm, AgnosticZooming, which discretizes the contract space into a finite number of regions, effectively treating each region as a single arm. This discretization is adaptively refined, so that more promising regions of the contract space are eventually discretized more finely. We provide a full analysis of this algorithm, showing that it achieves regret sublinear in the time horizon and substantially improves over non-adaptive discretization (which is the only competing approach in the literature).

1 Introduction

Crowdsourcing harnesses human intelligence and common sense to complete tasks that are difficult to accomplish using computers alone. Crowdsourcing markets, such as Amazon Mechanical Turk and Microsoft’s Universal Human Relevance System, are platforms designed to match available human workers with tasks to complete. Using these platforms, requesters may post tasks that they would like completed, along with the amount of money they are willing to pay. Workers then choose whether or not to accept the available tasks and complete the work.

Of course not all human workers are equal, nor is all human-produced work. Some tasks, such as proofreading a paragraph of English text, are easier for some workers than others, requiring less effort to produce high quality results. Additionally, some workers are more dedicated than others, willing to spend the extra time to make sure a task is completed properly. To encourage high quality results, requesters may set quality-contingent “bonus” payments on top of the base payment for each task, rewarding workers for producing valuable output. This can be viewed as offering workers a “contract” that specifies how much they will be paid based on the quality of their output.¹

In this paper, we examine the requester’s problem of dynamically setting quality-contingent payments for tasks. We consider a setting in which time evolves in rounds. In each round, the requester posts a new contract, a performance-contingent payment rule which specifies different levels of payment for different levels of output. A random, unidentifiable worker then arrives in the market and strategically decides whether to accept the requester’s task and how much effort to exert; the choice of effort level is not directly observable by the requester. After the worker completes

*Much of this research was completed while Ho was at Microsoft Research.

¹For some tasks, such as labeling websites as relevant to a particular search query or not, verifying the quality of work may be as difficult as completing the task. These tasks can be assigned in batches, with each batch containing one or more instances in which the correct answer is already known. Quality-contingent payments can then be based on the known instances.

the task (or chooses not to complete it), the requester observes the worker’s output, pays the worker according to the offered contract, and adjusts the contract for the next round. The goal of the requester is to maximize his own expected utility, the value he receives from completed work minus the payments made. We call it the *dynamic contract design* problem.

We treat this problem as a multi-armed bandit (MAB) problem, with each arm representing a potential contract. Since the action space is large (and in fact infinite) and has a well-defined real-valued structure, it is natural to consider an algorithm that uses *discretization*. Our algorithm, *AgnosticZooming*, divides the action space into regions, and chooses among these regions, effectively treating each region as a single “meta-arm.” The discretization is defined *adaptively*, so that the more promising areas of the action space are eventually discretized more finely than the less promising areas. While the general idea of adaptive discretization has appeared in prior work on MAB [12, 26, 41, 42], our approach to adaptive discretization is new and problem-specific. The main difficulty, compared to this prior work, is that an algorithm is not given any information that links the observable numerical structure of contracts and the expected utilities thereof.

To analyze performance, we propose a concept called “width dimension” which measures how “nice” a particular problem instance is. We show that *AgnosticZooming* achieves regret sublinear in the time horizon for problem instances with small width dimension. In particular, if the width dimension is d , it achieves regret $O(\log T \cdot T^{(d+1)/(d+2)})$ after T rounds. For problem instances with large width dimension, *AgnosticZooming* matches the performance of the naive algorithm which uniformly discretizes the space and runs a standard bandit algorithm. We illustrate our general results via some special cases, including an improvement over the prior work on online task pricing.

Related work. Our work builds on three areas of research. First, our model can be viewed as a multi-round version of the classical *principal-agent* model from contract theory [28]: the single round of our model corresponds to the basic principal-agent setting. However, our techniques are very different from those employed in contract theory.

Second, our methods build on those developed in the rich literature on MAB with continuous outcome spaces. The closest line of work is that on *Lipschitz MAB* [26], in which the algorithm is given a distance function on the arms, and the expected rewards of the arms are assumed to satisfy Lipschitz-continuity (or a relaxation thereof) with respect to this distance function [1, 3, 12, 23, 25, 26, 31, 32, 42]. Most related to our techniques is the idea of adaptive discretization [12, 26, 42], and in particular, the *zooming algorithm* [26, 42]. However, the zooming algorithm cannot be applied directly in our setting because the required numerical similarity information is not immediately available. This problem also arises in web search and advertising, where it is natural to assume that an algorithm can only observe a tree-shaped taxonomy on arms [27, 35, 36] which can be used to explicitly reconstruct relevant parts of the underlying metric space [41]. We take a different approach, using a notion of “virtual width” to estimate similarity information. Explicit comparisons between our results and prior MAB work are made throughout the paper.

Finally, our work follows several other theoretical papers on pricing in crowdsourcing markets. The problem closest to ours which has been studied in this context is the *online task pricing* problem in which a requester has a (possibly limited) supply of tasks to be completed and a budget B to spend on them [6, 7, 39, 40] (see [24] for the initial result on the budget-free version). The main difference between the budget-free version of their setting and ours is that in our setting the workers’ strategic choice of effort level is not directly observable, and therefore cannot be directly priced by the requester. A more thorough literature review can be found in Appendix A.

2 Our setting: the dynamic contract design problem

In this section, we formally define the problem that we set out to solve. We start by describing a *static model*, which captures what happens in a single round of interaction between a requester and a worker. As described above, this is a version of the standard *principal-agent* model [28]. We then define our *dynamic model*, an extension of the static model to multiple rounds, with a new worker arriving each round. We then detail the objective of our pricing algorithm and the simplifying assumptions that we make throughout the paper. Finally, we compare our setting to the classic multi-armed bandit problem.

Static model. We begin with a description of what occurs during each interaction between the requester and a single worker. The requester first posts a task which may be completed by the worker, and a *contract* specifying how the worker will be paid if she completes the task. If the task is completed, the requester pays the worker as specified in the contract, and the requester derives value from the completed task; for normalization, we assume that the value derived is in $[0, 1]$. The requester’s utility from a given task is this value minus the payment to the worker.

When the worker observes the contract and decides whether or not to complete the task, she also chooses a level of effort to exert, which in turn determines her cost (in terms of time, energy, or missed opportunities) and a distribution over the quality of her work. To model quality, we assume that there is a (small) set of possible *outcomes* that result from the worker completing the task (or choosing not to complete it), and that the realized outcome determines the value that the requester derives from the task. The realized outcome is observed by the requester, and the contract that the requester offers is a mapping from outcomes to payments for the worker.

The worker’s utility from a given task is the payment from the requester minus the cost corresponding to her chosen effort level. Given the contract she is offered, the worker chooses her effort level strategically so as to maximize her expected utility. Crucially, the chosen effort level is not directly observable by the requester.

By convention, the worker’s choice *not* to perform a task is modeled as a separate effort level of zero cost (called the *null* effort level) and a separate outcome of zero value and zero payment (called the *null* outcome) such that the null effort level deterministically leads to the null outcome, and it is the only effort level that can lead to this outcome.

The mapping from outcomes to the requester’s value is called the requester’s *value function*. The mapping from effort levels to costs is called the *cost function*, and the mapping from effort levels to distributions over outcomes is called the *production function*. For the purposes of this paper, a worker is completely specified by these two functions; we say that the cost function and the production function comprise the worker’s *type*. Unlike some traditional versions of the principal-agent problem, in our setting a worker’s type is not observable by the requester, nor is any prior given.

Dynamic model. The dynamic model we consider in this paper is a natural extension of the static model to multiple rounds and multiple workers. We are still concerned with just a single requester. In each round, a new worker arrives. We assume a *stochastic environment* in which the worker’s type in each round is an i.i.d. sample from some fixed and unknown distribution over types, called the *supply distribution*. The requester posts a new task and a contract for this task. All tasks are of the same type, in the sense that the set of possible effort levels and the set of possible outcomes are the same for all tasks. The worker strategically chooses her effort level so as to maximize her expected utility from this task. Based on the chosen effort level and the worker’s production function, an outcome is realized. The requester observes this outcome (but not the worker’s effort level) and pays the worker the amount specified by the contract. The type of the arriving worker is never revealed to the requester. The requester can adjust the contract from one round to another, and his total utility is the sum of his utility over all rounds. For simplicity, we assume that the number of rounds is known in advance, though this assumption can be relaxed using standard tricks.

The dynamic contract design problem. Throughout this paper, we take the point of view of the requester interacting with workers in the dynamic model. The algorithms we examine dynamically choose contracts to offer on each round with the goal of maximizing the requester’s expected utility. A problem instance consists of several quantities, some of which are known to the algorithm, and some of which are not. The known quantities are the number of outcomes, the requester’s value function, and the time horizon T (i.e., the number of rounds). The latent quantities are the number of effort levels, the set of worker types, and the supply distribution. The algorithm adjusts the contract from round to round and observes the realized outcomes but receives no other feedback.

We focus on contracts that are *bounded* (offer payments in $[0, 1]$), and *monotone* (assign equal or higher payments for outcomes with higher value for the requester). We compare a given algorithm against the maximal expected utility achievable by any bounded, monotone contract, denoted OPT, within a set of discretized contracts. More precisely, we are interested in bounding the algorithm’s *regret* $R(T)$, defined as $T \times \text{OPT}$ minus the expected utility of the algorithm.

Notation. Let $v(\cdot)$ be the value function of the requester, with $v(\pi)$ denoting the value of outcome π . Let \mathcal{O} be the set of all outcomes and let m be the number of non-null outcomes. We will index the outcomes as $\mathcal{O} = \{0, 1, 2, \dots, m\}$ in the order of increasing value (ties broken arbitrarily), with a convention that 0 is the null outcome.

Let $c_i(\cdot)$ and $f_i(\cdot)$ be the cost function and production function for type i . Then the cost of choosing effort level e is $c_i(e)$, and the probability of obtaining outcome π having chosen effort e is $f_i(\pi|e)$. Let $F_i(\pi|e) = \sum_{\pi' \geq \pi} f_i(\pi'|e)$.

Recall that a contract x is a function from outcomes to (non-negative) payments. If contract x is offered to a worker sampled i.i.d. from the supply distribution, $V(x)$ is the expected value to the requester, $P(x) \geq 0$ is the expected payment, and $U(x) = V(x) - P(x)$ is the expected utility of the requester.

Assumption: First-order stochastic dominance (FOSD). Given two effort levels e and e' , we say that e has FOSD over e' for type i if $F_i(\pi|e) \geq F_i(\pi|e')$ for all outcomes π , with a strict inequality for at least one outcome.² We say

²This mimics the standard notion of FOSD between two distributions over a linearly ordered set.

that type i satisfies the FOSD assumption if for any two distinct effort levels, one effort level has FOSD over the other for type i . We assume that all types satisfy this assumption.

Assumption: Consistent tie-breaking. If multiple effort levels maximize the expected utility of a given worker for a contract x , we assume the tie is broken consistently in the sense that this worker chooses the same effort level for any contract that leads to this particular tie. This assumption is minor; it can be avoided (with minor technical complications) by adding random perturbations to the contracts. This assumption is implicit throughout the paper.

Our benchmark. We compare a given algorithm with the *omniscient benchmark*, an algorithm which optimizes its actions over time given the full knowledge of the latent information in the problem instance. This is a standard benchmark for many machine learning problems. In our setting this benchmark reduces to the best fixed contract from a set, that is, the contract that maximizes the requester’s expected utility in any given round.

Our benchmark only considers contracts that are bounded and monotone within an arbitrary discretized set. In practice, restricting to such contracts may be appealing to all human parties involved. However, this restriction is not without loss of generality: there are problem instances in which monotone contracts are not optimal; see Appendix B for an example. Further, it is not clear whether bounded monotone contracts are optimal among monotone contracts.

Alternative worker models. One of the crucial tenets in our model is that the workers maximize their expected utility. This “rationality assumption” is very standard in Economics, and is often used to make the problem amenable to rigorous analysis. However, there is a considerable literature suggesting that in practice workers may deviate from this “rational” behaviour. Thus, it is worth pointing out that our results do not rely heavily on the rationality assumption. The FOSD assumption (which is also fairly standard) can be circumvented, too. In fact, all our assumptions regarding worker behavior serve *only* to guarantee that the collective worker behavior satisfies the following natural property: if the requester increases the “increment payment” (described in the next section) for a particular outcome, the probability of obtaining an outcome at least that good also increases. This property is used in the proof of Lemma 3.1.

Comparison to multi-armed bandits (MAB). Dynamic contract design can be modeled as special case of the MAB problem with some additional, problem-specific structure. The basic MAB problem is defined as follows. An algorithm repeatedly chooses actions from a fixed *action space* and collects rewards for the chosen actions; the available actions are traditionally called *arms*. More specifically, time is partitioned into rounds, so that in each round the algorithm selects an arm and receives a reward for the chosen arm. No other information, such as the reward the algorithm would have received for choosing an alternative arm, is revealed. In an MAB problem with *stochastic rewards*, the reward of each arm in a given round is an i.i.d. sample from some distribution which depends on the arm but not on the round. A standard measure of algorithm’s performance is regret with respect to the best fixed arm, defined as the difference in expected total reward between a benchmark (usually the best fixed arm) and the algorithm.

Thus, dynamic contract design can be naturally modeled as an MAB problem with stochastic rewards, in which arms correspond to monotone contracts. While the prior work on MAB with large / infinite action spaces often assumes explicit upper bounds on similarity between arms, in our setting such upper bounds are absent. On the other hand, our problem has some supplementary structure compared to the standard MAB setting. In particular, the algorithm’s reward decomposes into value and payment, both of which are determined by the outcome, which in turn is probabilistically determined by the worker’s strategic choice of the effort level. Effectively, this supplementary structure provides some “soft” information on similarity between contracts, in the sense that numerically similar contracts are usually (but not always) similar to one another.

3 Our algorithm: AgnosticZooming

In this section, we specify our algorithm. We call it *AgnosticZooming* because it “zooms in” on more promising areas of the action space, and does so without knowing a precise measure of the similarity between contracts. This zooming can be viewed as a dynamic form of discretization. Before stating the algorithm itself, we discuss the discretization of the action space in more detail, laying the groundwork for our approach.

3.1 Discretization of the action space

In each round, the *AgnosticZooming* algorithm partitions the action space into several regions and chooses among these regions, effectively treating each region as a “meta-arm.” In this section, we discuss which subsets of the action space are used as regions, and introduce some useful notions and properties of such subsets.

Candidate contracts. Our algorithm is parameterized by a subset X_{cand} of bounded, monotone contracts (for example, all bounded, monotone contracts with payments that are integer multiples of ψ , for some $\psi > 0$). Contracts in X_{cand} are called *candidate contracts*. The goal of the algorithm is to compete against the best candidate contract. There are two reasons we set this as our goal. First, particular crowdsourcing platforms may enforce that payments are multiples of some minimum unit (e.g., one cent), in which case it is natural to restrict attention to contracts satisfying this constraint. Second, it is not clear whether or not the discretization error, i.e., the difference between the best candidate contract and the best unconstrained contract, is bounded. In the most simple setting with only one non-null outcome, we can bound the discretization error and therefore bound the regret to the best unconstrained contract if we have a bound on regret to the best candidate contract. However, it is not trivial to extend this bound to the setting with multiple non-null outcomes. We hope to include more discussion of this subtle issue in the full version of this paper.

Increment space and cells. To describe our approach to discretization, it is useful to think of contracts in terms of *increment payments*. Specifically, we represent each monotone contract $x : \mathcal{O} \rightarrow [0, \infty)$ as a vector $\mathbf{x} \in [0, \infty)^m$, where m is the number of positive outcomes and $\mathbf{x}_\pi = x(\pi) - x(\pi - 1) \geq 0$ for each positive outcome π . (Recall that by convention 0 is the null outcome and $x(0) = 0$.) We call this vector the *increment representation* of contract x , and denote it $\text{incr}(x)$. Note that if x is bounded, then $\text{incr}(x) \in [0, 1]^m$. Conversely, call a contract *weakly bounded* if it is monotone and its increment representation lies in $[0, 1]^m$. Such a contract is not necessarily bounded.

We discretize the space of all weakly bounded contracts, viewed as a multi-dimensional unit cube. More precisely, we define the *increment space* as $[0, 1]^m$ with a convention that every vector represents the corresponding weakly bounded contract. Each region in the discretization is a closed, axis-aligned m -dimensional cube in the increment space; henceforth, such cubes are called *cells*. A cell is called *relevant* if it contains at least one candidate contract. A relevant cell is called *atomic* if it contains exactly one candidate contract, and *composite* otherwise.

In each composite cell C , the algorithm will only use two contracts: the *maximal corner*, denoted $x^+(C)$, in which all increment payments are maximal, and the *minimal corner*, denoted $x^-(C)$, in which all increment payments are minimal. These may or may not be in the set X_{cand} . In each atomic cell C , the algorithm will only use one contract: the unique candidate contract. Those contracts used by the algorithm are called the *anchors* of C .

Virtual width. To take advantage of the problem structure, it is essential to estimate how similar the contracts within a given composite cell C are. Ideally, we would like to know the maximal difference in expected utility:

$$\text{width}(C) = \sup_{x, y \in C} |U(x) - U(y)|.$$

We estimate the width using a proxy, called *virtual width*, which is expressed in terms of the anchors:

$$\text{VirtWidth}(C) = (V(x^+(C)) - P(x^-(C))) - (V(x^-(C)) - P(x^+(C))). \quad (1)$$

The virtual width is a crucial concept due to the following lemma, the proof of which is in Appendix C. As mentioned above, the proof of this lemma is the only place in the paper where we make use of the worker utility model, the FOSD assumption on types, and the consistent tie-breaking assumption. The remainder of the analysis holds for any model of worker behavior as long as Lemma 3.1 remains true.

Lemma 3.1. *Consider the dynamic contract design problem with consistent tie-breaking and all types satisfying the FOSD assumption. For each cell C , $\text{width}(C) \leq \text{VirtWidth}(C)$.*

3.2 Description of the algorithm

With these ideas in place, we are now ready to describe our algorithm. The high-level outline of *AgnosticZooming* is very simple. The algorithm maintains a set of *active* cells which cover the increment space at all times. Initially, there is only a single active cell comprising the entire increment space. In each round t , the algorithm chooses one active cell C_t using an upper confidence index and posts contract x_t sampled uniformly at random among the anchors of this cell. After observing the feedback, the algorithm may choose to *zoom in* on C_t , removing C_t from the set of active cells and activating all relevant quadrants thereof, where the *quadrants* of cell C are defined as the 2^m sub-cells of half the size for which one of the corners is the center of C . In the remainder of this section, we specify how the cell C_t is chosen (the *selection rule*), and how the algorithm decides whether to zoom in on C_t (the *zooming rule*).

Let us first introduce some notation. Consider cell C that is active in some round t . Let $U(C)$ be the expected utility from a single round in which C is chosen by the algorithm, i.e., the average expected utility of the anchor(s) of C . Let $n_t(C)$ be the number of times this cell has been chosen before round t . Consider all rounds in which C is chosen by

the algorithm before round t . Let $U_t(C)$ be the average utility over these rounds. For a composite cell C , let $V_t^+(C)$ and $P_t^+(C)$ be the average value and average payment over all rounds when anchor $x^+(C)$ is chosen. Similarly, let $V_t^-(C)$ and $P_t^-(C)$ be the average value and average payment over all rounds when anchor $x^-(C)$ is chosen. Accordingly, we can estimate the virtual width of composite cell C at time t as

$$W_t(C) = (V_t^+(C) - P_t^-(C)) - (V_t^-(C) - P_t^+(C)). \quad (2)$$

To bound the deviations, we define the *confidence radius* as

$$\text{rad}_t(C) = \sqrt{c_{\text{rad}} \log(T)/n_t(C)}, \quad (3)$$

for some absolute constant c_{rad} ; in our analysis, $c_{\text{rad}} \geq 6$ suffices. With high probability all sample averages defined above will stay within $\text{rad}_t(C)$ of the respective expectations. If this high probability event holds, the width estimate $W_t(C)$ will always be within $4 \text{rad}_t(C)$ of $\text{VirtWidth}(C)$.

Now we are ready to complete the algorithm. The selection rule is as follows. In each round t , the algorithm chooses an active cell C with maximal *index* $I_t(\cdot)$, defined as

$$I_t(C) = \begin{cases} U_t(C) + \text{rad}_t(C) & \text{if } C \text{ is an atomic cell,} \\ U_t(C) + W_t(C) + 5 \text{rad}_t(C) & \text{otherwise.} \end{cases} \quad (4)$$

$I_t(C)$ is an upper confidence bound on the expected utility of *any* candidate contract in C . The zooming rule is as follows. We zoom in on a composite cell C_t if $W_{t+1}(C_t) > 5 \text{rad}_{t+1}(C_t)$, i.e., if the uncertainty due to random sampling, expressed by the confidence radius, becomes sufficiently small compared to the uncertainty due to discretization, expressed by the virtual width. We never zoom in on atomic cells. The pseudocode is summarized in Algorithm 1.

Algorithm 1 AgnosticZooming

Parameters: subset X_{cand} of *candidate contracts*.

Data structure: Collection \mathcal{A} of cells. Initially, $\mathcal{A} = \{[0, 1]^m\}$.

For each round $t = 1$ to T

Let $C_t = \text{argmax}_{C \in \mathcal{A}} I_t(C)$, where $I_t(\cdot)$ is defined as in Equation (4).

Let x_t be the contract sampled u.a.r. among the anchors of C_t . \ \ \ Anchors are defined in Section 3.1.

Post contract x_t and observe feedback.

If $|C \cap X_{\text{cand}}| > 1$ **and** $5 \text{rad}_{t+1}(C_t) < W_{t+1}(C_t)$ **then**

$\mathcal{A} \leftarrow \mathcal{A} \cup \{\text{all relevant quadrants of } C_t\} \setminus \{C_t\}$. \ \ \ C is called *relevant* if $|C \cap X_{\text{cand}}| \geq 1$.

4 Regret bounds and discussion

We now formally state our results and discuss their significance. In particular, we present the main regret bound for AgnosticZooming (derived in Appendix D), state several corollaries, and compare our results to prior work.

The main result. We start with the main regret bound. Like the algorithm itself, this regret bound is parameterized by the set X_{cand} of candidate contracts; our goal is to bound the algorithm's regret with respect to candidate contracts.

Let $\text{OPT}(X_{\text{cand}}) = \sup_{x \in X_{\text{cand}}} U(x)$ be the optimal expected utility over candidate contracts. The algorithm's regret with respect to candidate contracts is $R(T|X_{\text{cand}}) = T \text{OPT}(X_{\text{cand}}) - U$, where T is the time horizon and U is the total expected utility of the algorithm.

Define the *badness* $\Delta(x)$ of a contract $x \in X$ as the difference in expected utility between an optimal candidate contract and x : $\Delta(x) = \text{OPT}(X_{\text{cand}}) - U(x)$. Let $X_\epsilon = \{x \in X_{\text{cand}} : \Delta(x) \leq \epsilon\}$.

We will only be interested in cells that can potentially be used by AgnosticZooming. Formally, we recursively define a collection of *feasible* cells as follows: (i) the cell $[0, 1]^m$ is feasible, (ii) for each feasible cell C , all relevant quadrants of C are feasible. Note that the definition of a feasible cell implicitly depends on the set X_{cand} of candidate contracts.

Let \mathcal{F}_ϵ denote the collection of all feasible, composite cells C such that $\text{VirtWidth}(C) \geq \epsilon$. For $Y \subset X_{\text{cand}}$, let $\mathcal{F}_\epsilon(Y)$ be the collection of all cells $C \in \mathcal{F}_\epsilon$ that overlap with Y , and let $N_\epsilon(Y) = |\mathcal{F}_\epsilon(Y)|$; sometimes we will write $N_\epsilon(Y|X_{\text{cand}})$ in place of $N_\epsilon(Y)$ to emphasize the dependence on X_{cand} .

Using the structure defined above, the main theorem is stated as follows. We prove this theorem in Appendix D.

Theorem 4.1. Consider the dynamic contract design problem with all types satisfying the FOSD assumption and a constant number of outcomes. Assume $T \geq \max(2^m + 1, 18)$. Consider `AgnosticZooming`, parameterized by some set X_{cand} of candidate contracts. There is an absolute constant $\beta_0 > 0$ such that for any $\delta > 0$,

$$R(T|X_{\text{cand}}) \leq \delta T + O(\log T) \sum_{\epsilon=2^{-j} \geq \delta: j \in \mathbb{N}} \frac{N_{\epsilon \beta_0}(X_\epsilon|X_{\text{cand}})}{\epsilon}. \quad (5)$$

Equation (5) has a shape similar to several other regret bounds in the literature, as discussed below. To make this more apparent, we observe that regret bounds in “bandits in metric spaces” are often stated in terms of covering numbers. (For a fixed collection \mathcal{F} of subsets of a given ground set X , the *covering number* of a subset $Y \subset X$ relative to \mathcal{F} is the smallest number of subsets in \mathcal{F} that is sufficient to cover Y .) The numbers $N_\epsilon(Y|X_{\text{cand}})$ are, essentially, about covering Y with feasible cells with virtual width close to ϵ . We make this point more precise as follows. Let an ϵ -minimal cell be a cell in \mathcal{F}_ϵ which does not contain any other cell in \mathcal{F}_ϵ . Let $N_\epsilon^{\min}(Y)$ be the covering number of Y relative to the collection of ϵ -minimal cells, i.e., the smallest number of ϵ -minimal cells sufficient to cover Y . Then

$$N_\epsilon(Y) \leq \lceil \log \frac{1}{\psi} \rceil N_\epsilon^{\min}(Y) \text{ for any } Y \subset X_{\text{cand}} \text{ and } \epsilon \geq 0, \quad (6)$$

where ψ is the smallest size of a feasible cell.³ Thus, Equation (5) can be easily restated using the covering numbers $N_\epsilon^{\min}(\cdot)$ instead of $N_\epsilon(\cdot)$.

Corollary: Polynomial regret. Literature on regret-minimization often states “polynomial” regret bounds of the form $R(T) = \tilde{O}(T^\gamma)$, $\gamma < 1$. While covering-number regret bounds are more precise and versatile, the exponent γ in a polynomial regret bound expresses algorithms’ performance in a particularly succinct and lucid way.

For “bandits in metric spaces” the exponent γ is typically determined by an appropriately defined notion of “dimension”, such as the covering dimension,⁴ which succinctly captures the difficulty of the problem instance. Interestingly, the dependence of γ on the dimension d is typically of the same shape: $\gamma = (d + 1)/(d + 2)$, for several different notions of “dimension”. In line with this tradition, we define the *width dimension*:

$$\text{WidthDim}_\alpha = \inf \{d \geq 0 : N_{\epsilon \beta_0}(X_\epsilon|X_{\text{cand}}) \leq \alpha \epsilon^{-d} \text{ for all } \epsilon > 0\}, \alpha > 0. \quad (7)$$

Note that the width dimension depends on X_{cand} and the problem instance, and is parameterized by a constant $\alpha > 0$. By optimizing the choice of δ in Equation (5), we obtain the following corollary.

Corollary 4.2. Consider the the setting of Theorem 4.1. For any $\alpha > 0$, let $d = \text{WidthDim}_\alpha$. Then

$$R(T|X_{\text{cand}}) \leq O(\alpha \log T) T^{(1+d)/(2+d)}. \quad (8)$$

The width dimension is similar to the “zooming dimension” in Kleinberg et al. [26] and “near-optimality dimension” in Bubeck et al. [12] in the work on “bandits in metric spaces.” See Appendix E for further discussion.

Comparison to prior work (non-adaptive discretization). One approach from prior work that is directly applicable to the dynamic contract design problem is *non-adaptive discretization*. This is an algorithm, call it `NonAdaptive`, which runs an off-the-shelf MAB algorithm, treating a set of candidate contracts X_{cand} as arms. For concreteness, and following the prior work [23, 24, 26], we use UCB1 [2] as an off-the-shelf MAB algorithm.

To compare `AgnosticZooming` with `NonAdaptive`, it is useful to derive several “worst-case” corollaries of Theorem 4.1, replacing $N_\epsilon(X_\epsilon)$ with various (loose) upper bounds.⁵

Corollary 4.3. In the setting of Theorem 4.1, the regret of `AgnosticZooming` can be upper-bounded as follows:

- (a) $R(T|X_{\text{cand}}) \leq \delta T + \sum_{\epsilon=2^{-j} \geq \delta: j \in \mathbb{N}} \tilde{O}(|X_\epsilon|/\epsilon)$, for each $\delta \in (0, 1)$.
- (b) $R(T|X_{\text{cand}}) \leq \tilde{O}(\sqrt{T} |X_{\text{cand}}|)$.

Here the $\tilde{O}()$ notation hides the logarithmic dependence on T , δ , and ψ .

³To prove Equation (6), observe that for each cell $C \in \mathcal{F}_\epsilon(Y)$ there exists an ϵ -minimal cell $C' \subset C$, and for each ϵ -minimal cell C' there exist at most $\lceil \log \frac{1}{\psi} \rceil$ cells $C \in \mathcal{F}_\epsilon(Y)$ such that $C' \subset C$.

⁴Given covering numbers $N_\epsilon(\cdot)$, the *covering dimension* of Y is the smallest $d \geq 0$ such that $N_\epsilon(Y) = O(\epsilon^{-d})$ for all $\epsilon > 0$.

⁵We use the facts that $X_\epsilon \subset X_{\text{cand}}$, $N_\epsilon(Y) \leq N_0(Y)$, and $N_0^{\min}(Y) \leq |Y|$ for all subsets $Y \subset X$.

The best known regret bounds for `NonAdaptive` coincide with those in Corollary 4.3 up to poly-logarithmic factors. When parameterized by the same X_{cand} , `AgnosticZooming` essentially matches the best known regret bounds for `NonAdaptive` in the worst case, and performs better when the bound in Theorem 4.1 is significantly better. We next substantiate this comparison with a concrete example; additional examples are deferred to the full version.

Applications and special cases. We illustrate the performance of `AgnosticZooming` via some special cases. The simplest case is when there is just one non-null outcome. Essentially, each worker makes a strategic choice whether to accept or reject a given task, and this choice is fully observable. This special case, sometimes called *dynamic procurement*, has been studied before [6, 7, 24, 40]. By using adaptive discretization, we achieve significant improvement over prior work for this problem and its “dual,” *dynamic pricing*, in which the principal is selling rather than buying [5, 7, 8, 24]. See Appendix F.

Next we consider a somewhat richer setting in which workers’ strategic decisions are *not* observable, a salient feature of dynamic contract design called *moral hazard* in the contract theory literature. We consider the simplest example that features moral hazard and multiple worker types, which we refer to as the **high-low example**. This example, which subsumes dynamic procurement, is defined as follows. There are two non-null outcomes (`low` and `high`), and two non-null effort levels (`low` and `high`). `Low` outcome brings zero value to the requester; `low` effort level inflicts zero cost on a worker. `Low` effort leads to `low` outcome with probability 1. `High` effort results in `high` outcome with probability θ (and `low` outcome with probability $1 - \theta$). The θ is the same for all worker types, but not known to the algorithm. We assume that workers break ties between effort levels in a consistent way: `high` better than `low` better than `null`. (Hence, as `low` effort incurs zero cost, the only possible outcomes are `low` and `high`.)

We show that the probability $\Pr[\text{high}|x]$ of obtaining the `high` outcome with a given contract x is determined by the price increment $p = x(\text{high}) - x(\text{low})$. Therefore we can write $\Pr[\text{high}|x] = F(p)$ for some function F . In particular, it is optimal for the requester to set $x(\text{low}) = 0$. The worker’s type is completely specified by a single number, the cost c_h of the `high` effort level. Thus, given the transition probability θ , the supply distribution (and, indirectly, the function F) is determined by the distribution D_h over the costs c_h in the worker population.

In the present version, we focus on a natural special case for D_h , in which this distribution has uniform density. More precisely, we assume that $F(\cdot)$ is Lipschitz-continuous and $F''(\cdot) < 0$; these properties are satisfied if D_h has uniform density. We show that `AgnosticZooming` out-performs `NonAdaptive` on this example. For a fair comparison, we assume that both algorithms are run with the same X_{cand} . We consider X_{cand} which consists of all bounded, monotone contracts such that all prices are integer multiples of ψ , for some $\psi > 0$.⁶ Our formal results are as follows (the proofs are omitted from this version). We prove that $\text{WidthDim}_\alpha = \frac{1}{2}$, for some absolute constant α . Consequently, `AgnosticZooming` achieves regret $R(T|X_{\text{cand}}) = \tilde{O}(T^{3/5})$ by Corollary 4.2. Moreover, using the worst-case corollary (Corollary 4.3(a)) we derive regret $R(T|X_{\text{cand}}) = \tilde{O}(T^{1/3}\psi^{-2/3})$. On the other hand, then the best available regret bound for `NonAdaptive` is the one derived from Corollary 4.3(a), same as above. We conclude that `AgnosticZooming` achieves a significantly better regret bound (namely, polynomial regret with a smaller exponent) when ψ is sufficiently small, namely $\psi = \tilde{O}(T^{-\gamma})$ for some $\gamma > \frac{2}{5}$.

In the full version of this paper, we plan to prove that the above regret bound for `NonAdaptive` is tight, i.e., that its regret is at least $R(T|X_{\text{cand}}) = \Omega(T^{1/3}\psi^{-2/3})$. Moreover, we plan to generalize this analysis to D_h with piecewise-uniform density. Further, we plan to analyze another important special case in which D_h is a mixture of two types.

5 Conclusions

Motivated by applications to crowdsourcing markets, we consider a multi-round version of the principal-agent model with unobservable strategic decisions. We design an algorithm for this problem and derive regret bounds which compare favorably to prior work. In fact, given the comparison to the work on Lipschitz MAB, we conjecture that our regret bounds cannot be significantly improved in the general case of the problem.

An immediate direction for future theoretical work is a more extensive analysis of the special cases. Such analysis would fall into the scope of the present paper, albeit it may require deeper insights into the structure of the principal-agent problem. A more far-reaching (and likely very difficult) direction is to incorporate budget constraints, extending the results on online task pricing [6, 7, 40] to settings with unobservable strategic decisions.

⁶If one is allowed to use the knowledge that the algorithms are run on the high-low example, then one could one can set $x(\text{low}) = 0$ for all candidate policies. However, we find that this change does not affect the resulting regret bounds.

References

- [1] Rajeev Agrawal. The continuum-armed bandit problem. *SIAM J. Control and Optimization*, 33(6):1926–1951, 1995.
- [2] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002. Preliminary version in *15th ICML*, 1998.
- [3] Peter Auer, Ronald Ortner, and Csaba Szepesvári. Improved Rates for the Stochastic Continuum-Armed Bandit Problem. In *20th Conf. on Learning Theory (COLT)*, pages 454–468, 2007.
- [4] Moshe Babaioff, Michal Feldman, and Noam Nisan. Combinatorial agency. In *EC*, 2006.
- [5] Moshe Babaioff, Shaddin Dughmi, Robert Kleinberg, and Aleksandrs Slivkins. Dynamic pricing with limited supply. In *13th ACM Conf. on Electronic Commerce (EC)*, 2012.
- [6] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Yaron Singer. Learning on a budget: posted price mechanisms for online procurement. In *13th ACM Conf. on Electronic Commerce (EC)*, pages 128–145, 2012.
- [7] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. In *54th IEEE Symp. on Foundations of Computer Science (FOCS)*, 2013.
- [8] Omar Besbes and Assaf Zeevi. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, 57:1407–1420, 2009.
- [9] Omar Besbes and Assaf J. Zeevi. Blind network revenue management. *Operations Research*, 60(6):1537–1550, 2012.
- [10] Avrim Blum, Vijay Kumar, Atri Rudra, and Felix Wu. Online learning in online auctions. In *14th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 202–204, 2003.
- [11] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.
- [12] Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvari. Online Optimization in X-Armed Bandits. *J. of Machine Learning Research (JMLR)*, 12:1587–1627, 2011. Preliminary version in *NIPS 2008*.
- [13] Sébastien Bubeck, Gilles Stoltz, and Jia Yuan Yu. Lipschitz bandits without the lipschitz constant. In *22nd Intl. Conf. on Algorithmic Learning Theory (ALT)*, pages 144–158, 2011.
- [14] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge Univ. Press, 2006.
- [15] Vincent Conitzer and Nikesh Garera. Online learning algorithms for online principal-agent problems (and selling goods online). In *ICML*, 2006.
- [16] Arpita Ghosh and Patrick Hummel. A game-theoretic analysis of rank-order mechanisms for user-generated content. In *EC*, 2011.
- [17] Arpita Ghosh and Patrick Hummel. Learning and incentives in user-generated content: Multi-armed bandits with endogenous arms. In *ICTS*, 2013.
- [18] Arpita Ghosh and Preston McAfee. Incentivizing high-quality user-generated content. In *WWW*, 2011.
- [19] John Gittins, Kevin Glazebrook, and Richard Weber. *Multi-Armed Bandit Allocation Indices*. John Wiley & Sons, 2011.
- [20] Chien-Ju Ho, Yu Zhang, Jennifer Wortman Vaughan, and Mihaela van der Schaar. Towards social norm design for crowdsourcing markets. In *HCOMP*, 2012.
- [21] John J. Horton and Lydia B. Chilton. The labor economics of paid crowdsourcing. In *EC*, 2010.
- [22] Shaili Jain, Yiling Chen, and David Parkes. Designing incentives for online question-and-answer forums. *Games and Economic Behavior*, 2012.
- [23] Robert Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *18th Advances in Neural Information Processing Systems (NIPS)*, 2004.
- [24] Robert Kleinberg and Tom Leighton. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *44th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 594–605, 2003.
- [25] Robert Kleinberg and Aleksandrs Slivkins. Sharp Dichotomies for Regret Minimization in Metric Spaces. In *21st ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2010.

- [26] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-Armed Bandits in Metric Spaces. In *40th ACM Symp. on Theory of Computing (STOC)*, pages 681–690, 2008.
- [27] Levente Kocsis and Csaba Szepesvari. Bandit Based Monte-Carlo Planning. In *17th European Conf. on Machine Learning (ECML)*, pages 282–293, 2006.
- [28] Jean-Jacques Laffont and David Martimort. *The Theory of Incentives: The Principal-Agent Model*. Princeton University Press, 2002.
- [29] Tze Leung Lai and Herbert Robbins. Asymptotically efficient Adaptive Allocation Rules. *Advances in Applied Mathematics*, 6:4–22, 1985.
- [30] Armando Levy and Tomislav Vukina. Optimal linear contracts with heterogeneous agents. In *European Review of Agricultural Economics*, 2002.
- [31] Tyler Lu, Dávid Pál, and Martin Pál. Showing Relevant Ads via Lipschitz Context Multi-Armed Bandits. In *14th Intl. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2010.
- [32] Odalric-Ambrym Maillard and Rémi Munos. Online Learning in Adversarial Lipschitz Environments. In *European Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 305–320, 2010.
- [33] Winter Mason and Duncan Watts. Financial incentives and the “performance of crowds”. In *HCOMP*, 2009.
- [34] Sanjog Misra, Harikesh S. Nair, and Oystein Daljord. Homogenous contracts for heterogeneous agents: Aligning salesforce composition and compensation. Working Paper, 2012.
- [35] Rémi Munos and Pierre-Arnaud Coquelin. Bandit algorithms for tree search. In *23rd Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2007.
- [36] Sandeep Pandey, Deepak Agarwal, Deepayan Chakrabarti, and Vanja Josifovski. Bandits for Taxonomies: A Model-based Approach. In *SIAM Intl. Conf. on Data Mining (SDM)*, 2007.
- [37] Yuliy Sannikov. A continuous-time version of the principal-agent problem. In *The Review of Economics Studies*, 2008.
- [38] Yuliy Sannikov. Contracts: The theory of dynamic principal-agent relationships and the continuous-time approach. In *10th World Congress of the Econometric Society*, 2012.
- [39] Yaron Singer and Manas Mittal. Pricing mechanisms in crowdsourcing markets. In *WWW*, 2013.
- [40] Adish Singla and Andreas Krause. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *22nd Intl. World Wide Web Conf. (WWW)*, pages 1167–1178, 2013.
- [41] Aleksandrs Slivkins. Multi-armed bandits on implicit metric spaces. In *25th Advances in Neural Information Processing Systems (NIPS)*, 2011.
- [42] Aleksandrs Slivkins. Contextual Bandits with Similarity Information. In *24th Conf. on Learning Theory (COLT)*, 2011. To appear in *J. of Machine Learning Research (JMLR)*, 2013.
- [43] William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285294, 1933.
- [44] Noah Williams. On dynamic principal-agent problems in continuous time. Working Paper, 2009.
- [45] Ming Yin, Yiling Chen, and Yu-An Sun. The effects of performance-contingent financial incentives in online labor markets. In *AAAI*, 2013.
- [46] Yu Zhang and Mihaela van der Schaar. Reputation-based incentive protocols in crowdsourcing applications. In *Infocom*, 2012.

Appendix

A Related work

This paper is related to three different areas: contract theory, market design for crowdsourcing, and online decision problems. Below we outline connections to each of these areas.

A.1 Contract theory

Our model can be viewed as an extension of the classic principal-agent model from contract theory [28]. In the most basic version of the classic model, a single principal interacts with a single agent whose type (specified by a cost function and production function, as described in Section 2) is generally assumed to be known. The principal specifies a contract mapping outcomes to payments that the principal commits to make to the agent. The agent then chooses an action (i.e., effort level) that stochastically results in an outcome in order to maximize his expected utility given the contract. The principal observes the outcome, but cannot directly observe the agent's effort level, creating a *moral hazard* problem. The goal of the principal is to design a contract to maximize her own expected utility, which is the difference between the utility she receives from the outcome and the payment she makes. This maximization can be written as a constrained optimization problem, and it can be shown that linear contracts are optimal.

The *adverse selection* variation of the principal-agent problem relaxes the assumption that the agent's type is known. Most existing literature on the principal-agent problem with adverse selection focuses on applying the revelation principle [28]. In this setting, the principal offers a menu of contracts, and the contract chosen by the agent reveals the agent's type. The problem of selecting a menu of contracts that maximizes the principal's expected utility can again be formulated as a constrained optimization.

Our work differs from the classic setting in that we consider a principal interacting with multiple agents, and the principal may adjust her contract over time in an online manner. Several other authors have considered extensions of the classic model to multiple agents. Levy and Vukina [30] show that with multiple agents it is optimal to set individual linear contracts for each agent rather than a single uniform contract for all agents, but offer a variety of descriptive explanations for why it is more common to see uniform contracts in practice. Babaioff et al. [4] consider a setting in which one principal interacts with multiple agents, but observes only a single outcome which is a function of all agents' effort levels. Misra et al. [34] consider a variant in which the algorithm must decide both how to set a uniform contract for many agents and how to select a subset of agents to hire.

Alternative online versions of the problem have been considered in the literature as well. In dynamic principal agent problem [37, 38, 44], a single principal interacts with a single agent repeatedly over a period of time. The agent can choose to exert different effort at different time, and the outcome at time t is a function of all the efforts exerts by the agent before t . The principal cannot observe the agent's efforts but can observe the outcome. The goal of the principal is to design an optimal contract over time to maximize his payoff. Our work is different from this line of work since we consider the setting with multiple agents with different, unknown types. Our algorithm needs to learn the distribution of agent types and design an optimal contract accordingly.

Conitzer and Garera [15] studies the online principal agent problem with a similar setting to ours. However, they focus on empirically comparing different online algorithms, including bandit approaches with uniform discretizations, gradient ascent, and Bayesian update approaches to the problem. Our goal is to provide an algorithm with nice theoretical guarantees.

A.2 Incentives in crowdsourcing systems

Researchers have recently begun to examine the design of incentive mechanisms to encourage high-quality work in crowdsourcing systems. Jain et al. [22] explore ways in which to award virtual points to users in online question-and-answer forums to improve the quality of answers. Ghosh and Hummel [16, 17] and Ghosh and McAfee [18] study how to distribute user generated content (e.g., Youtube videos) to users to encourage the production of high-quality internet content by people who are motivated by attention. Ho et al. [20] and Zhang and van der Schaar [46] consider the design of two-sided reputation systems to encourage good behavior from both workers and requesters in crowdsourcing markets. While we also consider crowdsourcing markets, our work differs in that it focuses on how to design contracts, perhaps the most natural incentive scheme, to incentivize workers to exert effort.

The problem closest to ours which has been studied in the context of crowdsourcing systems is the online task pricing problem in which a requester has an unlimited supply of tasks to be completed and a budget B to spend on them [6, 39]. Workers with private costs arrive online, and the requester sets a single price for each arriving worker. The goal is to learn the optimal single fixed price over time. Our work can be viewed as a generalization of the task pricing problem, which is a special case of our setting with the number of non-null outcomes m fixed at 1.

There has also been empirical work examining how workers’ behavior varies based on the financial incentives offered in crowdsourcing markets. Mason and Watts [33] study how workers react to changes of financial incentives. In their study, increasing financial incentives increases the number of tasks workers complete, but not the quality of their output. Ying et al. [45] provide a potential explanation for this phenomenon using the concept of “anchoring effect” — a worker’s cost for completing a task is influenced by the first price the worker sees for this task. They show that if a sequence of tasks is presented to a worker, worker performance changes as prices change. Horton and Chilton [21] run experiments to estimate workers’ reservation wage for completing tasks. They show that many workers respond rationally to offered contracts. However, some workers in their study do not behave entirely rationally, and instead appear to aim to earn some “target” amount of payment (e.g., a total payment that is evenly divisible by 5).

Overall, previous empirical work demonstrates that workers in crowdsourcing markets do respond to the change of financial incentives, but that their behavior does not always follow the traditional rational-worker model — similar to people in any real-world market. In our work, we start our analysis with the rational-worker assumption ubiquitous in economic theory, but demonstrate that our results can still hold without these assumptions as long as the collective worker behavior satisfies some natural properties (in particular, as long as Lemma 3.1 holds).

A.3 Online decision problems

In online decision problems, an algorithm makes sequential decisions over time. Two directions that are relevant to this paper are multi-armed bandits and dynamic pricing.

Multi-armed bandits (MAB). MAB have been studied since 1933 [43] in Operations Research, Economics, and several branches of Computer Science including machine learning, theoretical computer science, AI, and algorithmic economics. A survey of prior work on MAB is beyond the scope of this paper; the reader is encouraged to refer to Cesa-Bianchi and Lugosi [14] or Bubeck and Cesa-Bianchi [11] for background on prior-independent MAB, and to Gittins et al. [19] for background on Bayesian MAB. Below we briefly discuss the lines of work on MAB that are directly relevant to our paper.

Our setting can be modeled as prior-independent MAB with stochastic rewards: the reward of a given arm i is an i.i.d. sample of some time-invariant distribution, and neither this distribution nor a Bayesian prior on it are known to the algorithm. The basic formulation (with a small number of arms) is well-understood [2, 11, 29]. To handle problems with a large or infinite number of arms, one typically needs side information on similarity between arms. A typical way to model this side information, called *Lipschitz MAB* [26], is that an algorithm is given a distance function on the arms, and the expected rewards are assumed to satisfy Lipschitz-continuity (or a relaxation thereof) with respect to this distance function [1, 3, 12, 23, 25, 26, 31, 32, 42]. Most related to this paper is the idea of adaptive discretization which is often used in this setting [12, 26, 42], and particularly the *zooming algorithm* [26, 42]. In particular, the general template of our algorithm is similar to the one in the zooming algorithm (but our “selection rule” and “zooming rule” are very different, reflecting the lack of a priori known similarity information).

In some settings (including ours), the numerical similarity information required for Lipschitz MAB is not immediately available. For example, in applications to web search and advertising it is natural to assume that an algorithm can only observe a tree-shaped taxonomy on arms [27, 35, 36, 41]. In particular, Slivkins [41] explicitly reconstructs (the relevant parts of) the metric space defined by the taxonomy. In a different direction, Bubeck et al. [13] study a version of Lipschitz MAB where the Lipschitz constant is not known, and essentially recover the performance of NonAdaptive for this setting.

Dynamic pricing. Dynamic pricing (a.k.a. online posted price auctions) refers to settings in which a principal interacts with agents that arrive over time and offers each agent a price for a transaction, such as selling or buying an item. The version in which the principal *sells* items has been studied in Operations Research, typically in a Bayesian setting [8]. The study of prior-independent formulations has been initiated in Blum et al. [10] and Kleinberg and Leighton [24] and continued by several others [5, 7–9]. Further, Badanidiyuru et al. [6], Singla and Krause [40], and Badanidiyuru et al. [7] studied the version in which the principal *buys* items. It is worth noting that all work after the initial papers

[10, 24] has focused on models with constraints on the principal’s supply or budgets, and does not imply any improved results when specialized to unconstrained settings.

We will use the term *dynamic procurement* suggested in Badanidiyuru et al. [7]. In Appendix F, we identify an important family of problem instances for which `AgnosticZooming` out-performs `NonAdaptive`. Further, we extend this result to *dynamic inventory-pricing* – the “dual” version of dynamic procurement in which the principal is selling rather than buying.

B Monotone contracts may not be optimal

In this section we provide an example of a problem instance for which all monotone contracts are suboptimal (at least when restricting attention to only those contracts with non-negative payoffs). In this example, there are three non-null outcomes (i.e., $m = 3$), and two non-null effort levels, “low” effort and “high” effort, which we denote e_ℓ and e_h respectively. There is only a single worker type. Since there is only one type, we drop the subscript when describing the cost function c . We let $c(e_\ell) = 0$, and let $c(e_h)$ be any positive value less than $0.5(v(2) - v(1))$. If a worker chooses low effort, the outcome is equally likely to be 1 or 3. If the worker chooses high effort, it is equally likely to be 2 or 3. It is easy to verify that this type satisfies the FOSD assumption. Finally, for simplicity, we assume that all workers break ties between high effort and any other effort level in favor of high effort, and that all workers break ties between low effort and the null effort level in favor of low effort.

Let’s consider the optimal contract. Since there is just a single worker type and all workers of this type break ties in the same way, we can consider separately the best contract that would make all workers choose the null effort level, the best contract that would make all workers choose low effort, and the best contract that would make all workers choose high effort, and compare the requester’s expected value for each.

Since $c(e_\ell) = 0$ and workers break ties between low effort and null effort in favor of low effort, there is no contract that would cause workers to choose null effort; workers always prefer low effort to null effort.

It is easy to see that the best contract (in terms of requester expected value) that would make workers choose low effort would set $x(1) = x(3) = 0$ and $x(2)$ sufficiently low that workers would not be enticed to choose high effort; setting $x(2) = 0$ is sufficient. In this case, the expected value of the requester would be $0.5(v(1) + v(3))$.

Now let’s consider contracts that cause workers to choose high effort. If a worker chooses high effort, the expected value to the requester is

$$0.5(v(2) - x(2) + v(3) - x(3)). \quad (9)$$

Workers will choose high effort if and only if

$$0.5(x(1) + x(3)) \leq 0.5(x(2) + x(3)) - c(e_h)$$

or

$$0.5x(1) \leq 0.5x(2) - c(e_h). \quad (10)$$

So to find the contract that maximizes the requester’s expected value when workers choose high effort, we want to maximize Equation 9 subject to the constraint in Equation 10. Since $x(3)$ doesn’t appear in Equation 10, we can set it to 0 to maximize Equation 9. Since $x(1)$ does not appear in Equation 9, we can set $x(1) = 0$ to make Equation 10 as easy as possible to satisfy. We can then see that the optimal occurs when $x(2) = 2c(e_h)$.

Plugging this contact x into Equation 9, the expected utility in this case is $0.5(v(2) + v(3)) - c(e_h)$. Since we assumed that $c(e_h) < 0.5(v(2) - v(1))$, this is strictly preferable to the constant 0 contract, and is in fact the unique optimal contract. Since $x(2) > x(3)$, the unique optimal contract is not monotonic.

C Proof of Lemma 3.1

Let us restate Lemma 3.1 for convenience.

Lemma. *Consider the dynamic contract design problem with consistent tie-breaking and all types satisfying the FOSD assumption. For each cell C , $\text{width}(C) \leq \text{VirtWidth}(C)$.*

For two vectors $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^m$, write $\mathbf{x}' \succeq \mathbf{x}$ if \mathbf{x}' pointwise dominates \mathbf{x} , i.e., if $x'_j \geq x_j$ for all j . For two monotone contracts x, x' , write $x' \succeq x$ if $\text{incr}(x') \succeq \text{incr}(x)$.

Claim C.1. Consider a worker whose type satisfies the FOSD assumption and two weakly bounded contracts x, x' such that $x' \succeq x$. Let e (resp., e') be the effort levels exerted by this worker when he is offered contract x (resp., x'). Then e does not have FOSD over e' .

Proof. For the sake of contradiction, assume that e has FOSD over e' . Note that $e \neq e'$.

Let i be the worker's type. Recall that $F_i(\pi|e)$ denotes the probability of generating an outcome $\pi' \geq \pi$ given the effort level e . Define $\mathbf{F} = (F_i(1|e), \dots, F_i(m|e))$, and define \mathbf{F}' similarly for e' .

Let \mathbf{x} and \mathbf{x}' be the increment representations for x and x' . Given contract x , the worker's expected utility for effort level e is $U_i(x|e) = \mathbf{x} \cdot \mathbf{F} - c_i(e)$. Since e is the optimal effort level given this contract, we have $U_i(x|e) \geq U_i(x|e')$, and therefore

$$\mathbf{x} \cdot \mathbf{F} - \mathbf{x} \cdot \mathbf{F}' \geq c_i(e) - c_i(e').$$

Similarly, since e' is the optimal effort level given contract x' , we have

$$\mathbf{x}' \cdot \mathbf{F}' - \mathbf{x}' \cdot \mathbf{F} \geq c_i(e') - c_i(e).$$

Combining the above two inequalities, we obtain

$$(\mathbf{x} - \mathbf{x}') \cdot (\mathbf{F} - \mathbf{F}') \geq 0. \tag{11}$$

Note that if Equation (11) holds with equality then $U_i(x|e) = U_i(x|e')$ and $U_i(x'|e) = U_i(x'|e')$, so the worker breaks the tie between e and e' in a different way for two different contracts. This contradicts the consistent tie-breaking assumption. However, Equation (11) cannot hold with a strict equality, either, because $\mathbf{x}' \succeq \mathbf{x}$ and (since e has FOSD over e') we have $\mathbf{F} \succeq \mathbf{F}'$ and $\mathbf{F}_\pi > \mathbf{F}'_\pi$ for some outcome $\pi > 0$. Therefore we obtain a contradiction, completing the proof. \square

The proof of Claim C.1 is the only place in the paper where we directly use the consistent tie-breaking assumption. (But the rest of the paper relies on this claim.)

Claim C.2. Assume all types satisfy the FOSD assumption. Consider weakly bounded contracts x, x' such that $x' \succeq x$. Then $V(x') \geq V(x)$ and $P(x') \geq P(x)$.

Proof. Consider some worker, let i be his type. Let e and e' be the chosen effort levels for contracts \mathbf{x} and \mathbf{x}' , respectively. By the FOSD assumption, either $e = e'$, or e' has FOSD over e , or e has FOSD over e' . Claim C.1 rules out the latter possibility.

Define vectors \mathbf{F} and \mathbf{F}' as in the proof of Claim C.1. Note that $\mathbf{F}' \succeq \mathbf{F}$.

Then $P = \mathbf{x} \cdot \mathbf{F}$ and $P' = \mathbf{x}' \cdot \mathbf{F}'$ is the expected payment for contracts x and x' , respectively. Further, letting \mathbf{v} denote the increment representation of the requester's value for each outcome, $V = \mathbf{v} \cdot \mathbf{F}$ and $V' = \mathbf{v} \cdot \mathbf{F}'$ is the expected requester's value for contracts x and x' , respectively. Since $\mathbf{x}' \succeq \mathbf{x}$ and $\mathbf{F}' \succeq \mathbf{F}$, it follows that $P' \geq P$ and $V' \geq V$. Since this holds for each worker, this also holds in expectation over workers. \square

To finish the proof of Lemma 3.1, fix a contract $x \in C$ and observe that $V(x^+) \geq V(x) \geq V(x^-)$ and $P(x^+) \geq P(x) \geq P(x^-)$, where $x^+ = x^+(C)$ and $x^- = x^-(C)$ are the two anchors.

D Proof of the main regret bound (Theorem 4.1)

Our high-level approach is to define a *clean execution* of an algorithm as an execution in which some high-probability events are satisfied, and derive bounds on regret conditional on the clean execution. The analysis of a clean execution does not involve any "probabilistic" arguments. This approach tends to simplify regret analysis.

We start by listing some simple invariants enforced by `AgnosticZooming`:

Invariant D.1. In each round t of each execution of `AgnosticZooming`, the following holds:

- (a) All active cells are relevant,
- (b) Each candidate contract is contained in some active cell,
- (c) $W_t(C) \leq 5 \text{rad}_t(C)$ for each active composite cell C .

Note that the zooming rule is essential to ensure Invariant D.1(c).

D.1 Analysis of the randomness

Definition D.2 (Clean Execution). An execution of `AgnosticZooming` is called *clean* if for each round t and each active cell C it holds that

$$|U(C) - U_t(C)| \leq \text{rad}_t(C), \quad (12)$$

$$|\text{VirtWidth}(C) - W_t(C)| \leq 4 \text{rad}_t(C) \quad (\text{if } C \text{ is composite}). \quad (13)$$

Lemma D.3. Assume $c_{\text{rad}} \geq 16$ and $T \geq \max(1 + 2^m, 18)$. Then:

- (a) $\Pr [\text{Equation (12) holds } \forall \text{ rounds } t, \text{ active cells } C] \geq 1 - 2T^{-2}$.
- (b) $\Pr [\text{Equation (13) holds } \forall \text{ rounds } t, \text{ active composite cells } C] \geq 1 - 16T^{-2}$.

Consequently, an execution of `AgnosticZooming` is clean with probability at least $1 - \frac{1}{T}$.

Essentially, Lemma D.3 follows from the standard concentration inequality known as ‘‘Chernoff Bounds’’. However, one needs to be careful about conditioning and other details.

Proof of Lemma D.3(a). Consider an execution of `AgnosticZooming`. Let N be the total number of activated cells. Since at most 2^m cells can be activated in any one round, $N \leq 1 + 2^m T \leq T^2$. Let C_j be the $\min(j, N)$ -th cell activated by the algorithm. (If multiple ‘‘quadrants’’ are activated in the same round, order them according to some fixed ordering on the quadrants.)

Fix some feasible cell C and $j \leq T^2$. We claim that

$$\Pr [|U(C) - U_t(C)| \leq \text{rad}_t(C) \text{ for all rounds } t \mid C_j = C] \geq 1 - 2T^{-4}. \quad (14)$$

Let $n(C) = n_{1+T}(C)$ be the total number of times cell C is chosen by the algorithm. For each $s \in \mathbb{N}$: $1 \leq s \leq n(C)$ let U_s be the requester’s utility in the round when C is chosen for the s -th time. Further, let \mathcal{D}_C be the distribution of U_1 , conditional on the event $n(C) \geq 1$. (That is, the per-round reward from choosing cell C .) Let U'_1, \dots, U'_T be a family of mutually independent random variables, each with distribution \mathcal{D}_C . Then for each $n \leq T$, conditional on the event $\{C_j = C\} \wedge \{n(C) = n\}$, the tuple (U_1, \dots, U_n) has the same joint distribution as the tuple (U'_1, \dots, U'_n) . Consequently, applying Chernoff Bounds to the latter tuple, it follows that

$$\Pr \left[\left| U(C) - \frac{1}{n} \sum_{s=1}^n U_s \right| \leq \sqrt{\frac{1}{n} c_{\text{rad}} \log(T)} \mid \{C_j = C\} \wedge \{n(C) = n\} \right] \geq 1 - 2T^{-2c_{\text{rad}}} \geq 1 - 2T^{-5}.$$

Taking the Union Bound over all $n \leq T$, and plugging in $\text{rad}_t(C_j)$, $n_t(C_j)$, and $U_t(C_j)$, we obtain Equation (14).

Now, let us keep j fixed in Equation (14), and integrate over C . More precisely, let us multiply both sides of Equation (14) by $\Pr[C_j = C]$ and sum over all feasible cells C . We obtain, for all $j \leq T^2$:

$$\Pr [|U(C_j) - U_t(C_j)| \leq \text{rad}_t(C_j) \text{ for all rounds } t] \geq 1 - 2T^{-4}. \quad (15)$$

(Note that to obtain Equation (15), we do *not* need to take the Union Bound over all feasible cells C .) We complete the proof by taking the Union Bound over all $j \leq 1 + T^2$. \square

Proof Sketch of Lemma D.3(b). We show that

$$\Pr [|V^+(C) - V_t^+(C)| \leq \text{rad}_t(C) \quad \forall \text{ rounds } t, \text{ active composite cells } C] \geq 1 - 4T^{-2}, \quad (16)$$

and similarly for $V^-(C)$, $P^+(C)$ and $P^-(C)$. Each of these four statements is proved similarly, using the technique from Lemma D.3(a). In what follows, we sketch the proof for one of the four cases, namely for Equation (16).

For a given composite cell C , we are only interested in rounds in which anchor $x^+(C)$ is selected by the algorithm. Letting $n_t^+(C)$ be the number of times this anchor is chosen up to time t , let us define the corresponding notion of ‘‘confidence radius’’:

$$\text{rad}_t^+(C) = \frac{1}{2} \sqrt{\frac{c_{\text{rad}} \log T}{n_t^+(C)}}.$$

With the technique from the proof of Lemma D.3(a), we can establish the following high-probability event:

$$|V^+(C) - V_t^+(C)| \leq \text{rad}_t^+(C). \quad (17)$$

More precisely, we can prove that

$$\Pr [\text{Equation (17) holds } \forall \text{ rounds } t, \text{ active composite cells } C] \geq 1 - 2T^{-2}.$$

Further, we need to prove that w.h.p. the anchor $x^+(C)$ is played sufficiently often. Noting that $\mathbb{E}[n_t^+(C)] = \frac{1}{2} n_t(C)$, we establish an auxiliary high-probability event:⁷

$$n_t^+(C) \geq \frac{1}{2} n_t(C) - \frac{1}{4} \text{rad}_t(C). \quad (18)$$

More precisely, we can use Chernoff Bounds to show that, if $c_{\text{rad}} \geq 16$,

$$\Pr [\text{Equation (18) holds } \forall \text{ rounds } t, \text{ active composite cells } C] \geq 1 - 2T^{-2}. \quad (19)$$

Now, letting $n_0 = (c_{\text{rad}} \log T)^{1/3}$, observe that

$$\begin{aligned} n_t(C) \geq n_0 &\Rightarrow n_t^+(C) \geq \frac{1}{4} n_t(C) &\Rightarrow \text{rad}_t^+(C) \leq \text{rad}_t(C), \\ n_t(C) < n_0 &\Rightarrow \text{rad}_t(C) \geq 1 &\Rightarrow |V^+(C) - V_t^+(C)| \leq \text{rad}_t(C). \end{aligned}$$

Therefore, once Equations (17) and (18) hold, we have $|V^+(C) - V_t^+(C)| \leq \text{rad}_t(C)$. This completes the proof of Equation (16). \square

D.2 Analysis of a clean execution

The rest of the analysis focuses on a clean execution. Recall that C_t is the cell chosen by the algorithm in round t .

Claim D.4. *In any clean execution, $I(C_t) \geq \text{OPT}(X_{\text{cand}})$ for each round t .*

Proof. Fix round t , and let x^* be any candidate contract. By Invariant D.1(b), there exists an active cell, call it C_t^* , which contains x^* .

We claim that $I_t(C_t^*) \geq U(x^*)$. We consider two cases, depending on whether C_t^* is atomic. If C_t^* is atomic then the anchor is unique, so $U(C_t^*) = U(x^*)$, and $I_t(C_t^*) \geq U(x^*)$ by the clean execution. If C_t^* is composite then

$$\begin{aligned} I_t(C_t^*) &\geq U(C_t^*) + \text{VirtWidth}(C_t^*) && \text{by clean execution} \\ &\geq U(C_t^*) + \text{width}(C_t^*) && \text{by Lemma 3.1} \\ &\geq U(x^*) && \text{by definition of width, since } x^* \in C_t^*. \end{aligned}$$

We have proved that $I_t(C_t^*) \geq U(x^*)$. Now, by the selection rule we have $I_t(C_t) \geq I_t(C_t^*) \geq U(x^*)$. Since this holds for any candidate contract x^* , the claim follows. \square

Claim D.5. *In any clean execution, for each round t , the index $I_t(C_t)$ is upper-bounded as follows:*

- (a) *if C_t is atomic then $I(C_t) \leq U(C_t) + 2 \text{rad}_t(C_t)$.*
- (b) *if C_t is composite then $I(C_t) \leq U(x) + O(\text{rad}_t(C_t))$ for each contract $x \in C_t$.*

Proof. Fix round t . Part (a) follows because $I_t(C_t) = U_t(C_t) + \text{rad}_t(C_t)$ by definition of the index, and $U_t(C_t) \leq U(C_t) + \text{rad}_t(C_t)$ by clean execution.

For part (b), fix a contract $x \in C_t$. Then:

$$\begin{aligned} U_t(C_t) &\leq U(C_t) + \text{rad}_t(C_t) && \text{by clean execution} \\ &\leq U(x) + \text{width}(C_t) + \text{rad}_t(C_t) && \text{by definition of width} \\ &\leq U(x) + \text{VirtWidth}(C_t) + \text{rad}_t(C_t) && \text{by Lemma 3.1} \\ &\leq U(x) + W_t(C_t) + 5 \text{rad}_t(C_t) && \text{by clean execution.} \\ I_t(C_t) &= U_t(C_t) + W_t(C_t) + 5 \text{rad}_t(C_t) && \text{by definition of index} \\ &\leq U(x) + 2 W_t(C_t) + 10 \text{rad}_t(C_t) && \text{by Equation (20)} \\ &\leq U(x) + 20 \text{rad}_t(C_t) && \text{by Invariant D.1(c).} \end{aligned} \quad (20)$$

\square

⁷The constant $\frac{1}{4}$ in Equation (18) is there to enable a consistent choice of n_0 in the remainder of the proof.

For each relevant cell C , define badness $\Delta(C)$ as follows. If C is composite, $\Delta(C) = \sup_{x \in C} \Delta(x)$ is the maximal badness among all contracts in C . If C is atomic and $x \in C$ is the unique candidate contract in C , then $\Delta(C) = \Delta(x)$.

Claim D.6. *In any clean execution, $\Delta(C) \leq O(\text{rad}_t(C))$ for each round t and each active cell C .*

Proof. By Claims D.4 and D.5, $\Delta(C_t) \leq O(\text{rad}_t(C_t))$ for each round t . Fix round t and let C be an active cell in this round. If C has never be selected before round t , the claim is trivially true. Else, let s be the most recent round before t when C is selected by the algorithm. Then $\Delta(C) \leq O(\text{rad}_s(C))$. The claim follows since $\text{rad}_s(C) = \text{rad}_t(C)$. \square

Claim D.7. *In any clean execution, each cell C is selected at most $O(\log T / (\Delta(C))^2)$ times.*

Proof. By Claim D.6, $\Delta(C) \leq O(\text{rad}_T(C))$. The claim follows from the definition of rad_T in Equation (3). \square

Let $n(x)$ and $n(C)$ be the number of times contract x and cell C , respectively, are chosen by the algorithm. Then regret of the algorithm is

$$R(T | X_{\text{cand}}) = \sum_{x \in X} n(x) \Delta(x) \leq \sum_{\text{cells } C} n(C) \Delta(C). \quad (21)$$

The next result (Lemma D.8) upper-bounds the right-hand side of Equation (21) for a clean execution. By Lemma D.3, this suffices to complete the proof of Theorem 4.1

Lemma D.8. *Consider a clean execution of AgnosticZooming. For any $\delta \in (0, 1)$,*

$$\sum_{\text{cells } C} n(C) \Delta(C) \leq \delta T + O(\log T) \sum_{\epsilon=2^{-j} \geq \delta: j \in \mathbb{N}} \frac{|\mathcal{F}_\epsilon(X_{2\epsilon})|}{\epsilon}.$$

The proof of Lemma D.8 relies on some simple properties of $\Delta(\cdot)$ which we state below.

Claim D.9. *Consider two relevant cells $C \subset C_p$. Then:*

- (a) $\Delta(C) \leq \Delta(C_p)$.
- (b) *If $\Delta(C) \leq \epsilon$ for some $\epsilon > 0$, then C overlaps with X_ϵ .*

Proof. To prove part (a), one needs to consider two cases, depending on whether cell C_p is composite. If it is, the claim follows trivially. If C_p is atomic, then C is atomic, too, and so $\Delta(C) = \Delta(C_p) = \Delta(x)$, where x is the unique candidate contract in C_p .

For part (b), there exists a candidate contract $x \in C$. It is easy to see that $\Delta(x) \leq \Delta(C)$ (again, consider two cases, depending on whether C is composite.) Therefore, $x \in X_\epsilon$. \square

Proof of Lemma D.8. Let Σ denote the sum in question. Let \mathcal{A}^* be the collection of all cells ever activated by the algorithm. Among such cells, consider those with badness on the order of ϵ :

$$\mathcal{G}_\epsilon := \{ C \in \mathcal{A}^* : \Delta(C) \in [\epsilon, 2\epsilon] \}.$$

By Claim D.7, the algorithm chooses each cell $C \in \mathcal{G}_\epsilon$ at most $O(\log T / \epsilon^2)$ times, so $n(C) \Delta(C) \leq O(\log T / \epsilon)$.

Fix some $\delta \in (0, 1)$ and observe that all cells C with $\Delta(C) \leq \delta$ contribute at most δT to Σ . Therefore it suffices to focus on \mathcal{G}_ϵ , $\epsilon \geq \delta/2$. It follows that

$$\Sigma \leq \delta T + O(\log T) \sum_{\epsilon=2^{-i} \geq \delta/2} \frac{|\mathcal{G}_\epsilon|}{\epsilon}. \quad (22)$$

We bound $|\mathcal{G}_\epsilon|$ as follows. Consider a cell $C \in \mathcal{G}_\epsilon$. The cell is called a *leaf* if it is never zoomed in on (i.e., removed from the active set) by the algorithm. If C is activated in the round when cell C_p is zoomed in on, C_p is called the *parent* of C . We consider two cases, depending on whether or not C is a leaf.

- (i) Assume cell C is not a leaf. Since $\Delta(C) < 2\epsilon$, C overlaps with $X_{2\epsilon}$ by Claim D.9(b). Note that C is zoomed in on in some round, say in round $t - 1$. Then

$$\begin{aligned} 5 \operatorname{rad}_t(C) &\leq W_t(C) && \text{by the zooming rule} \\ &\leq \operatorname{VirtWidth}(C) + 4 \operatorname{rad}_t(C) && \text{by clean execution,} \end{aligned}$$

so $\operatorname{rad}_t(C) \leq \operatorname{VirtWidth}(C)$. Therefore, using Claim D.6, we have

$$\epsilon \leq \Delta(C) \leq O(\operatorname{rad}_t(C)) \leq O(\operatorname{VirtWidth}(C)).$$

It follows that $C \in \mathcal{F}_{\Omega(\epsilon)}(X_{2\epsilon})$.

- (ii) Assume cell C is a leaf. Let C_p be the parent of C . Since $C \subset C_p$, we have $\Delta(C) \leq \Delta(C_p)$ by Claim D.9(a). Therefore, invoking case (i), we have

$$\epsilon \leq \Delta(C) \leq \Delta(C_p) \leq O(\operatorname{VirtWidth}(C_p)).$$

Since $\Delta(C) < 2\epsilon$, C overlaps with $X_{2\epsilon}$ by Claim D.9(b), and therefore so does C_p . It follows that $C_p \in \mathcal{F}_{\Omega(\epsilon)}(X_{2\epsilon})$.

Combing these two cases, it follows that $|\mathcal{G}_\epsilon| \leq (2^m + 1) |\mathcal{F}_{\Omega(\epsilon)}(X_{2\epsilon})|$. Plugging this into (22) and making an appropriate substitution $\epsilon \rightarrow \Theta(\epsilon)$ to simplify the resulting expression, we obtain the regret bound in Theorem 4.1 \square

E Comparison to prior work on “bandits in metric spaces”

Consider a variant of dynamic contract design in which an algorithm is given a priori information on similarity between contracts: a function $\mathcal{D} : X_{\text{cand}} \times X_{\text{cand}} \rightarrow [0, 1]$ such that $|U(x) - U(y)| \leq \mathcal{D}(x, y)$ for any two candidate contracts x, y . If an algorithm is given this function \mathcal{D} (call such algorithm \mathcal{D} -aware), the machinery from “bandits in metric spaces” [12, 26] can be used to perform adaptive discretization and obtain a significant advantage over NonAdaptive. We argue that we obtain similar results with AgnosticZooming without knowing the \mathcal{D} .

In practice, the similarity information \mathcal{D} would be coarse, probably aggregated according to some pre-defined hierarchy. To formalize this idea, the hierarchy can be represented as a collection \mathcal{F} of subsets of X_{cand} , so that $\mathcal{D}(x, y)$ is a function of the smallest subset in \mathcal{F} containing both x and y . The hierarchy \mathcal{F} should be natural given the structure of the contract space. One such natural hierarchy is the collection of all feasible cells, which corresponds to splitting the cells in half in each dimension. Formally, $\mathcal{D}(x, y) = f(C_{x,y})$ for some f with $f(C_{x,y}) \geq \operatorname{width}(C_{x,y})$, where $C_{x,y}$ is the smallest feasible cell containing both x and y .

Given this shape of \mathcal{D} , let us state the regret bounds for \mathcal{D} -aware algorithms in Kleinberg et al. [26] and Bubeck et al. [12]. To simplify the notation, we assume that the action space is restricted to X_{cand} . The regret bounds have a similar “shape” as that in Theorem 4.1:

$$R(T|X_{\text{cand}}) \leq \delta T + O(\log T) \sum_{\epsilon=2^{-j} \geq \delta: j \in \mathbb{N}} \frac{N_{\Omega(\epsilon)}^*(X_\epsilon)}{\epsilon}, \quad (23)$$

where the numbers $N_\epsilon^*(\cdot)$ have a similar high-level meaning as $N_\epsilon(\cdot)$, and nearly coincide with $N_\epsilon^{\min}(\cdot)$ when $\mathcal{D}(x, y) = \operatorname{VirtWidth}(C_{x,y})$. One can use Equation (23) to derive a polynomial regret bound like Equation (8).

For a more precise comparison, we focus on the results in Kleinberg et al. [26]. (The regret bounds in Bubeck et al. [12] are very similar in spirit, but are stated in terms of a slightly different structure.) The “covering-type” regret bound in Kleinberg et al. [26] focuses on balls of radius at most ϵ according to distance \mathcal{D} , so that $N_\epsilon^*(Y)$ is the smallest number of such balls that is sufficient to cover Y . In the special case $\mathcal{D}(x, y) = \operatorname{VirtWidth}(C_{x,y})$ balls of radius $\leq \epsilon$ are precisely feasible cells of virtual width $\leq \epsilon$. This is very similar (albeit not technically the same) as the ϵ -minimal cells in the definition of $N_\epsilon^{\min}(\cdot)$.

Further, the covering numbers $N_\epsilon^*(Y)$ determine the “zooming dimension”:

$$\operatorname{ZoomDim}_\alpha = \inf \left\{ d \geq 0 : N_{\epsilon/8}^*(X_\epsilon) \leq \alpha \epsilon^{-d} \text{ for all } \epsilon > 0 \right\}, \quad \alpha > 0. \quad (24)$$

This definition coincides with the covering dimension in the worst case, and can be much smaller for “nice” problem instances in which X_ϵ is a significantly small subset of X_{cand} . With this definition, one obtains a polynomial regret bound which is version of Equation (8) with $d = \text{ZoomDim}_\alpha$.

We conclude that `AgnosticZooming` essentially matches the regret bounds for \mathcal{D} -aware algorithms, despite the fact that \mathcal{D} -aware algorithms have access to much more information.

F Application to dynamic procurement / pricing

We discuss dynamic procurement, the special case of dynamic contract design in which there is exactly one non-null outcome, which has received some attention in prior work [6, 7, 24, 40]. We identify an important family of problem instances for which `AgnosticZooming` out-performs `NonAdaptive`. Further, we extend this result to *dynamic inventory-pricing*, the “dual” version of dynamic procurement in which the principal is selling rather than buying [5, 7, 8, 24].⁸

Dynamic procurement: some background. The dynamic procurement problem, in its most basic version, is defined as follows. There is one *principal* (buyer) who sequentially interacts with multiple *agents* (sellers). In each round t , an agent arrives, with one item for sale. The principal offers price p_t for this item, and the agent agrees to sell if and only if $p_t \geq c_t$, where $c_t \in [0, 1]$ is the agent’s private *cost* for this item. The principal derives value v for each item bought; his utility is the value from bought items minus the payment. The time horizon T (the number of rounds) is known. Each private cost c_t is an independent sample from some fixed distribution, called the *supply distribution*. We are interested in the *prior-independent* version, where the supply distribution is not known to the principal. The algorithm’s goal is to choose the offered prices p_t so as to maximize the expected utility of the principal.

Dynamic procurement is precisely the special case of dynamic contract design in which there is exactly one non-null outcome (which corresponds to a sale). Indeed, in this special case there is exactly one non-null effort level e without loss of generality (because any non-null effort levels deterministically lead to the non-null outcome). Hence, the worker’s type i is summarized by a single number: $c_i = c_i(e)$. Each contract is also summarized by a single number: the offered price p for the non-null outcome. Let $U(p)$ be the corresponding expected utility of the algorithm.

One crucial simplification compared to the general case is that the discretization error can now be easily bounded from above: $\text{OPT} - \text{OPT}(X_{\text{cand}}) \leq \psi$, where any two candidate prices in X_{cand} differ by at most ψ . Thus, one can now derive regret relative to OPT rather than to $\text{OPT}(X_{\text{cand}})$ (i.e., the best candidate contract). Also, note that all contracts are trivially monotone and any optimal contract is bounded without loss of generality. It follows that $\text{OPT} = \sup_{p \geq 0} U(p)$, the optimal expected utility over all possible prices.

Worst-case regret bounds are implicit in prior work on dynamic inventory-pricing [24].⁹ Consider the set of all prices in $[0, 1]$ that are integer multiples of some $\psi > 0$. Call this set the *additive ψ -mesh* and denote it $X_{\text{mesh}}(\psi)$. Let `NonAdaptive`(ψ) denote algorithm `NonAdaptive` with $X_{\text{cand}} = X_{\text{mesh}}(\psi)$. Then, by the analysis in Kleinberg and Leighton [24], `NonAdaptive`(ψ) achieves regret $R(T) = \tilde{O}(\psi T + \psi^{-2})$. This is optimized to $R(T) = \tilde{O}(T^{2/3})$ if and only if $\psi = \tilde{O}(T^{1/3})$. Moreover, there is a matching lower bound: $R(T) = \Omega(T^{2/3})$ for any algorithm.

Our results: upper bounds. We focus on problem instances with piecewise-uniform costs and bounded density. Formally, we say that an instance of dynamic procurement has *k-piecewise-uniform costs* if the interval $[0, 1]$ is partitioned into $k \in \mathbb{N}$ sub-intervals such that the supply distribution is uniform on each sub-interval. A problem instance has *λ -bounded density*, $\lambda \geq 1$ if the supply distribution has a probability density function almost everywhere, and the density is between $\frac{1}{\lambda}$ and λ .

We use `AgnosticZooming` with $X_{\text{cand}} = X$, i.e. each contract is a candidate contract. Using the full power of Theorem 4.1, we obtain the following regret bound.

⁸The “selling” version is usually called “dynamic pricing” in the literature; we term it “dynamic inventory-pricing” to emphasize the distinction between buying and selling.

⁹The algorithmic result for dynamic procurement is an easy modification of the analysis in Kleinberg and Leighton [24] for dynamic inventory-pricing. The lower bound in Kleinberg and Leighton [24] can also be “translated” from dynamic inventory-pricing to dynamic procurement without introducing any new ideas. We omit the details from this version.

Theorem F.1. Consider the dynamic procurement problem with k -piecewise-uniform costs and λ -bounded density, for some $k \in \mathbb{N}$ and $\lambda > 1$. `AgnosticZooming` with $X_{\text{cand}} = X$ achieves regret

$$R(T) = \tilde{O}(\min(\lambda^{1/3}T^{2/3}, \lambda^{3/5}k^{2/5}T^{3/5})).$$

In particular, if k and λ are absolute constants, it achieves regret $R(T) = \tilde{O}(T^{3/5})$.

Proof Sketch. First, Let $S(p)$ be the probability of sale at price p , i.e., the probability that a random worker's cost is at most p . Since the supply distribution has density at most λ , it follows that $S(\cdot)$ is a Lipschitz-continuous function with Lipschitz constant λ . It follows that each cell of virtual width at least ϵ has diameter at least $\Omega(\epsilon/\lambda)$, for any $\epsilon > 0$. (Note that each "cell" is now simply a sub-interval $[p, q] \subset [0, 1]$, so its diameter is simply $q - p$.)

Second, we claim that X_ϵ is contained in a union of k intervals of diameter $O(\sqrt{\epsilon\lambda})$. To see this, consider the partition of $[0, 1]$ into k subintervals such that the supply distribution has a uniform density on each subinterval. Let $[p_j, q_j]$ be the j -th subinterval. Let p_j^* be the local optimum of $U(\cdot)$ on this subinterval, and let $X_{j,\epsilon} = \{x \in [p_j, q_j] : U(p_j^*) - U(x) \leq \epsilon\}$. Then $X_\epsilon \subset \cup_j X_{j,\epsilon}$. We can show that $X_{j,\epsilon} \subset [p_j^* - \delta, p_j^* + \delta]$ for some $\delta = O(\sqrt{\epsilon\lambda})$.

Recall that $N_{\epsilon\beta_0}(X_\epsilon)$ is the number of feasible cells of virtual width at least $\epsilon\beta_0$ which overlap with X_ϵ . It follows that $N_{\epsilon\beta_0}(X_\epsilon)$ is at most k times the maximal number of feasible cells of diameter at least $\Omega(\epsilon/\lambda)$ that overlap with an interval of diameter $O(\sqrt{\epsilon\lambda})$. Therefore: $N_{\epsilon\beta_0}(X_\epsilon) = O(k\lambda^{3/2}\epsilon^{-1/2} \log \frac{1}{\epsilon})$. Moreover, we have a less sophisticated upper bound on $N_{\epsilon\beta_0}(X_\epsilon)$: it is at most the number of feasible cell of diameter at least $\Omega(\epsilon/\lambda)$. So $N_{\epsilon\beta_0}(X_\epsilon) = O(\lambda/\epsilon)(\log \frac{1}{\epsilon})$. The theorem follows by plugging both upper bounds on $N_{\epsilon\beta_0}(X_\epsilon)$ into Equation (5). \square

Comparison with NonAdaptive. We argue that `AgnosticZooming` outperforms `NonAdaptive`(ψ) for problem instances with piecewise-uniform costs and bounded density. As we need to take into account all possible values of ψ , we prove a somewhat nuanced statement, which we state below.

Let \mathcal{F}_k be the family of all problem instances with k -piecewise-uniform costs and λ -bounded density, for some sufficiently large absolute constant λ ; say $\lambda = 4$ for concreteness. Let $\mathcal{F}_\infty = \cup_{k \in \mathbb{N}} \mathcal{F}_k$.

We argue that `AgnosticZooming` outperforms `NonAdaptive`(ψ) on \mathcal{F}_∞ . Informally, we prove that for any $\psi > 0$:

- regret of `AgnosticZooming` over \mathcal{F}_∞ is at most that of `NonAdaptive`(ψ), up to polylog factors.
- `AgnosticZooming` has a significantly smaller regret than `NonAdaptive`(ψ) either over \mathcal{F}_∞ or over \mathcal{F}_2 .

Formally, we prove the following two-pronged lower bound:

Lemma F.2. Let $R_\psi(T)$ be the regret of `NonAdaptive`(ψ) over \mathcal{F}_∞ . Then:

- $R_\psi(T) = \Omega(\psi T + \psi^{-2}) \geq \Omega(T^{2/3})$.
- If $R_\psi(T) \leq \tilde{O}(T^{2/3})$ then `NonAdaptive`(ψ) achieves regret $R(T) = \Omega(T^{2/3})$ on \mathcal{F}_2 .

(By part (a), $R_\psi(T) \leq \tilde{O}(T^{2/3})$ if and only if $\psi = \tilde{O}(T^{-1/3})$.)

Proof Sketch. Let $S(p)$ be the probability of sale at price p . Note that $S(p)$ is increasing in p . For piecewise-uniform costs, we have $S(0) = 0$ and $S(p) = 1$. Assume that the principal derives value $v = 1$ from each item. Then the expected utility from price p is $U(p) = S(p)(1 - p)$.

For part (a), fix $\psi > 0$ and consider `NonAdaptive` with $X_{\text{cand}} = X_{\text{mesh}}(\psi)$. Use the following problem instance. Let $\mathcal{P}_\delta = [\frac{2}{5}, \frac{3}{5}] \cap \{4j\psi + \delta : j \in \mathbb{N}\}$. Set $U(p) = \frac{1}{4}$ for each $p \in \mathcal{P}_0$. Further, pick some $p^* \in \mathcal{P}_{\psi/2}$ and set $U(p^*) = \frac{1}{4} + \Omega(\psi)$. This defines $S(p)$ for $p \in \mathcal{P} \cup \{0, 1, p^*\}$. For the rest of the prices, define $S(\cdot)$ via linear interpolation. This completes the description of the problem instance.

We show that X_ψ consists of $\Omega(\frac{1}{\psi})$ candidate contracts. Since each such contract contributes at least $\Omega(\frac{1}{\psi})$ to regret, we have $R(T) \geq \Omega(\psi^{-2})$. Further, we show that the discretization error is at least $\Omega(\psi)$, so $R(T) \geq \Omega(\psi T)$.

For part (b), fix $\psi = \tilde{O}(T^{-1/3})$. Recall that for $k = 2$ the supply distribution has density λ_1 on interval $[0, p_0]$, and density λ_2 on interval $[p_0, 1]$, for some numbers $\lambda_1, \lambda_2, p_0$. We pick p_0 so that it is sufficiently far from any point in $X_{\text{mesh}}(\psi)$. Note that the function $U(\cdot)$ is a parabola on each of the two intervals. We adjust the densities so that $U(\cdot)$

achieves its maximum at p_0 , and the maximum of either of the two parabolas is sufficiently far from p_0 . Then the discretization error of $X_{\text{mesh}}(\psi)$ is at least $\Omega(\psi)$, which implies regret $\Omega(\psi T)$. \square

Dynamic inventory-pricing. We extend the above results to dynamic inventory-pricing, the “dual” version of dynamic procurement in which the principal is selling rather than buying. The basic version of dynamic inventory-pricing is as follows. In each round t , an agent (buyer) arrives. The principal offers one item for sale at price p_t . The agent buys if and only if $p_t \leq v_t$, where $v_t \in [0, 1]$ is the agent’s value for an item. Like in dynamic procurement, the value v_t is *private* (not known to the principal). Each value v_t is sampled independently from a fixed distribution, called the *demand distribution*. We consider the prior-independent case in which the demand distribution is not known to the principal. The principal’s utility is simply the total payment; the principal has no supply contracts and derives no value from left-over items. The algorithm’s goal is to choose the offered prices p_t so as to optimize the expected utility of the principal. This problem has been addressed in Kleinberg and Leighton [24], achieving precisely the results as described above for dynamic procurement.

Dynamic inventory-pricing can be cast as a special case of the dynamic contract design problem. Namely, it is the case of two outcomes (with the non-null outcome corresponding to a sale), where the principal’s value is 0 and (by abuse of notation) the payments are negative. However, negative payments invalidate Lemma 3.1, and necessitate a different (but very natural) definition of virtual width. Recall that a cell C in this setting is simply an interval $[p^-, p^+] \in [0, 1]$, where (if C is composite) p^- and p^+ are the two anchors. Let $S(p)$ be the probability of the non-null outcome (i.e. a sale) at price p ; note that $S(p)$ is non-increasing in p . We define the two-outcome version of virtual width as

$$\text{VirtWidth}_2(C) = p^+ \cdot S(p^-) - p^- \cdot S(p^+). \quad (25)$$

It is easy to see that $\text{width}(C) \leq \text{VirtWidth}_2(C)$. Thus, `AgnosticZooming` and analysis thereof carry over word-by-word. In particular, Theorem F.1 applies word-by-word to dynamic inventory-pricing, with (k, λ) -uniform costs replaced by with (k, λ) -uniform values (which are defined in a similar way).

Theorem F.3. *Consider the dynamic inventory-pricing problem with (k, λ) -uniform values. `AgnosticZooming` with $X_{\text{cand}} = X$ achieves regret $R(T) = \tilde{O}(\min(\lambda^{1/3}T^{2/3}, \lambda^{3/5}k^{2/5}T^{3/5}))$.*

Further, the lower bounds for `NonAdaptive` (Lemma F.2) carry over, too. The constructions used in the proof of Lemma F.4 are similar to those in the proof of Lemma F.2, but slightly different technically. We omit the details.

Lemma F.4. *Consider the dynamic inventory-pricing problem with $(k, 4)$ -uniform costs.*

- (a) $R_\psi(T) = \Omega(\psi T + \psi^{-2}) \geq \Omega(T^{2/3})$.
- (b) `NonAdaptive`(ψ) achieves regret $R(T) = \Omega(T^{2/3})$ for $\psi = \tilde{O}(T^{-1/3})$, even if $k = 2$.