

Batched Pandora’s Box

Shaddin Dughmi ^{*} Yusuf Hakan Kalayci [†] Vasilis Livanos [‡] Aditya Prasad [§]

Abstract

Motivated by numerous parallelizable stochastic search problems, most notable and timely among them being LLM inference-time scaling, we propose and study batched versions of the Pandora’s Box problem of Weitzman. In particular, boxes are opened in capacity-constrained batches, each batch has a setup cost, and all rewards in a batch are revealed together. We consider two different variants, motivated by different application environments: one where boxes are reusable (i.e., can provide multiple i.i.d. samples) and another where they are not. For both variants we rule out most “simple” natural heuristics, and also formally prove NP-hardness of approximation in the traditional sense. We then relax the problem to allow bi-criteria approximations, with respect to both rewards and setup costs, where we exhibit constant approximation algorithms for both the reusable and non-reusable settings. This is obtained through a linear-programming relaxation of Pandora’s Box problem, followed by randomized or Pipage rounding.

1 Introduction

In several stochastic search problems, before we find a solution of sufficient quality we need to pay a search cost. In marketing, a platform tests different advertisements before committing to the best-performing one; in hyper-parameter tuning, a practitioner trains their model with a menu of settings before selecting a model; in inference-time reasoning, an inference system issues multiple LLM queries or decoding strategies before choosing the best response. The classical Pandora’s Box problem, originally studied by Weitzman [38], captures the sequential version of this tradeoff. Each option has an inspection cost and an unknown reward, and the decision maker adaptively decides which option to inspect next and when to stop.

In many of these applications, however, inspections are naturally parallelized. Ad variants in A-B testing are exposed to users in cohorts, experiments are tested on a pool of machines, and queries to an LLM are submitted together to reduce latency. This parallelism creates a batched information structure, as the policy must commit to a set of inspections before seeing any of their outcomes. A batch may also have a setup cost, capturing the overhead of launching a new experimental framework, allocating computing resources, or coordinating a marketing campaign. Thus batching trades off the cost of latency with the savings provided by adaptivity.

^{*}University of Southern California, Email: shaddin@usc.edu. Supported by the Air Force Office of Scientific Research under award number FA9550-24-1-0261. This work was done in part while the author was on sabbatical as the Carter and Tania Neild visiting professor at Northwestern University, as well as a visiting professor in the Data Science Institute at the University of Chicago.

[†]University of Southern California, Email: kalayci@usc.edu. Supported by the Air Force Office of Scientific Research under award number FA9550-24-1-0261.

[‡]University of Southern California, Email: vas.livanos@gmail.com. Supported by the Air Force Office of Scientific Research under award number FA9550-24-1-0261. This work was done in part while the author was visiting Northwestern University.

[§]University of Chicago, Email: adityaprasad@uchicago.edu.

While batched feedback has been studied in bandit problems [31, 18, 23, 36], it has seen comparatively less study in the Pandora’s Box problem. We initiate this direction by studying a batched version of the Pandora’s Box problem. In the *Batched Pandora’s Box model*, we are given n boxes, each with a distribution of a hidden reward V_i , as well as an opening cost c_i . A policy groups boxes to create the next batch to be opened, and a feasible batch contains at most k boxes for a given batch *capacity* k and costs a fixed setup cost T to open in addition to the individual opening costs. Once a batch is opened, all rewards in the batch are revealed simultaneously. One can then choose to stop and obtain the maximum reward observed over all opened batches or continue to open another batch.

We introduce two variants of this model: The *non-reusable setting* where an opened box is consumed and cannot be opened again, and the *reusable setting* where each box can be opened multiple times and provides a fresh i.i.d. sample each time. The distinction matches the difference between one-time opportunities, such as allocating a consumer cohort to an ad variant, and repeatable experiments, such as running additional inference trials of the same LLM. Although this model is a direct analogue of the classical problem with delayed feedback, the resulting optimization problem differs significantly from classical Pandora’s Box, requiring us to examine which elements of Weitzman’s solution to the classical problem carry over to the batched model.

1.1 Our Contributions

We organize our results around three questions raised by batched feedback.

What survives of Weitzman’s index rule under batched feedback? The answer depends on reusability. For non-reusable boxes, adaptive batch formation is useful: a policy can choose later batches after seeing earlier outcomes and thereby outperform every fixed menu of disjoint batches. This advantage is bounded, however. We prove that every adaptive non-reusable policy has a fixed-menu counterpart with at least half its value, using a random root-to-leaf path of the adaptive decision tree in the spirit of stochastic probing [25, 10]. For reusable boxes, we can recover Weitzman’s index structure. Each feasible batch $\mathbf{a}$ has a reservation value $\sigma(\mathbf{a})$, and an optimal policy repeatedly opens a maximum-reservation batch until its realized value exceeds that threshold.

Is exact or ordinary approximation tractable? The answer is negative, in a strong sense. Specifically, exact optimization is NP-hard for the non-reusable problem for every fixed finite batch capacity $k \geq 3$. Moreover, with variable batch capacity, and even with $k = \infty$, both the non-reusable and reusable models have a sign gap: it is NP-hard to distinguish $\text{OPT} = 0$ from $\text{OPT} > 0$. The reductions come from SUBSET PRODUCT [19], where the input consists of integers $b_1, \dots, b_n$ and a target Q , and the question is whether some subset has product exactly Q . For each b_i , we create a box with a two-point distribution that gives value M with probability $1 - 1/b_i$ and value 0 with probability $1/b_i$. Hence a batch corresponding to a subset S fails with probability $1/\prod_{i \in S} b_i$. By calibrating the setup and item costs, we make it so that a batch has positive surplus exactly when the product of the constituent boxes’ b_i ’s is the target Q . Therefore, finding any positive-value policy certifies a YES instance of SUBSET PRODUCT. This sign gap rules out ordinary multiplicative approximation under the original costs, since any positive approximation guarantee would distinguish zero-optimum instances from positive-optimum instances.

Can a discounted-cost relaxation recover useful algorithmic guarantees? The sign gap motivates a bi-criteria benchmark: compare against the original-cost optimum, but evaluate the

returned policy with discounted costs. For a policy π , with maximum observed value M_π and total cost C_π , let

$$\text{val}_\lambda(\pi) = \mathbb{E}[M_\pi] - \lambda \mathbb{E}[C_\pi].$$

An (α, λ) discounted-cost bi-criteria guarantee means that, for every $\varepsilon > 0$, there is an algorithm returning an *efficient* policy $\hat{\pi} = \hat{\pi}(\varepsilon)$ with

$$\text{val}_\lambda(\hat{\pi}) \geq \alpha \text{OPT}_1 - \varepsilon,$$

where OPT_1 is the expected value of the optimal adaptive policy with the original (non-discounted) costs. For finite-support distributions, our algorithms for both variants are based on rounding a linear programming relaxation of the batched problem. In the non-reusable model, the LP records opening probabilities, expected batch count, and probabilities that each box is the maximum in a batch, and upper bounds every nonempty adaptive policy. In a solution of this LP, the batch count can be fractional, and so we need to round it first to an integral value. Then, we use dependent rounding to partition the boxes into legal batches. We incur a loss of $2(\sqrt{2} - 1)$ from the batch count rounding and of $1 - 1/e$ from the correlation gap of the dependent rounding scheme, yielding in total

$$\text{val}_{\alpha_{\text{NR}}}(\hat{\pi}) \geq \alpha_{\text{NR}} \text{OPT}_1 - \varepsilon, \quad \text{where} \quad \alpha_{\text{NR}} = (1 - 1/e)2(\sqrt{2} - 1).$$

In the reusable model, we reduce the problem to finding a single batch, corresponding to a multi-set of boxes, to sample repeatedly until another sample no longer pays for its cost. We design a feasibility LP RLP_τ based on a candidate reservation value of the desired batch that is feasible when there is a batch certifying a reservation value at least τ . Binary search finds the largest certified threshold up to a small additive error, and independent slot rounding turns the certificate into an integral batch. In total, we only incur a rounding loss of $\alpha_{\text{R}} = 1 - 1/e$, giving a stronger bound of

$$\text{val}_{\alpha_{\text{R}}}(\hat{\pi}) \geq \text{OPT}_1 - \varepsilon.$$

1.2 Related Work

Pandora’s Box and costly inspection. Weitzman’s seminal work [38] initiated study of the Pandora’s Box setting and showed that independent boxes with inspection costs admit an optimal reservation-value policy. Subsequent work extends this costly-search view in several directions, including generalized objectives [30], market-search coordination [28], nonobligatory inspection [7, 17, 6, 33], combinatorial or online feasibility constraints [34, 8, 5, 9, 13, 12], and correlated or sample-access variants [15, 14, 22]. Another recent line studies online and learning variants of Pandora’s Box under online-arrival, contextual, bandit, semi-bandit, and LLM-motivated feedback models [16, 21, 3, 20, 1, 29, 27, 4]. These works mainly change the distributional knowledge, feedback available across repeated rounds, contextual structure, or feasible selections. In contrast, our model is a known-distribution, single-instance Pandora’s Box problem with independent rewards. We keep the best-observed-value objective and instead change the feedback structure: inspections must be launched in capacity-limited batches with setup costs.

Stochastic probing and rounding. Our structural and algorithmic tools also connect to stochastic probing and submodular rounding. Stochastic probing studies the value of adaptivity when random elements are probed subject to feasibility constraints, including constant adaptivity gaps for submodular and XOS objectives and near-optimal results for multi-value probing [25, 10]. Our non-reusable adaptivity-gap proof uses the same random-path perspective, but with Pandora-specific accounting through reservation values, capped rewards, and fixed-menu Pandora policies.

The LP algorithms use standard correlation-gap and dependent-rounding ideas, including pipage rounding and correlation-robust stochastic optimization [11, 2].

Delayed feedback. Delayed feedback has been studied as a natural model in bandits and online learning. One line of work studies how late observations affect regret in stochastic, Bayesian, or adversarial bandits, including general delay reductions, delayed conversions, anonymous aggregated rewards, and unrestricted delays [24, 26, 37, 32, 35, 39]. Another line studies batched or parallel experimentation, where actions are chosen in groups before feedback from the group is available, for regret minimization, best-arm identification, or Bayesian optimization [31, 18, 23, 36].

These works share the practical obstruction that decisions may have to be made before earlier outcomes are observed, but their objective is typically statistical: regret or sample complexity. Our model is a single-instance search problem. Here delayed feedback enters through capacity-constrained batches with setup costs, so the delay changes the economics and commitment structure of costly inspection rather than only the rate at which a learner receives information.

2 Preliminaries

In the classical Pandora’s Box problem, there are n boxes numbered $[n] = \{1, \dots, n\}$. Box i has an independent non-negative reward V_i with finite mean and a deterministic opening cost $c_i \geq 0$. Opening box i costs c_i and immediately reveals V_i . Weitzman’s rule assigns each box a reservation value

$$\tau_i = \sup \left\{ \tau \in \mathbb{R} : \mathbb{E}[(V_i - \tau)^+] \geq c_i \right\},$$

opens boxes in decreasing order of τ_i , and stops when the best observed reward is at least as large as the largest remaining reservation value [38].

Model. We study a batched version of this problem. We group the boxes into batches where each batch contains at most k boxes and, to open a nonempty batch, we have to pay a setup cost $T \geq 0$ in addition to the individual opening costs. When a batch is opened, all rewards in it are revealed together. We consider two variants: a non-reusable model, as in the classical problem, where opened boxes are consumed and thus batches partition the boxes, and a reusable model, where each box can provide multiple i.i.d. samples and thus batches correspond to multisets of boxes.

Definition 2.1 (Non-Reusable Batched Pandora’s Box, $\text{BP}(T, k)$). A feasible batch is a nonempty subset B of currently unopened boxes with $|B| \leq k$. Opening B costs

$$K(B) = T + \sum_{i \in B} c_i,$$

reveals $(V_i)_{i \in B}$, and consumes all boxes in B . If $k = \infty$, the cardinality constraint is omitted.

Since the batches partition the n boxes in the non-reusable model, $\text{BP}(T, k)$ is identical to $\text{BP}(T, n)$ whenever $k \geq n$.

Definition 2.2 (Reusable Batched Pandora’s Box, $\text{RBP}(T, k)$). Each box type $i \in [n]$ has a strictly positive sampling cost $c_i > 0$.¹ A feasible reusable batch is a vector $\mathbf{a} \in \mathbb{Z}_{\geq 0}^n$ of multiplicities with $1 \leq \|\mathbf{a}\|_1 \leq k$. Opening $\mathbf{a}$ costs

$$K(\mathbf{a}) = T + \sum_i a_i c_i$$

¹This assumption is relevant only when $k = \infty$; it rules out degenerate instances in which arbitrarily many copies of a zero-cost type can be included without increasing the batch cost.

and draws a_i i.i.d. samples of type i . The reward observed in batch $\mathbf{a}$ is

$$X_{\mathbf{a}} = \max_{i \in [n]} \max_{1 \leq r \leq a_i} V_{i,r},$$

where $V_{i,r}$ denotes the r -th sample of type i . Since boxes (and thus batches) are reusable, they remain available to be used in later rounds. If $k = \infty$, the cardinality constraint is omitted.

For a policy π in any of these models, let M_π be the maximum reward observed by the policy, with $M_\pi = 0$ if no box is opened, and let C_π be the total cost paid, including setup and individual opening costs. As in the classical problem, we evaluate the performance of a policy π by its expected maximum reward minus its expected total cost. We define the discounted-cost value

$$\text{val}_\lambda(\pi) = \mathbb{E}[M_\pi] - \lambda \mathbb{E}[C_\pi],$$

where $\lambda \in [0, 1]$. Thus, $\lambda = 1$ corresponds to the original-cost objective, while $\lambda < 1$ corresponds to evaluating the policy after discounting the costs. We refer to $\text{val}_1(\pi)$ as the *utility* of policy π . Similarly, the *utility* of a batch — also called its *one-shot surplus* — is its expected maximum reward minus its opening cost. We write OPT_1 for the optimal original-cost value, taking the empty policy when the optimum is zero. The source of the main departure of our batched model from the classical Pandora’s Box is the setup cost.

Observation 2.3. When $T = 0$, any chosen batch can be simulated by opening its boxes sequentially, ignoring intermediate observations until the batch is complete; the same applies to the fresh samples in a reusable batch. Since singleton batches (batches of size 1) are feasible, for every $k \geq 1$, $\text{BP}(0, k)$ has the classical singleton value (i.e. $\text{BP}(0, k) = \text{BP}(0, 1)$), and if $c_i > 0$ for every i , then $\text{RBP}(0, k)$ has the reusable singleton value (i.e. $\text{RBP}(0, k) = \text{RBP}(0, 1)$).

Computational considerations. We represent each distribution by a finite list of value-probability pairs $(v_{i\ell}, p_{i\ell})$, where $v_{i\ell} \geq 0$, $p_{i\ell} > 0$, and $\sum_\ell p_{i\ell} = 1$. We use L_i for the (non-zero) support of the distribution of box i , i.e. $\ell \in L_i \implies v_{i\ell} > 0$. The hardness reductions use two-point distributions, while all approximation statements are stated for this finite-support input model.

Submodular functions. For a set function $f : 2^{[n]} \rightarrow \mathbb{R}_{\geq 0}$, we say that f is normalized if $f(\emptyset) = 0$, monotone if $f(A) \leq f(B)$ for all $A \subseteq B$, and submodular if, for all $A \subseteq B \subseteq [n]$ and $i \notin B$,

$$f(A \cup \{i\}) - f(A) \geq f(B \cup \{i\}) - f(B).$$

The function $F(x) = \mathbb{E}[f(S_x)]$ is called the multilinear extension of f , where S_x includes each element i independently with probability x_i .

3 Structural Observations

We begin by examining how the index-based structure of Weitzman’s policy changes under batching. A natural first attempt greedily merges boxes in order of decreasing reservation values. We show that this type of sort-and-merge consecutive boxes policy fails badly: in our construction, the only positive utility batch combines a non-consecutive group of boxes. Here, the optimal policy’s batches can be made arbitrarily far apart. This rules out both greedy and dynamic programming approaches for partitioning the boxes into consecutive batches in the reservation-value order.

Next, we examine the strength of adaptivity. In the non-reusable model, we present an example that establishes a gap between the value attained by adaptive policies and fixed-menu policies that commit to a partition of boxes into batches and run Weitzman’s policy on the fixed batches. Then, we show that this adaptivity gap between the two types of policies can be at most a factor of 2. Finally, we show that in the reusable setting, there is no gap between the optimal adaptive and non-adaptive policy. As a nice byproduct, we recover an index principle in the style of Weitzman: specifically, the optimal policy repeats the batch with the highest reservation value until the batch value exceeds the reservation threshold.

3.1 Failure of Simple Batching Rules

A natural first attempt is to import Weitzman’s ordering rule into the batched problem. Compute each box’s reservation value as if it were opened alone, sort boxes in decreasing order of these values, and then form batches from consecutive blocks of this order. This preserves the intuition that high-index boxes should be inspected early and, in the batched setting, possibly inspected together. The next example shows that this rule misses a batched effect: setup costs can make two separated boxes valuable together even though every consecutive batch in the reservation-value order is unprofitable.

Take $T = 10$ and $k = 2$. Let A have item cost $c_A = 1$, with $\Pr[V_A = 100] = 0.1$ and $\Pr[V_A = 0] = 0.9$. Let B be deterministic with value 9, with item cost $c_B = 1$. Individually, neither box pays for the setup cost. However, together they have expected maximum

$$\mathbb{E}[\max\{V_A, V_B\}] = 0.1 \cdot 100 + 0.9 \cdot 9 = 18.1,$$

so the batch $\{A, B\}$ has positive utility. We will design the instance so that the target boxes A and B will be arbitrarily far apart under Weitzman’s reservation-value ordering.

Now consider a deterministic decoy box D . Let D have value 100 and item cost $c_D = 91$. Its reservation value is worse than A ’s but better than B ’s, so the sorted order is $A \succ D \succ B$. At the same time, D never contributes a useful net value: it duplicates the high outcome of A , but its item cost is almost as large as its value. One can verify that every feasible consecutive batch has negative utility, while $\{A, B\}$ has positive utility. Therefore, no consecutive batching algorithm, whether greedy or based on dynamic programming over intervals, can obtain positive utility on this example. The exact calculations are given in Section A.1, which also proves a stronger bi-criteria lower bound: for every batch capacity $k \geq 2$ and every $\alpha > 1/2$, no algorithm restricted to consecutive batches in the reservation-value order can guarantee an (α, α) -bi-criteria approximation.

3.2 Non-Reusable Model: Adaptivity Gap

A fixed-menu policy first commits to a disjoint family $\mathcal{F}$ of feasible batches. It then treats each batch $B \in \mathcal{F}$ as one box in a finite classical Pandora instance, with reward $X_B = \max_{i \in B} V_i$ and cost $K(B)$, and applies Weitzman’s rule. Let $\text{OPT}_{\text{fix}}(T, k)$ denote the best value of any such fixed-menu policy. This benchmark preserves the Pandora stopping rule while removing the branching structure of adaptive batch choice.

The natural question is whether the best fixed menu can match the value of the best adaptive policy. For $k = 1$ the answer is yes, by the classical Pandora rule. For larger batches, fixed menus lose the ability to branch on an early observation before deciding which later batch to open. The following example shows that this restriction can reduce value.

Example 3.1 (Adaptive branching can help). Consider three boxes with setup cost $T = 1/5$ and batch capacity $k = 2$. Box 1 is deterministic with value $1/2$. Box 2 is 0 with probability $2/5$, $1/2$

with probability $2/5$, and 9 with probability $1/5$. Box 3 is 10 with probability $1/5$, and 0 otherwise. The item costs are $c_1 = c_2 = \frac{1}{5}$ and $c_3 = 1$.

The adaptive policy first opens box 2. If box 2 realizes as 9, the policy stops. If box 2 realizes as $1/2$, it opens box 3 alone. If box 2 realizes as 0, it opens the batch $\{1, 3\}$. The point is that box 1 is useful only if box 3 realizes as 0, i.e. when we don't already have a guarantee from box 2.

A fixed menu cannot make this branch-dependent choice. It must decide in advance whether box 1 is grouped with box 3, separated from box 3, or grouped with box 2. For this instance, the adaptive policy has value $57/25 = 2.28$, whereas the best fixed-menu policy has value $279/125 = 2.232$. Thus adaptive branching strictly improves over every fixed menu. Appendix A.2 gives a tuned version of the same signal/fallback/risky-box template, with all parameters rounded to two decimal places; in that instance, the adaptive policy attains about 1.193 times the value of the best fixed-menu policy.

Nevertheless, the value of adaptive batch formation is bounded: every adaptive policy is within a constant factor of the best fixed-menu policy.

Theorem 3.2 (Non-Reusable Adaptivity Gap). *For every T and k ,*

$$\text{OPT}_1^{\text{BP}(T,k)} \leq 2 \cdot \text{OPT}_{\text{fix}}(T, k).$$

The proof is existential and is given in Section A. It utilizes a random-path rounding argument in the style of stochastic probing [25, 10]: one samples a root-to-leaf path of the adaptive decision tree on an independent “ghost” instance and then plays that fixed path on the real instance. For this style of argument, we must ensure that along any realization of box values, the batches consist of disjoint sets of boxes. Then, we use Weitzman’s capped-value analysis to turn this path into a fixed-menu policy losing at most a factor two.

3.3 Reusable boxes: one repeated batch

We turn our attention to the reusable setting. Since opening a reusable batch does not consume future access to it, each reusable batch can be viewed as a box with infinitely many independent copies. The dynamic problem therefore reduces to identifying the batch with the largest reservation value. For a reusable batch $\mathbf{a}$, recall that $X_{\mathbf{a}}$ is the best reward among the fresh copies drawn by opening $\mathbf{a}$, and $K(\mathbf{a})$ is its setup-plus-sampling cost. If $0 < K(\mathbf{a}) \leq \mathbb{E}[X_{\mathbf{a}}]$, define its reservation value $\sigma(\mathbf{a})$ by $K(\mathbf{a}) = \mathbb{E}[(X_{\mathbf{a}} - \sigma(\mathbf{a}))^+]$. Batches with $K(\mathbf{a}) > \mathbb{E}[X_{\mathbf{a}}]$ are overpriced. Let

$$\sigma^* = \max\left(\{0\} \cup \{\sigma(\mathbf{a}) : \mathbf{a} \text{ is feasible and not overpriced}\}\right).$$

Theorem 3.3 (Reusable Reservation Index). *Assume values lie in $[0, V_{\max}]$ and every non-empty feasible batch has positive opening cost. For $k = \infty$, also assume every $c_i > 0$. If $\sigma^* = 0$, opening nothing is optimal. If $\sigma^* > 0$, then for any feasible batch $\mathbf{a}^*$ with $\sigma(\mathbf{a}^*) = \sigma^*$, the policy that repeatedly opens $\mathbf{a}^*$ until it observes a reward exceeding σ^* is optimal.*

The theorem is the reusable analogue of Weitzman’s index rule. A copy of the best reservation batch remains available after every failed attempt, so the index policy continues to choose it. The boundedness and positive-cost assumptions ensure that the relevant batches that are not overpriced are finite in number; the formal details are in Section A.

The repeated-batch value equals the reservation value. If batch $\mathbf{a}$ has reservation value $\sigma(\mathbf{a})$ and $p = \Pr[X_{\mathbf{a}} > \sigma(\mathbf{a})]$, then repeatedly opening it until the first success gives

$$\begin{aligned} \mathbb{E}[X_{\mathbf{a}} \mid X_{\mathbf{a}} > \sigma(\mathbf{a})] - \frac{K(\mathbf{a})}{p} &= \frac{\mathbb{E}[X_{\mathbf{a}} \mathbf{1}\{X_{\mathbf{a}} > \sigma(\mathbf{a})\}]}{p} - \frac{K(\mathbf{a})}{p} = \sigma(\mathbf{a}) + \frac{\mathbb{E}[(X_{\mathbf{a}} - \sigma(\mathbf{a}))^+] - K(\mathbf{a})}{p} \\ &= \sigma(\mathbf{a}). \end{aligned}$$

The last equality uses $\mathbb{E}[(X_{\mathbf{a}} - \sigma(\mathbf{a}))^+] = K(\mathbf{a})$, by the definition of $\sigma(\mathbf{a})$.

4 Hardness Results

In this section, we show that the various settings of Batched Pandora’s Box are NP-hard. Broadly, our reductions construct instances of Batched Pandora’s Box in which even building a batch with positive utility encodes NP-hard search problems.

Product gadget. Our reductions use a common product gadget. Each box has the same positive prize M , and the box corresponding to label b_i fails with probability $1/b_i$. A batch whose labels multiply to weight $w = \prod_{i \in B} b_i$ then fails with probability $1/w$. We set the item costs and the batching cost carefully as a function of the weights of the boxes, ensuring a batch offers positive opening utility if and only if its labels multiply to a prescribed target value Q . Consequently, finding a batch with positive opening utility encodes an NP-hard search problem, yielding hardness for Batched Pandora’s Box. For clarity, we initially present the proof using exact logarithms, and then show that rational approximations to the logarithms (of polynomial length) still preserve this property. The formal proofs are presented in Section 6.

Non-reusable boxes. We first investigate the non-reusable problem. Exact optimization is already intractable for every fixed finite batch capacity at least three.

Theorem 4.1 (Fixed Finite-Batch-Capacity Exact Hardness). *For every fixed $k \geq 3$, exact optimization of $\text{BP}(T, k)$ is NP-hard, even when $T > 0$ and all boxes have support $\{0, M\}$ for a shared M . Equivalently, deciding whether $\text{OPT}_1 \geq \Theta$ for a rational threshold Θ is NP-hard.*

We reduce from the strongly NP-hard Numerical 3-Dimensional Matching problem. The instance consists of three classes of items X, Y, Z and a size function $s(e), e \in X \cup Y \cup Z$, and the goal is to partition the items into triples (x, y, z) satisfying $s(x) + s(y) + s(z) = B$.

For batch size $k = 3$, we encode each item e as a box whose label contains the factor $2^{s(e)}$. To encode the class constraint, we assign a distinct identifying prime to each of X, Y, Z , and include the corresponding prime in the label of every item from that class. We then set the target product to be the product of the three class primes times 2^B . By unique factorization, a batch has the target product if and only if it contains one item from each class whose sizes sum to B . For $k > 3$, we add dummy zero-size classes to fill the remaining positions in each batch.

The product gadget makes precisely the target tuples profitable. Since all target tuples have the same success probability and the same cost, the optimal value is a strictly increasing function of the maximum number of disjoint target tuples, and exact optimization distinguishes whether a perfect numerical matching exists.

The next question is whether one can at least obtain an ordinary constant-factor approximation. When batch capacity is part of the input, the product gadget gives a stronger obstruction: distinguishing zero value from positive value is already NP-hard.

Theorem 4.2 (Non-Reusable Sign-Gap Decision Hardness). *For $\text{BP}(T, k)$ with k part of the input, and for $\text{BP}(T, \infty)$, the promise problem of distinguishing*

$$\text{OPT}_1 = 0 \quad \text{from} \quad \text{OPT}_1 > 0$$

is NP-hard, even on common-prize two-point instances with $T > 0$.

Here the reduction is from SUBSET PRODUCT: given labels $a_1, \dots, a_n$ and target Q , decide whether some subset has product Q . The batch capacity is set to the number of source items. If a target subset exists, opening the corresponding batch once gives positive value; otherwise every nonempty batch has negative one-shot surplus. Because all boxes have a common prize, adaptive stopping cannot turn these negative-surplus attempts into positive value: before the first success the policy observes only zeros, and the expected reward from any sequence of attempts is dominated by its expected cost.

Reusable boxes. For reusable boxes, the same product idea applies, but we must prevent a target batch from using multiple copies of the same source item. Broadly, for each source item i , we introduce a private prime p_i and create two boxes whose labels correspond to p_i and $p_i a_i$. We set the target product to $Q \prod_i p_i$. Since p_i appears in no other label and has exponent one in the target, any multi-set whose labels multiply to the target must contain exactly one of p_i and $p_i a_i$ for each i . Choosing p_i corresponds to excluding a_i from the subset, while choosing $p_i a_i$ corresponds to including it. After canceling the private primes, the remaining condition enforces that the product of the selected a_i 's equals Q . Finally, the product gadget then gives the reusable sign gap.

Theorem 4.3 (Reusable Sign-Gap Decision Hardness). *For $\text{RBP}(T, k)$ with finite k as part of the input, and for $\text{RBP}(T, \infty)$, the promise problem of distinguishing*

$$\text{OPT}_1 = 0 \quad \text{from} \quad \text{OPT}_1 > 0$$

is NP-hard, even on common-prize two-point instances with $T > 0$.

For unbounded batch capacity, item costs make all sufficiently large multisets automatically overpriced, so the reduction only has to protect a finite range of multiplicities by the product separation.

5 Bi-criteria Approximation Results

The hardness results of the previous section preclude standard positive-factor approximation guarantees under the original costs. We therefore turn to discounted-cost bicriteria guarantees. Recall that $\text{val}_\lambda(\pi) = \mathbb{E}[M_\pi] - \lambda \mathbb{E}[C_\pi]$, and that an algorithm has an (α, λ) guarantee if, for every requested $\varepsilon > 0$, it returns a feasible policy $\hat{\pi} = \hat{\pi}(\varepsilon)$ with $\text{val}_\lambda(\hat{\pi}) \geq \alpha \text{OPT}_1 - \varepsilon$. Thus, the algorithm is compared with the original-cost optimum, while its returned policy is evaluated after discounting costs.

5.1 Non-Reusable Boxes

Consider first the non-reusable model. Although an adaptive policy may have a possibly exponential decision tree, before any values are realized, it induces only ex-ante statistics: the probability x_i that each box is ever opened and the expected number z of batches it opens. These statistics determine the expected cost. The relaxation therefore only needs to upper-bound the final reward that can be attributed to the opened boxes. For each box i , we omit the zero atom and, for $\ell \in L_i$, write

$$\Pr[V_i = v_{i\ell}] = p_{i\ell}, \quad \text{where} \quad v_{i\ell} > 0.$$

When box i is the final selected value and realizes as the point $v_{i\ell}$ in its support, we assign the probability mass of this event to a variable $y_{i\ell}$. This “winner” event can occur only if box i is

Algorithm 1 Non-reusable LP rounding

Input: Opening marginals x , fractional batch count $z \geq 1$, and capacity k

Set $\theta = z - \lfloor z \rfloor$

Draw $\bar{z} = \lfloor z \rfloor$ with probability $1 - \theta$, and $\bar{z} = \lceil z \rceil$ with probability θ

Set $q_i = \min\{1, (\bar{z}/z)x_i\}$ for every box i

Apply pipage rounding over the rank- $(k\bar{z})$ uniform matroid to obtain a set S with $|S| \leq k\bar{z}$

Partition S arbitrarily into at most $\bar{z}$ groups of size at most k each, open all nonempty groups, and stop

opened and realizes as $v_{i\ell}$, and at most one positive value is selected overall. These two observations give the following relaxation:

$$\begin{aligned} \max \quad & \sum_i \sum_{\ell \in L_i} v_{i\ell} y_{i\ell} - Tz - \sum_i c_i x_i \\ \text{s.t.} \quad & 1 \leq z \leq n, \\ & \sum_i x_i \leq kz, \\ & 0 \leq y_{i\ell} \leq p_{i\ell} x_i \quad \forall i, \ell \in L_i, \\ & 0 \leq x_i \leq 1 \quad \forall i, \\ & \sum_i \sum_{\ell \in L_i} y_{i\ell} \leq 1 \end{aligned} \tag{LP}$$

The constraint $\sum_i x_i \leq kz$ is the ex-ante version of batch capacity: if a policy opens z batches in expectation, it cannot open more than kz boxes in expectation.

At first glance, one could think that we could write n different versions of this LP, one per candidate value of z , since any policy always opens an integer number of batches, and take the maximum of these n LPs. This, in principle, would save us the extra loss we incur from rounding z . Unfortunately, such an approach does not constitute a valid upper bound to the optimal adaptive policy, since the latter could open a different number of batches depending on the history of observed values and thus, in expectation, a fractional number of batches overall. For this reason, allowing z to be fractional is crucial so that the LP can capture the value of the optimal adaptive algorithm. Also, the LP is a relaxation for the optimal adaptive algorithm, because the atom masses are allowed to be arranged fractionally into a final winning value. After fixing the opening marginals x , this upper bound on the expected reward has the equivalent threshold form

$$B(x) = \int_0^\infty \min \left\{ 1, \sum_i x_i \Pr[V_i > t] \right\} dt.$$

Sampling which boxes to open independently with probabilities given by the x_i 's guarantees a $(1 - 1/e)$ -fraction of this envelope at every threshold. Afterwards, pipage rounding algorithm converts the opening marginals x into an actual set of opened boxes, while respecting the batch budget certified by z , without decreasing the expected monotone submodular reward.

We also round z to one of its two neighboring integers and scale the opening marginals accordingly. Once the number of batches is integral, a selected set of at most $k\bar{z}$ boxes can be partitioned arbitrarily into $\bar{z}$ legal batches. Consequently, the rounding incurs only the batch-count factor $2(\sqrt{2} - 1)$ from rounding the batch count and no additional contention loss inside individual batches.

Theorem 5.1 (Finite-Support Non-Reusable Guarantee). *For every finite-support non-reusable instance and every $\varepsilon > 0$, there is a randomized polynomial-time algorithm that returns a policy $\hat{\pi} = \hat{\pi}(\varepsilon)$ satisfying*

$$\text{val}_{\alpha_{\text{NR}}}(\hat{\pi}) \geq \alpha_{\text{NR}} \text{OPT}_1 - \varepsilon, \quad \alpha_{\text{NR}} = \left(1 - \frac{1}{e}\right) 2(\sqrt{2} - 1).$$

The algorithm solves (LP), returns the empty policy if its optimum is non-positive, and otherwise applies Algorithm 1 to an optimal solution.

The proof follows this outline. The details are in Section 7.1.

5.2 Reusable Boxes

The reusable case starts from a different structural fact. By Theorem 3.3, an optimal original-cost policy repeats one maximum-reservation batch. Thus, it suffices to find one certifying multiset of at most k fresh i.i.d. samples. The case $T = 0$ is handled exactly by Observation 2.3, so assume $T > 0$ below and let $V_{\max}$ be the largest support value.

For a threshold τ , write $f_\tau(\mathbf{a}) = \mathbb{E}[(X_{\mathbf{a}} - \tau)^+]$. If one batch pays for its discounted cost using the expected excess above τ , then repeatedly opening it guarantees discounted value at least τ .

Lemma 5.2 (Threshold Certificate). *Fix $\lambda \in (0, 1]$. If a nonempty reusable batch $\mathbf{a}$ satisfies*

$$f_\tau(\mathbf{a}) \geq \lambda K(\mathbf{a}),$$

then repeatedly opening $\mathbf{a}$ until the first realization $X_{\mathbf{a}} > \tau$ has λ -discounted value at least τ .

This observation reduces the reusable algorithm to a threshold search. For a candidate τ , the task is to determine whether some batch can certify it, i.e. whether there is a batch with reservation value at least τ . The fractional relaxation is similar to the non-reusable LP, except that it is turned into a feasibility LP and x_i is now a fractional number of copies of type i in a single reusable batch:

$$\begin{aligned} &\text{find } x, y \\ &\text{s.t. } \sum_i x_i \leq k, \\ &\quad 0 \leq y_{i\ell} \leq p_{i\ell} x_i \quad \forall i, \ell \in L_i, \\ &\quad 0 \leq x_i \quad \forall i, \\ &\quad \sum_i \sum_{\ell \in L_i} y_{i\ell} \leq 1 \\ &\quad \sum_i \sum_{\ell \in L_i} (v_{i\ell} - \tau)^+ y_{i\ell} \geq T + \sum_i c_i x_i \end{aligned} \tag{RLP}_\tau$$

We say that τ is LP-certified if RLP_τ is feasible. The y -variables form a fractional one-winner certificate: the constraints $y_{i\ell} \leq p_{i\ell} x_i$ encode the marginal availability of atom $v_{i\ell}$, while $\sum_{i,\ell} y_{i\ell} \leq 1$ encodes that one batch contributes only one maximum value. Thus feasibility of RLP_τ certifies that the fractional excess above τ can pay for the batch cost. Lowering τ only increases the coefficients $(v_{i\ell} - \tau)^+$ and makes the certificate constraint easier to satisfy. Therefore, certification is monotone and binary search can find the largest certified threshold up to an additive error.

When a threshold is certified fractionally, we round by filling the k batch slots independently according to the copy counts x_i . At level $t \geq \tau$, one slot exceeds t with probability $\sum_i x_i \Pr[V_i > t]/k$, so the rounded batch achieves an $\alpha_R = 1 - 1/e$ fraction of the fractional excess envelope. The method of conditional expectations derandomizes the slot choices without losing the certificate.

Algorithm 2 Reusable threshold search

Input: A finite-support reusable RBP(T, k) instance and accuracy $\varepsilon > 0$

Set $\ell = 0$, $u = V_{\max}$, $\hat{\tau} = 0$, and $\hat{\mathbf{a}} = \mathbf{0}$

for $\lceil \log_2(V_{\max}/\varepsilon) \rceil$ iterations **do**

Set $\tau = (\ell + u)/2$ and solve RLP $_{\tau}$

if τ is LP-certified **then**

Round feasible LP solution into a certifying batch $\mathbf{a}_{\tau}$ using the method of conditional expectation

Set $\ell = \tau$, $\hat{\tau} = \tau$, and $\hat{\mathbf{a}} = \mathbf{a}_{\tau}$

else

Set $u = \tau$

end if

end for

if $\hat{\mathbf{a}} = \mathbf{0}$ **then**

return the empty policy

end if

Repeatedly open $\hat{\mathbf{a}}$ until its value exceeds $\hat{\tau}$

Theorem 5.3 (Finite-Support Reusable Guarantee). *For every finite-support reusable instance with finite batch capacity k and every $\varepsilon > 0$, Algorithm 2 runs in polynomial time and returns a policy $\hat{\pi} = \hat{\pi}(\varepsilon)$ satisfying*

$$\text{val}_{\alpha_R}(\hat{\pi}) \geq \text{OPT}_1 - \varepsilon, \quad \alpha_R = 1 - \frac{1}{e}.$$

Equivalently, the algorithm provides a $(1, \alpha_R)$ discounted-cost bicriteria guarantee.

The comparison with the optimum follows from the reusable reservation-index theorem. If $\mathbf{a}^*$ is an optimal original-cost batch with reservation value OPT_1 , then it certifies every threshold $\tau \leq \text{OPT}_1$ in the feasibility LP. Binary search therefore reaches a threshold within ε of the optimum, and the rounded certifying batch is converted into a repeated policy by Lemma 5.2. The formal proof appears in Section 7.2.

6 Proofs of Hardness Results

We will repeatedly use two numerical primitives. First, define

$$D(z) = \ln z + \frac{1}{z} - 1.$$

This function is minimized at $z = 1$. Moreover, when w and Q are positive integers with $w \neq Q$, the value $D(w/Q)$ is bounded away from zero by an inverse polynomial in Q .

Lemma 6.1 (Analytic Product Separation). *For every integer $Q \geq 8$ and every positive integer $w \neq Q$,*

$$D(w/Q) \geq \frac{1}{16 \cdot Q^2}.$$

Proof. We have

$$D(1) = D'(1) = 0, \quad D'(z) = \frac{z-1}{z^2}, \quad D''(z) = \frac{2-z}{z^3}.$$

Thus D decreases on $(0, 1)$ and increases on $(1, \infty)$. On $[1/2, 3/2]$, $D''(z) \geq 4/27$, so Taylor's theorem around 1 gives

$$D(z) \geq \frac{2}{27} \cdot (z - 1)^2.$$

If $w/Q \in [1/2, 3/2]$, then $w \neq Q$ and integrality imply $|w/Q - 1| \geq 1/Q$, hence

$$D(w/Q) \geq \frac{2}{27 \cdot Q^2} > \frac{1}{16 \cdot Q^2}.$$

Outside this interval, monotonicity gives

$$D(w/Q) \geq \min\{D(1/2), D(3/2)\} > \frac{1}{100} \geq \frac{1}{16 \cdot Q^2},$$

where the last inequality uses $Q \geq 8$. □

Second, we will use rational approximations to logarithms so that all constructed costs are rational and polynomially representable.

Lemma 6.2 (Rational Log Approximation). *Let $m_1, \dots, m_r$ be positive integers and let $\eta > 0$ be rational. In time polynomial in the input bit length and $\log(1/\eta)$, one can compute rational numbers $\ell_1, \dots, \ell_r$ such that*

$$|\ell_i - \ln m_i| \leq \eta \quad \text{for all } i.$$

Proof. The approximation claim is the standard bit-complexity computation of elementary functions: after range reduction, logarithms can be approximated to b bits by classical polynomial-time series or binary-splitting algorithms. □

6.1 Proof of Theorem 4.1: Fixed finite-capacity non-reusable boxes

Proof of Theorem 4.1. We first describe the reduction using exact logarithmic costs and ignore representation-size issues; the final paragraph replaces the logarithms by rational approximations and checks that the constructed instance has polynomial encoding length. We reduce from strongly NP-complete Numerical 3-Dimensional Matching [19]. An instance consists of three classes X, Y, Z , each of size m , positive integer sizes $s(e)$, and a target B . The question is whether the elements can be partitioned into m triples (x, y, z) satisfying

$$s(x) + s(y) + s(z) = B.$$

Since Numerical 3DM is strongly NP-complete, we may restrict to instances in which B and all sizes $s(e)$ are bounded by a polynomial in m . We also assume $m \geq 2$, since the case $m = 1$ is decidable directly.

Fix $k \geq 3$. For $k = 3$, use the three original classes. For $k > 3$, add $k - 3$ dummy classes, each containing m dummy elements of size zero. Choose k distinct odd primes $p_1, \dots, p_k$, one per class, and use 2 as a separate encoding prime. For an element e in class j , define

$$b_e = p_j \cdot 2^{s(e)}.$$

For dummy elements, $s(e) = 0$. Set

$$Q = \left(\prod_{j=1}^k p_j \right) \cdot 2^B.$$

Call a batch *target* if its label product is Q . By unique factorization, target batches are exactly the feasible k -tuples with one element from every class and original sizes summing to B . For $k > 3$, the dummy classes preserve the perfect-matching question: any collection of disjoint valid original triples can be extended using distinct dummy elements, and any disjoint family of target k -tuples projects to disjoint valid original triples.

Set

$$H = 1, \quad S = 100 \cdot H \cdot Q^2, \quad M = S \cdot Q.$$

For each element e , create an independent high-or-zero box equal to M with probability $1 - 1/b_e$ and zero otherwise. Give an element e in class j cost

$$c_e = S \cdot \ln b_e = S \cdot (\ln p_j + s(e) \ln 2),$$

and set the setup cost to

$$T = S \cdot (Q - 1 - \ln Q) - H = S \cdot \left(Q - 1 - \sum_{j=1}^k \ln p_j - B \ln 2 \right) - H.$$

For a nonempty batch A , write

$$w(A) = \prod_{e \in A} b_e, \quad K(A) = T + \sum_{e \in A} c_e.$$

The batch fails with probability $1/w(A)$, and its one-shot surplus is

$$\begin{aligned} M \cdot \left(1 - \frac{1}{w(A)} \right) - K(A) &= SQ \cdot \left(1 - \frac{1}{w(A)} \right) - \left(S(Q - 1 - \ln Q) - H + \sum_{e \in A} S \ln b_e \right) \\ &= SQ - \frac{SQ}{w(A)} - S(Q - 1 - \ln Q) + H - S \sum_{e \in A} \ln b_e \\ &= H + S \left(1 + \ln Q - \ln w(A) - \frac{Q}{w(A)} \right) \\ &= H - S \cdot \left(\ln \frac{w(A)}{Q} + \frac{Q}{w(A)} - 1 \right) \\ &= H - S \cdot D(w(A)/Q). \end{aligned}$$

Thus every target batch has surplus H . If A is non-target, then Lemma 6.1 gives

$$H - S \cdot D(w(A)/Q) \leq H - \frac{S}{16Q^2} = -\frac{21}{4} \cdot H,$$

so every non-target batch is overpriced.

Every target batch has the same item-cost sum

$$S \cdot \left(\sum_{j=1}^k \ln p_j + B \ln 2 \right),$$

and therefore the same total cost

$$K = S \cdot (Q - 1) - H,$$

the same failure probability $q = 1/Q$, and the same positive one-shot surplus

$$\Delta = M \cdot (1 - q) - K = H.$$

Let ν be the maximum number of pairwise-disjoint target batches. By Lemma A.2, an optimal policy opens only target batches. Until the first success, every observed value is zero, so the all-failure path of any deterministic policy is a sequence of disjoint target batches and has length at most ν . Conversely, any packing of r target batches can be opened in any order until the first success. Its expected utility is

$$\begin{aligned} M \cdot (1 - q^r) - K \cdot (1 + q + \dots + q^{r-1}) &= (M \cdot (1 - q) - K) \cdot (1 + q + \dots + q^{r-1}) \\ &= \Delta \cdot (1 + q + \dots + q^{r-1}). \end{aligned}$$

This expression is strictly increasing in r . Randomization and stopping before exhausting a packing cannot improve it, since every additional available target batch has the same positive conditional surplus. Hence

$$\text{OPT}_1 = \Delta \cdot (1 + q + \dots + q^{\nu-1}),$$

with the geometric sum interpreted as zero when $\nu = 0$.

A YES instance has $\nu = m$, while a NO instance has $\nu \leq m - 1$. Define

$$\Theta = \frac{1}{2} \cdot \left[\Delta \cdot (1 + q + \dots + q^{m-1}) + \Delta \cdot (1 + q + \dots + q^{m-2}) \right].$$

Then $\text{OPT}_1 \geq \Theta$ exactly in YES instances.

Representation. The construction above used exact logarithms. To make the input rational, let $s_{\max} = \max_e s(e)$ and

$$D_0 = 2 \cdot k + k \cdot s_{\max} + B.$$

By Lemma 6.2, choose rationals $\ell_1, \dots, \ell_k, \ell_{\text{bin}}$ such that

$$|\ell_j - \ln p_j| \leq \delta, \quad |\ell_{\text{bin}} - \ln 2| \leq \delta, \quad \delta = \frac{H}{100 \cdot S \cdot D_0}.$$

Replace the real costs by

$$c_e = S \cdot (\ell_j + s(e) \cdot \ell_{\text{bin}})$$

for an element e in class j , and set

$$T = S \cdot \left(Q - 1 - \sum_{j=1}^k \ell_j - B \cdot \ell_{\text{bin}} \right) - H.$$

For any batch of at most k elements, the change in its surplus is at most

$$S \cdot (2 \cdot k + k \cdot s_{\max} + B) \cdot \delta = H/100.$$

Target batches still have total cost exactly $K = S(Q - 1) - H$ and surplus exactly H , while every non-target batch remains overpriced. The setup cost remains positive because the real setup cost $S(Q - 1 - \ln Q) - H > 0$ for $Q \geq 8$ and the perturbation is at most $H/100$; the item costs are nonnegative by the same choice of precision.

Finally, the reduction has polynomial size. Since B and all sizes $s(e)$ are bounded by a polynomial in m , and k is fixed, the integers 2^B and $Q = (\prod_j p_j)2^B$ may be large as values but have polynomial binary encoding length. Hence the labels b_e also have polynomial binary encoding length. The probabilities $1 - 1/b_e$, the scale parameters S, M , the rational logarithmic approximations, and the rational costs all have polynomial encoding length. Also Q^{m-1} has bit length $O(m \log Q) = O(mB)$, so the threshold Θ is polynomially representable. Thus the rational instance and threshold are produced in polynomial time. $\square$

6.2 Proof of Theorem 4.2: Sign-gap hardness for non-reusable boxes

Before proving the non-reusable sign-gap theorem, we isolate the batch-level hardness used in the reduction. For a feasible batch B , let X_B be the best value observed inside the batch and let $K(B)$ be its opening cost. Its *one-shot surplus* is

$$\Delta(B) = \mathbb{E}[X_B] - K(B).$$

If $\Delta(B) < 0$, equivalently $K(B) > \mathbb{E}[X_B]$, the batch is *overpriced*.

The construction makes the probability of no success equal to the reciprocal of a product. Costs are logarithmic in the same labels. Thus the reward term contributes a reciprocal-product term, the cost term contributes a logarithmic term, and after calibration the surplus contains $D(w/Q)$, which is uniquely minimized at the target product $w = Q$.

Definition 6.3 (Restricted SUBSET PRODUCT). The input consists of integers $a_1, \dots, a_n \geq 2$ and an integer target $Q \geq 8$. The question is whether there is a subset $S \subseteq [n]$ such that

$$\prod_{i \in S} a_i = Q.$$

The usual SUBSET PRODUCT problem has no restrictions on a_i or Q . The restricted version above remains NP-hard.

Lemma 6.4 (Restricted Subset Product Remains NP-Hard). Restricted SUBSET PRODUCT is NP-hard.

Proof. Reduce from the standard SUBSET PRODUCT problem. Given an instance $(a_1, \dots, a_n, Q)$, delete all labels equal to 1, since they do not affect feasibility, and write

$$A = \prod_i a_i$$

for the product of the remaining labels, with $A = 1$ if none remain. Add one new label

$$P = 8 \cdot A$$

and set the new target to

$$Q' = P \cdot Q.$$

All labels in the new instance are at least 2, and $Q' \geq 8$.

If the original instance has a subset of product Q , then adding the new label P gives a subset of product Q' . Conversely, any subset of the new instance with product Q' must contain P : without it, the product is at most $A < Q'$. After removing P , the remaining selected labels have product Q . Thus the two instances are equivalent, and the transformation has polynomial bit length because $\log A$ is at most the total input length. $\square$

Lemma 6.5. For every rational $H > 0$, deciding whether a common-prize high-or-zero non-reusable instance contains a nonempty positive-surplus feasible batch is NP-hard, even when the batch capacity is part of the input and also when $k = \infty$, and even with $T > 0$.

Proof. We first describe the reduction using exact logarithmic costs and ignore representation-size issues; the final paragraph replaces the logarithms by rational approximations and checks that the

constructed instance has polynomial encoding length. We reduce from restricted SUBSET PRODUCT. Let the source instance be $(b_1, \dots, b_N, Q)$. Set

$$\Lambda = 100 \cdot H \cdot Q^2, \quad M = \Lambda \cdot Q.$$

For label b_j , create a high-or-zero box equal to M with probability $1 - 1/b_j$ and zero otherwise. A set A has failure probability

$$\prod_{j \in A} b_j^{-1} = \frac{1}{w(A)}, \quad w(A) = \prod_{j \in A} b_j.$$

Give label b_j opening cost

$$c_j = \Lambda \cdot \ln b_j, \quad T = \Lambda \cdot (Q - 1 - \ln Q) - H.$$

The one-shot surplus of A is

$$\begin{aligned} M \cdot \left(1 - \frac{1}{w(A)}\right) - T - \sum_{j \in A} c_j &= \Lambda Q \cdot \left(1 - \frac{1}{w(A)}\right) - \left(\Lambda(Q - 1 - \ln Q) - H + \Lambda \sum_{j \in A} \ln b_j\right) \\ &= H + \Lambda \left(1 + \ln Q - \ln w(A) - \frac{Q}{w(A)}\right) \\ &= H - \Lambda \cdot \left(\ln \frac{w(A)}{Q} + \frac{Q}{w(A)} - 1\right) \\ &= H - \Lambda \cdot D(w(A)/Q). \end{aligned}$$

It equals H when $w(A) = Q$. If $w(A) \neq Q$, then Lemma 6.1 gives

$$H - \Lambda \cdot D(w(A)/Q) \leq H - \frac{\Lambda}{16 \cdot Q^2} = -\frac{21}{4} \cdot H.$$

Representation. The construction above used exact logarithms. To make the input rational, apply Lemma 6.2 and choose rational numbers ℓ_j, ℓ_Q satisfying

$$|\ell_j - \ln b_j| \leq \delta, \quad |\ell_Q - \ln Q| \leq \delta, \quad \delta = \frac{H}{100 \cdot \Lambda \cdot (N + 1)}.$$

Replace the exact logarithmic costs by

$$c_j = \Lambda \cdot \ell_j, \quad T = \Lambda \cdot (Q - 1 - \ell_Q) - H.$$

For every nonempty set A , rationalization changes its surplus by at most

$$\Lambda \cdot \left(|\ell_Q - \ln Q| + \sum_{j \in A} |\ell_j - \ln b_j| \right) \leq \Lambda \cdot (N + 1) \cdot \delta = H/100.$$

Therefore a target batch has surplus in $[99 \cdot H/100, 101 \cdot H/100]$, while every off-target batch has surplus at most $-21H/4 + H/100 < -5H$. Thus the constructed instance contains a positive-surplus batch if and only if the restricted SUBSET PRODUCT instance is YES. The precision also keeps every ℓ_j positive, since $b_j \geq 2$. Since $Q \geq 8$, the quantity $Q - 1 - \ln Q$ is bounded away from zero, and the approximation error is less than $H/100$, so $T > 0$. The required precision has polynomial encoding length, so Lemma 6.2 makes the whole construction polynomial time. Set the batch capacity to $k = N$; since there are only N boxes, the same instance also proves the claim for $k = \infty$. $\square$

Proof of Theorem 4.2. Use the instances constructed in Lemma 6.5 with $H = 1$. In a YES instance, there is a batch with positive one-shot surplus, so opening it once gives $\text{OPT}_1 > 0$. In a NO instance, every nonempty feasible batch is overpriced. By Lemma A.2, there is an optimal non-reusable policy that never opens an overpriced batch. Hence the empty policy is optimal and $\text{OPT}_1 = 0$. Thus a polynomial-time solver for the promise problem would decide the NP-hard batch-existence problem from Lemma 6.5. $\square$

6.3 Proof of Theorem 4.3: Sign-gap hardness for reusable boxes

Lemma 6.6. *For every rational $H > 0$, deciding whether a common-prize high-or-zero reusable instance contains a nonempty positive-surplus feasible batch is NP-hard when finite k is part of the input, and also for $k = \infty$, even with $T > 0$.*

Proof. We first describe the reduction using exact logarithmic costs and ignore representation-size issues; the final paragraph replaces the logarithms by rational approximations and checks that the constructed instance has polynomial encoding length. We reduce from restricted SUBSET PRODUCT. Let the source instance be $(a_1, \dots, a_n, Q)$. The only additional difficulty in the reusable model is that a batch may contain repeated copies. We first encode each original item as a forced binary choice, so that any product hitting the target uses each source item at most once.

Encoding the subset choice. Choose pairwise distinct private primes $p_1, \dots, p_n$ that divide neither Q nor any input number a_j .

For each source item i , create two labels

$$b_{i,0} = p_i, \quad b_{i,1} = p_i \cdot a_i,$$

and set

$$Q' = Q \cdot \prod_i p_i.$$

The label $b_{i,0}$ means “skip item i ,” while $b_{i,1}$ means “take item i .” Both labels contain the private prime p_i , and no other label or source factor contains p_i .

For a reusable multiplicity vector u , define

$$w(u) = \prod_i b_{i,0}^{u_{i,0}} \cdot b_{i,1}^{u_{i,1}}.$$

The private-prime encoding gives the following equivalence:

$$w(u) = Q' \iff u_{i,0} + u_{i,1} = 1 \quad \forall i \quad \text{and} \quad \prod_{i: u_{i,1}=1} a_i = Q.$$

Indeed, comparing the exponent of each private prime p_i forces $u_{i,0} + u_{i,1} = 1$, and after canceling all private primes the remaining condition is exactly the source subset-product equation.

Creating the reusable sign gap. Let $L = n$ for the finite-batch-capacity instance, where we set $k = n$. For the unbounded-batch-capacity instance, set $L = L_0 = 4Q' + 1$.

Apply the product construction to the $2n$ labels $b_{i,0}, b_{i,1}$ with target Q' and multiplicity bound L . Set

$$\Lambda = 100 \cdot H \cdot (Q')^2, \quad M = \Lambda Q'.$$

For each label b_j , create a reusable high-or-zero type equal to M with probability $1 - 1/b_j$ and zero otherwise. Give label b_j opening cost

$$c_j = \Lambda \ln b_j, \quad T = \Lambda(Q' - 1 - \ln Q') - H.$$

For a multiplicity vector u with $\|u\|_1 \leq L$, the all-zero probability is $1/w(u)$, and its one-shot surplus is

$$\begin{aligned} M \cdot \left(1 - \frac{1}{w(u)}\right) - T - \sum_j u_j c_j &= \Lambda Q' \cdot \left(1 - \frac{1}{w(u)}\right) - \left(\Lambda(Q' - 1 - \ln Q') - H + \Lambda \sum_j u_j \ln b_j\right) \\ &= H - \Lambda \cdot \left(\ln \frac{w(u)}{Q'} + \frac{Q'}{w(u)} - 1\right) \\ &= H - \Lambda D(w(u)/Q'). \end{aligned}$$

Thus vectors with $w(u) = Q'$ have surplus H , while those with $w(u) \neq Q'$ have surplus at most $-21H/4$ by Lemma 6.1.

Since every label is at least 2, every item cost is at least $\Lambda/4$. In the finite-batch-capacity instance every feasible batch has $\|u\|_1 \leq n = L$. In the unbounded-batch-capacity instance, every batch with $\|u\|_1 > L_0$ has cost greater than

$$L_0 \cdot \frac{\Lambda}{4} = M + \frac{\Lambda}{4},$$

so its surplus is at most $-\Lambda/4 < -5H$ because its expected reward is at most M .

Representation. The construction above used exact logarithms. To make the input rational, apply Lemma 6.2 to the labels and to Q' , and choose rational approximations with

$$\delta = \min \left\{ \frac{H}{100 \cdot \Lambda \cdot (L+1)}, \frac{1}{4} \right\}.$$

Replace each $\ln b_j$ and $\ln Q'$ in the costs and setup cost by its rational approximation. Rationalizing the logarithms changes the surplus of every vector $\|u\|_1 \leq L$ by at most $H/100$. Hence target vectors have surplus in $[99 \cdot H/100, 101 \cdot H/100]$, and every non-target vector of size at most L has surplus at most $-21H/4 + H/100 < -5H$. Since $b_j \geq 2$ and $\delta \leq 1/4$, every item cost remains at least $\Lambda/4$. Therefore, in the unbounded-batch-capacity instance, every batch with $\|u\|_1 > L_0$ still has surplus at most $-\Lambda/4 < -5H$.

Combining this with the private-prime equivalence, the constructed rational reusable instance has a positive-surplus feasible batch if and only if the source SUBSET PRODUCT instance is YES. The private primes can be found in polynomial time: if L_{in} is the total source bit length, then $Q \cdot \prod_j a_j$ has at most L_{in} distinct prime divisors, so among the first $n + L_{\text{in}}$ primes at least n are available. Thus the primes, labels, and target Q' have polynomial binary encoding length. In the unbounded-batch-capacity case, $L_0 = 4Q' + 1$ may be numerically large, but its bit length is polynomial. The required precision has polynomial encoding length. The exact setup cost is positive for $Q' \geq 8$, and replacing $\ln Q'$ by its rational approximation changes it by at most $\Lambda\delta \leq H/100$, so the setup cost remains positive. Hence the rational instance is produced in polynomial time. $\square$

Proof of Theorem 4.3. Use the instances constructed in Lemma 6.6 with $H = 1$. In a YES instance, there is a reusable batch with positive one-shot surplus. For a common-prize reusable batch with success probability s , cost K , and prize M , positive surplus means $K < sM$, so its reservation value is $M - K/s > 0$. By Theorem 3.3, this gives $\text{OPT}_1 > 0$. In a NO instance, every nonempty reusable batch is overpriced. Then $\sigma^* = 0$, and Theorem 3.3 says opening nothing is optimal, so $\text{OPT}_1 = 0$. Thus a polynomial-time solver for the promise problem would decide the NP-hard batch-existence problem from Lemma 6.6. $\square$

7 Proofs of Bicriteria Approximation Results

7.1 Proof of Theorem 5.1: Non-reusable boxes

The proof follows the outline from Section 5.1. We first identify the finite-support reward envelope, then verify that every adaptive policy induces a feasible LP point, and finally analyze the rounding of the opening marginals and the fractional batch count.

For fixed opening marginals $x \in [0, 1]^n$, define the atomized one-winner envelope

$$B(x) = \max \left\{ \sum_i \sum_{\ell \in L_i} v_{i\ell} y_{i\ell} : 0 \leq y_{i\ell} \leq p_{i\ell} x_i \ \forall i, \ell, \quad \sum_i \sum_{\ell \in L_i} y_{i\ell} \leq 1 \right\}.$$

The useful form of this envelope is its threshold representation. As in the high-or-zero case, the identity is the same fractional-knapsack structure for a single winner; the only difference is that each box has been split into its positive atoms.

Proposition 7.1 (Threshold Form of the Atomized Envelope). *For every $x \in [0, 1]^n$,*

$$B(x) = \int_0^\infty \min \left\{ 1, \sum_i x_i \Pr[V_i > t] \right\} dt.$$

Proof. For each positive atom $a = (i, \ell)$, write $w_a = v_{i\ell}$ and $\mu_a = p_{i\ell} x_i$. Then

$$B(x) = \max \left\{ \sum_a w_a y_a : 0 \leq y_a \leq \mu_a \ \forall a, \quad \sum_a y_a \leq 1 \right\}.$$

For any feasible y ,

$$\sum_a w_a y_a = \int_0^\infty \sum_{a:w_a > t} y_a dt \leq \int_0^\infty \min \left\{ 1, \sum_{a:w_a > t} \mu_a \right\} dt.$$

Since $\sum_{a:w_a > t} \mu_a = \sum_i x_i \Pr[V_i > t]$, this gives the upper bound.

For the reverse inequality, order the atoms in nonincreasing value and greedily fill one unit of winner mass:

$$y_a = \min \left\{ \mu_a, \left(1 - \sum_{b < a} y_b \right)^+ \right\}.$$

For every threshold t that is not an atom value, the atoms with $w_a > t$ form a prefix of this order after grouping equal values, and the greedy construction attains

$$\sum_{a:w_a > t} y_a = \min \left\{ 1, \sum_{a:w_a > t} \mu_a \right\}.$$

The two sides can differ only at finitely many atom values, which do not affect the integral. $\square$

With this envelope in place, the next step is to check that the LP is indeed a relaxation of adaptive policies. This is where the nonanticipatory nature of a policy is used: the decision to open box i is made before the realization of V_i is observed.

When upper-bounding a positive-utility policy, it is without loss of generality to assume that the policy opens its first batch with probability one. Indeed, if a policy stops immediately with probability $1 - q$ and otherwise follows a nonempty policy π' , then

$$\text{val}_1(\pi) = q \text{val}_1(\pi').$$

Thus, whenever $\text{val}_1(\pi) > 0$, conditioning on the nonempty branch weakly increases utility.

Lemma 7.2 (General LP Upper Bound). *Every adaptive non-reusable policy π that opens its first batch with probability one induces a feasible LP solution with objective value $\mathbb{E}[M_\pi] - \mathbb{E}[C_\pi]$.*

Proof. Let x_i be the probability that box i is opened, and let z be the expected number of opened batches. Since π opens a first batch, uses nonempty batches, and consumes every opened box,

$$1 \leq z \leq n, \quad \sum_i x_i \leq kz.$$

Moreover,

$$\mathbb{E}[C_\pi] = Tz + \sum_i c_i x_i.$$

Break ties deterministically among boxes attaining the final maximum. For each positive atom, let $y_{i\ell}$ be the probability that box i is the selected positive winner and realizes value $v_{i\ell}$. The event that i is opened is determined before V_i is revealed and is independent of V_i , so

$$y_{i\ell} \leq \Pr[i \text{ is opened and } V_i = v_{i\ell}] = x_i p_{i\ell}.$$

At most one positive atom is selected, so $\sum_{i,\ell} y_{i\ell} \leq 1$, and

$$\mathbb{E}[M_\pi] = \sum_i \sum_{\ell \in L_i} v_{i\ell} y_{i\ell}.$$

Thus the induced point is feasible and has the claimed objective. $\square$

The upper bound above is fractional in the winner variables. To round it, we compare the envelope with the reward of an actual random set of opened boxes. For a set $S \subseteq [n]$, let

$$R(S) = \mathbb{E} \left[\max_{i \in S} V_i \right], \quad R(\emptyset) = 0,$$

and let F be its multilinear extension. The function R is normalized, monotone, and submodular.

The next lemma is the rank-one correlation-gap comparison between independent sampling with marginals x and the one-winner envelope $B(x)$.

Lemma 7.3 (Correlation Gap for the Winner Envelope). *For every $x \in [0, 1]^n$,*

$$F(x) \geq \left(1 - \frac{1}{e}\right) B(x).$$

Proof. If each box i is independently sampled with probability x_i , then

$$F(x) = \int_0^\infty \left(1 - \prod_i (1 - x_i \Pr[V_i > t])\right) dt.$$

For a threshold t , put $s_t = \sum_i x_i \Pr[V_i > t]$. Then

$$1 - \prod_i (1 - x_i \Pr[V_i > t]) \geq 1 - e^{-s_t} \geq \left(1 - \frac{1}{e}\right) \min\{1, s_t\}.$$

Integrating and applying Proposition 7.1 proves the claim. $\square$

The correlation-gap lemma assumes that the marginals x can be used directly. The algorithm must also convert the fractional batch count z into an integer number of batches. The following lemma isolates exactly the loss from that two-point rounding.

Lemma 7.4 (Random Batch-Count Rounding). *Fix $x \in [0, 1]^n$ and $z \geq 1$. Let $\bar{z}$ be obtained by rounding z to $\lfloor z \rfloor$ or $\lceil z \rceil$ as in Algorithm 1, and let $q_i = \min\{1, (\bar{z}/z)x_i\}$. Then*

$$\mathbb{E}_{\bar{z}}[F(q)] \geq \left(1 - \frac{1}{e}\right) h(z) B(x), \quad h(z) = \frac{\lfloor z \rfloor + (z - \lfloor z \rfloor)^2}{z}.$$

Moreover, $h(z) \geq 2(\sqrt{2} - 1)$ for all $z \geq 1$.

Proof. Write $r = \lfloor z \rfloor$ and $\theta = z - r$. If $\bar{z} = r$, then $q = (r/z)x$. Since R is monotone submodular, F is concave along nonnegative rays, so $F(q) \geq (r/z)F(x)$. If $\bar{z} = r + 1$, then $q \geq x$ coordinatewise, and monotonicity gives $F(q) \geq F(x)$. Therefore

$$\mathbb{E}_{\bar{z}}[F(q)] \geq \left((1 - \theta) \frac{r}{r + \theta} + \theta\right) F(x) = h(z) F(x).$$

Combining with Lemma 7.3 gives the first claim.

For the batch-count factor, keep $r \geq 1$ fixed and vary $\theta \in [0, 1]$:

$$h(z) = \frac{r + \theta^2}{r + \theta}.$$

Its minimum on this interval is attained at $\theta^* = \sqrt{r^2 + r} - r$ and equals

$$2 \left(\sqrt{r^2 + r} - r \right) = \frac{2}{\sqrt{1 + 1/r} + 1}.$$

This expression increases with r , so the global minimum is the case $r = 1$, equal to $2(\sqrt{2} - 1)$. $\square$

After the number of batches is integral, the remaining requirement is the ex-post capacity bound. Since capacity is uniform and batches can be formed after the set is chosen, pipage rounding supplies the final conversion from fractional marginals to a feasible set.

Lemma 7.5 (Uniform-Capacity Pipage Rounding). *Let $f : 2^{[n]} \rightarrow \mathbb{R}_{\geq 0}$ be normalized, monotone, and submodular, with multilinear extension F . If $q \in [0, 1]^n$ and $\sum_i q_i \leq K$, then randomized pipage rounding returns S with $|S| \leq K$, $\Pr[i \in S] = q_i$, and*

$$\mathbb{E}[f(S)] \geq F(q).$$

This is the standard pipage-rounding guarantee for monotone submodular functions under a uniform matroid constraint [11].

We now combine the ingredients: the LP upper bound for the benchmark, the correlation-gap comparison for reward, the batch-count rounding loss, pipage rounding for hard capacity, and the cost accounting.

Theorem 5.1 (Finite-Support Non-Reusable Guarantee). *For every finite-support non-reusable instance and every $\varepsilon > 0$, there is a randomized polynomial-time algorithm that returns a policy $\hat{\pi} = \hat{\pi}(\varepsilon)$ satisfying*

$$\text{val}_{\alpha_{\text{NR}}}(\hat{\pi}) \geq \alpha_{\text{NR}} \text{OPT}_1 - \varepsilon, \quad \alpha_{\text{NR}} = \left(1 - \frac{1}{e}\right) 2(\sqrt{2} - 1).$$

The algorithm solves (LP), returns the empty policy if its optimum is non-positive, and otherwise applies Algorithm 1 to an optimal solution.

Proof. Let Φ be the optimum value of the LP. By Lemma 7.2, Φ upper-bounds every nonempty adaptive policy at original costs.

If $\Phi \leq 0$, every nonempty policy has nonpositive original-cost value, and the empty policy is optimal. Assume $\Phi > 0$, and let (x, z, y) be an optimal solution. For this fixed (x, z) , replacing y by an optimizer in the definition of $B(x)$ can only increase the objective, so we may write

$$\Phi = B(x) - Tz - \sum_i c_i x_i.$$

Apply Algorithm 1. For each realized $\bar{z}$,

$$\sum_i q_i \leq \frac{\bar{z}}{z} \sum_i x_i \leq k\bar{z},$$

so Lemma 7.5 applies with the rank- $k\bar{z}$ uniform matroid. Together with Lemma 7.4, the expected reward is at least

$$\alpha_{\text{NR}} B(x), \quad \alpha_{\text{NR}} = \left(1 - \frac{1}{e}\right) 2(\sqrt{2} - 1).$$

The expected setup cost is at most $T\mathbb{E}[\bar{z}] = Tz$. For every item,

$$\Pr[i \in S] = (1 - \theta) \frac{\lfloor z \rfloor}{z} x_i + \theta \min \left\{ 1, \frac{\lfloor z \rfloor + 1}{z} x_i \right\} \leq x_i,$$

so the expected item cost is at most $\sum_i c_i x_i$. Hence

$$\text{val}_{\alpha_{\text{NR}}}(\hat{\pi}) \geq \alpha_{\text{NR}} \left(B(x) - Tz - \sum_i c_i x_i \right) = \alpha_{\text{NR}} \Phi.$$

Since $\Phi \geq \text{OPT}_1$ when the optimum is nonempty, and the empty optimum has value zero, the theorem follows. Solving the LP only to additive accuracy ε loses the corresponding additive amount.

The LP has polynomially many variables and constraints in the finite-support input representation. Moreover, the multilinear extension can be evaluated in polynomial time by summing over the finitely many support thresholds. Hence the pipage-rounding procedure is implementable in polynomial time. $\square$

7.2 Proof of Theorem 5.3: Reusable boxes

The reusable proof follows the same order as the overview in Section 5.2. First, a threshold certificate is converted into an actual repeated policy. Then we show that the feasibility LP contains every integral batch certificate, and finally that a fractional certificate can be rounded to an integral batch while losing only the discounted-cost factor.

Proof of Lemma 5.2. Let $p = \Pr[X_{\mathbf{a}} > \tau]$. The certificate implies $p > 0$. Repeating $\mathbf{a}$ until the first threshold crossing uses a geometric number of openings, so its λ -discounted value is

$$\mathbb{E}[X_{\mathbf{a}} \mid X_{\mathbf{a}} > \tau] - \lambda \frac{K(\mathbf{a})}{p} = \tau + \frac{f_{\tau}(\mathbf{a}) - \lambda K(\mathbf{a})}{p} \geq \tau.$$

$\square$

The threshold certificate is useful only if the relaxation certifies every integral batch that could serve as the optimal repeated batch. The next lemma checks this by assigning the selected excess above threshold to the atom that realizes it.

Lemma 7.6 (RLP $_{\tau}$ Relaxes Integral Batches). *Let $\mathbf{a}$ be a feasible reusable batch with reservation value $\sigma(\mathbf{a})$. Then, for every $\tau \leq \sigma(\mathbf{a})$, there exist variables $y_{i\ell}$ such that $x_i = a_i$ and (x, y) is feasible for RLP $_{\tau}$.*

Proof. Set $x_i = a_i$. After one opening of $\mathbf{a}$, break ties deterministically among copies whose value is the batch maximum and exceeds τ . If the maximum is at most τ , select no atom. Otherwise, let the selected winning copy have type i and atom $v_{i\ell}$, and define $y_{i\ell}$ as the probability of this event. Then $\sum_{i,\ell} y_{i\ell} \leq 1$, and

$$\sum_i \sum_{\ell \in L_i} (v_{i\ell} - \tau)^+ y_{i\ell} = f_{\tau}(\mathbf{a}).$$

The event counted by $y_{i\ell}$ implies that at least one of the a_i fresh copies of type i realizes atom $v_{i\ell}$, so the union bound gives

$$y_{i\ell} \leq a_i p_{i\ell} = p_{i\ell} x_i.$$

Moreover, since $\tau \leq \sigma(\mathbf{a})$,

$$f_{\tau}(\mathbf{a}) \geq f_{\sigma(\mathbf{a})}(\mathbf{a}) = K(\mathbf{a}).$$

Therefore the certificate constraint of RLP $_{\tau}$ is satisfied, and (x, y) , with $x_i = a_i$, is feasible for RLP $_{\tau}$. □

Conversely, a certified fractional solution must be implemented as a true batch. We fill the k slots independently according to the fractional copy counts and compare the resulting threshold tails with the LP winner envelope.

Lemma 7.7 (Reusable Slot Rounding). *Let (x, y) be feasible for RLP $_{\tau}$. Fill k independent slots by choosing type i with probability x_i/k and a dummy type otherwise. Let $\mathbf{A}$ be the resulting random multiplicity vector. Then*

$$\mathbb{E}[f_{\tau}(\mathbf{A})] \geq \alpha_R \sum_i \sum_{\ell \in L_i} (v_{i\ell} - \tau)^+ y_{i\ell}, \quad \mathbb{E}\left[\sum_i c_i A_i\right] = \sum_i c_i x_i.$$

Proof. For $t \geq \tau$, set

$$q_x(t) = \sum_i x_i \Pr[V_i > t].$$

Since $\sum_i x_i \leq k$, we have $0 \leq q_x(t) \leq k$. One rounded slot exceeds t with probability $q_x(t)/k$, so

$$\Pr[X_{\mathbf{A}} > t] = 1 - \left(1 - \frac{q_x(t)}{k}\right)^k \geq \alpha_R \min\{1, q_x(t)\}, \quad \alpha_R = 1 - \frac{1}{e}.$$

The LP winner mass above the same level is bounded by the same envelope:

$$\sum_{i,\ell: v_{i\ell} > t} y_{i\ell} \leq \min\left\{1, \sum_i x_i \Pr[V_i > t]\right\}.$$

Therefore, for every $t \geq \tau$,

$$\Pr[X_{\mathbf{A}} > t] \geq \alpha_R \sum_{i,\ell: v_{i\ell} > t} y_{i\ell}.$$

Integrating over thresholds gives the excess-reward bound. The cost identity is linear because each slot chooses type i with probability x_i/k . □

It remains to combine threshold monotonicity, slot rounding, and the structural characterization of optimal reusable policies by a single reservation-index batch.

Theorem 5.3 (Finite-Support Reusable Guarantee). *For every finite-support reusable instance with finite batch capacity k and every $\varepsilon > 0$, Algorithm 2 runs in polynomial time and returns a policy $\hat{\pi} = \hat{\pi}(\varepsilon)$ satisfying*

$$\text{val}_{\alpha_R}(\hat{\pi}) \geq \text{OPT}_1 - \varepsilon, \quad \alpha_R = 1 - \frac{1}{e}.$$

Equivalently, the algorithm provides a $(1, \alpha_R)$ discounted-cost bicriteria guarantee.

Proof. The case $T = 0$ is handled exactly by Observation 2.3, so assume $T > 0$. Certification is monotone: if τ is LP-certified, then every $\tau' \leq \tau$ is also LP-certified, because lowering the threshold increases the coefficients $(v_{i\ell} - \tau)^+$ and makes the certificate constraint easier to satisfy.

Suppose RLP_τ has a feasible solution (x, y) . By Lemma 7.7,

$$\mathbb{E} \left[f_\tau(\mathbf{A}) - \alpha_R \left(T + \sum_i c_i A_i \right) \right] \geq \alpha_R \left(\sum_i \sum_{\ell \in L_i} (v_{i\ell} - \tau)^+ y_{i\ell} - T - \sum_i c_i x_i \right) \geq 0.$$

Since the input distributions have finite support, the conditional expectation of the displayed certificate can be computed exactly after any partial assignment of slots. The method of conditional expectations returns a realization $\mathbf{a}$ satisfying

$$f_\tau(\mathbf{a}) \geq \alpha_R K(\mathbf{a}).$$

Because $T > 0$, the all-dummy realization has negative certificate value, so the returned batch is nonempty. By Lemma 5.2, the repeated policy for this batch has α_R -discounted value at least τ .

If $\text{OPT}_1 = 0$, the algorithm either returns the empty policy or a certified repeated batch, both of which have nonnegative α_R -discounted value. Suppose $\text{OPT}_1 > 0$. By Theorem 3.3, there is an optimal reusable batch $\mathbf{a}^*$ with reservation value $\sigma^* = \text{OPT}_1$. For every $\tau \leq \sigma^*$,

$$f_\tau(\mathbf{a}^*) \geq f_{\sigma^*}(\mathbf{a}^*) = K(\mathbf{a}^*).$$

Lemma 7.6 therefore implies that every $\tau \leq \text{OPT}_1$ is LP-certified. Monotonicity of certification lets binary search return a certified threshold $\hat{\tau} \geq \text{OPT}_1 - \varepsilon$. The policy returned for $\hat{\tau}$ satisfies

$$\text{val}_{\alpha_R}(\hat{\pi}) \geq \hat{\tau} \geq \text{OPT}_1 - \varepsilon. \quad \square$$

AI Disclosure. OpenAI’s ChatGPT (GPT-5.5) was used to support the writing and mathematical development of the paper, including editing, improving the exposition, and formalizing parts of the analysis. In particular, it was used to develop illustrative examples and to extend results from high-or-zero distributions to general finite-support distributions. The authors made all final decisions regarding the mathematical statements and proofs, and independently re-derived, verified, and revised all AI-assisted mathematical content. The authors assume responsibility for all content.

References

- [1] Arpit Agarwal, Rohan Ghuge, and Viswanath Nagarajan. Semi-bandit learning for monotone stochastic optimization*. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1260–1274, 2024.

- [2] Shipra Agrawal, Yichuan Ding, Amin Saberi, and Yinyu Ye. Correlation robust stochastic optimization. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA. SIAM, 2010.
- [3] Alexia Atsidakou, Constantine Caramanis, Evangelia Gergatsouli, Orestis Papadigenopoulos, and Christos Tzamos. Contextual pandora’s box. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI’24/IAAI’24/EAAI’24. AAAI Press, 2024.
- [4] Alexandre Belloni, Yan Chen, and Yehua Wei. Online pandora’s box for contextual llm cascading, 2026.
- [5] Ben Berger, Tomer Ezra, Michal Feldman, and Federico Fusco. Pandora’s problem with combinatorial cost. In *Proceedings of the 24th ACM Conference on Economics and Computation*, EC, 2023.
- [6] Hedyeh Beyhaghi and Linda Cai. Pandora’s problem with nonobligatory inspection: Optimal structure and a ptas. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, page 803–816, New York, NY, USA, 2023. Association for Computing Machinery.
- [7] Hedyeh Beyhaghi and Robert Kleinberg. Pandora’s problem with nonobligatory inspection. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, EC ’19, page 131–132, New York, NY, USA, 2019. Association for Computing Machinery.
- [8] Shant Boodaghians, Federico Fusco, Philip Lazos, and Stefano Leonardi. Pandora’s Box problem with order constraints. *Mathematics of Operations Research*, 48(1):498–519, 2024.
- [9] Robin Bowers and Bo Waggoner. Matching with nested and bundled pandora boxes. In Marios Mavronicolas, Qi Qi, and Grant Schoenebeck, editors, *Web and Internet Economics*, pages 61–83, Cham, 2026. Springer Nature Switzerland.
- [10] Domagoj Bradač, Sahil Singla, and Goran Žužić. (Near) Optimal Adaptivity Gaps for Stochastic Multi-Value Probing. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, APPROX/RANDOM, pages 49:1–49:21, 2019.
- [11] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a Matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.
- [12] Shuchi Chawla, Dimitris Christou, and Trung Dang. Commitment gap via correlation gap, 2025. arXiv:2508.20246.
- [13] Shuchi Chawla, Dimitris Christou, Amit Harlev, and Ziv Scully. Combinatorial selection with costly information, 2024. arXiv:2412.03860.
- [14] Shuchi Chawla, Evangelia Gergatsouli, Jeremy McMahan, and Christos Tzamos. Approximating Pandora’s Box with Correlations. In Nicole Megow and Adam Smith, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2023)*, volume 275 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:24, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

- [15] Shuchi Chawla, Evangelia Gergatsouli, Yifeng Teng, Christos Tzamos, and Ruimin Zhang. Pandora’s box with correlations: Learning and approximation. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1214–1225, 2020.
- [16] Hossein Esfandiari, Mohammad Taghi Haji Aghayi, Brendan Lucier, and Michael Mitzenmacher. Online pandora’s boxes and bandits. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI’19/IAAI’19/EAAI’19*. AAAI Press, 2019.
- [17] Hu Fu, Jiawei Li, and Daogao Liu. Pandora box problem with nonobligatory inspection: Hardness and approximation scheme. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, page 789–802, New York, NY, USA, 2023. Association for Computing Machinery.
- [18] Zijun Gao, Yanjun Han, Zhimei Ren, and Zhengqing Zhou. Batched multi-armed bandits problem. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [19] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [20] Khashayar Gatmiry, Thomas Kesselheim, Sahil Singla, and Yifan Wang. Bandit algorithms for prophet inequality and pandora’s box. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 462–500, 2024.
- [21] Evangelia Gergatsouli and Christos Tzamos. Online learning for min sum set cover and pandora’s box. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 7382–7403. PMLR, 17–23 Jul 2022.
- [22] Evangelia Gergatsouli and Christos Tzamos. Weitzman’s rule for pandora’s box with correlations. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS ’23, Red Hook, NY, USA, 2023. Curran Associates Inc.
- [23] Aditya Grover, Todor Markov, Peter Attia, Norman Jin, Nicolas Perkins, Bryan Cheong, Michael Chen, Zi Yang, Stephen Harris, William Chueh, and Stefano Ermon. Best arm identification in multi-armed bandits with delayed feedback. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 833–842. PMLR, 09–11 Apr 2018.
- [24] Sudipto Guha, Kamesh Munagala, and Martin Pal. Multiarmed bandit problems with delayed feedback, 2010. arXiv:1011.1161.
- [25] Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. Adaptivity gaps for stochastic probing: Submodular and XOS functions. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA, pages 1688–1702. SIAM, 2017.

- [26] Pooria Joulani, András György, and Csaba Szepesvári. Online learning under delayed feedback. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*. PMLR, 2013.
- [27] Yusuf Kalayci, Vinod Raman, and Shaddin Dughmi. Optimal stopping vs best-of- n for inference time optimization, 2025.
- [28] Robert Kleinberg, Bo Waggoner, and E. Glen Weyl. Descending price optimally coordinates search. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, EC '16, page 23–24, New York, NY, USA, 2016. Association for Computing Machinery.
- [29] Junyan Liu, Ziyun Chen, Kun Wang, Haipeng Luo, and Lillian J. Ratliff. Improved regret and contextual linear extension for pandora’s box and prophet inequality. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.
- [30] Wojciech Olszewski and Richard Weber. A more general pandora rule? *Journal of Economic Theory*, 160(C):429–437, 2015.
- [31] Vianney Perchet, Philippe Rigollet, Sylvain Chassang, and Erik Snowberg. Batched bandit problems. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 1456–1456, Paris, France, 03–06 Jul 2015. PMLR.
- [32] Ciara Pike-Burke, Shipra Agrawal, Csaba Szepesvari, and Steffen Grunewalder. Bandits with delayed, aggregated anonymous feedback. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4105–4113. PMLR, 10–15 Jul 2018.
- [33] Ziv Scully and Laura Doval. Local hedging approximately solves Pandora’s Box problems with nonobligatory inspection, 2024. arXiv:2410.19011.
- [34] Sahil Singla. The price of information in combinatorial optimization. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA. SIAM, 2018.
- [35] Tobias Sommer Thune, Nicolò Cesa-Bianchi, and Yevgeny Seldin. *Nonstochastic multiarmed bandits with unrestricted delays*. Curran Associates Inc., Red Hook, NY, USA, 2019.
- [36] Arun Verma, Zhongxiang Dai, and Bryan Kian Hsiang Low. Bayesian optimization under stochastic delayed feedback. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 22145–22167. PMLR, 17–23 Jul 2022.
- [37] Claire Vernade, Olivier Cappé, and Vianney Perchet. Stochastic Bandit Models for Delayed Conversions. In *Conference on Uncertainty in Artificial Intelligence*, Sydney, Australia, August 2017.
- [38] Martin L. Weitzman. Optimal search for the best alternative. *Econometrica*, 47(3):641–654, 1979.
- [39] Han Wu and Stefan Wager. Thompson sampling with unrestricted delays. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, EC '22, page 937–955, New York, NY, USA, 2022. Association for Computing Machinery.

A Additional Structural Details

A.1 A lower bound for consecutive batching

We sort boxes by their individual reservation values computed with respect to item costs only: σ_i is defined by

$$\mathbb{E}[(V_i - \sigma_i)^+] = c_i.$$

Fix an ordering of the boxes. A consecutive-batch policy is a policy that only opens intervals in the current remaining order.

We first give the full calculation for the example from the main text. The setup cost is $T = 10$, the batch capacity is $k = 2$, and the boxes are ordered as A, D, B . Box A has item cost $c_A = 1$, with $\Pr[V_A = 100] = 0.1$ and $\Pr[V_A = 0] = 0.9$. Box B is deterministic with value 9 and item cost $c_B = 1$. Box D is deterministic with value 100 and item cost $c_D = 91$.

The reservation values are

$$\sigma_A = 100 - \frac{1}{0.1} = 90, \quad \sigma_D = 100 - 91 = 9, \quad \sigma_B = 9 - 1 = 8.$$

Thus the reservation-value order is $A \succ D \succ B$. The utilities of the consecutive feasible batches are

S	$\{A\}$	$\{D\}$	$\{B\}$	$\{A, D\}$	$\{D, B\}$
$u(S)$	-1	-1	-2	-2	-2.

Indeed, $u(\{A\}) = 10 - 10 - 1 = -1$, $u(\{D\}) = 100 - 10 - 91 = -1$, $u(\{B\}) = 9 - 10 - 1 = -2$, and both pairs containing D have maximum value 100 and total cost 102. On the other hand,

$$u(\{A, B\}) = \mathbb{E}[\max\{V_A, V_B\}] - 10 - 1 - 1 = (0.1 \cdot 100 + 0.9 \cdot 9) - 10 - 1 - 1 = 6.1 > 0.$$

So the non-consecutive batch $\{A, B\}$ is profitable, while every consecutive batch is unprofitable.

Moreover, every consecutive-batch policy has value at most zero. If such a policy ever opens D , then its final reward is at most 100, while opening D alone already costs $T + c_D = 101$. Thus any policy opening D has negative value. If the policy never opens D , then it can only open A and B as singleton batches. The expected marginal reward from opening A is at most $10 < T + c_A = 11$, and the expected marginal reward from opening B is at most $9 < T + c_B = 11$. Hence opening either singleton cannot increase expected value, and the empty policy is optimal among consecutive-batch policies.

We now prove the bicriteria lower bound.

Theorem A.1. *Fix $k \geq 2$ and $\alpha \in (1/2, 1]$. There is a finite-support non-reusable instance with setup cost $T = 1$ and batch capacity k such that the reservation-value order is*

$$A \succ D_1 \succeq \dots \succeq D_k \succ B,$$

where $D_1, \dots, D_k$ are identical decoy boxes, and every consecutive-batch policy π satisfies

$$\text{val}_\alpha(\pi) \leq 0,$$

while $\text{OPT}_1 > 0$. Consequently, no algorithm restricted to consecutive batches in this order can guarantee an (α, α) -bicriteria approximation for any $\alpha > 1/2$.

Proof. The construction has two useful boxes and a block of decoy boxes. The useful boxes A and B are profitable only when opened together: A is a rare large prize, B is a moderate sure prize, and their maximum pays for one setup cost. The decoy boxes only serve to separate A from B in the reservation value order. They are chosen to have reservation values between those of A and B , but their total expected reward is arbitrarily small.

Choose β with $1/2 < \beta < \alpha$. This will be the expected value of each useful box. Then choose $d > 0$ such that $\beta + d < \alpha$; d will upper-bound the total expected value of all decoys. Since $\beta > 1/2$, choose $p \in (0, 1)$ small enough that

$$(2 - p)\beta > 1.$$

Finally choose $\varepsilon > 0$ small enough that

$$(2 - p)\beta - 1 - 2\varepsilon > 0 \quad \text{and} \quad \varepsilon < \beta.$$

Next, we create the boxes. Box A has item cost $c_A = \varepsilon$ and

$$\Pr[V_A = \beta/p] = p, \quad \Pr[V_A = 0] = 1 - p.$$

Box B has item cost $c_B = \varepsilon$ and deterministic value $V_B = \beta$. Thus

$$\mathbb{E}[V_A] = \mathbb{E}[V_B] = \beta.$$

Their reservation values are

$$\sigma_A = \frac{\beta - \varepsilon}{p}, \quad \sigma_B = \beta - \varepsilon,$$

so $\sigma_A > \sigma_B$.

It remains to insert k boxes between A and B without giving a consecutive policy useful value. Choose any target reservation value σ_D with $\sigma_A > \sigma_D > \sigma_B$. Each decoy is a rare high value: choose $G > \max\{\sigma_D, d/k\}$, let

$$\Pr[V_D = G] = \frac{d}{kG}, \quad \Pr[V_D = 0] = 1 - \frac{d}{kG},$$

and set

$$c_D = \frac{d}{kG}(G - \sigma_D).$$

Thus, each decoy box has expected value d/k . At the same time, its reservation value is exactly the prescribed value σ_D , since

$$\mathbb{E}[(V_D - \sigma_D)^+] = \frac{d}{kG}(G - \sigma_D) = c_D.$$

We create independent copies $D_1, \dots, D_k$ of the same box type D . Therefore the decreasing reservation value order is

$$A \succ D_1 \succeq \dots \succeq D_k \succ B.$$

Now consider any consecutive-batch policy π . The key point is that every nonempty batch pays one setup cost, while a consecutive policy can extract at most one useful box, plus negligible decoy box value, per opened batch. Let N be the number of nonempty batches opened, let I_A, I_B indicate whether A, B are opened, and let I_{D_j} indicate whether the j -th decoy box is opened. If both A and B are opened, then at least two batches must be opened: in the initial order, any interval containing both useful boxes also contains all k decoys and therefore has size $k + 2$. Hence, pathwise,

$$I_A + I_B \leq N.$$

Also, if $N = 0$, no decoy box is opened, while if $N \geq 1$, the total expected value of all opened decoy boxes is at most the total expected value d of all decoys. Therefore, pathwise,

$$\frac{d}{k} \sum_{j=1}^k I_{D_j} \leq dN.$$

Let M_π be the maximum reward observed by π . We upper-bound this maximum by the sum of all opened rewards. Since rewards are nonnegative,

$$M_\pi \leq I_A V_A + I_B V_B + \sum_{j=1}^k I_{D_j} V_{D_j}.$$

The event that a box is opened is determined before that box's value is observed, and values are independent. Therefore,

$$\begin{aligned} \mathbb{E}[M_\pi] &\leq \beta \mathbb{E}[I_A + I_B] + \frac{d}{k} \sum_{j=1}^k \mathbb{E}[I_{D_j}] \\ &\leq (\beta + d) \mathbb{E}[N]. \end{aligned}$$

The setup cost alone is N , and item costs are nonnegative, so $C_\pi \geq N$. Thus

$$\text{val}_\alpha(\pi) = \mathbb{E}[M_\pi] - \alpha \mathbb{E}[C_\pi] \leq (\beta + d - \alpha) \mathbb{E}[N] \leq 0.$$

On the other hand, the unrestricted policy can open the non-consecutive batch $\{A, B\}$, which is feasible because $k \geq 2$. Since $\beta/p > \beta$,

$$\mathbb{E}[\max\{V_A, V_B\}] = p \cdot \frac{\beta}{p} + (1-p)\beta = (2-p)\beta.$$

Therefore the original-cost value of opening $\{A, B\}$ is

$$(2-p)\beta - 1 - 2\varepsilon > 0.$$

Hence $\text{OPT}_1 > 0$, while every consecutive-batch policy has α -discounted value at most zero. This proves the theorem. $\square$

A.2 Numerical details for the strict adaptivity-gap example

We tuned parameters of the instance used in Example 3.1 with at most two decimal places. The setup cost is $T = 0.44$, the batch capacity is $k = 2$, and the boxes are as follows. Box 1 is deterministic and has value $V_1 = 0.83$ and cost $c_1 = 0.40$. Box 2 has distribution

$$\Pr[V_2 = 0] = 0.48, \quad \Pr[V_2 = 0.83] = 0.43, \quad \Pr[V_2 = 100] = 0.09,$$

and cost $c_2 = 8.50$. Box 3 has distribution

$$\Pr[V_3 = 100] = 0.05, \quad \Pr[V_3 = 0] = 0.95,$$

and cost $c_3 = 4.16$.

Consider the adaptive policy that first opens box 2. If $V_2 = 0$, it opens $\{1, 3\}$. If $V_2 = 0.83$, it opens $\{3\}$. If $V_2 = 100$, it stops. On the low and medium branches, the expected final reward after opening the second batch is

$$0.05 \cdot 100 + 0.95 \cdot 0.83 = 5.7885.$$

Therefore the expected reward of the adaptive policy is

$$\mathbb{E}[M] = (0.48 + 0.43) \cdot 5.7885 + 0.09 \cdot 100 = 14.267535.$$

The relevant batch costs are

$$K(\{2\}) = 0.44 + 8.50 = 8.94, \quad K(\{1, 3\}) = 0.44 + 0.40 + 4.16 = 5.00,$$

and

$$K(\{3\}) = 0.44 + 4.16 = 4.60.$$

Hence

$$\mathbb{E}[C] = 8.94 + 0.48 \cdot 5.00 + 0.43 \cdot 4.60 = 13.318.$$

Thus the adaptive policy has value

$$\mathbb{E}[M] - \mathbb{E}[C] = 0.949535.$$

We now compute the best fixed-menu value. Since a fixed-menu policy can ignore available batches, it suffices to enumerate maximal disjoint menus. For each fixed menu $\mathcal{F}$, we treat every batch $B \in \mathcal{F}$ as a Pandora box with reward $X_B = \max_{i \in B} V_i$ and cost $K(B)$, run Weitzman's rule, and record the resulting fixed-menu utility $\text{val}_1(\mathcal{F})$.

For $k = 2$, the maximal disjoint menus and their values are

$\mathcal{F}$	$\text{val}_1(\mathcal{F})$
$\{\{1\}, \{2\}, \{3\}\}$	0.796055
$\{\{1, 2\}, \{3\}\}$	0.794535
$\{\{1, 3\}, \{2\}\}$	0.795380
$\{\{2, 3\}, \{1\}\}$	0.789055.

Therefore

$$\text{OPT}_{\text{fix}}(T, 2) = 0.796055.$$

Since

$$\frac{0.949535}{0.796055} > 1.1928,$$

this gives a strict adaptivity gap of about 1.193.

For $k \geq 3$, the only additional maximal menu is the single batch $\{\{1, 2, 3\}\}$. Its one-shot value is

$$\mathbb{E}[\max\{V_1, V_2, V_3\}] - K(\{1, 2, 3\}) = 14.267535 - 13.50 = 0.767535,$$

which is still below 0.796055. Hence the same rounded instance gives a strict adaptive-versus-fixed-menu gap for every $k \geq 2$.

A.3 Capped-value reduction for non-reusable policies

For a feasible batch B , let $X_B = \max_{i \in B} V_i$. If $K(B) = 0$, define

$$\kappa_B = X_B, \quad \beta_B = 0.$$

If $0 < K(B) \leq \mathbb{E}[X_B]$, let σ_B solve

$$K(B) = \mathbb{E}[(X_B - \sigma_B)^+].$$

For such positive-cost batches, define

$$\kappa_B = \min\{X_B, \sigma_B\}, \quad \beta_B = (X_B - \sigma_B)^+.$$

A positive-cost batch with $K(B) > \mathbb{E}[X_B]$ is called overpriced.

Lemma A.2 (Overpriced Batches Can Be Skipped). *In every finite non-reusable instance, there is an optimal policy that never opens an overpriced batch.*

Proof. Let $W(U, m)$ be the optimal expected continuation payoff when the unopened boxes are U and the current best observed value is m . Increasing the incumbent cannot hurt, and increasing it by d can improve the final selected value by at most d . Thus $W(U, \cdot)$ is nondecreasing and 1-Lipschitz.

Suppose a policy is about to open an overpriced batch $B \subseteq U$. Its continuation payoff is at most

$$\begin{aligned} -K(B) + \mathbb{E}W(U \setminus B, \max\{m, X_B\}) &\leq W(U \setminus B, m) - K(B) + \mathbb{E}[(X_B - m)^+] \\ &\leq W(U \setminus B, m) - K(B) + \mathbb{E}[X_B] \\ &< W(U \setminus B, m). \end{aligned}$$

Discarding B without opening it and then following the same continuation is therefore strictly better than opening B . $\square$

Lemma A.3 (Capped-Value Upper Bound for Adaptive Policies). *Let O_π be the random family of batches opened by a finite non-reusable policy π that uses no overpriced positive-cost batches. Then*

$$\text{val}_1(\pi) \leq \mathbb{E} [\max(\{0\} \cup \{\kappa_B : B \in O_\pi\})].$$

Proof. Index the possible rounds by t . Let $\mathcal{H}_t$ be the history immediately before round t , let I_t indicate that the round is reached, and let B_t be the batch selected at that history. Both I_t and B_t are $\mathcal{H}_t$ -measurable. Since B_t contains only unopened boxes, its primitive values are independent of $\mathcal{H}_t$. Therefore, on every reached history,

$$\mathbb{E}[\beta_{B_t} \mid \mathcal{H}_t] = \mathbb{E}[(X_{B_t} - \sigma_{B_t})^+ \mid \mathcal{H}_t] = K(B_t),$$

and hence $\mathbb{E}[I_t \cdot \beta_{B_t}] = \mathbb{E}[I_t \cdot K(B_t)]$.

Let A_t indicate, with deterministic tie-breaking, that round t supplies the final maximum. Then $A_t \leq I_t$, and

$$\begin{aligned} \mathbb{E}[M_\pi] - \mathbb{E}[C_\pi] &= \mathbb{E} \sum_t A_t \cdot (\kappa_{B_t} + \beta_{B_t}) - \mathbb{E} \sum_t I_t \cdot K(B_t) \\ &\leq \mathbb{E} \sum_t A_t \cdot \kappa_{B_t} \\ &\leq \mathbb{E} [\max(\{0\} \cup \{\kappa_B : B \in O_\pi\})]. \end{aligned}$$

All sums are finite because each primitive box can be opened at most once. $\square$

Proposition A.4 (Fixed Menus Are Classical Pandora Instances). *Fix a finite disjoint family $\mathcal{F}$ of feasible batches, each of which is either zero-cost or non-overpriced. Treat each batch $B \in \mathcal{F}$ as one box in a finite classical Pandora instance, with value distribution X_B and opening cost $K(B)$. These induced boxes are independent, so Weitzman's rule is optimal, and its value is*

$$\mathbb{E} [\max(\{0\} \cup \{\kappa_B : B \in \mathcal{F}\})].$$

Proof. Disjointness makes the induced box prizes $(X_B)_{B \in \mathcal{F}}$ independent. The claim is Weitzman's theorem applied to these boxes [38]. $\square$

A.4 Proof of the non-reusable adaptivity gap

Lemma A.5 (Random-Path Rounding). *Let $Y_1, \dots, Y_N$ be independent latent variables. Each action e has a support $S(e) \subseteq [N]$, reveals a measurable observation of $(Y_i)_{i \in S(e)}$, and carries a nonnegative integrable mark determined by that observation. Assume every feasible action sequence has pairwise-disjoint supports. For every finite-depth adaptive decision tree, there is a deterministic feasible path whose expected maximum mark is at least half the adaptive expected maximum mark.*

Proof. It suffices to prove the claim for the randomized path obtained by running the tree on an independent ghost instance and then probing the resulting path on a fresh real instance; one deterministic path attains at least the average.

Fix a threshold $\tau \geq 0$. Let $a_\tau(\mathcal{T})$ be the probability that the adaptive tree $\mathcal{T}$ observes a mark at least τ , and let $r_\tau(\mathcal{T})$ be the same probability for the ghost-path policy. We prove $r_\tau(\mathcal{T}) \geq a_\tau(\mathcal{T})/2$ by induction on the tree depth.

Let the root action be e , with mark X_e , observation O_e , and continuation tree $\mathcal{T}_o$ after observation o . Put $p = \mathbb{P}[X_e \geq \tau]$. Then

$$a_\tau(\mathcal{T}) = p + \mathbb{E} \left[\mathbf{1}_{\{X_e < \tau\}} a_\tau(\mathcal{T}_{O_e}) \right].$$

For the ghost-path policy, the real root mark is independent of the ghost observation $\tilde{O}_e$. Moreover, every feasible continuation has support disjoint from the root support. The induction hypothesis gives

$$r_\tau(\mathcal{T}) \geq p + \frac{1-p}{2} \cdot \mathbb{E} \left[a_\tau(\mathcal{T}_{\tilde{O}_e}) \right].$$

Since $\tilde{O}_e$ and O_e have the same marginal distribution,

$$\mathbb{E} \left[a_\tau(\mathcal{T}_{\tilde{O}_e}) \right] \geq \mathbb{E} \left[\mathbf{1}_{\{X_e < \tau\}} a_\tau(\mathcal{T}_{O_e}) \right] =: B.$$

Therefore

$$r_\tau(\mathcal{T}) - \frac{1}{2} \cdot a_\tau(\mathcal{T}) \geq \frac{p}{2} \cdot (1 - B) \geq 0.$$

Integrating over thresholds using $\mathbb{E}[Z] = \int_0^\infty \mathbb{P}[Z \geq \tau] d\tau$ proves the expected maximum bound. $\square$

Proof of Theorem 3.2. The definitions above already handle zero-cost batches through $\kappa_B = X_B$ and $\beta_B = 0$. By Lemma A.2, take an optimal non-reusable policy that never opens an overpriced positive-cost batch. Lemma A.3 upper-bounds its net value by the expected maximum capped value among the batches it opens.

Apply Lemma A.5 to the adaptive decision tree whose actions are feasible batches, whose supports are the primitive boxes in the batch, and whose mark is the capped value κ_B . Along every realized non-reusable path these supports are disjoint. Hence some fixed disjoint family has expected maximum capped value at least half the adaptive capped value. Proposition A.4 identifies this quantity with the value of the corresponding fixed-menu Pandora instance. Thus

$$\text{OPT}_1^{\text{BP}(T,k)} \leq 2 \cdot \text{OPT}_{\text{fix}}(T, k).$$

$\square$

A.5 Proof of Theorem 3.3: Reusable reservation index

Proof of Theorem 3.3. For finite k , there are only $\binom{n+k}{k} - 1$ feasible reusable batches. For $k = \infty$, every non-overpriced batch $\mathbf{a}$ satisfies

$$T + \sum_i a_i \cdot c_i = K(\mathbf{a}) \leq \mathbb{E}[X_{\mathbf{a}}] \leq V_{\max}.$$

Since every $c_i > 0$, this implies $a_i \leq (V_{\max} - T)_+ / c_i$ for each i , so only finitely many non-overpriced batches remain. Hence σ^* is attained whenever it is positive.

For every feasible reusable batch $\mathbf{a}$ and every $s \geq \sigma^*$,

$$\mathbb{E}[(X_{\mathbf{a}} - s)^+] \leq K(\mathbf{a}).$$

For a non-overpriced batch this follows because the excess function decreases in s and $s \geq \sigma(\mathbf{a})$. For an overpriced batch it follows from $\mathbb{E}[(X_{\mathbf{a}} - s)^+] \leq \mathbb{E}[X_{\mathbf{a}}] < K(\mathbf{a})$.

Set $W(m) = \max\{m, \sigma^*\}$. If $m < \sigma^*$, opening any batch $\mathbf{a}$ and then continuing according to W gives at most

$$-K(\mathbf{a}) + \mathbb{E}W(\max\{m, X_{\mathbf{a}}\}) = \sigma^* - K(\mathbf{a}) + \mathbb{E}[(X_{\mathbf{a}} - \sigma^*)^+] \leq \sigma^* = W(m).$$

If $m \geq \sigma^*$, the corresponding bound is

$$-K(\mathbf{a}) + \mathbb{E}\max\{m, X_{\mathbf{a}}\} = m - K(\mathbf{a}) + \mathbb{E}[(X_{\mathbf{a}} - m)^+] \leq m = W(m).$$

Backward induction bounds every finite-horizon policy by $W(m)$. For an admissible infinite-horizon policy, truncate after h openings. Bounded convergence applies to the terminal reward and monotone convergence applies to the accumulated cost, so the same upper bound survives as $h \rightarrow \infty$.

If $\sigma^* > 0$, let $\mathbf{a}^*$ attain it and put $p = \mathbb{P}[X_{\mathbf{a}^*} > \sigma^*]$. Positive opening cost implies $p > 0$. Repeating $\mathbf{a}^*$ until the first threshold crossing uses a geometric number of openings and has value

$$\mathbb{E}[X_{\mathbf{a}^*} \mid X_{\mathbf{a}^*} > \sigma^*] - \frac{K(\mathbf{a}^*)}{p} = \sigma^*,$$

by the reservation equation. This attains $W(m)$ whenever $m < \sigma^*$; for $m \geq \sigma^*$, stopping attains $W(m)$. $\square$