

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

Chien-Ju Ho

Cornell University, Ithaca, NY, USA

CH624@CORNELL.EDU

Aleksandrs Slivkins

Microsoft Research, New York, NY, USA

SLIVKINS@MICROSOFT.COM

Jennifer Wortman Vaughan

Microsoft Research, New York, NY, USA

JENN@MICROSOFT.COM

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 analyze 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). Our results advance the state of art on several different topics: the theory of crowdsourcing markets, principal-agent problems, multi-armed bandits, and dynamic pricing.

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 CrowdFlower, 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, requiring 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 quality. 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 properties of a random worker (formally: the distribution over the workers’ types) are not known to the requester, but may be learned over time. 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.

For concreteness, consider a special case in which a worker can strategically choose to perform a task with low effort or with high effort, and the task may be completed either at low quality or at high quality. The low effort incurs no cost and results in low quality, which in turn brings no value to the requester. The high effort leads to high quality with some positive probability (which may vary from one worker to another, and is unknown to the requester). The requester only observes the quality of completed tasks, and therefore cannot always infer the effort level. This example captures the two main tenets of our model: that the properties of a random worker are unknown to the requester and that workers’ strategic decisions are unobservable.

We treat the dynamic contract design 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 (Kleinberg, Slivkins, & Upfal, 2008; Bubeck, Munos, Stoltz, & Szepesvari, 2011a; Slivkins, 2014, 2011), 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

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 (often called “gold” data). Quality-contingent payments can then be based on the known instances.

standard bandit algorithm. We illustrate our general results via some corollaries and special cases, including the high-low example described above. We support the theoretical results with simulations.

Further, we consider a special case of our setting where each worker only chooses whether to accept or reject a given task. This special case corresponds to a dynamic task pricing problem previously studied in the literature. Our results significantly improve over the prior work on this problem.

Our contributions can be summarized as follows. We define a broad, practically important setting in crowdsourcing markets; 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 contributions are the model itself and the adaptive discretization approach mentioned above. Finally, this paper prompts further research on dynamic contract design along several directions that we outline in the conclusion.

1.1 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 (Laffont & Martimort, 2002). A single round of our model corresponds to the basic principal-agent setting, with *adverse selection* (unknown worker’s type) and *moral hazard* (unobservable worker’s decisions). Unlike much of the existing work in contract theory, the prior over worker types is not known to the principal, but may be learned over time. Accordingly, 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; Kleinberg, 2004; Auer, Ortner, & Szepesvári, 2007; Kleinberg et al., 2008; Bubeck et al., 2011a; Slivkins, 2014). Most related to our techniques is the idea of adaptive discretization (Kleinberg et al., 2008; Bubeck et al., 2011a; Slivkins, 2014), and in particular, the *zooming algorithm* (Kleinberg et al., 2008; Slivkins, 2014). 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 & Szepesvari, 2006; Munos & Coquelin, 2007; Pandey, Agarwal, Chakrabarti, & Josifovski, 2007) which can be used to explicitly reconstruct relevant parts of the underlying metric space (Slivkins, 2011; Bull, 2013). 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 (Kleinberg & Leighton, 2003; Badanidiyuru, Kleinberg, & Singer, 2012; Singer & Mittal, 2013; Singla & Krause, 2013; Badanidiyuru, Kleinberg, & Slivkins, 2013). In

particular, Badanidiyuru et al. (2012) and Singla and Krause (2013) study a version of our setting with simple, single-price contracts (independent of the output), where the focus is on dealing with a global budget constraint.

A more thorough literature review (including a discussion of some related empirical work) can be found in Section 9.

2. Our Setting: The Dynamic Contract Design Problem

In this section, we formally define the problem that we set out to solve and discuss the implications of several aspects of our model.

2.1 Our Model

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 & 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.

2.1.1 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 *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.

2.1.2 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 the standard “doubling trick” (Cesa-Bianchi & Lugosi, 2006) in which full executions of the algorithm are repeated in “phases” with exponentially increasing time horizons.

2.1.3 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” $X_{\text{cand}} \subset X$. Letting $\text{OPT}(X_{\text{cand}})$ be the optimal utility over

all contracts in X_{cand} , the goal is to minimize the algorithm’s *regret* $R(T|X_{\text{cand}})$, defined as $T \times \text{OPT}(X_{\text{cand}})$ minus the algorithm’s expected utility.

The subset X_{cand} may be finite or infinite, possibly $X_{\text{cand}} = X$. The most natural example of a finite X_{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 $X_{\text{cand}}(\psi)$.

2.1.4 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 $F_i(\pi|e) = \sum_{\pi' \geq \pi} f_i(\pi'|e)$ be the probability of obtaining an outcome at least as good as π having chosen effort 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 $\text{OPT}(X_{\text{cand}}) = \sup_{x \in X_{\text{cand}}} U(x)$.

2.1.5 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 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.

2.1.6 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.

2.2 Discussion

Before jumping into our results, we discuss the implications of several aspects of our model in more detail.

2.2.1 NUMBER OF OUTCOMES

Our results assume a small number of outcomes. This regime is important in practice for several reasons. First, some tasks naturally have only a small number of outcomes.

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

For example, a binary labeling task can only have four possible outcomes if completed: $\{\text{yes/no}\} \times \{\text{correct/incorrect}\}$. Second, it often makes sense to group together multiple outcomes with similar value to the requester (such as false positives and false negatives) if the value is not known very precisely. This has the added benefit that contracts become simpler from the workers’ perspective. Third, even if a task can be completed in many different ways, the quality may be difficult to evaluate in fine granularity; a good example is a translation of a sentence. Fourth, even if a fine-grained quality evaluation exists, such as the error count in speech transcription tasks, it may be difficult to make it consistent across different tasks.

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.

2.2.2 THE BENCHMARK

Our benchmark $\text{OPT}(\cdot)$ 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 Appendix A for an example. Further, it is not clear whether bounded monotone contracts are optimal among monotone contracts.

Our benchmark $\text{OPT}(X_{\text{cand}})$ is relative to a given set X_{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 the same constraint. Second, achieving guarantees relative to $\text{OPT}(X)$ 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.³ 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 $\text{OPT}(X)$ is to start with some finite $X_{\text{cand}} \subset X$, design an algorithm with guarantees relative to $\text{OPT}(X_{\text{cand}})$, and then, as a separate result, bound the discretization error $\text{OPT}(X) - \text{OPT}(X_{\text{cand}})$. Then the choice of X_{cand} drives the tradeoff between the discretization error and regret $R(T|X_{\text{cand}})$, and one can choose X_{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.

2.2.3 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

3. A particularly relevant analogy is contextual bandits with policy sets (Dudik, Hsu, Kale, Karampatziakis, Langford, Reyzin, & Zhang, 2011).

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 a natural “increment payment property” used in the proof of Lemma 3.1: if the requester increases the “increment payment” for a particular outcome (as described in the next section), the probability of obtaining an outcome at least that good also increases. In particular, this property is consistent with worker behavior that takes into account long-term effects such as changes in reputation scores. It is also consistent with workers acting upon subjective (and possibly incorrect) beliefs about the offered contract, such as beliefs about how the “guaranteed” base payment may actually depend on the quality of submitted work.⁴

2.2.4 MINIMUM WAGE

For ethical or legal reasons one may want to enforce some form of minimum wage. This can be expressed within our model as a *minimal payment* θ for a completed task, i.e., for any non-null outcome. Our algorithm can be easily modified to accommodate this constraint. Essentially, it suffices to restrict the action space to contracts that pay at least θ for a completed task. Formally, the “increment space” defined in Section 3 should be $[\theta, 1] \times [0, 1]^{m-1}$ rather than $[0, 1]^m$, and the “quadrants” of each “cell” are defined by splitting the cell in half in each dimension. All our results easily carry over to this version (restricting X_{cand} to contracts that pay at least θ for a completed task). We omit further discussion of this issue for the sake of simplicity.

2.2.5 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 an 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 or infinite action spaces often assumes known upper bounds on the 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

4. A worker model that incorporates such subjective beliefs has been suggested by Ho, Slivkins, Suri, and Vaughan (2015) based on experimental evidence, and this model satisfies the increment payment property mentioned above.

\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 auxiliary 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 auxiliary structure provides some “soft” information on similarity between contracts, in the sense that numerically similar contracts usually (but not always) induce similar response from the workers.

2.2.6 APPLICABILITY OF THE MODEL

Despite a considerable generality, our model is somewhat idealized. Let us discuss several potential concerns regarding how applicable and realistic the model is.

An implicit intuition behind using performance-based payments is that they can incentivize better quality. A growing empirical literature on incentives in crowdsourcing markets finds that it happens for some types of tasks but not for others. In particular, the experiments in the work of Ho et al. (2015) suggest that it happens if and only if the task is *effort-responsive*, in the sense that one can obtain higher quality work by increasing effort, at effort levels that are not too costly for the worker. This observation is consistent with our worker model: indeed, effort-responsiveness of a task is a joint property of the production function and the cost function which implies a significant response to sufficiently increased quality-based payments. Ho et al. propose pilot experiments to determine whether a given type of tasks is effort-responsive, which in turn would shed light on whether to use quality-based payments for these tasks in practice. A more comprehensive discussion of related empirical work can be found in Section 9. We also observe that our model and results for a single non-null outcome are applicable and novel even if the task is not effort-responsive.

Following the bulk of prior work on dynamic pricing and MAB, we assume that the collective worker response to a given contract, as given by a distribution over the outcomes, does not depend on the contracts that have been offered in the past, or on the algorithm used to choose future contracts. Thus, we do not model the possibility that price experimentation may alter future worker responses, or that the workers may try to “game the system.” Both effects are not easy to model and extremely difficult to analyze, even in the relatively simple scenario of a single non-null outcome with no emphasis on adaptive discretization. Additionally, the available empirical work does not provide sufficient guidance on how to choose a realistic model for these effects among theoretically plausible alternatives. We leave this to future work.

From the MAB point of view, our model does not incorporate the possibility that the worker response may intrinsically change over time, or the fact that requesters may have hard budget constraints on the total amount of money that they can spend. This reflects the limitations of state-of-the-art work on MAB: adaptive discretization, budgets, and adversarial change over time are fairly well-understood separately, but any two of them (let alone all three) have not been studied jointly. That said, we conjecture that our techniques would be useful in generalizing dynamic pricing and dynamic contract design to these richer settings.

5. Such upper bound is informative if and only if $\mathcal{D}(x, y) < 1$.

Likewise, we do not model a scenario when the requester’s stream of tasks overwhelms the crowdsourcing market and causes a drastic change in the available worker population (and therefore in the worker response). In particular, we assume that the worker pool is sufficiently large to accommodate the requester. This is a relatively benign assumption for a large crowdsourcing system.

To deploy dynamic selection of prices or contracts in practice (regardless of the particular algorithm used) the crowdsourcing platform needs to enable requesters to change their prices/contracts relatively fast in response to observed worker responses. While this feature is not currently instrumented on commercial platforms such as Amazon Mechanical Turk, it appears easily implementable from an engineering point of view. We believe the main hurdle would be to incorporate dynamic price/contract selection into the overall economic design of the market. Given the multitude of existing crowdsourcing markets and a relative ease of deploying new market designs, we believe this direction is well worth studying.

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.

3.1.1 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 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 $\text{incr}(x)$. 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*. The *size* of a cell is the length of any one side. 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 . Note that anchors are not necessarily candidate contracts.

3.1.2 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:

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

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 Section 3.3).

Lemma 3.1. *If all types satisfy the FOSD assumption and consistent tie-breaking holds, then $\text{width}(C) \leq \text{VirtWidth}(C)$ 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

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.

3.2.3 NOTES ON 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 supremum 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.

3.3 Proof of Lemma 3.1 (virtual width)

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

Claim 3.2. *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

$$(\mathbf{x} - \mathbf{x}') \cdot (\mathbf{F} - \mathbf{F}') \geq 0. \tag{5}$$

Note that if Equation (5) 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 (5) 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 $F_i(\pi|e) > F_i(\pi|e')$ for some outcome $\pi > 0$. Therefore we obtain a contradiction, completing the proof. \square

The proof of Claim 3.2 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 3.3. *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 3.2 rules out the latter possibility.

Define vectors \mathbf{F} and \mathbf{F}' as in the proof of Claim 3.2. 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, consider a composite cell C with anchors $x^+ = x^+(C)$ and $x^- = x^-(C)$, and fix a contract $x \in C$. Since $x^+ \succeq x \succeq x^-$, by Claim 3.3 it follows that $V(x^+) \geq V(x) \geq V(x^-)$ and $P(x^+) \geq P(x) \geq P(x^-)$. Therefore $|U(x) - U(y)| \leq \text{VirtWidth}(C)$. Taking the supremum over all $x \in C$ over, we obtain $\text{width}(C) \leq \text{VirtWidth}(C)$, as claimed.

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.

4.1 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.

Recall that $\text{OPT}(X_{\text{cand}}) = \sup_{x \in X_{\text{cand}}} U(x)$ is 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 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 : $\Delta(x) = \text{OPT}(X_{\text{cand}}) - U(x)$. 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: by definition, a feasible cell is one that contains a candidate contract.

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 Section 6.

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

$$R(T|X_{\text{cand}}) \leq \delta T + O(\log T) \sum_{\epsilon=2^{-j} \geq \delta: j \in \mathbb{N}} \frac{N_{\epsilon \beta_0}(X_\epsilon|X_{\text{cand}})}{\epsilon}. \quad (6)$$

Remark 1. As discussed in Section 2.2, 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 2. 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 (6) 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 $N_\epsilon^{\text{min}}(Y)$ 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

$$N_\epsilon(Y) \leq \lceil \log \frac{1}{\psi} \rceil N_\epsilon^{\text{min}}(Y) \text{ for any } Y \subset X_{\text{cand}} \text{ and } \epsilon \geq 0, \quad (7)$$

where ψ is the smallest size of a feasible cell.⁶ Thus, Equation (6) can be easily restated using the covering numbers $N_\epsilon^{\text{min}}(\cdot)$ instead of $N_\epsilon(\cdot)$.

4.2 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

6. To prove Equation (7), 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 $\epsilon > 0$.

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*:

$$\text{WidthDim}_\alpha = \inf \left\{ d \geq 0 : N_{\epsilon\beta_0}(X_\epsilon | X_{\text{cand}}) \leq \alpha \epsilon^{-d} \text{ for all } \epsilon > 0 \right\}, \alpha > 0. \quad (8)$$

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 (6), 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*

$$R(T | X_{\text{cand}}) \leq O(\alpha \log T) T^{(1+d)/(2+d)}. \quad (9)$$

The width dimension is similar to the “zooming dimension” in the work of Kleinberg et al. (2008) and “near-optimality dimension” in the work on “bandits in metric spaces” (Bubeck et al., 2011a).

4.3 Comparison to Prior Work

Below we compare our results with previous work in non-adaptive discretization and bandits in metric spaces.

4.3.1 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.⁸ For concreteness, and following the prior work (Kleinberg & Leighton, 2003; Kleinberg, 2004; Kleinberg et al., 2008), we use a well-known algorithm `UCB1` (Auer, Cesa-Bianchi, & Fischer, 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 4.3. *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 and δ .

The best known regret bounds for `NonAdaptive` coincide with those in Corollary 4.3 up to poly-logarithmic factors. However, the regret bounds in Theorem 4.1 may be significantly better than the ones in Corollary 4.3. We further discuss this in the next section, in the context of a specific example.

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

4.3.2 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” (Kleinberg et al., 2008; Bubeck et al., 2011a) 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 predefined 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 $f(C_{x,y}) \geq \text{width}(C_{x,y})$, 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 the work of Kleinberg et al. (2008) and Bubeck et al. (2011a). 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:

$$R(T|X_{\text{cand}}) \leq \delta T + O(\log T) \sum_{\epsilon=2^{-j} \geq \delta: j \in \mathbb{N}} \frac{N_{\Omega(\epsilon)}^*(X_\epsilon)}{\epsilon}, \quad (10)$$

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 $\mathcal{D}(x, y) = \text{VirtWidth}(C_{x,y})$. One can use Equation (10) to derive a polynomial regret bound like Equation (9).

For a more precise comparison, we focus on the results in the work of Kleinberg et al. (2008). (The regret bounds in Bubeck et al., 2011a are very similar in spirit, but are stated in terms of a slightly different structure.) The “covering-type” regret bound in the work of Kleinberg et al. (2008) 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 $\mathcal{D}(x, y) = \text{VirtWidth}(C_{x,y})$ 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”:

$$\text{ZoomDim}_\alpha = \inf \left\{ d \geq 0 : N_{\epsilon/8}^*(X_\epsilon) \leq \alpha \epsilon^{-d} \text{ for all } \epsilon > 0 \right\}, \quad \alpha > 0. \quad (11)$$

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 a version of Equation (9) 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.

5. A Special Case: The “High-Low Example”

We apply the machinery in Section 4 on a special case, and we show that **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 (Kleinberg & Leighton, 2003; Badanidiyuru et al., 2012; Singla & Krause, 2013; Badanidiyuru et al., 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 Section 8 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.

In this example, the worker’s type consists of a pair (c_h, θ_h) , where $c_h \geq 0$ is the cost for **high** effort and $\theta_h \in [0, 1]$ is the probability of **high** outcome given **high** effort. Note that dynamic task pricing is equivalent to the special case $\theta_h = 1$.

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 the probability of obtaining **high** outcome given contract x $\Pr[\text{high outcome} | \text{contract } x]$ depends only on $p = x(\text{high}) - x(\text{low})$; denote this probability by $S(p)$. Moreover, $S(p)$ is non-decreasing in p . Therefore:*

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

To bound the discretization error, it is essential that $S(p)$ is non-decreasing in p .

Recall that $X_{\text{cand}}(\psi)$, 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 $x(\text{low}) = 0$, in which case $U(x) = \tilde{U}(p)$. Thus, if an algorithm knows that it is given a high-low example, it can set $x(\text{low}) = 0$, 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 $x(\text{low}) = 0$.

5.1 “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 θ_h is the same for all worker types and the marginal distribution of c_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 $X_{\text{cand}} \subset X$, **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 $X_{\text{cand}} \subset X$. Therefore, **AgnosticZooming** with this X_{cand} has regret $R(T|X_{\text{cand}}) = O(\log T) T^{3/5}$.*

We contrast this with the performance of **NonAdaptive**, parameterized with the natural choice $X_{\text{cand}} = X_{\text{cand}}(\psi)$. 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_{cand} , whereas **NonAdaptive** cannot do better than $R(T|X) = O(T^{3/4})$ for any $X_{\text{cand}} = X_{\text{cand}}(\psi)$, $\psi > 0$.

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

- AgnosticZooming** with $X_{\text{cand}} \supset X_{\text{cand}}(T^{-2/5})$ has regret $R(T|X) = O(T^{3/5} \log T)$.
- NonAdaptive** with $X_{\text{cand}} = X_{\text{cand}}(\psi)$ cannot achieve regret $R(T|X) < o(T^{3/4})$ over all problem instances, for any $\psi > 0$.¹⁰

5.2 Proofs

Proof of Claim 5.1. Consider a contract x with $x(\text{low}) = b$ and $x(\text{high}) = b + p$, and a worker of type (c_h, θ_h) . If the worker exerts **high** effort, she pays cost c_h and receives expected payment $\theta_h(p + b) + (1 - \theta_h)b$, for a total expected payoff $p\theta_h + b - c_h$. Her expected payoff for exerting **low** effort is b . Therefore she will choose to exert **high** effort if and only if $p\theta_h + b - c_h \geq b$, i.e., if $c_h/\theta_h \leq p$, and choose to exert **low** effort otherwise. Therefore

$$\Pr[\text{high outcome} | \text{contract } x] = \mathbb{E}_{(c_h, \theta_h)} [\theta_h \mathbf{1}_{\{c_h/\theta_h \leq p\}}].$$

This is a function of p , call it $S(p)$. Moreover, this is a non-decreasing function simply because the expression inside the expectation is non-decreasing in p .

It trivially follows that $U(x) = S(p)(v - p) - x(\text{low})$.

We can upper-bound the discretization error using a standard approach from the work on dynamic pricing (Kleinberg & Leighton, 2003). Fix discretization granularity $\psi > 0$. For any $\epsilon > 0$, there exists a contract $x^* \in X$ such that $\text{OPT}(X) - U(x^*) < \epsilon$. Round $x^*(\text{high})$

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

and $x^*(\text{low})$ up and down, respectively, to the nearest integer multiple of ψ ; let $x \in X_{\text{cand}}(\psi)$ be the resulting contract. Denoting $p = x(\text{high}) - x(\text{low})$ and $p^* = x^*(\text{high}) - x^*(\text{low})$, we see that $p^* \leq p \leq p^* + 2\psi$. It follows that

$$U(x) \geq U(x^*) - 3\psi \geq \text{OPT}(X) - \epsilon - 3\psi.$$

Since this holds for any $\epsilon > 0$, we conclude that $\text{OPT}(X) - \text{OPT}(X_{\text{cand}}(\psi)) \leq 3\psi$. \square

Proof of Lemma 5.2. To calculate the width dimension, we need to count the number of feasible cells in the increment space which (i) have virtual width larger than or equal to $O(\epsilon)$ and (ii) overlap with X_ϵ , the set of contracts with badness smaller than ϵ .

We first characterize X_ϵ . We use $x_{p,b}$ to denote the contract with $x(\text{high}) = p + b$ and $x(\text{low}) = b$. The benefit of this representation is that, p and b would then be the two axes in the increment space. Let $x_{p^*,0}$ be an optimal contract. Since $U(x_{p,b})$ is strongly concave in p , we know that for any b , there exist constants c_1 and c_2 such that for any $p \in [0, 1]$, $c_1(p^* - p)^2 \leq U(x_{p^*,b}) - U(x_{p,b}) \leq c_2(p^* - p)^2$. Also we know that $U(x_{p^*,b}) = U(x_{p^*,0}) - b$. Therefore.

$$X_\epsilon = \{x_{p,b} : (p - p^*)^2 + b \leq O(\epsilon)\}$$

We can also write it as

$$X_\epsilon = \{x_{p,b} : p^* - \theta_{\text{h}}(\sqrt{\epsilon}) \leq p \leq p^* + \theta_{\text{h}}(\sqrt{\epsilon}) \text{ and } b \leq O(\epsilon)\}$$

Intuitively, X_ϵ contains contracts $\{x_{p,b}\}$ with p not $O(\sqrt{\epsilon})$ away from p^* and b not $O(\epsilon)$ away from $b^* = 0$.

Next we characterize the virtual width of a cell. We use $C_{p,b,d}$ to denote the cell with size d and with anchors $\{x_{p,b}, x_{(p+d),(b+d)}\}$. We can derive the expected payment and value on the two anchors as:

- $P^+(C_{p,b,d}) = (p + d)S(p + d) + b + d$
- $V^+(C_{p,b,d}) = vS(p + d)$
- $P^-(C_{p,b,d}) = pS(p) + b$
- $V^-(C_{p,b,d}) = vS(p)$

By definition, we can get that (we use d_F to represent $S(p + d) - S(p)$ for simplification)

$$\text{VirtWidth}(C_{p,b,d}) = (v + p)d_F + dS(p) + d d_F + d.$$

Now we can count the number of feasible cells with virtual width larger than $\theta_{\text{h}}(\epsilon)$ which overlaps with X_ϵ . Note that since the total number of feasible cells $C_{p,b,d}$ with large d is small, we can treat the number of cells with large d as a constant. Also, for any relevant cell $C_{p,b,d}$, we have $p \approx p^*$. Therefore, we only care about feasible cells $C_{p,b,d}$ with small d and when p is close to p^* .

Since $S(p)$ is Lipschitz, we have $d_F = O(d)$. Therefore, for any relevant cell $C_{p,d}$,

$$\text{VirtWidth}(C_{p,b,d}) = O(d)$$

Given the above two arguments, we know that the number of cells with virtual width larger than ϵ which also overlaps with X_ϵ is $O(\epsilon/\epsilon) \times O(\sqrt{\epsilon}/\epsilon) = O(\epsilon^{-1/2})$. Therefore the width dimension is $1/2$. \square

Proof Sketch of Lemma 5.3(b). Consider a version of `NonAdaptive` that runs an off-the-shelf MAB algorithm `ALG` on candidate contracts $X_{\text{cand}} = X_{\text{cand}}(\psi)$. For `ALG`, the “arms” are the candidate contracts; recall that the arms are randomly permuted before they are given to `ALG`.

Fix $\psi > 0$. It is easy to construct a problem instance with discretization error $\mathbf{Error} \triangleq \text{OPT}(X) - \text{OPT}(X_{\text{cand}}(\psi)) \geq \Omega(\psi)$. Note that X_{cand} contains $N = \Omega(\psi^{-2})$ suboptimal contracts that are suboptimal w.r.t. $\text{OPT}(X_{\text{cand}})$. (For example, all contracts x with $x(\text{low}) > 0$ are suboptimal.)

Fix any problem instance \mathcal{I} of MAB with N suboptimal arms. Using standard lower-bound arguments for MAB, one can show that if one runs `ALG` on a problem instance obtained by randomly permuting the arms in \mathcal{I} , then the expected regret in T rounds is at least $\Omega(\sqrt{NT})$.

Therefore, $R(T|X_{\text{cand}}) \geq \Omega(\sqrt{NT})$. It follows that

$$R(T|X) \geq \Omega(\sqrt{NT}) + \mathbf{Error} \cdot T \geq \Omega(\sqrt{T}/\psi + \psi T) \geq \Omega(T^{3/4}). \quad \square$$

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 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) *$W_t(C) \leq 5 \text{rad}_t(C)$ for each active composite cell C .*

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

Throughout, we say that the algorithm *activates* a cell if this cell is added to the collection of active cells. A cell stays active once it is activated.

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

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

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

Lemma 6.3. *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 $1 - 1/T$.

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

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

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

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

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 .) 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

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

and similarly for $V^-(C)$, $P^+(C)$ and $P^-(C)$. Each of these four statements is proved similarly, using the technique from Lemma 6.3(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 6.3(a), we can establish the following high-probability event:

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

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:¹¹

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

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

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

Therefore, once Equations (17) and (18) hold, we have $|V^+(C) - V_t^+(C)| \leq \mathbf{rad}_t(C)$. This completes the proof of Equation (16). \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, $I(C_t) \geq \text{OPT}(X_{\text{cand}})$ 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 $I_t(C_t^*) \geq U(x^*)$ by the clean execution. If C_t^* is composite then

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

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

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

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

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

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

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

□

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, $\Delta(C) \leq O(\mathbf{rad}_t(C))$ for each round t and each active cell C .*

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

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, $\Delta(C) \leq O(\mathbf{rad}_T(C))$. The claim follows from the definition of \mathbf{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

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

The next result (Lemma 6.8) upper-bounds the right-hand side of Equation (21) 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 $\delta \in (0, 1)$,*

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

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

Claim 6.9. *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.) So, $x \in X_\epsilon$. \square

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

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

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

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

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 6.9(a). Therefore, invoking case (i), we have

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

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 $C_p \in \mathcal{F}_{\Omega(\epsilon)}(X_{2\epsilon})$.

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

7. Simulations

We evaluate the performance of **AgnosticZooming** through simulations. **AgnosticZooming** is compared 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. In both algorithms, in each round a numerical score (called *index*) is computed for each arm, and an arm with a maximal index is chosen. In **UCB1**, the index of an arm is a high-confidence upper bound on the expected reward of this arm. In Thompson Sampling, the index is sampled independently from the Bayesian posterior distribution of the arm’s expected reward.

7.1 Setup

We consider a generalized version of the high-low example from Section 5 in which the requester’s value of the **low** outcome could be nonzero. In the results reported below, we set the requester’s values to $V(\mathbf{high}) = 1$ and $V(\mathbf{low}) = .3$, and the probability of obtaining **high** outcome given **high** effort to $\theta_h = .8$. While we do not explicitly report the results, we additionally tried a wide range of alternative values of $V(\mathbf{high})$, $V(\mathbf{low})$, and θ_h and found that they were similar qualitatively. Intuitively, varying the requester’s values and θ_h only changes which contracts the algorithms converge to (that is, the optimal arms), but does not impact the problem structure; the width dimension is the same for all settings.

In this generalized high-low example, the worker’s type is characterized by the cost c_h for **high** effort. We consider three supply distributions:

- *Uniform*: c_h is uniformly distributed on $[0, 1]$.
- *Homogeneous*: c_h is the same for every worker.
- *Two-type*: c_h is uniformly distributed over two values, c'_h and c''_h .

These first two distributions represent the extreme cases in which workers are either extremely homogeneous or extremely diverse. The third distribution is one way to get at the middle ground. For each distribution, we run each algorithm 100 times.¹² For the Homogeneous Supply Distribution, c_h is drawn uniformly at random from $[0, 1]$ for each run. For the Two-Type Supply Distribution, c'_h and c''_h are drawn independently and uniformly from $[0, 1]$ on each run.

For both **UCB1** and **AgnosticZooming**, we replace the logarithmic confidence terms with small constants. We find this beneficial in practice for both algorithms, which is consistent with prior work (Radlinski, Kleinberg, & Joachims, 2008; Slivkins, Radlinski, & Gollapudi, 2013). For both algorithms, we tried several different constants and found that performance is not very sensitive to the particular constant used as long as it is on the order of 1. In the results reported below, we set these confidence terms equal to 1. For **UCB1**, this means that if a given arm a has been played n_a times, its index is the average reward plus $1/\sqrt{n_a}$. For **AgnosticZooming**, it means that $\mathbf{rad}_t(\cdot) = \sqrt{1/n_t(\cdot)}$.

All three algorithms are run with $X_{\text{cand}} = X_{\text{cand}}(\psi)$, where $\psi > 0$ is a parameter specifying the granularity of discretization.

12. The standard errors in all plots are in the order of 0.001 or less. (Note that each point is not only the average of 100 runs but also the average of all previous rounds.)

7.2 Overview of the Results.

Across all simulations, **AgnosticZooming** either outperforms or nearly matches **NonAdaptive**. Its performance does not appear to suffer from the 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 of contract space. When ψ is large, **AgnosticZooming** converges more slowly than **NonAdaptive**, but eventually achieves similar performance. Further, we find that **AgnosticZooming** with small ψ performs well compared to **NonAdaptive** with larger ψ . In particular, it is 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 near-optimal performance. However, in real applications approximately optimal ψ may be difficult to compute, and the T may not be known in advance. **AgnosticZooming** performs consistently well with a wide range of ψ and therefore does not require prior knowledge of T or careful tuning of ψ .

7.3 Detailed Results

For each algorithm, we compute the time-averaged cumulative utility after T rounds given granularity ψ , denoted $\widehat{U}(T, \psi)$, for various values of T and ψ .

First, we fix the time horizon T to 5,000 rounds, and study how $\widehat{U}(T, \psi)$ changes with ψ . The results are shown in Figure 1. We observe that **AgnosticZooming** either closely matches or outperforms both versions of **NonAdaptive** across all supply distributions and all values of ψ . **AgnosticZooming** performs consistently well with different ψ while the performance of both versions of **NonAdaptive** decreases rapidly when ψ is small.

Second, we study how the three algorithms perform over time. Specifically, we plot T vs. $\widehat{U}(T, \psi)$, for three values of ψ , namely 0.02, 0.08, and 0.2. Since setting $\psi = 0.08$ is close to optimal in our examples, these values of ψ represent, respectively, values that are too small, adequate, and too large. The results are shown in Figure 2. For small values of ψ , **AgnosticZooming** quickly “zooms in” on promising regions of the contract space, leading to faster converge than the alternatives. However, when ψ is large, **AgnosticZooming** converges more slowly, but eventually achieves similar performance. In this regime, **AgnosticZooming** does not reap the benefits of adaptive discretization because the mesh of candidate contracts is too sparse, but still suffers the overhead. This suggests that if the time horizon T is known in advance and one can optimize ψ given this T , then **NonAdaptive** can achieve near-optimal performance. **AgnosticZooming** performs consistently with different choices of ψ and therefore does not require either the prior knowledge of T or the careful tuning of ψ .

To further demonstrate the benefit of not having to know T or tune ψ , we compare the performance of **AgnosticZooming** with small ψ against that of **NonAdaptive** with different values of ψ . For each algorithm and each choice of ψ , we plot T vs. $\widehat{U}(T, \psi)$. See Figure 3. We only show the results for the Uniform Supply Distribution since the results for other distributions are very similar. Additionally, we omit the results for Thompson Sampling since **UCB1** performed better in these experiments.¹³ We find that for small T , **AgnosticZooming**

13. We conjecture that this is because we replaced the logarithmic confidence term in **UCB1** with 1.

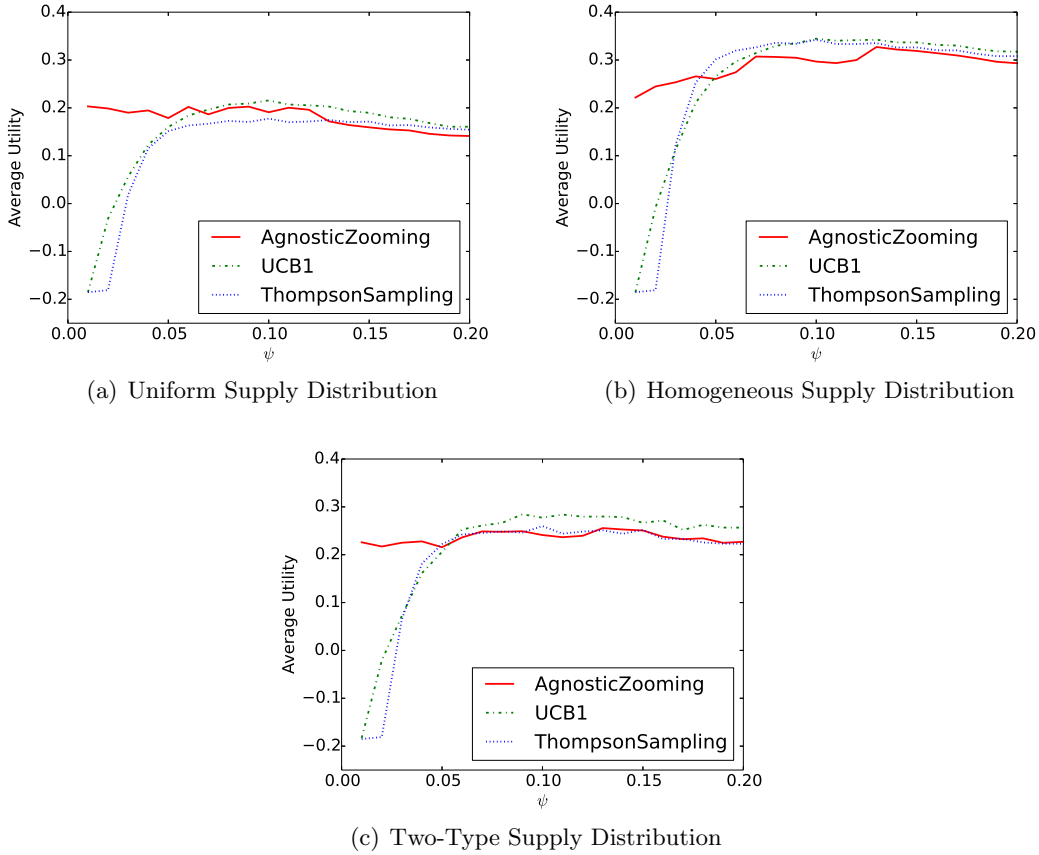


Figure 1: The requester’s average per-round utility after 5,000 rounds vs. the choice of initial discretization ψ .

with small ψ converges nearly as fast as `NonAdaptive` with larger ψ . When T is large, `AgnosticZooming` with small ψ matches `NonAdaptive` with the optimal ψ .

Finally, in Figure 4, we confirm the intuition that $\text{OPT}(X_{\text{cand}}(\psi))$ decreases with the granularity ψ . To this end, we run `AgnosticZooming` for 50,000 rounds (so the algorithm has time to nearly converge to the optimal contract), and examine the average utility over the last 5,000 rounds. As expected, we see that the average requester utility achievable when ψ is small is significantly higher than the utility achievable when ψ is larger.

Our simulation results suggest that `AgnosticZooming` performs well across different supply distributions and different settings of ψ , not requiring careful tuning of the algorithm parameters. Given that the smaller the value of ψ , the better the payoff of the optimal contract $\text{OPT}(X_{\text{cand}}(\psi))$, `AgnosticZooming` with small ψ is a good algorithm for a variety of settings.

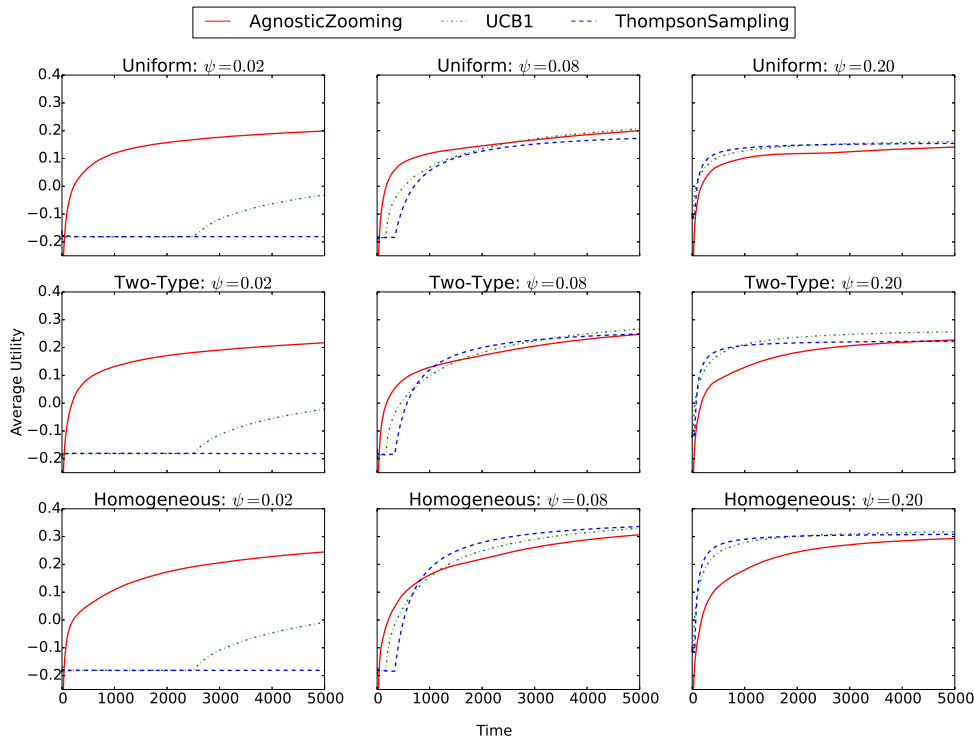


Figure 2: The requester’s average per-round utility over time under different supply distributions and discretization sizes.

8. Application to Dynamic Task Pricing

We discuss *dynamic task pricing*, which can be seen as the special case of dynamic contract design in which there is exactly one non-null outcome. We identify an important family of problem instances for which AgnosticZooming out-performs NonAdaptive.

8.1 Background

The dynamic task pricing 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.

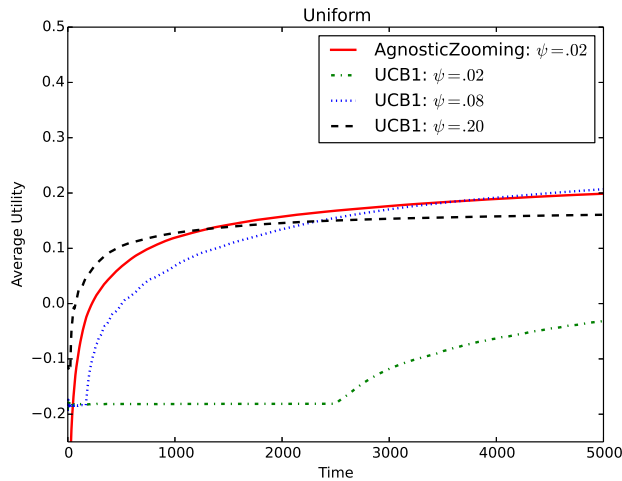


Figure 3: The requester’s average per-round utility over time using **AgnosticZooming** with small ψ compared with **NonAdaptive** with three different values of ψ .

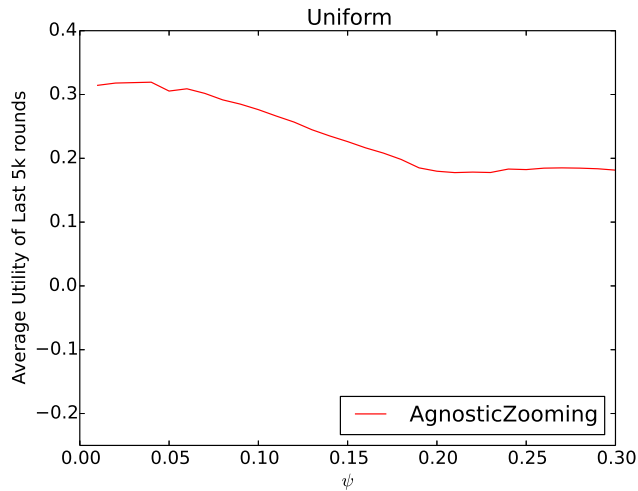


Figure 4: Average requester utility over the last 5,000 rounds in a 50,000-round run of **AgnosticZooming** for different values of ψ .

Dynamic task pricing can be seen as 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).

One crucial simplification compared to the full generality of dynamic contract design is that the discretization error can now be easily bounded from above: ¹⁴

$$\text{OPT}(X) - \text{OPT}(X_{\text{cand}}(\psi)) \leq \psi \quad \text{for each } \psi > 0.$$

Worst-case regret bounds are implicit in prior work on dynamic inventory-pricing (Kleinberg & Leighton, 2003).¹⁵ Let $\text{NonAdaptive}(\psi)$ denote algorithm `NonAdaptive` with $X_{\text{cand}} = X_{\text{cand}}(\psi)$. Then, by the analysis in the work of Kleinberg and Leighton (2003), $\text{NonAdaptive}(\psi)$ 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.

Further, it is a folklore result that $\text{NonAdaptive}(\psi)$ achieves regret $R(T) = \tilde{O}(T^{2/3})$ if and only if $\psi = \tilde{\Theta}(T^{-1/3})$. (We sketch a lower-bounding example in the proof of Lemma 8.4, to make the paper more self-contained.)

8.2 Preliminaries

Each contract is summarized by a single number: the offered price p for the non-null outcome. Let $F(p)$ be the probability of a worker accepting a task at price p , and let $U(p) = F(p)(v - p)$ be the corresponding expected utility of the algorithm.

Note that all contracts are trivially monotone and any optimal contract is bounded without loss of generality. It follows that $\text{OPT}(X) = \sup_{p \geq 0} U(p)$, the optimal expected utility over all possible prices.

A cell C is just a price interval $C = [p, p'] \subset [0, 1]$, and its virtual width is

$$\text{VirtWidth}(C) = (v F(p') - p F(p)) - (v F(p) - p' F(p')).$$

8.3 Our Results: The General Case

We will be using `AgnosticZooming` with $X_{\text{cand}} = X$.

First, let us prove that this is a reasonable choice in the worst case: namely, that we achieve the optimal $\tilde{O}(T^{2/3})$ regret.

Lemma 8.1. *Consider the dynamic task pricing problem. `AgnosticZooming` with $X_{\text{cand}} = X$ achieves regret $O(T^{2/3} \log T)$.*

Proof Sketch. Fix $\epsilon > 0$. The key observation is that if $\text{VirtWidth}(C) \geq \epsilon$ then either $p' - p \geq \frac{\epsilon}{4}$, or $F(p') - F(p) \geq \frac{\epsilon}{4}$. Call C a *red* cell if the former happens, and *blue* cell otherwise. Therefore in any collection of mutually disjoint cells of virtual width $\geq \epsilon$ there can be at most $O(\frac{1}{\epsilon})$ red cells and at most $O(\frac{1}{\epsilon})$ blue cells, hence at most $O(\frac{1}{\epsilon})$ cells total. It follows that there can be at most $O(\frac{1}{\epsilon})$ active cells of virtual width $\geq \epsilon$.

So, in the notation of Theorem 4.1 we have $N_\epsilon(\cdot) \leq O(\frac{1}{\epsilon})$. It follows that the width dimension is at most 1, which in turn implies the desired regret bound. \square

14. Recall that $X_{\text{cand}}(\psi)$ denotes the set of all prices in $[0, 1]$ that are integer multiples of a given $\psi > 0$; call this set the *additive ψ -mesh*.

15. The algorithmic result for dynamic task pricing is an easy modification of the analysis in the work of Kleinberg and Leighton (2003) for dynamic inventory-pricing. The lower bound in the work of Kleinberg and Leighton can also be “translated” from dynamic inventory-pricing to dynamic task pricing without introducing any new ideas. We omit the details from this version.

8.4 Our Results: “Nice” Problem Instances

We focus on problem instances with piecewise-uniform costs and bounded density. Formally, we say that an instance of dynamic task pricing 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 . Using the full power of Theorem 4.1, we obtain the following regret bound.

Theorem 8.2. *Consider the dynamic task pricing problem with k -piecewise-uniform costs and λ -bounded density, for some absolute constants $k \in \mathbb{N}$ and $\lambda > 1$. **AgnosticZooming** with $X_{\text{cand}} = X$ achieves regret $R(T) = \tilde{O}(T^{3/5})$.*

Proof Sketch. Since the supply distribution has density at most λ , it follows that $F(\cdot)$ 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 (6). \square

8.5 Comparison with NonAdaptive

Consider **NonAdaptive**(ψ_0), where $\psi_0 = \tilde{\Theta}(T^{-1/3})$ is the granularity required for the optimal worst-case performance. Call a problem instance *nice* if it has 2-piecewise-uniform costs and λ -bounded density, for some sufficiently large absolute constant λ ; say $\lambda = 4$ for concreteness. We claim that **AgnosticZooming** outperforms **NonAdaptive**(ψ_0) on the “nice” problem instances.

Lemma 8.3. ***NonAdaptive**(ψ_0) achieves regret $R(T) = \Omega(T^{2/3})$ in the worst case over all “nice” problem instances.*

Proof Sketch. 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 $X_{\text{cand}}(\psi_0)$. 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 $X_{\text{cand}}(\psi_0)$ is at least $\Omega(\psi_0)$, which implies regret $\Omega(\psi_0 T)$. \square

8.6 Lower Bound for NonAdaptive

We provide a specific lower-bounding example for the worst-case performance of $\text{NonAdaptive}(\psi)$, for an arbitrary $\psi > 0$. Let \mathcal{F} be the family of all problem instances with k -piecewise-uniform costs and λ -bounded density, for all $k \in \mathbb{N}$ and $\lambda = 4$.

Lemma 8.4. *Let $R_\psi(T)$ be the maximal regret of $\text{NonAdaptive}(\psi)$ over all problem instances in \mathcal{F} . Then $R_\psi(T) = \Omega(\psi T + \sqrt{T/\psi}) \geq \Omega(T^{2/3})$.*

Proof Sketch. For piecewise-uniform costs, we have $F(0) = 0$ and $F(p) = 1$. Assume that the principal derives value $v = 1$ from each item. Then the expected utility from price p is $U(p) = F(p)(1 - p)$.

Fix $\psi > 0$. 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 $F(p)$ for $p \in \mathcal{P} \cup \{0, 1, p^*\}$. For the rest of the prices, define $F(\cdot)$ via linear interpolation. This completes the description of the problem instance.

We show that X_ψ consists of $N = \Omega(\frac{1}{\psi})$ candidate contracts. Therefore, using standard lower-bounding arguments for MAB, we obtain $R(T|X_{\text{cand}}) \geq \Omega(\sqrt{TN}) = \Omega(\sqrt{T/\psi})$. Further, we show that the discretization error is at least $\Omega(\psi)$, implying that $R(T) \geq R(T|X_{\text{cand}}) + \Omega(\psi T)$. \square

9. 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.

9.1 Contract Theory

Our model can be viewed as an extension of the classic principal-agent model from contract theory (Laffont & Martimort, 2002). 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 (Laffont & Martimort, 2002). 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 (2002) 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, Feldman, and Nisan (2006) 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, Nair, and Daljord (2012) 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 (Sannikov, 2008; Williams, 2009; Sannikov, 2012), 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 exerted 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 (2006) study 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 discretization, gradient ascent, and Bayesian update approaches to the problem. Our goal is to provide an algorithm with nice theoretical guarantees.

Bohren and Kravitz (2013) study the setting when the outcome is unverifiable. To address this issue, they propose to assign a bundle of tasks to each worker. To verify the outcome, each task in the bundle is chosen as a verifiable task with some non-trivial probability. A verifiable task can either be a gold standard task with known answer or a task which is assigned to multiple workers for verification. The payment for a task bundle is then conditional only on the outcome of verified tasks. In our setting, we assume the task outcome is verifiable. We can relax this assumption by adopting similar approaches.

9.2 Incentives in Crowdsourcing Systems

Researchers have recently begun to examine the design of incentive mechanisms to encourage high-quality work in crowdsourcing systems. Jain, Chen, and Parkes (2012) 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 (2011, 2013) and Ghosh and McAfee (2011) 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, Zhang, Vaughan, and van der Schaar (2012) and Zhang and van der Schaar (2012) 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 monetary 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 (Badanidiyuru et al., 2012; Singer & Mittal, 2013). 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 (2009) study how workers react to changes of performance-independent financial incentives. In their study, increasing financial incentives increases the number of tasks workers complete, but not the quality of their output. Yin, Chen, and Sun (2013) 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. Horton and Chilton (2010) run experiments to estimate workers’ reservation wage for completing tasks. They show that many workers respond rationally to offered contracts, whereas some of the workers appeared to have some “target payment” in mind.

Some recent research studies the effects of performance-based payments (PBPs). Harris (2011) runs MTurk experiments on resume screening, where workers can get a bonus if they perform well. He concludes that the quality of work is better with PBPs than with uniform payments. Yin et al. (2013) show that varying the magnitude of the bonus does not have much effect in certain settings. Ho et al. (2015) perform a more comprehensive set of experiments aimed at determining whether, when, and why PBPs increase the quality of submitted work. Their results suggest that PBPs can increase quality on tasks for which increased time or effort leads to higher quality work. Their results also suggest that workers may interpret a contract as performance-based even if it is not stated as such (since requesters always have the option to reject work). Based on this evidence, they propose a new model of worker behavior that extends the principal-agent model to explicitly reflect workers’ subjective beliefs about their likelihood of being paid.

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 (namely, as long as Lemma 3.1 holds). We note that our results hold under the generalized worker model proposed by Ho et al. (2015), which is consistent with their experimental evidence as discussed above.

9.3 Sequential Decision Problems

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

MAB have been studied since 1933 (Thompson, 1933) 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 the work of Cesa-Bianchi and Lugosi (2006) or Bubeck and Cesa-Bianchi (2012) for background on prior-independent MAB, and to the work of Gittins, Glazebrook, and Weber (2011) 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 (Lai & Robbins, 1985; Auer et al., 2002; Bubeck & Cesa-Bianchi, 2012). 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* (Kleinberg et al., 2008), 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 this distance function (Agrawal, 1995; Kleinberg, 2004; Auer et al., 2007; Kleinberg et al., 2008; Bubeck et al., 2011a; Slivkins, 2014). Most related to this paper is the idea of adaptive discretization which is often used in this setting (Kleinberg et al., 2008; Bubeck et al., 2011a; Slivkins, 2014), and particularly the *zooming algorithm* (Kleinberg et al., 2008; Slivkins, 2014). 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 (Kocsis & Szepesvari, 2006; Munos & Coquelin, 2007; Pandey et al., 2007; Slivkins, 2011; Bull, 2013). In particular, Slivkins (2011) and Bull (2013) explicitly reconstruct (the relevant parts of) the metric space defined by the taxonomy. In a different direction, Bubeck, Stoltz, and Yu (2011b) study a version of Lipschitz MAB where the Lipschitz constant is not known, and essentially recover the performance of `NonAdaptive` for this setting.

In MAB with partial monitoring (Audibert & Bubeck, 2010; Bartók, Foster, Pál, Rakhlin, & Szepesvári, 2014; Antos, Bartók, Pál, & Szepesvári, 2013), in each round the algorithm receives auxiliary feedback about rewards in this round (along with the reward of the chosen arm), and the goal is to take advantage of this auxiliary feedback. Dynamic task pricing can be cast in this framework: if a given price p is accepted, then any higher price would be too, and if it is rejected, then any lower price would be. However, we are not aware of any way to link dynamic task pricing to existing results on partial monitoring

via this connection. The general version of dynamic contract design does not appear to fit the partial monitoring framework, essentially due to moral hazard.

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 extensively studied in operations research, typically in a Bayesian setting; see the work of den Boer (2015) for a thorough literature review. The study of prior-independent, non-parameterized formulations has been initiated in the work of Blum, Kumar, Rudra, and Wu (2003) and Kleinberg and Leighton (2003) and continued by several others (Besbes & Zeevi, 2009; Babaioff, Dughmi, Kleinberg, & Slivkins, 2015; Besbes & Zeevi, 2012; Wang, Deng, & Ye, 2014; Badanidiyuru et al., 2013; Badanidiyuru, Langford, & Slivkins, 2014). Further, Badanidiyuru et al. (2012) and Singla and Krause (2013) studied the version in which the principal *buys* items, or equivalently commissions tasks; we call this version *dynamic task pricing*. Modulo budget constraints, this is essentially the special case of our setting where in each round a worker is offered the chance to perform a task at a specified price, and can either accept or reject this offer. In particular, the worker’s strategic choice is directly observable. More general settings have been studied (Badanidiyuru et al., 2013, 2014; Agrawal & Devanur, 2014; Agrawal, Devanur, & Li, 2015).¹⁶ However, all this work (after the initial papers, see Blum et al., 2003 and Kleinberg & Leighton, 2003) 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.

10. 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 treat this problem as a multi-armed bandit problem, design an algorithm for this problem, and derive regret bounds which compare favorably to prior work. Our main conceptual contribution, aside from identifying the model, 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. The generality and the shortcomings of our model are discussed in Section 2.2.

We believe that the dynamic contract design problem deserves further study, in several directions that we outline below.

1. It is not clear whether our provable results can be improved, perhaps using substantially different algorithms and relative to different problem-specific structures. In particular, one needs to establish *lower bounds* in order to argue about optimality; no lower bounds for dynamic contract design are currently known.
2. Our adaptive discretization approach may be fine-tuned to improve its performance in practice. In particular, the definition of the “index” $I_t(C)$ of a given feasible cell C

16. The papers by Badanidiyuru et al. (2014) and Agrawal and Devanur (2014) are concurrent and independent work with respect to the conference publication of this paper, and the work of Agrawal et al. (2015) is subsequent work.

may be re-defined in several different ways. First, it can use the information from C in a more sophisticated way, similar to the more sophisticated indices for the basic K -armed bandit problem; for example, see the work of Garivier and Cappé (2011). Second, the index can incorporate information from other cells. Third, it can be defined in a “smoother”, probabilistic way, e.g., as in Thompson Sampling (Thompson, 1933).

3. Deeper insights into the structure of the (static) principal-agent problem are needed, primarily in order to optimize the choice of X_{cand} , the set of candidate contracts. The most natural target here is the uniform mesh $X_{\text{cand}}(\epsilon)$. To optimize the granularity ϵ , one needs to upper-bound the discretization error $\text{OPT}(X_{\text{cand}}) - \text{OPT}(X_{\text{cand}}(\epsilon))$ in terms of some function $f(\epsilon)$ such that $f(\epsilon) \rightarrow 0$ as $\epsilon \rightarrow 0$. The first-order open question is to resolve whether this can be done in the general case, or provide a specific example when it cannot. A related open question concerns the effect of increasing the granularity: upper-bound the difference $\text{OPT}(X_{\text{cand}}(\epsilon)) - \text{OPT}(X_{\text{cand}}(\epsilon'))$, $\epsilon > \epsilon' > 0$, in terms of some function of ϵ and ϵ' . Further, it is not known whether the optimal mesh of contracts is in fact a uniform mesh.

Also of interest is the effect of restricting our attention to monotone contracts. While we prove that monotone contracts may not be optimal (Appendix A), the significance of this phenomenon is unclear. One would like to characterize the scenarios when restricting to monotone contracts is alright (in the sense that the best monotone contract is as good, or not much worse, than the best contract), and the scenarios when this restriction results in a significant loss. For the latter scenarios, different algorithms may be needed.

4. A much more extensive analysis of special cases is in order. Our general results are difficult to access (which appears to be an inherent property of the general problem), so the most immediate direction for special cases is deriving lucid corollaries from the current regret bounds. In particular, it is desirable to optimize the choice of candidate contracts. Apart from “massaging” the current results, one can also design improved algorithms and derive specialized lower bounds. Particularly appealing special cases concern supply distributions that are mixtures of a small number of types, and supply distributions that belong to a (simple) parameterized family with unknown parameter.

Going beyond our current model, a natural direction is to incorporate a budget constraint, extending the corresponding results on dynamic task pricing. The main difficulty for such settings is that a *distribution* over two contracts may perform much better than any fixed contract; see the work of Badanidiyuru et al. (2013) for discussion. Effectively, an algorithm needs to optimize over the distributions. As a first step, one can use non-adaptive discretization in conjunction with the general algorithms for bandits with budget constraints, sometimes called “bandits with knapsacks” (Badanidiyuru et al., 2013; Agrawal & Devanur, 2014). However, it is not clear how to choose an optimal mesh of contracts (as we discussed throughout the paper), and this mesh is not likely to be uniform (because it is not uniform for the special case of dynamic task pricing with a budget; see Badanidiyuru et al., 2013 for discussion). The eventual target in this research direction is to marry adaptive discretization and the techniques from prior work on “bandits with knapsacks.”

Acknowledgments

We thank the anonymous reviewers for their useful comments. 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.

Appendix A. 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

$$0.5(v(2) - x(2) + v(3) - x(3)). \quad (23)$$

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

$$0.5x(1) \leq 0.5x(2) - c(e_h). \quad (24)$$

So to find the contract that maximizes the requester’s expected value when workers choose high effort, we want to maximize Equation 23 subject to the constraint in Equation 24. Since $x(3)$ doesn’t appear in Equation 24, we can set it to 0 to maximize Equation 23.

Since $x(1)$ does not appear in Equation 23, we can set $x(1) = 0$ to make Equation 24 as easy as possible to satisfy. We can then see that the optimal occurs when $x(2) = 2c(e_h)$.

Plugging this contract x into Equation 23, 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.

References

- Agrawal, R. (1995). The continuum-armed bandit problem. *SIAM J. Control and Optimization*, 33(6), 1926–1951.
- Agrawal, S., & Devanur, N. R. (2014). Bandits with concave rewards and convex knapsacks. In *15th ACM Conf. on Economics and Computation (EC)*.
- Agrawal, S., Devanur, N. R., & Li, L. (2015). Contextual bandits with global constraints and objective.. Technical report, arXiv:1506.03374.
- Antos, A., Bartók, G., Pál, D., & Szepesvári, C. (2013). Toward a classification of finite partial-monitoring games. *Theor. Comput. Sci.*, 473, 77–99.
- Audibert, J., & Bubeck, S. (2010). Regret Bounds and Minimax Policies under Partial Monitoring. *J. of Machine Learning Research (JMLR)*, 11, 2785–2836.
- Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem.. *Machine Learning*, 47(2-3), 235–256.
- Auer, P., Ortner, R., & Szepesvári, C. (2007). Improved Rates for the Stochastic Continuum-Armed Bandit Problem. In *20th Conf. on Learning Theory (COLT)*, pp. 454–468.
- Babaioff, M., Dughmi, S., Kleinberg, R. D., & Slivkins, A. (2015). Dynamic pricing with limited supply. *ACM Trans. on Economics and Computation*, 3(1), 4.
- Babaioff, M., Feldman, M., & Nisan, N. (2006). Combinatorial agency. In *7th ACM Conf. on Electronic Commerce (EC)*.
- Badanidiyuru, A., Kleinberg, R., & Singer, Y. (2012). Learning on a budget: posted price mechanisms for online procurement. In *13th ACM Conf. on Electronic Commerce (EC)*, pp. 128–145.
- Badanidiyuru, A., Kleinberg, R., & Slivkins, A. (2013). Bandits with knapsacks. In *54th IEEE Symp. on Foundations of Computer Science (FOCS)*.
- Badanidiyuru, A., Langford, J., & Slivkins, A. (2014). Resourceful contextual bandits. In *27th Conf. on Learning Theory (COLT)*.
- Bartók, G., Foster, D. P., Pál, D., Rakhlin, A., & Szepesvári, C. (2014). Partial monitoring - classification, regret bounds, and algorithms. *Math. Oper. Res.*, 39(4), 967–997.
- Besbes, O., & Zeevi, A. (2009). Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, 57, 1407–1420.
- Besbes, O., & Zeevi, A. J. (2012). Blind network revenue management. *Operations Research*, 60(6), 1537–1550.

- Blum, A., Kumar, V., Rudra, A., & Wu, F. (2003). Online learning in online auctions. In *14th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pp. 202–204.
- Bohren, J. A., & Kravitz, T. (2013). Incentives for spot market labor when output is unverifiable. Working paper.
- Bubeck, S., & Cesa-Bianchi, N. (2012). Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Foundations and Trends in Machine Learning*, 5(1), 1–122.
- Bubeck, S., Munos, R., Stoltz, G., & Szepesvari, C. (2011a). Online Optimization in X-Armed Bandits. *J. of Machine Learning Research (JMLR)*, 12, 1587–1627.
- Bubeck, S., Stoltz, G., & Yu, J. Y. (2011b). Lipschitz bandits without the lipschitz constant. In *22nd Intl. Conf. on Algorithmic Learning Theory (ALT)*, pp. 144–158.
- Bull, A. D. (2013). Adaptive-treed bandits. Tech. rep. 1302.2489, arxiv.org.
- Cesa-Bianchi, N., & Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge Univ. Press.
- Conitzer, V., & Garera, N. (2006). Online learning algorithms for online principal-agent problems (and selling goods online). In *International Conference on Machine Learning (ICML)*.
- den Boer, A. V. (2015). Dynamic pricing and learning: Historical origins, current research, and new directions. *Surveys in Operations Research and Management Science*. Forthcoming.
- Dudik, M., Hsu, D., Kale, S., Karampatziakis, N., Langford, J., Reyzin, L., & Zhang, T. (2011). Efficient optimal learning for contextual bandits. In *27th Conf. on Uncertainty in Artificial Intelligence (UAI)*.
- Garivier, A., & Cappé, O. (2011). The KL-UCB Algorithm for Bounded Stochastic Bandits and Beyond. In *24th Conf. on Learning Theory (COLT)*.
- Ghosh, A., & Hummel, P. (2011). A game-theoretic analysis of rank-order mechanisms for user-generated content. In *12th ACM Conf. on Electronic Commerce (EC)*.
- Ghosh, A., & Hummel, P. (2013). Learning and incentives in user-generated content: Multi-armed bandits with endogenous arms. In *Proc. 4th Conference on Innovations in Theoretical Computer Science (ITCS)*.
- Ghosh, A., & McAfee, P. (2011). Incentivizing high-quality user-generated content. In *20th Intl. World Wide Web Conf. (WWW)*.
- Gittins, J., Glazebrook, K., & Weber, R. (2011). *Multi-Armed Bandit Allocation Indices*. John Wiley & Sons.
- Harris, C. G. (2011). You’re hired! an examination of crowdsourcing incentive models in human resource tasks. In *CSDM*.
- Ho, C., Slivkins, A., Suri, S., & Vaughan, J. W. (2015). Incentivizing high quality crowd-work. In *24th Intl. World Wide Web Conf. (WWW)*.
- Ho, C.-J., Zhang, Y., Vaughan, J. W., & van der Schaar, M. (2012). Towards social norm design for crowdsourcing markets. In *HCOMP*.

- Horton, J. J., & Chilton, L. B. (2010). The labor economics of paid crowdsourcing. In *11th ACM Conf. on Electronic Commerce (EC)*.
- Jain, S., Chen, Y., & Parkes, D. (2012). Designing incentives for online question-and-answer forums. *Games and Economic Behavior*.
- Kleinberg, R. (2004). Nearly tight bounds for the continuum-armed bandit problem. In *18th Advances in Neural Information Processing Systems (NIPS)*.
- Kleinberg, R., & Leighton, T. (2003). 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)*, pp. 594–605.
- Kleinberg, R., Slivkins, A., & Upfal, E. (2008). Multi-armed bandits in metric spaces. In *40th ACM Symp. on Theory of Computing (STOC)*, pp. 681–690.
- Kleinberg, R. D., & Leighton, F. T. (2003). The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *IEEE Symp. on Foundations of Computer Science (FOCS)*.
- Kocsis, L., & Szepesvari, C. (2006). Bandit Based Monte-Carlo Planning. In *17th European Conf. on Machine Learning (ECML)*, pp. 282–293.
- Laffont, J.-J., & Martimort, D. (2002). *The Theory of Incentives: The Principal-Agent Model*. Princeton University Press.
- Lai, T. L., & Robbins, H. (1985). Asymptotically efficient Adaptive Allocation Rules. *Advances in Applied Mathematics*, 6, 4–22.
- Levy, A., & Vukina, T. (2002). Optimal linear contracts with heterogeneous agents. In *European Review of Agricultural Economics*.
- Mason, W., & Watts, D. (2009). Financial incentives and the “performance of crowds”. In *HCOMP*.
- Misra, S., Nair, H. S., & Daljord, O. (2012). Homogenous contracts for heterogeneous agents: Aligning salesforce composition and compensation. Working Paper.
- Munos, R., & Coquelin, P.-A. (2007). Bandit algorithms for tree search. In *23rd Conf. on Uncertainty in Artificial Intelligence (UAI)*.
- Pandey, S., Agarwal, D., Chakrabarti, D., & Josifovski, V. (2007). Bandits for Taxonomies: A Model-based Approach. In *SIAM Intl. Conf. on Data Mining (SDM)*.
- Radlinski, F., Kleinberg, R., & Joachims, T. (2008). Learning diverse rankings with multi-armed bandits. In *25th Intl. Conf. on Machine Learning (ICML)*, pp. 784–791.
- Sannikov, Y. (2008). A continuous-time version of the principal-agent problem. In *The Review of Economics Studies*.
- Sannikov, Y. (2012). Contracts: The theory of dynamic principal-agent relationships and the continuous-time approach. In *10th World Congress of the Econometric Society*.
- Singer, Y., & Mittal, M. (2013). Pricing mechanisms in crowdsourcing markets. In *22nd Intl. World Wide Web Conf. (WWW)*.
- Singla, A., & Krause, A. (2013). Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *22nd Intl. World Wide Web Conf. (WWW)*.

- Slivkins, A. (2011). Multi-armed bandits on implicit metric spaces. In *25th Advances in Neural Information Processing Systems (NIPS)*.
- Slivkins, A. (2014). Contextual bandits with similarity information. *J. of Machine Learning Research (JMLR)*, 15(1), 2533–2568. Preliminary version in *COLT 2011*.
- Slivkins, A., Radlinski, F., & Gollapudi, S. (2013). Ranked bandits in metric spaces: Learning optimally diverse rankings over large document collections. *J. of Machine Learning Research (JMLR)*, 14(Feb), 399–436. Preliminary version in *27th ICML*, 2010.
- Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples.. *Biometrika*, 25(3-4), 285–294.
- Wang, Z., Deng, S., & Ye, Y. (2014). Close the gaps: A learning-while-doing algorithm for single-product revenue management problems. *Operations Research*, 62(2), 318–331.
- Williams, N. (2009). On dynamic principal-agent problems in continuous time. Working Paper.
- Yin, M., Chen, Y., & Sun, Y.-A. (2013). The effects of performance-contingent financial incentives in online labor markets. In *AAAI*.
- Zhang, Y., & van der Schaar, M. (2012). Reputation-based incentive protocols in crowdsourcing applications. In *Infocom*.