Adaptive Contract Design for Crowdsourcing Markets: Bandit Algorithms for Repeated Principal-Agent Problems

CHIEN-JU HO, UCLA, Los Angeles, CA, USA
ALEKSANDRS SLIVKINS, Microsoft Research, New York, NY, USA
JENNIFER WORTMAN VAUGHAN, Microsoft Research, New York, NY, USA

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).

Categories and Subject Descriptors: F.1.2 [**Modes of Computation**]: Online Computation; J.4 [**Social and Behavioral Sciences**]: Economics

Additional Key Words and Phrases: Crowdsourcing; principal-agent; dynamic pricing; multi-armed bandits; regret

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 English text, are easier for some workers than others, re-

---

All missing proofs and the plots for the simulations can be found in the full version, which is available at `arxiv.org`. The full version also presents an application to dynamic pricing of tasks, a more detailed discussion of related work, and other supplementary material.

Much of this research was completed while Ho was an intern at Microsoft Research. This research was partially supported by the NSF under grant IIS-1054911. Any opinions, findings, conclusions, or recommendations are those of the authors alone.

EC'14, June 8–12, 2014, Stanford, CA, USA. ACM 978-1-4503-2565-3/14/06 ...$15.00.

http://dx.doi.org/10.1145/2600057.2602880

quiring less effort to produce high quality results. Additionally, some workers are more dedicated than others, willing to spend 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]

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 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 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 (potentially 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 [Bubeck et al. 2011; Kleinberg et al. 2008; Slivkins 2011a,b], 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 dynamic task pricing. We support the theoretical results with simulations.

Our contributions can be summarized as follows. We define a broad, practically important setting; identify novel problem-specific structure, for both the algorithm and the regret bounds; distill ideas from prior work to work with these structures; argue that our approach is productive by deriving corollaries and comparing to prior work; and identify and analyze specific examples where our theory applies. The main conceptual contribution is the adaptive discretization approach mentioned above.

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

[1]For 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.

theory [Laffont and Martimort 2002]: 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* [Kleinberg et al. 2008], 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, [Agrawal 1995; Auer et al. 2007; Bubeck et al. 2011; Kleinberg 2004; Kleinberg et al. 2008; Slivkins 2011a]. Most related to our techniques is the idea of adaptive discretization [Bubeck et al. 2011; Kleinberg et al. 2008; Slivkins 2011a], and in particular, the *zooming algorithm* [Kleinberg et al. 2008; Slivkins 2011a]. 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 [Kocsis and Szepesvari 2006; Munos and Coquelin 2007; Pandey et al. 2007] which can be used to explicitly reconstruct relevant parts of the underlying metric space [Slivkins 2011b]. 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 *dynamic task pricing* , which is essentially the special case of our setting where in each round a worker is offered to perform a task at a specified price, and can either accept or reject this offer [Badanidiyuru et al. 2012, 2013; Kleinberg and Leighton 2003; Singer and Mittal 2013; Singla and Krause 2013]. [2] In particular, in this setting the worker's strategic choice is directly observable.

A more thorough literature review can be found in the full version.

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 [Laffont and Martimort 2002]. 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) finite set of possible

[2] In Badanidiyuru et al. [2013], this problem is called "dynamic procurement".

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.

We emphasize two crucial (and related) features of the principal-agent model: that the mapping from effort level to outcomes can be randomized, and that the effort level is not directly observed by the requester. This is in line with a standard observation in crowdsourcing that even honest, high-effort workers occasionally make errors.

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.

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). Let X

be the set of all bounded, monotone contracts. We compare a given algorithm against a given subset of "candidate contracts" $X_{\texttt{cand}} \subset X$. Letting $\texttt{OPT}(X_{\texttt{cand}})$ be the optimal utility over all contracts in $X_{\texttt{cand}}$, the goal is to minimize the algorithm's *regret* $R(T|X_{\texttt{cand}})$, defined as $T \times \texttt{OPT}(X_{\texttt{cand}})$ minus the algorithm's expected utility.

The subset $X_{\texttt{cand}}$ may be finite or infinite, possibly $X_{\texttt{cand}} = X$. The most natural example of a finite $X_{\texttt{cand}}$ is the set of all bounded, monotone contracts with payments that are integer multiples of some $\psi > 0$; we call it the *uniform mesh* with granularity ψ, and denote it $X_{\texttt{cand}}(\psi)$.

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 $\mathcal{O} = \{0, 1, 2, \ldots, m\}$ 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. Let $\texttt{OPT}(X_{\texttt{cand}}) = \sup_{x \in X_{\texttt{cand}}} U(x)$.

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.[3] We say 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.

2.1. Discussion

Number of outcomes. Our results assume a small number of outcomes. This regime is important in practice, as the quality of submitted work is typically difficult to evaluate in a very fine granularity. Even with $m = 2$ non-null outcomes, our setting has not been studied before. The special case $m = 1$ is equivalent to the dynamic pricing problem from Kleinberg and Leighton [2003]; we obtain improved results for it, too.

The benchmark. Our benchmark $\texttt{OPT}(\cdot)$ only considers contracts that are bounded and monotone. 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 the full version for an example. Further, it is not clear whether bounded monotone contracts are optimal among monotone contracts.

Our benchmark $\texttt{OPT}(X_{\texttt{cand}})$ is relative to a given set $X_{\texttt{cand}}$, which is typically a finite discretization of the contract space. There are two reasons for this. First, crowdsourcing platforms may require the payments to be multiples of some minimum unit (e.g., one cent), in which case it is natural to restrict our attention to contracts satisfying

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

the same constraint. Second, achieving guarantees relative to $\texttt{OPT}(X)$ for the full generality of our problem appears beyond the reach of our techniques. As in many other machine learning scenarios, it is useful to consider a restricted "benchmark set" – set of alternatives to compare to.[4] In such settings, it is considered important to handle *arbitrary* benchmark sets, which is what we do.

One known approach to obtain guarantees relative to $\texttt{OPT}(X)$ is to start with some finite $X_{\texttt{cand}} \subset X$, design an algorithm with guarantees relative to $\texttt{OPT}(X_{\texttt{cand}})$, and then, as a separate result, bound the discretization error $\texttt{OPT}(X) - \texttt{OPT}(X_{\texttt{cand}})$. Then the choice of $X_{\texttt{cand}}$ drives the tradeoff between the discretization error and regret $R(T|X_{\texttt{cand}})$, and one can choose $X_{\texttt{cand}}$ to optimize this tradeoff. However, while one can upper-bound the discretization error in some (very) simple special cases (see Section 5), it is unclear whether this can be extended to the full generality of dynamic contract design.

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" behavior. 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 enable us to prove Lemma 3.1, and more specifically to guarantee that the collective worker behavior satisfies the following natural property (which is used in the proof of Lemma 3.1): if the requester increases the "increment payment" (as described in the next section) for a particular outcome, the probability of obtaining an outcome at least that good also increases.

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. The prior work on MAB with large / infinite action spaces often assumes known upper bounds on similarity between arms. More precisely, this prior work would assume that an algorithm is given a metric $\mathcal{D}$ on contracts such that expected rewards are Lipschitz-continuous with respect to $\mathcal{D}$, i.e., we have upper bounds $|U(x) - U(y)| \leq \mathcal{D}(x, y)$ for any two contracts x, y.[5] However, 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 struc-

[4]A particularly relevant analogy is contextual bandits with policy sets, e.g., Dudik et al. [2011].
[5]Such upper bound is informative if and only if $\mathcal{D}(x, y) < 1$.

ture 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.

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 $x : \mathcal{O} \to [0, \infty)$ as a vector $\mathbf{x} \in [0, \infty)^m$, where m is the number of non-null outcomes and $\mathbf{x}_\pi = x(\pi) - x(\pi-1) \geq 0$ for each non-null 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 $\texttt{incr}(x)$. Note that if x is bounded, then $\texttt{incr}(x) \in [0, 1]^m$. 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 two contracts are called the *anchors* of C. In each atomic cell C, the algorithm will only use one contract: the unique candidate contract, also called the *anchor* 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:

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

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

$$\texttt{VirtWidth}(C) = \big(V(x^+(C)) - P(x^-(C))\big) - \big(V(x^-(C)) - P(x^+(C))\big)\,. \tag{1}$$

This definition is one crucial place where the problem structure is used. (Note that it is *not* the difference in utility at the anchors.) It is useful due to the following lemma (proved in the full version).

LEMMA 3.1. *If all types satisfy the FOSD assumption and consistent tie-breaking holds, then* $\texttt{width}(C) \leq \texttt{VirtWidth}(C)$ *for each composite cell* C.

Recall that the proof of this lemma is the only place in the paper where we use our assumptions on worker behavior. All further developments hold for any model of worker behavior which satisfies Lemma 3.1.

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

$$W_t(C) = \left(V_t^+(C) - P_t^-(C)\right) - \left(V_t^-(C) - P_t^+(C)\right). \tag{2}$$

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

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

for some absolute constant $c_{\texttt{rad}}$; in our analysis, $c_{\texttt{rad}} \geq 16$ suffices. We will show that with high probability all sample averages defined above will stay within $\texttt{rad}_t(C)$ of the respective expectations. If this high probability event holds, the width estimate $W_t(C)$ will always be within $4\,\texttt{rad}_t(C)$ of $\texttt{VirtWidth}(C)$.

Selection rule. 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* $I_t(\cdot)$. $I_t(C)$ is an upper confidence bound on the expected utility of *any* candidate contract in C, defined as

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

Zooming rule. We zoom in on a composite cell C_t if

$$W_{t+1}(C_t) > 5\,\texttt{rad}_{t+1}(C_t),$$

i.e., 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`

Inputs: subset $X_{\texttt{cand}} \subset X$ 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).
  Sample contract x_t u.a.r. among the anchors of C_t. \\ Anchors are defined in Section 3.1.
  Post contract x_t and observe feedback.
  If $|C \cap X_{\texttt{cand}}| > 1$ **and** $5\, \texttt{rad}_{t+1}(C_t) < W_{t+1}(C_t)$ **then**
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{\text{all relevant quadrants of } C_t\} \setminus \{C_t\}$. \\ C is *relevant* if $|C \cap X_{\texttt{cand}}| \geq 1$.

Integer payments. In practice it may be necessary to only allow contracts in which all payments are integer multiples of some amount ψ, e.g., whole cents. (In this case we can assume that candidate contracts have this property, too.) Then we can redefine the two anchors of each composite cell: the maximal (resp., minimal) anchor is the nearest allowed contract to the maximal (resp., minimal) corner. Width can be redefined as a $\sup$ over all allowed contracts in a given cell. With these modifications, the analysis goes through without significant changes. We omit further discussion of this issue.

4. REGRET BOUNDS AND DISCUSSION

We present the main regret bound for `AgnosticZooming`. Formulating this result requires some new, problem-specific structure. Stated in terms of this structure, the result is somewhat difficult to access. To explain its significance, we 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 $X_{\texttt{cand}}$ of candidate contracts; our goal is to bound the algorithm's regret with respect to candidate contracts.

Recall that $\texttt{OPT}(X_{\texttt{cand}}) = \sup_{x \in X_{\texttt{cand}}} U(x)$ is the optimal expected utility over candidate contracts. The algorithm's regret with respect to candidate contracts is $R(T|X_{\texttt{cand}}) = T\, \texttt{OPT}(X_{\texttt{cand}}) - U$, where T is the time horizon and U is the expected cumulative 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: $\Delta(x) = \texttt{OPT}(X_{\texttt{cand}}) - U(x)$. Let $X_\epsilon = \{x \in X_{\texttt{cand}} :\ \Delta(x) \leq \epsilon\}$.

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 $X_{\texttt{cand}}$ of candidate contracts.

Let $\mathcal{F}_\epsilon$ denote the collection of all feasible, composite cells C such that $\texttt{VirtWidth}(C) \geq \epsilon$. For $Y \subset X_{\texttt{cand}}$, 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 $N_\epsilon(Y|X_{\texttt{cand}})$ in place of $N_\epsilon(Y)$ to emphasize the dependence on $X_{\texttt{cand}}$.

Using the structure defined above, the main theorem is stated as follows. We prove this theorem in Section 6.

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* $X_{\texttt{cand}}$ *of can-*

didate contracts. There is an absolute constant $\beta_0 > 0$ such that for any $\delta > 0$,

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

Remark 4.2. As discussed in Section 2.1, we target the practically important case of a small number of outcomes. The impact of larger m is an exponential dependence on m in the $O()$ notation, and, more importantly, increased number of candidate policies (typically exponential in m for a given granularity).

Remark 4.3. Our regret bounds do not depend on the number of worker types, in line with prior work on dynamic pricing. Essentially, this is because bandit approaches tend to depend only on expected reward of a given "arm" (and perhaps also on the variance), not the finer properties of the distribution.

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 $N_\epsilon(Y|X_{\texttt{cand}})$ 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 $\mathcal{F}_\epsilon$ which does not contain any other cell in $\mathcal{F}_\epsilon$. Let $N^{\texttt{min}}_\epsilon(Y)$ 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

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

where ψ is the smallest size of a feasible cell.[6] Thus, Equation (5) can be easily restated using the covering numbers $N^{\texttt{min}}_\epsilon(\cdot)$ 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,[7] 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*:

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

Note that the width dimension depends on $X_{\texttt{cand}}$ 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 4.4. *Consider the the setting of Theorem 4.1. For any $\alpha > 0$, let $d =$ $\texttt{WidthDim}_\alpha$. Then*

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

---

[6]To 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 $C' \subset C$.

[7]Given 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 $\epsilon > 0$.

The width dimension is similar to the "zooming dimension" in Kleinberg et al. [2008] and "near-optimality dimension" in Bubeck et al. [2011] in the work on "bandits in metric spaces." See the full version 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 $X_{\texttt{cand}}$ as arms.[8] For concreteness, and following the prior work [Kleinberg 2004; Kleinberg and Leighton 2003; Kleinberg et al. 2008], we use a well-known algorithm `UCB1` [Auer et al. 2002] 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.[9]

COROLLARY 4.5. *In the setting of Theorem 4.1, the regret of* `AgnosticZooming` *can be upper-bounded as follows:*

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

(b) $R(T|X_{\texttt{cand}}) \leq \tilde{O}(\sqrt{T\ |X_{\texttt{cand}}|})$.

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

The best known regret bounds for `NonAdaptive` coincide with those in Corollary 4.5 up to poly-logarithmic factors. However, the regret bounds in Theorem 4.1 may be significantly better than the ones in Corollary 4.5. We further discuss this in the next section, in the context of a specific example.

5. A SPECIAL CASE: THE "HIGH-LOW EXAMPLE "

We show an application of the machinery in Section 4 on a specific example. On this example `AgnosticZooming` significantly outperforms `NonAdaptive`.

The most basic special 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 (where "reject" corresponds to the null effort level), and this choice is fully observable. This setting has been studied before [Badanidiyuru et al. 2012, 2013; Kleinberg and Leighton 2003; Singla and Krause 2013]; we will call it *dynamic task pricing* . Here the contract is completely specified by the price p for the non-null outcome. The supply distribution is summarized by the function $S(p) = \Pr[\text{accept}|p]$, so that the corresponding expected utility is $U(p) = S(p)(v-p)$, where v is the value for the non-null outcome. This special case is already quite rich, because $S(\cdot)$ can be an arbitrary non-decreasing function. By using adaptive discretization, we achieve significant improvement over prior work; see the full version for further discussion.

We consider a somewhat richer setting in which workers' strategic decisions are *not* observable; this is a salient feature of our setting, called *moral hazard* in the contract theory literature. 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, while `high` outcome brings value $v > 0$. `Low` effort level inflicts zero cost on a worker and leads to low outcome with probability 1. 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 will call this the ***high-low example***; it is perhaps the simplest example that features moral hazard.

---

[8]To simplify the proofs of the lower bounds, we assume that the candidate contracts are randomly permuted when given to the MAB algorithm.

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

In this example, the worker's type consists of a pair $(c_{\mathtt{h}}, \theta_{\mathtt{h}})$, where $c_{\mathtt{h}} \geq 0$ is the cost for `high` effort and $\theta_{\mathtt{h}} \in [0,1]$ is the probability of `high` outcome given `high` effort. Note that dynamic task pricing is equivalent to the special case $\theta_{\mathtt{h}} = 1$.

The following claim states a crucial property of the high-low example.

CLAIM 5.1. *Consider the high-low example with a fixed supply distribution. Then* $\Pr[\mathtt{high}$ *outcome* $|$ *contract* $x]$ *depends only on* $p = x(\mathtt{high}) - x(\mathtt{low})$*; denote this probability by* $S(p)$*. Moreover,* $S(p)$ *is non-decreasing in* p*. Therefore:*

- *expected utility is* $U(x) = S(p)(v-p) - x(\mathtt{low})$*.*
- *discretization error* $\mathtt{OPT}(X) - \mathtt{OPT}(X_{\mathtt{cand}}(\psi))$ *is at most* 3ψ*, for any* $\psi > 0$*.*

Recall that $X_{\mathtt{cand}}(\psi)$, the uniform mesh with granularity $\psi > 0$, consists of all bounded, monotone contracts with payments in $\psi\mathbb{N}$.

For our purposes, the supply distribution is summarized via the function $S(\cdot)$. Denote $\tilde{U}(p) = S(p)(v-p)$. Note that $U(x)$ is maximized by setting $x(\mathtt{low}) = 0$, in which case $U(x) = \tilde{U}(p)$. Thus, if an algorithm knows that it is given a high-low example, it can set $x(\mathtt{low}) = 0$, thereby reducing the dimensionality of the search space. Then the problem essentially reduces to dynamic task pricing with the same $S(\cdot)$.

However, in general an algorithm does not *know* whether it is presented with the high-low example (because the effort levels are not observable). So in what follows we will consider algorithms that do not restrict themselves to $x(\mathtt{low}) = 0$.

"Nice" supply distribution. We focus on a supply distribution D that is "nice", in the sense that $S(\cdot)$ satisfies the following two properties:

- $S(p)$ is Lipschitz-continuous: $|S(p) - S(p')| \leq L|p - p'|$ for some constant L.
- $\tilde{U}(p)$ is *strongly concave*, in the sense that $\tilde{U}''(\cdot)$ exists and satisfies $\tilde{U}''(\cdot) \leq C < 0$.

Here L and C are absolute constants. We call such D *strongly Lipschitz-concave*.

The above properties are fairly natural. For example, they are satisfied if $\theta_{\mathtt{h}}$ is the same for all worker types and the marginal distribution of $c_{\mathtt{h}}$ is piecewise uniform such that the density is between $\frac{1}{\lambda}$ and λ, for some absolute constant $\lambda \geq 1$.

We show that for any choice $X_{\mathtt{cand}} \subset X$, `AgnosticZooming` has a small width dimension in this setting, and therefore small regret.

LEMMA 5.2. *Consider the high-low example with a strongly Lipschitz-concave supply distribution. Then the width dimension is at most* $\frac{1}{2}$*, for any given* $X_{\mathtt{cand}} \subset X$*. Therefore,* `AgnosticZooming` *with this* $X_{\mathtt{cand}}$ *has regret* $R(T|X_{\mathtt{cand}}) = O(\log T)\, T^{3/5}$.

We contrast this with the performance of `NonAdaptive`, parameterized with the natural choice $X_{\mathtt{cand}} = X_{\mathtt{cand}}(\psi)$. We focus on $R(T|X)$: regret w.r.t. the best contract in X. We show that `AgnosticZooming` achieves $R(T|X) = \tilde{O}(T^{3/5})$ for a wide range of $X_{\mathtt{cand}}$, whereas `NonAdaptive` cannot do better than $R(T|X) = O(T^{3/4})$ for any $X_{\mathtt{cand}} = X_{\mathtt{cand}}(\psi)$, $\psi > 0$.

LEMMA 5.3. *Consider the setting of Lemma 5.2. Then:*

(a) `AgnosticZooming` *with* $X_{\mathtt{cand}} \supset X_{\mathtt{cand}}(T^{-2/5})$ *has regret* $R(T|X) = O(T^{3/5} \log T)$*.*

(b) `NonAdaptive` *with* $X_{\mathtt{cand}} = X_{\mathtt{cand}}(\psi)$ *cannot achieve regret* $R(T|X) < o(T^{3/4})$ *over all problem instances, for any* $\psi > 0$*.* [10]

6. PROOF OF THE MAIN REGRET BOUND (THEOREM 4.1)

We now prove the main result from Section 4. Our high-level approach is to define a *clean execution* of an algorithm as an execution in which some high-probability events

[10]This lower bound holds even if `UCB1` in `NonAdaptive` is replaced with any other MAB algorithm.

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 6.1. *In each round t of each execution of* `AgnosticZooming`*:*
(a) All active cells are relevant,
(b) Each candidate contract is contained in some active cell,
(c) $W_t(C) \leq 5\,\texttt{rad}_t(C)$ for each active composite cell C.

Note that the zooming rule is essential to ensure Invariant 6.1(c).

6.1. Analysis of the randomness

DEFINITION 6.2 (CLEAN EXECUTION). *An execution of* `AgnosticZooming` *is called* clean *if for each round t and each active cell C it holds that*

$$|U(C) - U_t(C)| \leq \texttt{rad}_t(C), \tag{9}$$

$$|\texttt{VirtWidth}(C) - W_t(C)| \leq 4\,\texttt{rad}_t(C) \qquad \textit{(if } C \textit{ is composite).} \tag{10}$$

LEMMA 6.3. *Assume $c_{\texttt{rad}} \geq 16$ and $T \geq \max(1+2^m, 18)$. Then:*
(a) $\Pr[$ Equation (9) holds $\;\forall$ rounds t, active cells $C\,] \geq 1 - 2\,T^{-2}$.
(b) $\Pr[$ Equation (10) holds $\;\forall$ rounds t, active composite cells $C\,] \geq 1 - 16\,T^{-2}$.
Consequently, an execution of `AgnosticZooming` *is clean with probability at least $1-1/T$.*

Lemma 6.3 follows from the standard concentration inequality known as "Chernoff Bounds". However, one needs to be careful about conditioning and other details.

PROOF OF LEMMA 6.3(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

$$\Pr[\,|U(C) - U_t(C)| \leq \texttt{rad}_t(C) \text{ for all rounds } t \mid C_j = C\,] \geq 1 - 2\,T^{-4}. \tag{11}$$

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 $n(S) \geq 1$. (That is, the per-round reward from choosing cell C.) Let $U'_1, \ldots, U'_T$ 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 $(U_1, \ldots, U_n)$ has the same joint distribution as the tuple $(U'_1, \ldots, U'_n)$. Consequently, applying Chernoff Bounds to the latter tuple, it follows that

$$\Pr\left[\; \left|U(C) - \tfrac{1}{n}\textstyle\sum_{s=1}^{n} U_s\right| \leq \sqrt{\tfrac{1}{n}\, c_{\texttt{rad}}\, \log(T)} \;\middle|\; \{C_j = C\} \wedge \{n(C) = n\} \;\right]$$
$$\geq 1 - 2\,T^{-2c_{\texttt{rad}}} \geq 1 - 2\,T^{-5}.$$

Taking the Union Bound over all $n \leq T$, and plugging in $\texttt{rad}_t(C_j)$, $n_t(C_j)$, and $U_t(C_j)$, we obtain Equation (11).

Now, let us keep j fixed in Equation (11), and integrate over C. More precisely, let us multiply both sides of Equation (11) by $\Pr[C_j = C]$ and sum over all feasible cells C. We obtain, for all $j \leq T^2$:

$$\Pr[\,|U(C_j) - U_t(C_j)| \leq \texttt{rad}_t(C_j) \text{ for all rounds } t\,] \geq 1 - 2\,T^{-4}. \tag{12}$$

(Note that to obtain Equation (12), we do *not* need to take the Union Bound over all feasible cells C.) To conclude, we take the Union Bound over all $j \leq 1 + T^2$. $\square$

Proof Sketch of Lemma 6.3(b). We show that

$$\Pr\left[\, \left|V^+(C) - V_t^+(C)\right| \leq \mathtt{rad}_t(C)\ \forall \text{ rounds } t, \text{ active composite cells } C \,\right] \geq 1 - \tfrac{4}{T^2}, \quad (13)$$

and similarly for $V^-()$, $P^+()$ and $P^-()$. Each of these four statements is proved similarly, using the technique from Lemma 6.3(a). In what follows, we sketch the proof for one of the four cases, namely for Equation (13).

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":

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

With the technique from the proof of Lemma 6.3(a), we can establish the following high-probability event:

$$\left|V^+(C) - V_t^+(C)\right| \leq \mathtt{rad}_t^+(C). \quad (14)$$

More precisely, we can prove that

$$\Pr\left[\, \text{Equation (14) holds} \quad \forall \text{ rounds } t, \text{ active composite cells } C \,\right] \geq 1 - 2\,T^{-2}.$$

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:[11]

$$n_t^+(C) \geq \tfrac{1}{2}\,n_t(C) - \tfrac{1}{4}\,\mathtt{rad}_t(C). \quad (15)$$

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

$$\Pr\left[\, \text{Equation (15) holds} \quad \forall \text{ rounds } t, \text{ active composite cells } C \,\right] \geq 1 - 2\,T^{-2}. \quad (16)$$

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

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

Therefore, once Equations (14) and (15) hold, we have $\left|V^+(C) - V_t^+(C)\right| \leq \mathtt{rad}_t(C)$. This completes the proof of Equation (13). $\square$

6.2. 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 6.4. *In any clean execution, $I(C_t) \geq \mathtt{OPT}(X_{\mathtt{cand}})$ for each round t.*

Proof. Fix round t, and let x^* be any candidate contract. By Invariant 6.1(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 $I_t(C_t^*) \geq U(x^*)$

[11] The constant $\frac{1}{4}$ in Equation (15) is there to enable a consistent choice of n_0 in the remainder of the proof.

by the clean execution. If C_t^* is composite then

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

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. □

CLAIM 6.5. *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 $I(C_t) \leq U(C_t) + 2\,\texttt{rad}_t(C_t)$.
(b) if C_t is composite then $I(C_t) \leq U(x) + O(\texttt{rad}_t(C_t))$ for each contract $x \in C_t$.

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

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

$$\begin{aligned} U_t(C_t) &\leq U(C_t) + \texttt{rad}_t(C_t) && \text{by clean execution} \\ &\leq U(x) + \texttt{width}(C_t) + \texttt{rad}_t(C_t) && \text{by definition of width} \\ &\leq U(x) + \texttt{VirtWidth}(C_t) + \texttt{rad}_t(C_t) && \text{by Lemma 3.1} \\ &\leq U(x) + W_t(C_t) + 5\,\texttt{rad}_t(C_t) && \text{by clean execution.} \qquad (17) \\ I_t(C_t) &= U_t(C_t) + W_t(C_t) + 5\,\texttt{rad}_t(C_t) && \text{by definition of index} \\ &\leq U(x) + 2\,W_t(C_t) + 10\,\texttt{rad}_t(C_t) && \text{by Equation (17)} \\ &\leq U(x) + 20\,\texttt{rad}_t(C_t) && \text{by Invariant 6.1(c).} \end{aligned}$$
□

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

PROOF. By Claims 6.4 and 6.5, $\Delta(C_t) \leq O(\texttt{rad}_t(C_t))$ 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 $\Delta(C) \leq O(\texttt{rad}_s(C))$. The claim follows since $\texttt{rad}_s(C) = \texttt{rad}_t(C)$. □

CLAIM 6.7. *In a clean execution, each cell C is selected $\leq O(\log T / (\Delta(C))^2)$ times.*

PROOF. By Claim 6.6, $\Delta(C) \leq O(\texttt{rad}_T(C))$. The claim follows from the definition of $\texttt{rad}_T$ in Equation (3). □

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

$$R(T|X_{\texttt{cand}}) = \textstyle\sum_{x \in X} n(x)\ \Delta(x) \leq \sum_{\text{cells } C}\ n(C)\,\Delta(C). \qquad (18)$$

The next result (Lemma 6.8) upper-bounds the right-hand side of Equation (18) for a clean execution. By Lemma 6.3, this suffices to complete the proof of Theorem 4.1

LEMMA 6.8. *Consider a clean execution of* `AgnosticZooming`. *For any $\delta \in (0,1)$,*

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

The proof of Lemma 6.8 relies on some simple properties of $\Delta(\cdot)$, stated below.

CLAIM 6.9. *Consider two relevant cells $C \subset C_{\text{p}}$. Then:*

(a) $\Delta(C) \leq \Delta(C_{\mathrm{p}})$.
(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 $\Delta(C) = \Delta(C_{\mathrm{p}}) = \Delta(x)$, 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.) So, $x \in X_\epsilon$. □

PROOF OF LEMMA 6.8. 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 6.7, 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 $\mathcal{G}_\epsilon$, $\epsilon \geq \delta/2$. It follows that

$$\Sigma \leq \delta T + O(\log T) \textstyle\sum_{\epsilon = 2^{-i} \geq \delta/2} \frac{|\mathcal{G}_\epsilon|}{\epsilon}. \tag{19}$$

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 6.9(b). Note that C is zoomed in on in some round, say in round $t-1$. Then

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

so $\mathtt{rad}_t(C) \leq \mathtt{VirtWidth}(C)$. Therefore, using Claim 6.6, we have

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

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 $C \subset C_{\mathrm{p}}$, we have $\Delta(C) \leq \Delta(C_{\mathrm{p}})$ by Claim 6.9(a). Therefore, invoking case (i), we have

$$\epsilon \leq \Delta(C) \leq \Delta(C_{\mathrm{p}}) \leq O(\mathtt{VirtWidth}(C_{\mathrm{p}})).$$

Since $\Delta(C) < 2\epsilon$, C overlaps with $X_{2\epsilon}$ by Claim 6.9(b), and therefore so does C_{p}. It follows that $C_{\mathrm{p}} \in \mathcal{F}_{\Omega(\epsilon)}(X_{2\epsilon})$.

Combing these two cases, it follows that $|\mathcal{G}_\epsilon| \leq (2^m + 1)\left|\mathcal{F}_{\Omega(\epsilon)}(X_{2\epsilon})\right|$. Plugging this into (19) and making an appropriate substitution $\epsilon \to \Theta(\epsilon)$ to simplify the resulting expression, we obtain the regret bound in Theorem 4.1 □

7. SIMULATIONS

We evaluate the performance of `AgnosticZooming` through simulations. We compare `AgnosticZooming` with two versions of `NonAdaptive` that use, respectively, two standard bandit algorithms: `UCB1` [Auer et al. 2002] and Thompson Sampling [Thompson 1933] (with Gaussian priors). For both `UCB1` and `AgnosticZooming`, we replace the logarithmic confidence terms with small constants.[12] All three algorithms are run with $X_{\texttt{cand}} = X_{\texttt{cand}}(\psi)$, where $\psi > 0$ is the granularity of the discretization.

Setup. We consider a version of the high-low example, as described in Section 5. We set the requester's values to $V(\texttt{high}) = 1$ and $V(\texttt{low}) = .3$. The probability of obtaining `high` outcome given `high` effort is set to $\theta_{\texttt{h}} = .8$. Thus, the worker's type is characterized by the cost $c_{\texttt{h}}$ for `high` effort. We consider three supply distributions:

— *Uniform Worker Market* : $c_{\texttt{h}}$ is uniformly distributed on $[0, 1]$.
— *Homogeneous Worker Market* : $c_{\texttt{h}}$ is the same for every worker.
— *Two-Type Market* : $c_{\texttt{h}}$ is uniformly distributed over two values, $c'_{\texttt{h}}$ and $c''_{\texttt{h}}$.

These first two markets represent the extreme cases when workers are extremely homogeneous or extremely diverse, and the third market is one way to represent the middle ground. For each market, we run each algorithm 100 times. For Homogeneous Worker Market, $c_{\texttt{h}}$ is drawn uniformly at random from $[0, 1]$ for each run. For Two-Type Market, $c'_{\texttt{h}}$ and $c''_{\texttt{h}}$ are drawn independently and uniformly from $[0, 1]$ on each run.

Results. The plots can be found in the full version. Across all simulations, `AgnosticZooming` performs comparably to or better than `NonAdaptive`. In particular, its performance does not appear to suffer from large "hidden constants" that appear in the analysis. We find that `AgnosticZooming` converges faster than `NonAdaptive` when ψ is near-optimal or smaller; this is consistent with the intuition that `AgnosticZooming` focuses on exploring the more promising regions. When ψ is large, `AgnosticZooming` converges slower than `NonAdaptive`, but eventually achieves the same performance. Further, we find that `AgnosticZooming` with small ψ performs well compared to `NonAdaptive` with larger ψ: not much worse initially, and much better eventually.

Our simulations suggest that *if* time horizon T is known in advance *and* one can tune ψ to T, then `NonAdaptive` can achieve similar performance as `AgnosticZooming`. However, in real applications approximately optimal ψ may be difficult to compute, and the T may not be known in advance.

8. CONCLUSIONS

Motivated by applications to crowdsourcing markets, we define the *dynamic contract design problem*: 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. Our main conceptual contribution is the adaptive discretization approach that does not rely on Lipschitz-continuity assumptions. We provably improve on the uniform discretization approach from prior work, both in the general case and in some illustrative special cases. These theoretical results are supported by simulations.

We believe that the dynamic contract design problem deserves further study. First, it is not clear whether our provable results can be improved, perhaps using substantially different algorithms and relative to other problem-specific structures. In particular, no lower bounds are currently known. Second, our adaptive discretization approach may be fine-tuned, in several different ways, to improve its performance in practice. One possible area of improvement is selecting a feasible cell in a "smoother", probabilistic way, e.g., as in Thompson Sampling [Thompson 1933]. Third, one needs deeper insights into the structure of the (static) principal-agent problem, primarily in order to upper-bound discretization errors of the form $\texttt{OPT}(X_{\texttt{cand}}) - \texttt{OPT}(X_{\texttt{cand}}(\epsilon))$ and $\texttt{OPT}(X_{\texttt{cand}}(\epsilon)) - \texttt{OPT}(X_{\texttt{cand}}(\epsilon'))$, $\epsilon > \epsilon' > 0$. Also of interest is the effect of restricting our attention

[12] We find such changes beneficial in practice, for both algorithms; this observation is consistent with prior work [Radlinski et al. 2008; Slivkins et al. 2013].

to monotone contracts. Fourth, a (much) more extensive analysis of special cases is in order. The most immediate direction is deriving lucid corollaries from the current regret bounds, preferably also optimizing the choice of candidate contracts. One can also design improved algorithms and derive specialized lower bounds.

Going beyond our current model, a natural (but probably very difficult) direction is to incorporate budget constraints, extending the results on online task pricing [Badanidiyuru et al. 2012, 2013; Singla and Krause 2013].

REFERENCES

Rajeev Agrawal. 1995. The continuum-armed bandit problem. *SIAM J. Control and Optimization* 33, 6 (1995), 1926–1951.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. 2002. Finite-time Analysis of the Multi-armed Bandit Problem. *Machine Learning* 47, 2-3 (2002), 235–256. Preliminary version in *15th ICML*, 1998.

Peter Auer, Ronald Ortner, and Csaba Szepesvári. 2007. Improved Rates for the Stochastic Continuum-Armed Bandit Problem. In *20th COLT*. 454–468.

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Yaron Singer. 2012. Learning on a budget: posted price mechanisms for online procurement. In *13th ACM EC*. 128–145.

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. 2013. Bandits with Knapsacks. In *54th IEEE FOCS*.

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvari. 2011. Online Optimization in X-Armed Bandits. *J. of Machine Learning Research (JMLR)* 12 (2011), 1587–1627. Preliminary version in *NIPS 2008*.

Miroslav Dudik, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. 2011. Efficient Optimal Leanring for Contextual Bandits. In *27th UAI*.

Robert Kleinberg. 2004. Nearly Tight Bounds for the Continuum-Armed Bandit Problem.. In *18th NIPS*.

Robert Kleinberg and Tom Leighton. 2003. The Value of Knowing a Demand Curve: Bounds on Regret for Online Posted-Price Auctions.. In *44th IEEE FOCS*. 594–605.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. 2008. Multi-Armed Bandits in Metric Spaces. In *40th ACM STOC*. 681–690.

Levente Kocsis and Csaba Szepesvari. 2006. Bandit Based Monte-Carlo Planning. In *17th ECML*. 282–293.

Jean-Jacques Laffont and David Martimort. 2002. *The Theory of Incentives: The Principal-Agent Model*. Princeton University Press.

Rémi Munos and Pierre-Arnaud Coquelin. 2007. Bandit algorithms for tree search. In *23rd UAI*.

Sandeep Pandey, Deepak Agarwal, Deepayan Chakrabarti, and Vanja Josifovski. 2007. Bandits for Taxonomies: A Model-based Approach. In *SDM*.

Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. 2008. Learning diverse rankings with multi-armed bandits. In *25th ICML*. 784–791.

Yaron Singer and Manas Mittal. 2013. Pricing Mechanisms in Crowdsourcing Markets. In *WWW*.

Adish Singla and Andreas Krause. 2013. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *22nd WWW*. 1167–1178.

Aleksandrs Slivkins. 2011a. Contextual Bandits with Similarity Information. In *24th COLT*. To appear in J. of Machine Learning Research (JMLR), 2013.

Aleksandrs Slivkins. 2011b. Multi-armed bandits on implicit metric spaces. In *25th NIPS*.

Aleksandrs Slivkins, Filip Radlinski, and Sreenivas Gollapudi. 2013. Learning optimally diverse rankings over large document collections. *J. of Machine Learning Research (JMLR)* 14, Feb (2013), 399–436. Preliminary version in *27th ICML*, 2010.

William R. Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25, 3-4 (1933), 285294.