OPTIMALLY PROCURING HOMOGENEOUS GOODS FROM MULTIPLE AGENTS WITH PRIVATE COSTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Procuring data for machine learning often requires compensating providers whose costs are privately known and vary with the quantity supplied. Motivated by this, we study how a principal procures *homogeneous goods* (*i.e.*, her value depends only on the total amount procured) using payment menus that specify compensation for each production level. Each agent produces at most M units at a cost drawn independently from a known distribution over K types. The principal wishes to maximize her expected value less payments.

We first study the single-agent setting, establishing NP-hardness via a reduction from revenue-optimal unit-demand pricing. We then give an exact algorithm running in time $(M+1)^K \mathrm{poly}(K, M)$: it enumerates candidate production profiles across the K types and solves a linear program for the optimal menu inducing each. Under an ordering condition on the types' cost curves (satisfied by linear costs as a special case), we obtain an exact $\mathrm{poly}(K, M)$-time algorithm.

In the multi-agent setting, posting one menu to all agents simultaneously fails: to avoid overproducing, the principal must equally distribute the task among agents; each agent suffers from high startup costs due to economy of scale. We therefore study a sequential model in which agents arrive over T rounds and the principal's value depends on their total production. With *adaptive* menus, where the principal may revise the menu on each round, an exact dynamic program over single-agent subproblems runs in time $(M+1)^K \mathrm{poly}(K, M, T)$, and in $\mathrm{poly}(K, M, T)$ time under the ordering condition. With *static* menus, where the principal commits to a fixed menu but may close the market at any round, a tree representation of candidate menus yields a $(K+1)^{K-1}(M+1)^{K-1}\,\mathrm{poly}(K, M, T)$ algorithm.

1 INTRODUCTION

In many applications of interest, a principal wishes to procure *homogeneous goods*, *i.e.*, goods for which her value depends only on the quantity procured. A canonical example is data collection, arising in clinical studies (Weiner et al., 2017), market research (Fader & Hardie, 2009), ecological monitoring (Sullivan et al., 2009), and spectrum monitoring (Calvo-Palomino et al., 2017). The same structure appears well beyond data: a cloud provider procures accelerators from hardware vendors for her data center, a firm contracts workers for person-hours on a project, and a government agency purchases vaccine doses from multiple producers ahead of a season.

This work. Motivated by such settings, we study how a principal may optimally procure homogeneous goods. She posts a menu of payments specifying how much she will pay an agent as a function of the quantity supplied. Each agent then best responds, producing the quantity that maximizes her payment less her costs. The principal seeks to maximize her own utility, *i.e.*, the value she derives from the goods minus payments. This setting poses three distinct challenges. First, the principal knows only the distribution from which agents' costs are drawn, not the realized costs; a payment sufficient to induce production by one agent may be too low for another, while higher payments erode her utility. Second, the payments offered at different quantities must be chosen jointly, since a quantity-payment pair intended for one agent is also available to every other. Compensating each agent for her intended production costs therefore does not by itself induce the desired quantities; we must account for how each agent responds to the entire menu. Third, the menu may depend only on the quantity an agent supplies, and cannot condition payment on what others produce; for instance, in clinical data collection or chip procurement, an agent is not party to another's arrangements.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1.1 Summary of our Contributions

We now summarize our key contributions. We first design optimal menus for when there is a single agent, whose costs are unknown, before tackling the multi-agent setting.

Single-agent model. A principal procures a homogeneous good from a single agent via an anonymous payment menu. The agent's type is private and drawn from a known distribution over K nondecreasing cost functions $c_1, \dots, c_K$, where $c_i : \{0, \dots, M\} \to \mathbb{R}_{\geq 0}$, where M is the maximum amount an agent can produce. The principal posts a menu π specifying a payment $\pi(m)$ for each production level $m \in \{0, \dots, M\}$; the agent best responds by maximizing $\pi(m) - c_i(m)$, with ties resolved in the principal's favor. The principal chooses π to maximize her expected utility $\mathbb{E}[V(m) - \pi(m)]$, where the expectation is over the agent's type.

We are particularly interested in the regime $K \ll M$, which reflects applications of interest.. In Alzheimer's data collection, a clinic's cost is largely set by its scanner and its staffing: only a handful of scanner models are in use, and staffing costs are driven by the clinic's city. There are far fewer scanner-city pairs than the number of data M a clinic might contribute. Chip manufacturing is similar, with cost set by a few process nodes while one foundry can supply several accelerators.

Hardness results. We first give polynomial-time reductions in both directions between single-agent procurement and revenue-optimal unit-demand item pricing, preserving the optimal objective value (§3). The reduction identifies production levels with items, the agent's utility with the buyer's, and the principal's utility with the seller's revenue. This reduction transfers strong NP-hardness (Chawla et al., 2022), along with a poly-time approximation scheme for fixed M (Hartline & Koltun, 2005).

Methods and results. However, as we are interested in the large M regime, we design an exactly optimal algorithm running in $(M+1)^K \mathrm{poly}(K, M)$ time (§4.1). It enumerates *production profiles*, each specifying an intended level for every type, and solves a linear program per profile for the menu of minimum expected payment making those levels best responses. Since an optimal menu induces one of the enumerated profiles, this is exact, and polynomial for any fixed number of types.

Under an *ordered-cost condition*, *i.e.*, when types can be ordered so that a lower-indexed type has no higher marginal cost than a higher-indexed one (satisfied by linear costs $c_i(m) = \theta_i m$ as a special case), we obtain an exact algorithm polynomial in both K and M (§4.2). We show that some optimal menu induces a monotone profile, *i.e.*, its best responses satisfy $m_1 \geq \cdots \geq m_K$, and characterize the cheapest payments implementing such a profile recursively. Substituting these into the principal's objective yields a dynamic program using $\mathcal{O}(KM)$ operations.

Multi-agent model. With multiple agents, we show that presenting the same menu to all agents simultaneously can fail spectacularly: to avoid overproducing, the principal must equally distribute the task among agents; each agent could suffer from high startup costs due to economy of scale. Hence, even the optimal common menu can yield lower expected utility as more agents participate.

Sequential arrivals. We therefore study a sequential model over T rounds, with one agent arriving per round and types drawn independently from the same known distribution. The principal maximizes the expected value of *total* production less *total* payments. We consider two settings: *(i) adaptive menus:* she may revise the menu before each arrival, conditioning on cumulative production and rounds remaining; *(ii) static menus:* she commits to a single menu but may close the market at a round of her choice. In both, an agent's payment depends only on her own production.

Adaptive menus. We design an exact algorithm running in $(M+1)^K \mathrm{poly}(K, M, T)$ time (§6.1) via a dynamic program (DP). The DP's state records cumulative production and rounds remaining, and choosing a menu at a state reduces to a single-agent problem under a modified valuation that accounts for optimal future procurement. Under the ordered-cost condition, invoking the faster single-agent algorithm reduces the running time to $\mathcal{O}(KT^2M^2)$.

Static menus. The key challenge in the static model is that a menu may give a type several tied best responses, and which one the principal prefers changes with cumulative production and the number of remaining rounds, so enumerating fixed profiles does not capture every useful choice. We instead show that every menu can be replaced, without loss in principal utility, by one in a finite family. Each menu in this family can be captured by K tight constraints forming a tree. Enumerating these trees and their associated menus, then evaluating each menu by dynamic programming, yields an optimal static menu and its policy in $(K+1)^{K-1}(M+1)^{K-1}\,\mathrm{poly}(K, M, T)$ time (§6.2).

1.2 Related Work

Due to space, we discuss most relevant work here, and differ additional discussions to Appendix A.

Screening, procurement, contract design. Our setting is a monopsonist posting a nonlinear payment menu to suppliers with private costs when her own valuation is also nonlinear. The classical screening and procurement literature (Mussa & Rosen, 1978; Maskin & Riley, 1984; Myerson, 1981; Iyengar & Kumar, 2008; Laffont & Martimort, 2002) takes the principal's value to be additive in what agents deliver, so the problem decomposes into independent single-agent problems. Under nonlinear valuations, we enjoy no such decomposition, as the worth of one agent's contribution depends on what others have already submitted. Procurement mechanisms that do couple agents elicit bids and condition an agent's payment on the others' reports (Singer, 2010; Anari et al., 2018; Jarman & Meisner, 2017; Eden et al., 2026), whereas our menu is posted anonymously before any agent acts. *Multi-agent contract design* studies incentives for hidden actions through outcome-contingent payments (Cacciamani et al., 2024; Castiglioni et al., 2023), whereas we study private costs of verifiable production. Online contract and procurement design (Ho et al., 2016; Badanidiyuru et al., 2012; Charalampopoulos et al., 2025; Lavi et al., 2026) sequences interactions to *learn* an unknown population, whereas we sequence to avoid over-procuring.

Procuring and selling data. Principal-agent models for data collection assume known costs (Cai et al., 2015), or target data quality rather than quantity (Ananthakrishnan et al., 2026). Work purchasing data from privately-costed agents (Chen et al., 2018; Roth & Schoenebeck, 2012; Abernethy et al., 2015) fixes a budget without designing a menu over production quantities. Work on *homogeneous-data pricing* likewise distinguishes a small number of types from many possible quantities, but studies learning prices under an unknown type distribution (Chen et al., 2024).

Connections to our techniques. Our reduction in §3 draws on *unit-demand item pricing* to obtain hardness and approximation guarantees (Chawla et al., 2022; Hartline & Koltun, 2005). Our ordered-cost algorithm in §4.2 uses monotonicity and adjacent incentive constraints familiar from *screening under adverse selection* (Laffont & Martimort, 2002). Our profile algorithms in §4 and tree-based static algorithm in §6.2 draw on *automated mechanism design* and graph representations of incentive constraints (Conitzer & Sandholm, 2004; Rochet, 1987; Heydenreich et al., 2009). Finally, our sequential models connect to *dynamic mechanism design* (Bergemann & Välimäki, 2019), with interactions across rounds arising through cumulative production and the remaining horizon.

2 Single-Agent Problem Setting

We first study optimal menus for procuring homogeneous goods, when there is a single agent.

Model. A principal wishes to incentivize a single agent to produce homogeneous goods. The agent belongs to one of K *types*, and is drawn from a publicly known distribution $q = \{q_i\}_{i\in[K]}$, where type i occurs with probability q_i. The agent may choose a *production level* $m \in \{0, 1, \dots, M\}$, indicating the number of homogeneous units produced (*e.g.*, data points collected, GPUs manufactured, or units of effort exerted). No agent can produce more than M units, indicating hard resource constraints (*e.g.*, finite compute, capital, or time). A type-i agent incurs cost $c_i(m)$ for production level m, where $c_i : \{0, 1, \dots, M\} \to \mathbb{R}_{\geq 0}$. We assume c_i to be nondecreasing and $c_i(0) = 0$, since producing at a higher level should cost weakly more and there is no cost to produce nothing.

Payment menu. The principal posts an anonymous (*i.e.*, non-discriminatory) payment menu $\pi : \{0, 1, \dots, M\} \to \mathbb{R}_{\geq 0}$ where $\pi(m)$ is the payment to an agent producing m units. The *utility* of a type-i agent producing m goods is her payment minus her cost, *i.e.*, $\pi(m) - c_i(m)$. Hence, given a menu of payments, an agent chooses a production level which maximizes her utility. In general, this argmax is the set $\widetilde{m}_i$ defined below in (1), and we allow the principal to break ties and choose a specific production level she desires. Precisely, we assume that when the principal publishes π, the agent will report $\widetilde{m}_i$ to the principal, who will then choose some $m \in \widetilde{m}_i(\pi)$. We have:

$$\widetilde{m}_i(\pi) := \operatorname*{argmax}_{m\in\{0,\dots,M\}} \big\{\pi(m) - c_i(m)\big\}. \tag{1}$$

Principal's utility and objective. The principal has valuation $V : \{0, 1, \dots, M\} \to \mathbb{R}_{\geq 0}$ where $V(m)$ is the value obtained from m goods. We assume V to be nondecreasing and $V(0) = 0$, since

acquiring more goods should weakly increase her valuation. If the principal receives m goods, her utility is her value minus payment, *i.e.*, $V(m) - \pi(m)$. The principal's objective is to maximize her expected utility over q. As the principal will choose the most favorable production level from the set $\widetilde{m}_i$ an agent provides, we can write this formally as shown below in (2). For what follows, we will denote a maximizer inside the expectation as $m_i(\pi)$. We have:

$$\mathrm{OPT} := \max_{\pi} \mathbb{E}_{i\sim q}\left[\max_{m\in\widetilde{m}_i(\pi)} (V(m) - \pi(m))\right] = \max_{\pi} \mathbb{E}_{i\sim q}\left[\big(V\big(m_i(\pi)\big) - \pi\big(m_i(\pi)\big)\big)\right]. \quad (2)$$

It is worth drawing attention to two aspects of our model before proceeding. First, our methods for the most general setting, make no assumptions on V and c_i beyond weak monotonicity, such as linearity or concavity. We will however show that under an ordering condition on agents' cost curves, the run time can be significantly improved. Second, we also do not impose any restrictions on the payment menu π beyond non-negativity. However, it is straightforward to show that without loss of generality (WLOG), we can assume that π is nondecreasing and $\pi(m) \leq V(m)$ for all m (see Lemmas 5 and 6 in Appendix B). In particular, this implies that $\pi(0) = V(0) = c_i(0) = 0$, which makes $m = 0$ an explicit nonparticipating option for any agent type.

3 Hardness Results and Connection to Item Pricing

We first establish a two-way reduction between our problem and revenue-optimal item pricing with unit-demand buyers, showing the two are computationally equivalent. NP-hardness then follows from known hardness results for item pricing. We begin by describing unit-demand item pricing.

Unit-demand item pricing. Consider a seller offering M distinct items to a single buyer. The buyer has type $i \in [K]$, drawn from a known distribution q with probability q_i, and values item m at $v_i(m) \geq 0$. The seller posts a price $p(m) \geq 0$ for each item. The buyer purchases at most one item, choosing one which maximizes her utility (value minus price), provided it gives nonnegative utility. In general, multiple items may be utility-maximizing, and ties are broken in favor of the seller, *i.e.*, by choosing the item with the highest price. Writing $m = 0$ for purchasing nothing, with $v_i(0) = p(0) = 0$, the buyer's purchase decision $m_i(p)$ under price curve p is as shown below. The seller seeks to design p that maximizes her expected revenue $\mathrm{Rev}(p)$. We have:

$$\widetilde{m}_i(p) = \operatorname*{argmax}_{m\in\{0,\ldots,M\}} \{v_i(m) - p(m)\}, \quad m_i(p) = \operatorname*{argmax}_{m\in\widetilde{m}_i(p)} p(m), \quad \mathrm{Rev}(p) = \textstyle\sum_{i=1}^{K} q_i\, p\big(m_i(p)\big).$$

We establish the following reduction theorem, whose full proof is given in Appendix C.1.

Theorem 1 (Reduction to item pricing)**.** *The single-agent procurement problem and unit-demand item pricing admit polynomial-time reductions in both directions that preserve optimal objective values and allow solutions to be converted back in polynomial time.*

Proof sketch. The reductions retain the distribution of types, and identify the M positive production levels with M items. Given a procurement instance and a payment menu π, consider the pricing instance with $v_i(m) = V(m) - c_i(m)$ and $p(m) = V(m) - \pi(m)$. The buyer's value for item m is the surplus generated by production, *i.e.*, the principal's value minus the agent's cost, while its price is the principal's value minus her payment. For the sketch, suppose these quantities are nonnegative; we handle negative values in the full proof. The buyer's utility for item m is then $v_i(m) - p(m) = \pi(m) - c_i(m)$, which is exactly the agent's utility for producing at level m in the procurement problem. Therefore, the utility-maximizing choices for the buyer and the agent coincide. Moreover, the seller's revenue from selling item m is $p(m) = V(m) - \pi(m)$, which is exactly the principal's utility at production level m. Thus the seller and the principal always break ties the same way, and the seller's expected revenue equals the principal's expected utility.

For the reverse reduction, we start with buyer valuations v_i and choose a sufficiently steep linear valuation V (specified in the full proof). Setting $c_i = V - v_i$ then gives nonnegative, nondecreasing production costs. Given pricing p, the inverse transformation sets $\pi = V - p$. Then the same identities as above establish the correspondence between the two instances. □

The reduction allows us to transfer two results from item pricing, stated as corollaries below. First, the strong NP-hardness of ordered item pricing established by Chawla et al. (2022) implies the same

for our problem. Second, applying the approximation scheme of Hartline & Koltun (2005) gives an algorithm when the production range M is small. Their proofs are given in Appendix C.2.

Corollary 2 (Computational hardness)**.** *Computing an optimal single-agent payment menu is strongly NP-hard, even when the principal's valuation V is linear.*

Corollary 3 (Approximation for small production range)**.** *For every $\epsilon \in (0, 1)$, there is an algorithm that computes a payment menu with expected principal utility at least $\mathrm{OPT}/(1+\epsilon)$. Its running time has exponential dependence in M, but polynomial dependence in K and $1/\epsilon$.*

Implications for computation. The hardness result shows that optimizing the payment menu is already difficult with a single agent. Therefore, we do not expect an exact algorithm polynomial in both K and M. The approximation scheme in Corollary 3 offers a solution when M is small. However, as discussed in §1.1, in the applications that motivate our work, we have $K \ll M$, making the exponential dependence on M prohibitive. Therefore, next, we describe an algorithm to compute an *exact* optimal payment menu whose exponential dependence is confined to K.

4 Exact Algorithms for a Single Agent

4.1 An Exact Algorithm for General Cost Curves

At a high level, our method enumerates all $(M+1)^K$ *production profiles*, each specifying an intended level for every type, finds the cheapest menu implementing each, and returns the best one. We begin by defining a production profile formally below.

Definition 1 (Production profile)**.** *A* production profile *is a vector $\boldsymbol{m} = (m_1, \ldots, m_K)$ where $m_i \in \{0, \ldots, M\}$ denotes the production level intended for type i. We say that a payment menu π* implements *$\boldsymbol{m}$ if $m_i \in \widetilde{m}_i(\pi)$ for every $i \in [K]$.*

Fix a production profile $\boldsymbol{m}$ and consider the principal's utility when each type selects its designated level. Under a fixed profile, the expected production value, $\sum_{i=1}^K q_i V(m_i)$, is fixed. Maximizing principal utility then amounts to finding the payment menu that implements this profile with the lowest expected payment $\sum_{i=1}^K q_i \pi(m_i)$. This can be computed via the LP in (3), which we refer to as Profile-LP($\boldsymbol{m}$). The incentive constraints in the last line ensure that every type i weakly prefers its designated production level m_i to every alternative. In the sense of Definition 1, these constraints implement $\boldsymbol{m}$. Since $c_i(0) = 0$ and the payments are nonnegative, the agent has at least one option with non-negative utility; this guarantees that m_i gives type i nonnegative utility. If Profile-LP($\boldsymbol{m}$) is infeasible, no payment menu implements the profile $\boldsymbol{m}$.

$$\begin{aligned} \min_{\pi(0),\ldots,\pi(M)} \quad & \sum_{i=1}^K q_i \pi(m_i) && (3)\\ \text{s.t.} \quad & \pi(m) \geq 0, && \forall\, m,\\ & \pi(m_i) - c_i(m_i) \geq \pi(m) - c_i(m), && \forall\, i,\ m. \end{aligned}$$

Algorithm. Our algorithm enumerates all $(M+1)^K$ profiles and solves Profile-LP($\boldsymbol{m}$) for each, discarding infeasible profiles. For each feasible profile $\boldsymbol{m}$, let $\pi_{\boldsymbol{m}}$ be an optimal LP solution. We evaluate this candidate menu by $\sum_{i=1}^K q_i \big[V(m_i) - \pi_{\boldsymbol{m}}(m_i)\big]$. The algorithm returns the candidate menu with the largest expected utility. The following theorem states its optimality and runtime.

Theorem 4. *The algorithm above gives an exact optimal solution to the single-agent procurement problem with runtime $(M+1)^K \mathrm{poly}(K, M)$.*

Proof. Let $\pi^\star$ be an optimal payment menu and $m_i^\star$ the level selected for type i under principal-favorable tie-breaking (fixing any one maximizer if several are tied). These choices define a production profile $\boldsymbol{m}^\star = (m_1^\star, \ldots, m_K^\star)$ with $\mathrm{OPT} = \sum_{i=1}^K q_i \big[V(m_i^\star) - \pi^\star(m_i^\star)\big]$.

Our enumeration of all profiles considers $\boldsymbol{m}^\star$. Each $m_i^\star$ is a best response under $\pi^\star$, so $\pi^\star$ is feasible for Profile-LP($\boldsymbol{m}^\star$). Thus an LP minimizer $\pi_{\boldsymbol{m}^\star}$ exists and satisfies $\sum_{i=1}^K q_i \pi_{\boldsymbol{m}^\star}(m_i^\star) \leq \sum_{i=1}^K q_i \pi^\star(m_i^\star)$. Under $\pi_{\boldsymbol{m}^\star}$, every $m_i^\star$ remains a best response. The principal can thus select these same production levels, obtaining the same production value while making no larger payment. Thus the enumerated candidates include a payment menu whose utility is at least OPT.

Run time. There are $(M+1)^K$ profiles. Each Profile-LP has $\mathcal{O}(M)$ variables and $\mathcal{O}(KM)$ constraints, so it can be solved in $\mathrm{poly}(K, M)$ time (Grötschel et al., 1988). □

4.2 Polynomial-Time Algorithm for Ordered Cost Curves

Next, we show that the exponential dependence on K can be removed under additional structure on the agents' cost curves. In particular, we say that the types satisfy an *ordered-cost condition* if

$$c_i(m') - c_i(m) \le c_j(m') - c_j(m), \qquad \forall i < j, \quad 0 \le m < m' \le M. \tag{4}$$

Intuitively, the type index orders agents from lowest to highest cost, in the sense that a lower-indexed agent's marginal cost is always weakly below that of a higher-indexed one. The condition covers the family of *scaled cost curves*, in which each agent's cost function is determined by a single efficiency parameter θ_i, with $0 \le \theta_1 \le \cdots \le \theta_K$. Precisely, $c_i(m) = \theta_i g(m)$, where g is nondecreasing with $g(0) = 0$. Linear production costs, $c_i(m) = \theta_i m$, are a particularly simple special case.

Under (4), the single-agent problem can be solved in polynomial time. The proof sketch below presents the key lemmas and the resulting algorithm; the full proof is deferred to Appendix D.

Theorem 5. *Suppose the types satisfy the ordered-cost condition in (4). Then an optimal payment menu can be computed in* $\mathrm{poly}(K, M)$ *time. In particular, the optimal profile can be computed by a dynamic program in* $\mathcal{O}(KM)$ *arithmetic operations.*

Proof sketch. Under (4), we first observe the following lemma, which states that for any payment menu π, the principal may choose utility maximizers $m_i(\pi)$ in a monotone manner.

Lemma 1 (Monotone profiles)**.** *For every payment menu π, there is a principal-favorable selection of best responses forming a monotone production profile $m_1 \ge m_2 \ge \cdots \ge m_K$.*

This is derived by juxtaposing the incentive constraints between arbitrary types $i < j$: i's constraint for choosing m_i over m_j, and j's constraint for choosing m_j over m_i. Combining these with (4) implies $m_i \ge m_j$ when the principal always chooses a larger production level if multiple levels give her the same utility, thus establishing Lemma 1. Hence, it suffices to enumerate monotone profiles. There are still exponentially many of these, so we need a second ingredient: the following lemma characterizes the cheapest payments implementing a monotone profile.

Lemma 2 (Cheapest payments for a monotone profile)**.** *Fix a monotone production profile $\boldsymbol{m}$ where $m_1 \ge m_2 \ge \cdots \ge m_K$. The minimum expected payment at the designated levels among all payment menus implementing $\boldsymbol{m}$ is achieved by*

$$\pi(m_K) = c_K(m_K), \qquad \pi(m_i) = \pi(m_{i+1}) + c_i(m_i) - c_i(m_{i+1}), \quad i = 1, \dots, K-1. \tag{5}$$

We need at least the payments in (5) to satisfy a subset of incentive constraints. In particular, these payments cover type K's production cost, and ensure that every other type i does not prefer m_{i+1} over m_i. We call these *local* incentive constraints. In fact, these payments satisfy *all* incentive constraints because the incentive constraints between arbitrary $i < j$ are implied by a telescope sum of local incentive constraints under the ordered-cost condition (4).

Lemma 2 allows us to eliminate the payment variables entirely. Through expanding the recursion, we derive the following *virtual cost curves* ψ_i satisfying $\sum_{i=1}^K q_i \pi(m_i) = \sum_{i=1}^K q_i \psi_i(m_i)$:

$$\psi_i(m) := c_i(m) + \frac{Q_{i-1}}{q_i}\big(c_i(m) - c_{i-1}(m)\big), \qquad Q_i := \sum\nolimits_{j=1}^{i} q_j.$$

Note that $Q_0 := 0$ and the second term of $\psi_1(m)$ vanishes. With this, optimizing the principal's utility is equivalent to computing $\max_{m_1 \ge \cdots \ge m_K} \sum_{i=1}^K q_i \left[V(m_i) - \psi_i(m_i)\right]$. The only interaction between the choices m_i is the monotonicity constraint. This permits a dynamic program as follows.

Algorithm. For $i \in [K]$ and $m \in \{0, \dots, M\}$, let $F_i(m)$ denote the maximum contribution from types $1, \dots, i$, subject to type i producing exactly m. The base case is $F_1(m) = q_1\big[V(m) - \psi_1(m)\big]$, and for $i \ge 2$, the Bellman equation is $F_i(m) = q_i\big[V(m) - \psi_i(m)\big] + \max_{m' \ge m} F_{i-1}(m')$. The optimal objective value is $\max_{m \in \{0,\dots,M\}} F_K(m)$. Each layer can be computed in $\mathcal{O}(M)$ time by suffix maxima, and standard backtracking recovers a maximizing profile after $\mathcal{O}(KM)$ arithmetic operations. Its payment menu is reconstructed using (5), with 0 payments at undesignated levels. □

5 MULTI-AGENT PROBLEM SETTING

We now turn to designing menus for multiple agents. A natural extension of the single-agent setting is to offer the same menu to all agents simultaneously.[1] With concave costs, however, this fails. The principal must honor the menu for every agent, which might result in excess production or paying repeatedly for expensive first units. Example 2 of Appendix E makes this concrete.

Sequential arrivals. We therefore adopt a *sequential* model, in which agents arrive one at a time. This also matches many applications of interest; for instance, a sponsor running a multi-site clinical trial negotiates enrollment targets site by site, with later terms shaped by how many patients earlier sites have committed. We consider a horizon of T rounds, with one agent arriving per round. Types $i_1, \dots, i_T$ are drawn independently from the distribution q. On round t, the agent produces $x_t \in \{0, \dots, M\}$, incurring cost $c_{i_t}(x_t)$. Let $z_t = \sum_{s=1}^{t} x_s$ denote the cumulative production level at round t, and let $z_0 = 0$. We also extend the principal's nondecreasing valuation to $V : \{0, \dots, TM\} \to \mathbb{R}_{\geq 0}$, with $V(0) = 0$.

Principal's policy. On round t, the principal posts a menu $\pi_t : \{0, \dots, M\} \to \mathbb{R}_{\geq 0}$ before the agent's type is realized, and an agent of type i_t reports her utility maximizing set $\widetilde{m}_{i_t}(\pi_t)$ to the principal. The principal then applies a tie-breaking rule x_t^{tb}, mapping the history to a production level $x_t \in \widetilde{m}_{i_t}$. Unlike in the single-round setting, where tie-breaking is straightforward (*i.e.*, choose the level which maximizes the principal's utility), here the principal needs to account for the future as well. After $\tau \in \{0, \dots, T\}$ completed rounds, the principal closes the market.

Adaptive and static menus. We will consider two types of menus in this work, both reflecting different practical use cases: *(i) Adaptive menus.* Here, π_t may change from round to round based on history. Hence, the principal's policy $\mathcal{A} = (\{\pi_t\}_{t=1}^{T}, \{x_t^{\mathrm{tb}}\}_{t=1}^{T}, \tau)$ consists of time-adapted menus and tie-breaking rules along with the stopping time. *(ii) Static menus.* Here, the principal chooses a single contract π, and hence her policy is $\mathcal{A} = (\pi, \{x_t^{\mathrm{tb}}\}_{t=1}^{T}, \tau)$. For static menus, $\pi_t = \pi$ for all $t \leq \tau$. In both models, $\pi_t = 0$ and $x_t^{\mathrm{tb}} = 0$ for all $t > \tau$. In both cases, the principal wishes to maximize her utility $U(\mathcal{A}) = \mathbb{E}\big[V(z_T) - \sum_{t=1}^{T} \pi_t(x_t)\big] = \mathbb{E}\big[V(z_\tau) - \sum_{t=1}^{\tau} \pi_t(x_t)\big]$.

6 ALGORITHMS FOR MULTI-AGENT ROUND-BASED MODELS

6.1 ADAPTIVE MENUS

To compute the optimal policy in the adaptive model, we can use a backward dynamic program (DP), in which each state requires solving a single-agent menu problem with a modified valuation. For $t \in \{0, \dots, T\}$ and $z \in \{0, \dots, tM\}$, let $A_t(z)$ be the maximum expected terminal valuation minus future payments, starting after t completed rounds with cumulative production z:

$$A_t(z) := \max_{\{(\pi_s, x_s^{\mathrm{tb}})\}_{s>t},\, t \leq \tau \leq T} \mathbb{E}\left[V(z_\tau) - \sum_{s=t+1}^{\tau} \pi_s(x_s) \,\middle|\, z_t = z \right].$$

Above, only payments made after round t are subtracted in this definition. Subtracting the payments from the first t rounds from $A_t(z)$ gives the optimal expected total utility conditional on the full history. As these payments do not affect the continuation and future types are independent of earlier rounds, cumulative production and the number of completed rounds suffice to describe the state.

After round T, there are no future payments, so $A_T(z) = V(z)$. For $t < T$, the Bellman equation is

$$A_t(z) = \max\left\{ V(z),\ \max_{\pi} \sum_{i=1}^{K} q_i \max_{m \in \widetilde{m}_i(\pi)} \big[A_{t+1}(z+m) - \pi(m) \big] \right\}. \tag{6}$$

The first term closes the market immediately. The second posts a menu for round $t+1$ before the next agent's type is realized. The inner maximization implements principal-favorable tie-breaking using the continuation value. Since A_{t+1} already accounts for payments after round $t+1$, the transition subtracts only the payment in round $t+1$.

[1] Alternatively, a menu could condition on total production, *i.e.*, pay $\pi(x_i, \sum_{j \neq i} x_j)$, but this makes payoffs interdependent and the induced game admits poor equilibria (Segal, 1999; 2003). This is also impractical in our applications; for instance, a clinic will not accept payment contingent on patient enrollment elsewhere.

Reduction of each state to a single-agent problem. Once A_{t+1} is known, define

$$W_{t,z}(m) := A_{t+1}(z+m) - A_{t+1}(z), \qquad m \in \{0, \dots, M\}.$$

Each A_t is nondecreasing in z. This follows by backward induction: increasing z weakly increases $V(z)$ and every continuation payoff in (6). Consequently, $W_{t,z}$ is nondecreasing in m and satisfies $W_{t,z}(0) = 0$, as required of a single-agent valuation curve. The continuation term in (6) is exactly $A_{t+1}(z)$ plus the optimal single-agent utility with valuation $W_{t,z}$, costs c_i, and type probabilities q_i. Subtracting the constant $A_{t+1}(z)$ changes neither the optimal menu nor the principal's preferred best responses. Each state can therefore be solved by the production-profile algorithm of §4.1. This gives the following theorem.

Theorem 6 (Optimal adaptive menus)**.** *An optimal adaptive menu and stopping policy can be computed in time* $(M+1)^K \cdot \mathrm{poly}(K, M, T)$.

Proof. Initialize the terminal values using $A_T(z) = V(z)$ and evaluate (6) backward for $t = T-1, \dots, 0$. At each state, solve the corresponding single-agent problem exactly and record an optimal stopping decision or a menu together with its selected best responses. The Bellman equation considers every available decision and combines it with optimal continuation values, so backward induction establishes that the resulting policy has expected utility $A_0(0)$ and is optimal. There are $\mathcal{O}(T^2 M)$ states. Each state uses one call to the §4.1 algorithm, which takes $(M+1)^K \,\mathrm{poly}(K, M)$ time. The total runtime is therefore $(M+1)^K \,\mathrm{poly}(K, M, T)$. □

Improvement under ordered-costs. If the costs satisfy the ordered-cost condition from §4.2, every single-agent problem on each round above satisfies the same assumption: the costs and their type ordering are unchanged, and only the principal's valuation varies. We can therefore replace production-profile enumeration at each state with the dynamic program from §4.2. This routine takes $\mathcal{O}(KM)$ operations per state. Across the $\mathcal{O}(T^2 M)$ states, the resulting adaptive algorithm takes $\mathcal{O}(KT^2M^2)$ operations, and hence runs in $\mathrm{poly}(K, M, T)$ time while retaining exact optimality. The corollary below states this formally.

Corollary 7. *Suppose the types satisfy the ordered-cost condition in (4). Then the optimal multi-round adaptive payment menus can be computed in* $\mathrm{poly}(K, M, T)$ *time.*

6.2 Static menus

In the static model, the principal chooses a fixed payment menu before the first round and may stop at any round. We first dismiss a naive algorithm using production profiles.

A naive first attempt. The single-agent algorithm from §4.1 suggests: *(i)* Enumerate every production profile $\boldsymbol{m}$, and solve Profile-LP to obtain a minimum-payment menu $\pi_{\boldsymbol{m}}$ for the feasible ones; *(ii)* Keep the profile responses m_i fixed across rounds and use dynamic programming to find the optimal stopping rule. Return the profile, menu, and stopping rule with the largest principal utility. The key issue with this approach is that m_i is fixed whenever type i arrives. However, if π offers i multiple best responses, the principal might benefit from tie-breaking differently at different rounds. This marks a key difference between the static model and the single-agent/adaptive models. The following example shows that this restriction can lose utility even with one type and two rounds.

Example 1. *Let* $K = 1$, $M = T = 2$, *with* $c(0) = 0$, $c(1) = 0.6$, $c(2) = 1$, *and* $V(z) = \min\{z, 3\}$. *Implementing a production level* m *requires a payment of at least* $c(m)$, *so a fixed response* m *earns the principal at most* $\max_{\tau \in \{0,1,2\}} \{V(\tau m) - \tau c(m)\}$. *This gives* 0 *utility for* $m = 0$, *and* 0.8 *and* 1 *for* $m = 1$ *and* $m = 2$, *both at* $\tau = 2$. *Thus a fixed-profile policy has utility at most* 1*: collecting 2 units in each round pays for 4 while the valuation saturates at 3.*

Now consider the static menu $\pi(m) = c(m)$. *All three levels give the agent zero utility, so* $\widetilde{m}(\pi) = \{0, 1, 2\}$, *and the principal can select two units in the first round and one in the second, obtaining* $V(3) - \pi(2) - \pi(1) = 3 - 1 - 0.6 = 1.4$. *The menu is unchanged across rounds; only the element in the best response set selected via tie-breaking varies. This policy is therefore feasible in the static model but missed by every fixed-profile evaluation.*

We instead construct a finite family of menus that retains all the production choices needed by an optimal policy. The gives the following theorem, whose full proof is deferred to Appendix F.

Theorem 8 (Optimal static menus)**.** *An optimal static menu and its associated policy can be computed in time* $(K+1)^{K-1}(M+1)^{K-1} \,\mathrm{poly}(K, M, T)$. *This is polynomial for each fixed* K.

Proof sketch. Let $J(\pi)$ be the principal's optimal expected utility under menu π, with optimal stopping rule and tie-breaking decisions. For a utility vector $u \in \mathbb{R}^K_{\geq 0}$, define its *envelope menu* by

$$\pi_u(m) := \min_{j \in [K]} \{c_j(m) + u_j\}. \tag{7}$$

This menu gives type i utility at most u_i at every production level. We first show that the envelope menu of an appropriate $\widehat{u}$ can preserve all the choices available under an arbitrary original menu.

Lemma 3 (A replacement preserving every best response)**.** *For every menu π^0, there is a vector $\widehat{u} \geq 0$ whose envelope menu $\widehat{\pi} = \pi_{\widehat{u}}$ satisfies $\widetilde{m}_i(\pi^0) \subseteq \widetilde{m}_i(\widehat{\pi})$ and $\widehat{\pi}(m) \leq \pi^0(m)$ for all $i \in [K]$, $m \in \widetilde{m}_i(\pi^0)$. Consequently, $J(\widehat{\pi}) \geq J(\pi^0)$.*

We show that a $\widehat{u}$ exists. In particular, consider this replacement LP: minimize $\sum_i u_i$ subject to $0 \leq u_i \leq u_i^0$ and $c_i(m) + u_i \leq c_j(m) + u_j$ for every $m \in \widetilde{m}_i(\pi^0)$ and every type j, where u_i^0 is type i's maximum utility under π^0. The original utility vector u^0 is feasible and the feasible region is compact, so a minimizer exists. We claim this minimizer is a suitable $\widehat{u}$. The constraints ensure that every original best response attains utility $\widehat{u}_i$ under the envelope menu $\widehat{\pi}$, while $\widehat{u}_i \leq u_i^0$ ensures a weakly lower payment. The principal can therefore reproduce every tie-breaking choice and stopping rule without paying more. This LP is used only in the proof; the next lemma allows us to construct candidate menus without knowing the original menu or its best responses.

Lemma 4 (A tree representation of the replacement)**.** *For every menu π^0, a replacement vector $\widehat{u}$ from Lemma 3 can be represented by a tree on $\{\star, 1, \ldots, K\}$, oriented away from $\star$, with a production label on every edge between types, through the following equations: $u_i = 0$ if $\star \to i$ is a tree edge; $u_j = u_i + c_i(m) - c_j(m)$ if $i \xrightarrow{m} j$ is a tree edge.*

We view the replacement LP's lower bound constraints as directed edges, and retain those that are tight at a minimizer $\widehat{u}$. Every type must be reachable from $\star$ through tight edges: otherwise all unreachable coordinates could be decreased slightly, preserving feasibility and decreasing the objective, contradicting the minimality of $\widehat{u}$. A spanning tree of these tight edges gives the tree equations above, which determine $\widehat{u}$ along paths from the root.

Algorithm. Enumerate every tree on the $K+1$ vertices and every assignment of production labels from $\{0, \ldots, M\}$ to the tree edges between types. Solve the tree equations above, discard vectors with a negative coordinate, and construct the remaining menus using (7). For each candidate π, compute its actual best-response sets and evaluate it by the following dynamic program. Let $F_t^\pi(z)$ be the optimal expected terminal valuation minus future payments after t completed rounds, with cumulative production z and the market still open. The Bellman equations are $F_T^\pi(z) = V(z)$, and

$$F_t^\pi(z) = \max \left\{ V(z),\ \sum_{i=1}^K q_i \max_{m \in \widetilde{m}_i(\pi)} \left[F_{t+1}^\pi(z+m) - \pi(m) \right] \right\}, \qquad \forall t < T. \tag{8}$$

The two terms correspond to closing the market and keeping it open for the next arrival. The inner maximum implements principal-favorable tie-breaking at each state. Return the candidate menu π maximizing $J(\pi) = F_0^\pi(0)$, together with the policy recovered from the DP.

By Lemmas 3 and 4, the candidate family contains a menu with utility at least that of any original menu π^0. Backward induction establishes that (8) evaluates each candidate optimally, so the returned menu and policy are optimal. By Cayley's formula, there are $(K+1)^{K-1}$ trees on $K+1$ vertices, each with at most $K-1$ edges requiring production labels. Thus the algorithm generates at most $(K+1)^{K-1}(M+1)^{K-1}$ candidates. The DP for each candidate has $O(T^2 M)$ states, with $O(K(M+1))$ operations per state, giving the stated runtime. □

7 Conclusion

We studied the optimal procurement of homogeneous goods from agents with private, quantity-dependent costs. A two-way reduction to unit-demand item pricing gives strong NP-hardness, and our profile- and tree-based algorithms are exact and polynomial for any fixed number of types K. Under the ordered-cost condition, the single-agent and adaptive algorithms become polynomial in both K and M; whether static menus admit a comparable speedup remains open, as the principal's tie-breaking varies with the state and the monotone-profile argument does not directly apply.

AI USE DISCLOSURE

We used ChatGPT (OpenAI) as a supporting tool during the research and preparation of this manuscript. This included assistance with drafting and editing text, finding and understanding related literature, discussing technical questions, and drafting and refining proofs. The authors evaluated and revised AI-generated content, checked the arguments included in the paper, and take full responsibility for its results and final presentation.

REFERENCES

Jacob Abernethy, Yiling Chen, Chien-Ju Ho, and Bo Waggoner. Low-cost learning via active data procurement. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*. ACM, 2015. doi: 10.1145/2764468.2764519. URL https://arxiv.org/abs/1502.05774.

Nivasini Ananthakrishnan, Alireza Fallah, and Michael I. Jordan. Signaling in data markets via free samples, 2026. URL https://arxiv.org/abs/2602.16919. arXiv preprint arXiv:2602.16919.

Nima Anari, Gagan Goel, and Afshin Nikzad. Budget feasible procurement auctions. *Operations Research*, 66(3):637–652, 2018.

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Yaron Singer. Learning on a budget: Posted price mechanisms for online procurement. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC)*, pp. 128–145, 2012.

Dirk Bergemann and Juuso Välimäki. Dynamic mechanism design: An introduction. *Journal of Economic Literature*, 57(2):235–274, 2019. doi: 10.1257/jel.20180892. URL https://www.aeaweb.org/articles?id=10.1257/jel.20180892.

Federico Cacciamani, Martino Bernasconi, Matteo Castiglioni, and Nicola Gatti. Multi-agent contract design beyond binary actions, 2024. URL https://arxiv.org/abs/2402.13824. arXiv preprint arXiv:2402.13824.

Yang Cai, Constantinos Daskalakis, and Christos Papadimitriou. Optimum statistical estimation with strategic data sources. In *Proceedings of the 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pp. 280–296. PMLR, 2015. URL https://proceedings.mlr.press/v40/Cai15.html.

Roberto Calvo-Palomino, Domenico Giustiniano, Vincent Lenders, and Aymen Fakhreddine. Crowdsourcing spectrum data decoding. In *IEEE INFOCOM 2017 – IEEE Conference on Computer Communications*, pp. 1–4, 2017. doi: 10.1109/INFOCOM.2017.8057078.

Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Multi-agent contract design: How to commission multiple agents with individual outcomes. In *Proceedings of the 24th ACM Conference on Economics and Computation (EC)*, pp. 412–448, 2023.

Andreas Charalampopoulos, Dimitris Fotakis, Panagiotis Patsilinakos, and Thanos Tolias. A competitive posted-price mechanism for online budget-feasible auctions. In *Proceedings of the 26th ACM Conference on Economics and Computation (EC)*, pp. 1046–1075, 2025. doi: 10.1145/3736252.3742668.

Shuchi Chawla, Jason D. Hartline, and Robert Kleinberg. Algorithmic pricing via virtual valuations. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, pp. 243–251. ACM, 2007. doi: 10.1145/1250910.1250946. URL https://arxiv.org/abs/0808.1671.

Shuchi Chawla, Rojin Rezvan, Yifeng Teng, and Christos Tzamos. Pricing ordered items. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 722–735, 2022.

Keran Chen, Joon Suk Huh, and Kirthevasan Kandasamy. Learning to price homogeneous data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2407.05484.

Yiling Chen, Nicole Immorlica, Brendan Lucier, Vasilis Syrgkanis, and Juba Ziani. Optimal data acquisition for statistical estimation. In *Proceedings of the 2018 ACM Conference on Economics and Computation (EC)*, pp. 27–44, 2018. doi: 10.1145/3219166.3219195.

Vincent Conitzer and Tuomas Sandholm. Self-interested automated mechanism design and implications for optimal combinatorial auctions. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, pp. 132–141. ACM, 2004.

Alon Eden, Kira Goldner, Eldar Kerner, and Thodoris Tsilivis. From welfare to utility: Generalized objectives in budget-feasible procurement. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. arXiv:2607.00101.

Peter S Fader and Bruce GS Hardie. Probability models for customer-base analysis. *Journal of interactive marketing*, 23(1):61–69, 2009.

Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

Jason D. Hartline and Vladlen Koltun. Near-optimal pricing in near-linear time. In *Algorithms and Data Structures: 9th International Workshop, WADS 2005*, volume 3608 of *Lecture Notes in Computer Science*, pp. 422–431. Springer, 2005. doi: 10.1007/11534273_37.

Birgit Heydenreich, Rudolf Müller, Marc Uetz, and Rakesh V. Vohra. Characterization of revenue equivalence. *Econometrica*, 77(1):307–316, 2009. doi: 10.3982/ECTA7168. URL https://doi.org/10.3982/ECTA7168.

Chien-Ju Ho, Aleksandrs Slivkins, and Jennifer Wortman Vaughan. Adaptive contract design for crowdsourcing markets: Bandit algorithms for repeated principal-agent problems. *Journal of Artificial Intelligence Research*, 55:317–359, 2016.

Garud Iyengar and Anuj Kumar. Optimal procurement mechanisms for divisible goods with capacitated suppliers. *Review of Economic Design*, 12(2):129–154, 2008. doi: 10.1007/s10058-008-0046-7.

Felix Jarman and Vincent Meisner. Ex-post optimal knapsack procurement. *Journal of Economic Theory*, 171:35–63, 2017. doi: 10.1016/j.jet.2017.06.001.

Jean-Jacques Laffont and David Martimort. *The Theory of Incentives: The Principal-Agent Model*. Princeton University Press, 2002. URL https://press.princeton.edu/books/paperback/9780691091846/the-theory-of-incentives.

Elad Lavi, Hadas Shachnai, and Inbal Talgam-Cohen. Online contract design. *arXiv preprint arXiv:2602.07385*, 2026.

Eric Maskin and John Riley. Monopoly with incomplete information. *RAND Journal of Economics*, 15(2):171–196, 1984.

Michael Mussa and Sherwin Rosen. Monopoly and product quality. *Journal of Economic Theory*, 18(2):301–317, 1978.

Roger B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

Jean-Charles Rochet. A necessary and sufficient condition for rationalizability in a quasi-linear context. *Journal of Mathematical Economics*, 16(2):191–200, 1987.

Aaron Roth and Grant Schoenebeck. Conducting truthful surveys, cheaply. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC)*, pp. 826–843, 2012. doi: 10.1145/2229012.2229076.

Ilya Segal. Contracting with externalities. *The Quarterly Journal of Economics*, 114(2):337–388, 1999.

Ilya Segal. Coordination and discrimination in contracting with externalities: Divide and conquer? *Journal of Economic Theory*, 113(2):147–181, 2003. doi: 10.1016/S0022-0531(03)00114-5.

Yaron Singer. Budget feasible mechanisms. In *51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 765–774, 2010.

Brian L. Sullivan, Christopher L. Wood, Marshall J. Iliff, Rick E. Bonney, Daniel Fink, and Steve Kelling. eBird: A citizen-based bird observation network in the biological sciences. *Biological Conservation*, 142(10):2282–2292, 2009.

Michael W Weiner, Dallas P Veitch, Paul S Aisen, Laurel A Beckett, Nigel J Cairns, Robert C Green, Danielle Harvey, Clifford R Jack Jr, William Jagust, John C Morris, et al. Recent publications from the alzheimer's disease neuroimaging initiative: Reviewing progress toward improved ad clinical trials. *Alzheimer's & Dementia*, 13(4):e1–e85, 2017.

A Related Work

Screening under adverse selection Our problem belongs to the literature on *screening under adverse selection*, where a principal offers a menu to an agent with privately known characteristics. Contract theory encompasses both this setting and *moral hazard*, where the principal cannot directly contract on the agent's action; Laffont & Martimort (2002) develop both frameworks. In our model, production is verifiable and the agent's cost curve is private, so the incentive problem concerns screening costs through quantity choices. Our ordered-cost condition in §4 is a discrete increasing-differences condition, closely related to the single-crossing conditions used in screening theory. The associated monotonicity, reduction to adjacent incentive constraints, and information-rent adjustment of costs follow this classical approach.

Automated mechanism design Our computational approach in §4 and §6 connects to *automated mechanism design*, which optimizes mechanisms for explicitly specified preferences and type distributions (Conitzer & Sandholm, 2004). Fixing the quantity assigned to each type leaves a linear optimization problem over incentive-compatible payments. More generally, the interpretation of incentive constraints through cycles, difference constraints, and graph potentials is classical (Rochet, 1987; Heydenreich et al., 2009). Our general single-agent algorithm exploits the $(M+1)^K$ possible production profiles to obtain polynomial running time for fixed K. The static sequential setting requires an additional consideration: under one fixed menu, the principal's preferred choice among an agent's tied best responses may vary with accumulated production. The tree representation in §6 uses the graph structure of incentive constraints to obtain a finite family of candidate menus whose coverage accounts for these state-dependent choices.

Item pricing The single-agent problem is also closely connected to *unit-demand item pricing*. Hartline & Koltun (2005) give approximation schemes when the number of items is fixed, while Chawla et al. (2022) establish strong NP-hardness even when all buyer types share a common ordering of the items. The two-way, objective-preserving reduction in §3 transfers both results to our procurement problem. However, there are limitations to this approach. First, fixing the number of items corresponds to fixing the number of production levels M, which could be large under $K \ll M$, whereas our exact algorithms allow M to grow while fixing the number of types K. Second, our reduction does not necessarily induce a common ordering of the transformed values $V(m) - c_i(m)$, so ordered-item approximation results in Chawla et al. (2022) do not apply automatically. Another line of work uses virtual valuations to approximate unit-demand pricing under independently distributed item values (Chawla et al., 2007); but our finite distribution over cost curves can induce correlated values across production levels.

On the data-pricing side, Chen et al. (2024) study anonymous quantity-price curves for homogeneous data and finitely many buyer types. They exploit this structure to discretize pricing curves and learn from purchases when the type distribution is unknown. We share the distinction between the number of types and the number of quantities, but study exact procurement optimization under a known type distribution. In our sequential model, the principal's value depends on total acquired production, so successive transactions are also coupled through the value of the accumulated data.

Contract design Recent algorithmic work on *multi-agent contract design* often studies hidden actions that jointly determine observable outcomes. For example, Cacciamani et al. (2024) allow general action spaces and analyze deterministic contracts through Nash equilibria of the induced agent game. Such contracts condition payments on an observable outcome generated by the joint action profile, rather than directly on the unobservable actions. Individual-outcome payments also appear in this literature: Castiglioni et al. (2023) study separately observed agent outcomes, including a Bayesian setting with private types. Our setting combines verifiable production, private quantity-dependent costs, and an anonymous payment menu offered before the current agent's type is known. Conditional on this menu, the agent chooses her quantity independently of other agents' actions and the principal pays her accordingly. In the adaptive model, past production can affect the menu offered, but cannot subsequently change the promised payment for a chosen quantity.

Data procurement Our data-collection motivation connects to *strategic data procurement*. Abernethy et al. (2015) use sequential posted prices to procure data for learning under a budget, allowing acquisition costs to correlate with the data itself. Cai et al. (2015) design payments that induce

effort affecting data quality, including payments based on comparisons with other workers' data. Ananthakrishnan et al. (2026) study sellers with private data quality and per-sample costs, combining free-sample signaling with a subsequent procurement auction. These works address statistical learning, quality incentives, and information about data quality. Our model abstracts the benefit of homogeneous production into a known, potentially nonlinear valuation of total quantity and focuses on computing optimal quantity-payment menus for heterogeneous cost curves.

Dynamic mechanism design Finally, sequential procurement is related to *dynamic mechanism design*, which includes both evolving private information and changing populations of agents (Bergemann & Välimäki, 2019). Our model has a fresh, independently drawn agent in each round, and each agent participates once. Thus, the dynamic coupling comes from the principal's accumulated production and remaining horizon, while each agent faces a single menu-choice problem. The adaptive algorithm uses backward induction with a single-agent procurement problem at each state. Committing to a static menu instead couples these statewise decisions, motivating the finite candidate-menu representation used by our static algorithm.

B Proofs in §2

B.1 Structural properties of optimal payment menus

Our model does not require the payment curve π to be monotone. Nevertheless, when agent costs are nondecreasing, we may restrict attention to nondecreasing payment menus without loss of optimality.

Lemma 5. *For every payment menu π, there exists a nondecreasing payment menu $\bar{\pi}$ whose expected principal utility is at least that of π.*

Proof. For every $m \in \{0, \dots, M\}$, define

$$\bar{\pi}(m) := \max_{m' \le m} \pi(m').$$

Clearly, $\bar{\pi}$ is nondecreasing and $\bar{\pi}(0) = \pi(0)$.

Fix a type i, and let $u_i^* := \max_m\{\pi(m) - c_i(m)\}$ be her maximum utility under π. For any production level m, there exists $m' \le m$ such that $\bar{\pi}(m) = \pi(m')$. Since c_i is nondecreasing,

$$\bar{\pi}(m) - c_i(m) = \pi(m') - c_i(m) \le \pi(m') - c_i(m') \le u_i^*.$$

In other words, type i cannot achieve higher utility under $\bar{\pi}$ than under π. Now let m^* be a production level that achieves u_i^* under π. Since $\bar{\pi}(m) \ge \pi(m)$ for all m,

$$\bar{\pi}(m^*) - c_i(m^*) \ge \pi(m^*) - c_i(m^*) = u_i^*.$$

Thus m^* remains utility-maximizing under $\bar{\pi}$ and $\bar{\pi}(m^*) = \pi(m^*)$. Therefore, under principal-favorable tie-breaking, the principal can retain the same production level and payment as under π, and hence obtain at least the same utility. □

We can also rule out payment menus that pay more than the principal's value at any production level.

Lemma 6. *For every payment menu π, there exists a payment menu $\hat{\pi}$ satisfying $0 \le \hat{\pi}(m) \le V(m)$ for every m, whose expected principal utility is at least that of π.*

Proof. Define

$$\hat{\pi}(m) := \min\{\pi(m), V(m)\}.$$

Consider a type i. If the production level selected under π satisfies $\pi(m_i(\pi)) \le V(m_i(\pi))$, then its payment is unchanged, while payments at other production levels can only decrease. Hence it remains utility-maximizing under $\hat{\pi}$, and principal-favorable tie-breaking gives the principal at least as much utility as before.

Otherwise,

$$\pi(m_i(\pi)) > V(m_i(\pi)),$$

so the principal's utility from type i under π is negative. Under $\hat{\pi}$, $V(m) - \hat{\pi}(m) \ge 0$, so every production level m gives the principal nonnegative utility. Thus the principal's utility from this type increases. □

Since V is nondecreasing, applying Lemma 6 followed by Lemma 5 preserves the upper bound $\pi(m) \le V(m)$. Hence there always exists an optimal payment menu satisfying

$$0 = \pi(0) \le \pi(1) \le \cdots \le \pi(M), \qquad \pi(m) \le V(m).$$

C Proofs in §3

C.1 Proof of Theorem 1

From payment menus to item pricing. Given a production instance, retain its type distribution and define

$$v_i(m) = \max\{V(m) - c_i(m), 0\}, \qquad m \in \{0, \dots, M\}.$$

There is one item for every positive production level and an outside option $m = 0$. As in Section 2, we may restrict to menus satisfying $0 \le \pi(m) \le V(m)$, with $\pi(0) = 0$. For completeness, the upper bound can be enforced without decreasing expected principal utility: replace every payment above $V(m)$ by zero. For any type whose original choice gives nonnegative principal utility, that choice remains available at its original payment; any newly selected option either retains its payment and was already a utility-maximizer, or has payment zero and cost zero. In the latter case its original payment was strictly positive, so this switch could not displace an original best response with nonnegative utility. Favorable tie-breaking therefore preserves the principal's utility for such types. For a type whose original choice gives negative principal utility, every option after the modification gives nonnegative principal utility. Thus the modification cannot decrease the objective.

For a bounded menu define $p(m) = V(m) - \pi(m)$. If $V(m) \ge c_i(m)$, buyer and agent utilities coincide:

$$v_i(m) - p(m) = \pi(m) - c_i(m).$$

If $V(m) < c_i(m)$, agent utility is strictly negative, whereas buyer utility is $-p(m) \le 0$. Such an item can be a buyer's best response only when $p(m) = 0$ and the maximum utility is zero. It then contributes zero revenue, and the outside option is also a best response. Consequently, clipping negative valuations introduces only possible zero-revenue ties with the outside option. Since the principal and seller break ties to maximize $V(m) - \pi(m) = p(m)$, their optimal payoffs among best responses agree for every type. In particular, the mapped pricing has exactly the same expected revenue as the bounded menu's expected principal utility.

Conversely, all the constructed buyer valuations satisfy $v_i(m) \le V(m)$. We may therefore cap an arbitrary price at $V(m)$ without decreasing revenue. Indeed, an item priced above this bound gives every buyer strictly negative utility. After capping, it still gives nonpositive utility, so it can enter a best-response set only as a zero-utility option. For any buyer with maximum utility zero, the previous best responses remain available, and favorable tie-breaking ensures that revenue cannot decrease. After capping all prices, set $\pi(m) = V(m) - p(m)$. The preceding argument about clipped valuations again shows equality of the expected objectives. This proves equality of the optimal values and gives a conversion of any pricing solution without a decrease in its objective.

From item pricing to payment menus. Given an arbitrary unit-demand pricing instance, define

$$H = \max_{i \in [K],\, m \in [M]} v_i(m), \qquad V(m) = Hm, \qquad c_i(m) = Hm - v_i(m),$$

with $v_i(0) = V(0) = c_i(0) = 0$. If $H = 0$, both problems have optimal value zero, so suppose $H > 0$. For $m \ge 1$, $Hm \ge H \ge v_i(m)$, giving nonnegative costs. Also, for $m \in \{0, \dots, M-1\}$,

$$c_i(m+1) - c_i(m) = H - \big(v_i(m+1) - v_i(m)\big) \ge 0.$$

Thus the costs satisfy the production model even if the buyer valuations are not ordered. The principal's valuation is linear and nondecreasing.

Cap prices at H without decreasing revenue, by the same zero-utility tie argument as above. Then $\pi(m) = Hm - p(m) \ge 0$ for every positive production level, and $\pi(0) = 0$. For every type and every option,

$$\pi(m) - c_i(m) = v_i(m) - p(m), \qquad V(m) - \pi(m) = p(m).$$

The objectives therefore agree type by type, including favorable tie-breaking. Conversely, normalize any payment menu to satisfy $\pi \le V$, then set $p = V - \pi$; the same identities apply. All constructions

and normalizations take polynomial time and preserve the type distribution. The numerical values in the constructed production instance are at most HM, so their encoding lengths are polynomial in the input length.

C.2 Proofs of the Corollaries

Proof of Corollary 2. In *ordered item pricing*, all buyer types weakly prefer higher-ranked items:

$$0 \le v_i(1) \le \cdots \le v_i(M) \qquad \text{for every } i \in [K].$$

The seller otherwise solves the unit-demand item pricing problem defined in Section 3. Chawla et al. (Chawla et al., 2022, Theorem 4.5) prove that this problem is strongly NP-hard by a reduction from Max-Cut. Apply the pricing-to-production construction above with

$$H = \max_i v_i(M), \qquad V(m) = Hm, \qquad c_i(m) = Hm - v_i(m).$$

The resulting instance satisfies all production assumptions, has linear principal valuation, and has the same optimal objective value. The source hardness construction has polynomially bounded numerical parameters. Our transformation leaves the type probabilities unchanged and introduces valuations and costs of magnitude at most HM, preserving these bounds. Strong NP-hardness follows.

Proof of Corollary 3. Apply Theorem 1 to obtain a pricing instance with M items and K buyer types. It remains to explain why the approximation scheme of Hartline and Koltun Hartline & Koltun (2005) applies to weighted types. We give an explicit adaptation, using their price discretization and enumeration algorithm. This also avoids expanding rational type probabilities into a potentially much larger list of equally weighted buyers.

Discard zero-probability types. Let

$$h_i = \max_{m \in [M]} v_i(m), \qquad H = \max_i h_i, \qquad L = \max_i q_i h_i.$$

If $H = 0$, the zero menu is optimal. Otherwise $0 < L \le H$. Posting the same price h_i for every item earns at least $q_i h_i$, because type i buys at that price under favorable tie-breaking. Hence $L \le \mathrm{OPT}_{\mathrm{Pricing}}$.

Fix $\delta \in (0, 1)$. As before, we may cap prices at H. Raise every price below δL to δL. A type whose previously selected item cost at least δL still has that best response available, and increasing other prices cannot cause its revenue contribution to decrease under favorable tie-breaking. The total revenue previously contributed by the other types is at most

$$\sum_{i:\, p(m_i(p)) < \delta L} q_i \, \delta L \le \delta L \le \delta \, \mathrm{OPT}_{\mathrm{Pricing}} .$$

Thus some pricing in $[\delta L, H]^M$ earns at least $(1 - \delta)\,\mathrm{OPT}_{\mathrm{Pricing}}$. This is the weighted counterpart of the truncation step in Hartline and Koltun's Lemma 1.

Their Lemma 2 discretizes a price interval using geometric scales and a finer grid within each scale. Applied to $[\delta L, H]$, it gives a one-dimensional grid of size

$$s = O\left(\frac{M}{\delta} \log \frac{H}{\delta L}\right)$$

such that, for every price vector in this interval, some vector in the product grid preserves at least a $1/(1+\delta)$ fraction of its revenue. Their rounding proof establishes this guarantee separately for each buyer, so it continues to hold after multiplying by the nonnegative weights q_i and summing. Enumerating all s^M grid vectors and evaluating their expected revenues takes $O(KMs^M)$ arithmetic operations. Choose $\delta = \epsilon/4$. For $\epsilon \in (0, 1)$,

$$\frac{1-\delta}{1+\delta} \ge \frac{1}{1+\epsilon},$$

so the resulting pricing achieves the required approximation. The reduction converts it back to a payment menu with no loss in objective.

More explicitly, this implementation has running time

$$O\left(KM\left[\frac{CM}{\epsilon}\log\frac{4H}{\epsilon L}\right]^{M}\right)$$

for a universal constant C. If $q_{\min}$ is the smallest positive type probability, then $H/L \le 1/q_{\min}$. Therefore the logarithmic range factor is polynomial in the encoding length of the probabilities and in $\log(1/\epsilon)$. The algorithm is polynomial in the input length and $1/\epsilon$ for fixed M. As usual, a bit-complexity bound also includes polynomial factors for rational arithmetic.

For comparison, in Hartline and Koltun's original formulation with n equally weighted consumers and a fixed number M of items, their faster implementation has running time

$$O\left(n\log\left(\frac{1}{\epsilon}\log\frac{n}{\epsilon}\right)+\left(\frac{1}{\epsilon}\log\frac{n}{\epsilon}\right)^{M(M+1)}\right),$$

where constants may depend on the fixed M. It preprocesses an arrangement of hyperplanes and counts consumers in each cell. Weighted types can instead contribute their probabilities to these counters, with the price truncation adjusted as above. The simpler enumeration bound already suffices for our corollary and makes its dependence on M explicit.

D Proofs in §4.2

We give the full proof of Theorem 5. Throughout this appendix, each cost curve $c_i : \{0,\dots,M\} \to \mathbb{R}_{\ge 0}$ is nondecreasing with $c_i(0) = 0$, and $q_i > 0$ for every type WLOG. The types satisfy the ordered-cost condition (4):

$$c_i(m') - c_i(m) \le c_j(m') - c_j(m), \qquad i < j, \quad 0 \le m < m' \le M.$$

The same inequality holds trivially when $m = m'$. Taking $m = 0$ also gives $c_i(m') \le c_j(m')$ whenever $i < j$. We first establish monotonicity of profiles through Lemma 1.

Proof of Lemma 1. For each type i, choose $m_i := \max \operatorname{argmax}_{m\in\widetilde{m}_i(\pi)}\big[V(m)-\pi(m)\big]$. In other words, any remaining ties in the principal's utility are resolved in favor of the largest production level. This choice is well-defined because the set of best responses $\widetilde{m}_i(\pi)$ is finite, and it attains the principal's highest utility for that type.

Consider any two types $i < j$, and suppose for a contradiction that $m_i < m_j$. Their incentive constraints imply that they don't prefer each other's designated level over their own:

$$\pi(m_i) - c_i(m_i) \ge \pi(m_j) - c_i(m_j), \qquad \pi(m_j) - c_j(m_j) \ge \pi(m_i) - c_j(m_i).$$

Rearranging gives $c_j(m_j) - c_j(m_i) \le \pi(m_j) - \pi(m_i) \le c_i(m_j) - c_i(m_i)$. Condition (4) gives the reverse inequality between the two endpoints for $m = m_i$ and $m' = m_j$, so all three expressions are equal. Consequently, both m_i and m_j are best responses for both types. The principal chooses m_i over m_j for i, and m_j over m_i for j. These choices imply $V(m_i) - \pi(m_i) = V(m_j) - \pi(m_j)$. However, since we assumed $m_j > m_i$, choosing m_i instead of m_j for type i contradicts how the principal breaks ties. Therefore $m_i \ge m_j$. □

Note that an arbitrary implementable profile need not be monotone, but Lemma 1 provides the property needed for optimization: every menu has a monotone selection attaining its principal-favorable utility. Given this, it is sufficient to enumerate all monotone profiles. Next, we give a characterization of the cheapest implementing payments through Lemma 2.

Proof of Lemma 2. First consider any payment menu implementing $\boldsymbol{m}$. The incentive constraint comparing type i's designated production m_i with m_{i+1} requires

$$\pi(m_i) - c_i(m_i) \ge \pi(m_{i+1}) - c_i(m_{i+1}) \iff \pi(m_i) - \pi(m_{i+1}) \ge c_i(m_i) - c_i(m_{i+1}).$$

For the highest-cost type, comparison with production level 0, together with $c_K(0) = 0$ and non-negative payments, gives $\pi(m_K) - c_K(m_K) \ge \pi(0) - c_K(0) \ge 0$. Setting these inequalities to

equality gives exactly (5). Starting with type K and applying the lower bounds recursively shows that these payments are componentwise smallest among any implementing payment menu.

It remains to show that these payments are actually feasible, since their definition only satisfies a subset of incentive constraints. For any $i < j$, telescoping (5) gives

$$\pi(m_i) - \pi(m_j) = \sum_{\ell=i}^{j-1} \big[c_\ell(m_\ell) - c_\ell(m_{\ell+1})\big].$$

For each $\ell = i, \dots, j-1$, monotonicity of the profile and (4) imply

$$c_i(m_\ell) - c_i(m_{\ell+1}) \le c_\ell(m_\ell) - c_\ell(m_{\ell+1}) \le c_j(m_\ell) - c_j(m_{\ell+1}).$$

Note that the sums on the left and right telescope. Therefore, summing over ℓ gives

$$c_i(m_i) - c_i(m_j) \le \pi(m_i) - \pi(m_j) \le c_j(m_i) - c_j(m_j).$$

The first inequality ensures that type i weakly prefers m_i to m_j, while the second ensures that type j weakly prefers m_j to m_i. Thus all incentive constraints between designated production levels are satisfied. They also imply, for every type i,

$$\pi(m_i) - c_i(m_i) \ge \pi(m_K) - c_i(m_K) = c_K(m_K) - c_i(m_K) \ge 0,$$

where the last inequality follows from the ordering of the cost curves. Thus each designated choice is at least as good as the outside option.

Finally, we can extend the designated payments to a payment menu on every production level by

$$\pi_{\boldsymbol{m}}(m) := \begin{cases} \pi(m_i), & \text{if } m = m_i \text{ for some } i \in [K], \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

The recursion assigns equal payments to repeated production levels, so this menu is well-defined. All designated payments are nonnegative by (5), hence the entire menu is nonnegative. Since every designated choice gives nonnegative utility and every undesignated level pays zero, no type strictly prefers an undesignated level to her designated choice. Together with the incentive constraints above, this shows that $\pi_{\boldsymbol{m}}$ implements $\boldsymbol{m}$. The payments in (5) therefore achieve the minimum possible expected payment. □

Lemma 2 allows us to eliminate the payment variables entirely. Expanding the recursion gives

$$\pi(m_i) = c_i(m_i) + \sum_{j=i+1}^{K} \big[c_j(m_j) - c_{j-1}(m_j)\big].$$

Let $Q_i := \sum_{j=1}^{i} q_j$ for $i = 0, \dots, K$, where the empty sum gives $Q_0 = 0$, and use the convention $c_0 \equiv 0$. Interchanging the finite sums, the expected payment at the designated levels under the cheapest menu implementing $\boldsymbol{m}$ is

$$\begin{aligned} \sum_{i=1}^{K} q_i \pi(m_i) &= \sum_{i=1}^{K} q_i c_i(m_i) + \sum_{j=2}^{K} \left(\sum_{i=1}^{j-1} q_i \right) \big[c_j(m_j) - c_{j-1}(m_j)\big] \\ &= \sum_{i=1}^{K} \Big[q_i c_i(m_i) + Q_{i-1}\big(c_i(m_i) - c_{i-1}(m_i)\big) \Big], \end{aligned} \tag{10}$$

where the second term vanishes for $i = 1$ because $Q_0 = 0$. Define the *virtual cost curves* by

$$\psi_i(m) := c_i(m) + \frac{Q_{i-1}}{q_i} \big[c_i(m) - c_{i-1}(m)\big], \qquad i = 1, \dots, K.$$

In particular, $\psi_1 = c_1$. Equation (10) becomes $\sum_{i=1}^{K} q_i \pi(m_i) = \sum_{i=1}^{K} q_i \psi_i(m_i)$. Consequently, the optimization over monotone profiles is

$$\max_{m_1 \ge \cdots \ge m_K} \sum_{i=1}^{K} q_i \big[V(m_i) - \psi_i(m_i)\big], \qquad m_i \in \{0, \dots, M\}. \tag{11}$$

Algorithm Note that (11) has an objective separable across types, with the monotonicity constraint providing the only interaction between their production levels. This structure admits the following dynamic program. For $i \in [K]$ and $m \in \{0, \dots, M\}$, let $F_i(m)$ denote the maximum contribution from types $1, \dots, i$ over monotone profiles with type i producing exactly m. The base case is $F_1(m) = q_1\big[V(m) - \psi_1(m)\big]$, and for $i \geq 2$, the Bellman equation is

$$F_i(m) = q_i\big[V(m) - \psi_i(m)\big] + \max_{m' \geq m} F_{i-1}(m'). \tag{12}$$

The optimal objective value is $\max_{m \in \{0,\dots,M\}} F_K(m)$. To justify the recurrence, any monotone profile counted by $F_i(m)$ has a preceding level $m_{i-1} = m' \geq m$. Its first $i-1$ types contribute at most $F_{i-1}(m')$. Conversely, choosing m' attaining the maximum in (12) and appending m to a maximizing profile for $F_{i-1}(m')$ preserves monotonicity and attains the right-hand side. Together with the base case, this proves the recurrence by induction.

For each layer $i \geq 2$, define the suffix maxima $S_{i-1}(m) := \max_{m' \in \{m,\dots,M\}} F_{i-1}(m')$. They can be computed by scanning backward: initialize $S_{i-1}(M) = F_{i-1}(M)$, then set

$$S_{i-1}(m) = \max\{F_{i-1}(m), S_{i-1}(m+1)\}, \qquad m = M-1, \dots, 0.$$

Each entry $F_i(m)$ can then be computed in constant time using $S_{i-1}(m)$. Thus a layer takes $\mathcal{O}(M)$ arithmetic operations, and the entire dynamic program takes $\mathcal{O}(KM)$. Standard backtracking recovers a maximizing production profile in $\mathcal{O}(K)$ time. The designated payments and the payment menu are then reconstructed using (5) and (9) respectively.

Proof of Theorem 5. Write $U(\pi)$ for the principal's expected utility under principal-favorable tie-breaking, and consider any payment menu π. By Lemma 1, there is a monotone production profile $\boldsymbol{m}$ attaining its principal-favorable utility $U(\pi)$. By Lemma 2, the cheapest menu implementing this profile has expected payment no larger than that of π. Its profile payoff is therefore at least $U(\pi)$. Thus the maximum in (11) is at least the utility of every payment menu.

The dynamic program (12) optimizes over all monotone production profiles. For its maximizing profile $\widehat{\boldsymbol{m}}$, Lemma 2 constructs a menu $\pi_{\widehat{\boldsymbol{m}}}$ implementing every designated choice. Every designated level is a best response under the constructed menu, so principal-favorable tie-breaking gives

$$U(\pi_{\widehat{\boldsymbol{m}}}) \geq \sum_{i=1}^{K} q_i\big[V(\widehat{m}_i) - \pi_{\widehat{\boldsymbol{m}}}(\widehat{m}_i)\big] = \sum_{i=1}^{K} q_i\big[V(\widehat{m}_i) - \psi_i(\widehat{m}_i)\big].$$

Here $\widehat{m}_i$ is type i's designated level in $\widehat{\boldsymbol{m}}$; the equality follows from (10). Since the RHS is already at least the utility of every menu, $\pi_{\widehat{\boldsymbol{m}}}$ is optimal.

Computing the cumulative probabilities takes $\mathcal{O}(K)$ arithmetic operations, and evaluating all virtual costs takes $\mathcal{O}(KM)$. Together with the dynamic program and reconstruction above, this gives $\mathcal{O}(KM)$ arithmetic operations. □

E Example in §5

The following example shows that offering the same menu to all agents simultaneously can reduce the principal's optimal expected utility as the number of agents increases, even under joint principal-favorable tie-breaking.

Example 2. *Let $M = 2$, and suppose each agent independently has type L (low cost) or H (high cost) with equal probability. The costs and the principal's valuation are*

m	0	1	2
$c_L(m)$	0	0.3	0.4
$c_H(m)$	0	0.4	0.5

$$V(z) = \min\{z/2, 1\}.$$

With one agent, the menu $\pi(0) = \pi(1) = 0$ and $\pi(2) = 0.5$ induces two units from either type under principal-favorable tie-breaking, yielding utility 0.5. With two simultaneous agents, however, the optimal expected utility is only 0.45, even if the principal chooses the agents' best responses jointly.

To verify this, first consider a menu under which positive production is a best response for type H. The principal must then pay at least 0.4 for one unit or 0.5 for two units. Since type L costs 0.1 less at either positive quantity, her maximum utility is at least $\pi(0) + 0.1$. Thus, every best response for her is positive and requires payment at least 0.4 for one unit or 0.5 for two units. For any type realization, the principal's utility is at most 0.5: collecting at most one unit gives valuation at most 0.5, while collecting two or more units requires total payment at least 0.5. When both agents have type L, both produce positively and each receives payment at least 0.4, yielding utility at most 0.2. The expected utility is therefore at most $\frac{1}{4}(0.2) + \frac{3}{4}(0.5) = 0.425$.

Otherwise, only type L can produce positively. The probability of having at least one type-L agent is $3/4$. In that event, the principal's utility is at most 0.6: collecting at most one unit gives valuation at most 0.5, while collecting two or more units requires total payment at least 0.4. If neither agent has type L, her utility is nonpositive. This gives an expected utility upper bound of 0.45. It is attained by $\pi(0) = \pi(1) = 0$ and $\pi(2) = 0.4$: whenever a type-L agent is present, the principal selects two units from one such agent and zero from the other agent; otherwise, both produce zero.

The example combines bounded principal valuation with concave cost curves. Excluding type H leads to a loss in expected utility, but a payment sufficient to attract type H makes positive production strictly preferable to zero production for type L. Several low-cost agents may then produce more than the principal needs, while splitting the desired quantity among them repeats the more expensive first unit. Such decreasing marginal costs can arise in data collection when an agent must first configure equipment or access a data source, after which additional samples are cheaper to collect. Thus, the principal can be worse off when multiple agents arrive simultaneously and she must honor the menu for every agent.

F Proofs in §6.2

This section gives the full proof of Theorem 8. We give expanded statements of the two lemmas to specify the replacement LP and the candidate family used in their proofs.

Write $\mathcal{M} = \{0, \dots, M\}$ and, for a menu π, define

$$u_i(\pi) := \max_{m \in \mathcal{M}} \{\pi(m) - c_i(m)\}, \qquad \widetilde{m}_i(\pi) := \arg\max_{m \in \mathcal{M}} \{\pi(m) - c_i(m)\}.$$

Let $J(\pi)$ be the principal's optimal expected utility under menu π, with optimal stopping rule and tie-breaking decisions.

Replacing an arbitrary menu. For a utility vector $u \in \mathbb{R}^K_{\geq 0}$, define its envelope menu by (7):

$$\pi_u(m) := \min_{j \in [K]} \{c_j(m) + u_j\}.$$

This menu is nonnegative and gives type i utility at most u_i at every production level, since $\pi_u(m) \leq c_i(m) + u_i$ for all m. Any level m satisfying $\pi_u(m) = c_i(m) + u_i$ gives type i utility exactly u_i and is therefore a best response for her.

Expanded statement of Lemma 3. Fix an arbitrary menu π^0, and write $u_i^0 = u_i(\pi^0)$ and $\widetilde{m}_i^0 = \widetilde{m}_i(\pi^0)$. Consider the linear program

$$\begin{aligned} \min \quad & \sum_{i=1}^{K} u_i \\ \text{s.t.} \quad & 0 \leq u_i \leq u_i^0 && \forall i \in [K], \\ & c_i(m) + u_i \leq c_j(m) + u_j && \forall i, j \in [K],\ m \in \widetilde{m}_i^0. \end{aligned} \tag{13}$$

This program has a minimizer $\widehat{u}$ with $\min_i \widehat{u}_i = 0$. Its envelope menu $\widehat{\pi} = \pi_{\widehat{u}}$ satisfies

$$u_i(\widehat{\pi}) = \widehat{u}_i, \qquad \widetilde{m}_i^0 \subseteq \widetilde{m}_i(\widehat{\pi}), \qquad \widehat{\pi}(m) \leq \pi^0(m) \quad \forall i \in [K], m \in \widetilde{m}_i^0.$$

Consequently, $J(\widehat{\pi}) \geq J(\pi^0)$.

Proof of Lemma 3. The original utility vector u^0 is feasible for (13). Indeed, whenever $m \in \widetilde{m}_i^0$, m is a best response for i under π^0 but not necessarily for j. Thus,

$$c_i(m) + u_i^0 = \pi^0(m) \le c_j(m) + u_j^0 \qquad \forall j.$$

The feasible region is nonempty and compact, so a minimizer $\widehat{u}$ exists. If all its coordinates were positive, subtracting $\min_i \widehat{u}_i$ from every coordinate would preserve every difference constraint and every bound while decreasing the objective. Hence $\min_i \widehat{u}_i = 0$, and $\widehat{\pi}$ is nonnegative.

Now fix a type i and an original best response $m \in \widetilde{m}_i^0$. The corresponding constraints in (13) ensure that the minimum defining $\widehat{\pi}(m)$ is attained by type i. Therefore:

$$\widehat{\pi}(m) = c_i(m) + \widehat{u}_i \le c_i(m) + u_i^0 = \pi^0(m). \tag{14}$$

By (7), every production level gives type i utility at most $\widehat{u}_i$ under $\widehat{\pi}$, while every level in the nonempty set $\widetilde{m}_i^0$ gives exactly $\widehat{u}_i$. Therefore, $u_i(\widehat{\pi}) = \widehat{u}_i$ and $\widetilde{m}_i^0 \subseteq \widetilde{m}_i(\widehat{\pi})$, *i.e.*, every old best response for type i remains a best response under $\widehat{\pi}$.

To compare the principal's utilities, consider any policy under π^0. Before closing the market, all of its production choices remain available under $\widehat{\pi}$, and each has a weakly lower payment by (14). The principal can therefore reproduce the same tie-breaking choices and stopping rule. For every realization of the arriving types, the reproduced policy obtains the same terminal production with no larger total payment. Optimizing the policy under $\widehat{\pi}$ can only improve the principal's utility. Thus $J(\widehat{\pi}) \ge J(\pi^0)$. □

The preceding lemma preserves each type's original best-response set. It therefore applies even when the same type is assigned different production levels at different rounds. The linear program (13) is used to prove that a suitable replacement exists; the algorithm will enumerate a family containing such a replacement without knowing π^0 or its best-response sets.

Constructing the candidate family. We first describe a finite enumeration using only information from the cost curves. Each enumerated item, given below, will specify exactly one candidate vector u. Then in Lemma 4, we show that the collection of these u's must include a minimizer $\widehat{u}$ for the LP derived from any π^0.

1. Enumerate every spanning tree $\mathcal{T}$ on $K+1$ nodes $\{\star, 1, \dots, K\}$, and orient its edges away from the root $\star$.
2. Assign a production label $m \in \mathcal{M}$ to every edge in the tree, except for edges incident to the root.
3. Fix $u_\star = 0$ and solve the tree equations:

$$\begin{aligned} u_i &= 0 && \text{if } \star \to i \text{ is a tree edge,} \\ u_j &= u_i + c_i(m) - c_j(m) && \text{if } i \xrightarrow{m} j \text{ is a tree edge.} \end{aligned} \tag{15}$$

4. Discard u if any coordinate is negative. Otherwise retain u and its envelope menu π_u.

By basic tree properties, there is a unique path from the root to any node. Thus every tree with its assigned production labels has exactly one solution u to (15). Different labeled trees may produce the same vector.

Let $\mathcal{U}$ be the set of retained vectors and $\mathcal{P} = \{\pi_u : u \in \mathcal{U}\}$ be the corresponding menu family. By Cayley's formula, there are $(K+1)^{K-1}$ spanning trees on $K+1$ vertices. If the root has degree d, then only $K-d$ edges have production labels, giving $(M+1)^{K-d}$ production labelings. Since $d \ge 1$, the enumeration generates at most $(K+1)^{K-1}(M+1)^{K-1}$ candidates before discarding negative vectors or removing duplicates.

The candidate utility vectors parameterize menus; they need not satisfy the utility LP for a prescribed original menu π^0. The following lemma shows that the enumeration includes the minimizer $\widehat{u}$ associated with every π^0.

Expanded statement of Lemma 4. For every menu π^0, the minimizer $\widehat{u}$ constructed in the proof of Lemma 3 belongs to $\mathcal{U}$. Consequently, $\widehat{\pi} = \pi_{\widehat{u}}$ belongs to $\mathcal{P}$ and satisfies $J(\widehat{\pi}) \geq J(\pi^0)$.

Proof of Lemma 4. Fix π^0 and its minimizer $\widehat{u}$. We will find one tree and production labeling in the enumeration whose equations have solution exactly $\widehat{u}$. Represent the lower-bound constraints in (13) by a directed graph on $\{\star, 1, \ldots, K\}$. Include a root edge $\star \to i$ for $u_i \geq 0$, and, for distinct types i, j and every $m \in \widetilde{m}_i^0$, an edge $i \xrightarrow{m} j$ for $u_j \geq u_i + c_i(m) - c_j(m)$. Call an edge *tight* if its inequality holds with equality at $\widehat{u}$. We claim that every type is reachable from $\star$ through tight edges induced by $\widehat{u}$.

Let R be the vertices reachable from $\star$ using tight edges. Suppose some types are outside R, and denote their set by S. Every $i \in S$ has $\widehat{u}_i > 0$; otherwise the tight edge $\star \to i$ would put it in R. Moreover, every constraint edge from a type in R to a type in S has positive slack. A tight edge of this form would make its endpoint reachable as well.

Decrease every coordinate in S by the same sufficiently small $\varepsilon > 0$, leaving the other coordinates unchanged. This preserves nonnegativity because all coordinates in S are positive. It also preserves every difference constraint: if both endpoints are in S, both sides decrease by ε; if both are outside S, neither side changes; if the edge leaves S, its right-hand side decreases, which preserves the "$\geq$" inequality; and if the edge enters S, its initially positive slack absorbs a sufficiently small decrease. There are finitely many constraints, so one positive ε can satisfy all these requirements. All upper bounds with respect to u_i^0 remain satisfied as well. The resulting vector is feasible for (13), but its objective is smaller by $|S|\varepsilon$. This contradicts the optimality of $\widehat{u}$. Hence every type is reachable from $\star$ through tight edges.

Starting at $\star$, repeatedly choose a tight edge from an already reached vertex to a new vertex. Since every vertex is reachable, this constructs a directed spanning tree. Each edge between types carries a production label from $\mathcal{M}$. The enumeration therefore includes this tree with these labels. All its edges are tight at $\widehat{u}$, so $\widehat{u}$ satisfies its tree equations. The tree equations have a unique solution, so the candidate utility vector computed for this tree is exactly $\widehat{u}$. Since $\widehat{u} \geq 0$, it is retained in $\mathcal{U}$. Its menu belongs to $\mathcal{P}$, and Lemma 3 gives the asserted utility guarantee. □

Evaluating the candidate menus. For each $\pi \in \mathcal{P}$, compute its best-response sets $\widetilde{m}_i(\pi)$ and evaluate it by dynamic programming. The algorithm returns the candidate menu with the largest principal utility.

Let $F_t^\pi(z)$ be the optimal expected terminal valuation minus future payments after t completed rounds, with cumulative production z and the market still open. Past payments are sunk costs, and future types are independent of the history, so (t, z) suffices to describe the continuation state.

While the market remains open, the principal chooses whether to close it or retain the menu for the next round. The Bellman equations are $F_T^\pi(z) = V(z)$, and for $t < T$:

$$F_t^\pi(z) = \max\left\{ V(z),\ \sum_{i=1}^{K} q_i \max_{m \in \widetilde{m}_i(\pi)} \left[F_{t+1}^\pi(z+m) - \pi(m) \right] \right\}.$$

The first term closes the market immediately. The second keeps it open for the next arrival, with the inner maximum selecting the arriving agent's best response that gives the largest continuation utility to the principal. Thus $J(\pi) = F_0^\pi(0)$. Recording the maximizers gives the optimal market-closing and tie-breaking policies for this menu.

Proof of Theorem 8. Let $\mathcal{P}$ be the finite candidate family produced by the enumeration, and return a menu maximizing J over $\mathcal{P}$. For any menu π^0, Lemma 3 gives a replacement $\widehat{\pi}$ with $J(\widehat{\pi}) \geq J(\pi^0)$. Lemma 4 ensures that $\widehat{u} \in \mathcal{U}$ and $\widehat{\pi} \in \mathcal{P}$. Therefore $\max_{\pi \in \mathcal{P}} J(\pi) \geq J(\widehat{\pi}) \geq J(\pi^0)$. Since this holds for every π^0, the returned menu is optimal.

Recall that the enumeration generates at most $(K+1)^{K-1}(M+1)^{K-1}$ candidates. Constructing each menu and its best-response sets takes $\mathcal{O}(K(M+1))$ operations. The dynamic program has $\mathcal{O}(T^2 M)$ states and uses $\mathcal{O}(K(M+1))$ operations per state. Thus each candidate can be evaluated in $\mathrm{poly}(K, M, T)$ time, proving the stated bound. □