# Pandora Box Problem with Nonobligatory Inspection: Hardness and Approximation Scheme

Hu Fu * Jiawei Li † Daogao Liu ‡

**Abstract**

Weitzman (1979) introduced the Pandora Box problem as a model for sequential search with inspection costs, and gave an elegant index-based policy that attains provably optimal expected payoff. In various scenarios, the searching agent may select an option without making a costly inspection. The variant of the Pandora box problem with non-obligatory inspection has attracted interest from both economics and algorithms researchers. Various simple algorithms have proved suboptimal, with the best known 0.8-approximation algorithm due to Guha et al. (2008). No hardness result for the problem was known.

In this work, we show that it is NP-hard to compute an optimal policy for Pandora's problem with nonobligatory inspection. We also give a polynomial-time approximation scheme (PTAS) that computes policies with an expected payoff at least $(1-\epsilon)$-fraction of the optimal, for arbitrarily small $\epsilon > 0$. On the side, we show the decision version of the problem to be in NP.

## 1 Introduction

Weitzman [19] introduced the Pandora Box problem in 1979 as a model for sequential search with inspection costs. An agent is to select from $n$ options, viewed as locked boxes. Each box $i$ contains a value $v_i$ drawn independently from a known distribution $F_i$, but $v_i$ is revealed only if the agent opens box $i$, incurring a search cost $c_i$. At any step, the agent may choose to either select a box that has been opened and quit, or to open another box. Her goal is to maximize the expected value of the box selected, minus the search costs accrued along the way.

A policy for such a stochastic sequential problem may conceivably be adaptive in intricate ways. It may therefore come as a surprise that Weitzman showed the problem of admitting a simple and elegant optimal policy: there are indices, one for each box $i$, computable from $F_i$ and $c_i$, such that a ranking policy based on these indices maximizes the expected payoff.

Weitzman's formulation and the index-based policy have been highly influential, and serve as the basis for many models that involve search frictions (see e.g. Armstrong [2] for a survey, and Derakhshan et al. [7] for a recent example). It was later recognized that the index-based policy was a special case of Gittins [13]'s optimal algorithm for Bayesian bandits, an algorithm important in its own right, with many applications.

In Weitzman [19]'s motivating scenario, one searches for a technology among various alternatives; to be able to adopt any technology, research expenditure (the search cost) must be incurred before one sees the technology's benefit. In many other scenarios, however, it is more natural to allow the agent the possibility to select an option without inspecting it. For example, in a wireless system with multiple channels, a user intending to transmit a control packet may spend energy and time probing the transmission states of

*ITCS, Shanghai University of Finance and Economics. fuhu@mail.shufe.edu.cn
†University of Texas at Austin. davidlee@cs.utexas.edu
‡University of Washington. dgliu@uw.edu

channels, but may also decide to use a channel without knowing its exact state [14]; a student making a school choice may not always pay a campus visit when she is confident enough that one choice is superior [8]. This problem variant is known as *Pandora's problem with non-obligatory inspection* (PNOI). In contrast to the original Pandora Box problem, various simple ranking policies are not optimal [8]; in fact, optimal policies may be truly adaptive, in the sense that the order in which two options are inspected should depend on the outcome of the inspection of a third option. One may easily attain at least $\frac{1}{2}$ of the optimal payoff by using the better of two simple policies: (i) Weitzman's index-based policy, and (ii) selecting the box with the highest expected value without any inspection. Nontrivial algorithms with better approximation ratios have been proposed [14, 4], the best approximation ratio known so far being $0.8$ [14]. On the other hand, it has been unresolved whether computing optimal policies is intractable — the problem could be anywhere between P and PSPACE-complete [4].

In this work, we show that PNOI is NP-hard, giving the first hardness result for the problem (Theorem 3.2). We also give a polynomial time approximation scheme (PTAS) for PNOI (Theorem 4.5). On the side, we show the decision version of PNOI to be in NP (Corollary 2.4).

**Computational Hardness for PNOI** Before discussing the main idea of our hardness result, it is helpful to first relate a structure theorem on optimal policies that we strengthen from Guha et al. [14], The structure theorem by [14] is crucial for the $0.8$-approximation algorithm, and shows the existence of an optimal policy for which a unique box is possibly taken without inspection. We follow their arguments a step further, and show the existence of an optimal policy $\mathcal{A}^*$ with a simple description (Theorem 2.3): $\mathcal{A}^*$ commits to a subset $T^*$ of boxes, an ordering $\sigma$ on $T^*$, and a threshold $V_i$ for each box $i \in T^*$; $\mathcal{A}^*$ opens the boxes in $T^*$ in the ordering $\sigma$, until either (a) a box $i$ yields a value at least $V_i$, at which point $\mathcal{A}^*$ switches to running the index-based policy on the rest of the boxes (taking the best value seen so far as a free outside option), or (b) none of the first $|T^*| - 1$ boxes yield values passing their thresholds, at which point $\mathcal{A}^*$ takes the last box in $T^*$ without inspection. As special cases, if $T^* = \emptyset$, $\mathcal{A}^*$ is Weitzman's index policy; if $|T^*| = 1$, $\mathcal{A}^*$ takes the box in $T^*$ without inspection. Our strengthened structure theorem shows the existence of a succinctly representable optimal policy, and implies the decision version of PNOI to be in NP. To take another interesting perspective, Doval [8] observed that, in an optimal policy for a PNOI instance, the outcome from probing a box may influence the order in which the other boxes are probed. Our structure theorem shows that, such order switching follows a simple structure: only during the stage of probing boxes in $T^*$ can the order of future queries change (from $\sigma$), and that change is triggered only when a value high enough is revealed, at which point the policy switches to the index policy.

To show hardness, we study a family of PNOI instances where it is easy to determine $T^*$ and the thresholds $V_1, \cdots, V_n$ in the above description of the optimal policy, so that the hardness is solely from deciding the ordering $\sigma$. In these instances, each value distribution $F_i$ is supported on $\{0, \frac{1}{2}, 1\}$, with expectation less than $\frac{1}{2}$, and index (in Weitzman's sense) at least $\frac{1}{2}$. It can be shown that for such instances an optimal policy must be of the form stated in the structure theorem, with $T^*$ being the set of all boxes, and $V_i = \frac{1}{2}$ for each $i$.

The possibility of switching to the index policy after each inspection adds difficulty to calculating the policy's expected payoff, but this calculation is necessary for a reduction. Key to our analysis is to observe that a closely related policy has the *non-exposed* property, a property that was first crystallized by Kleinberg, Waggoner, and Weyl (2016) and has been instrumental in several works in optimal search (e.g., [18]; [11]; [4]). This property allows us to derive a relatively clean expression for the *difference* between a policy's expected payoff and that of Weitzman's index-based policy (Lemma 3.6). Computing an optimal policy boils down to finding an ordering $\sigma$ that maximizes this difference.

Finally, we give a fairly technical reduction from the classical Partition problem: given a set $S$ of $n$ positive integers, decide whether they can be partitioned into two subsets with equal sums. We embed the

$n$ integers in the parameters of $n$ boxes, and add two auxiliary boxes, $B_{n+1}$ and $B_{n+2}$. Box $B_{n+2}$ has both high index and high cost, so designed that $B_{n+2}$ is the unique box possibly selected without inspection, but is the first to be inspected if a value $\frac{1}{2}$ is found. This creates an exquisite balance between, on one hand, the saving in cost when a high-cost box is selected without inspection, and, on the other, the motive to inspect a high-index box early on. The time point at which to switch to the index-based policy is affected by the position of $B_{n+1}$. We are able to set the parameters so that the most balanced partition of $S$ is realized in the ordering $\sigma$ of the optimal policy: the boxes before $B_{n+1}$ and those after form the partition. The reduction is fairly involved technically due to the need for various approximations — the expected payoff even for such simple instances of PNOI is still complex, and takes a fair amount of massaging to have terms bearing resemblance to the sums in the Partition problem.

**Polynomial Time Approximation Scheme (PTAS)** Our PTAS is built on a framework by Fu et al. [9] that gives PTAS for a broad class of stochastic sequential optimization problems. The main idea of the framework, to put it very roughly, is to start by considering systems with only an $O(1)$ number of possible states. For such systems, one can show that, in the decision tree of a policy, nodes may be grouped into a small number of blocks — within each block, the system's state remains the same and the ordering of the actions matters little for the eventual objective. One may therefore use dynamic programming to exhaustively optimize over decision trees consisting of such blocks, with a loss of only a small fraction of the payoff. A natural way to cast PNOI in this framework is to let the state of the system be the highest value revealed so far. Further manipulations enable us to inherit the main theorem of Fu et al. and to obtain a PTAS for PNOI when there are only $O(1)$ possible values (Proposition 4.2).

To generalize from these restricted instances, it is natural to consider discretizing the values before applying the framework. For $\epsilon > 0$, standard discretization can reduce the support size of value distributions to $\mathrm{poly}(1/\epsilon)$ and lose $O(\epsilon)$ fraction of the payoff *if all values are within* $\mathrm{poly}(1/\epsilon)$ *factor of the optimal expected payoff*. Let's call a value *large* if it is at least $\frac{1}{\epsilon}$ times the optimal payoff. If one simply ignores large values, a sizable fraction of the payoff may be lost. A major technical contribution of ours is a separate method to discretize large values. We observe that, once a large value is found, the expected additional payoff is upper-bounded and approximated by a well-behaved additive function. We use this function to define $O(1/\epsilon)$ discretization points to which we round up the large values. These discretization points are potentially far apart, and we cannot round the values in the usual way, as that again may introduce too much error. Crucially, we only round up values yielded by opened boxes, i.e., overestimate what one currently has, but do not round up values of boxes to be opened. In other words, in the discretized problem, when a box yields a large value $v$, our payoff is $v$ minus the current highest value (if $v$ is larger) and the search cost, but then pretends from this point on that the highest value is the discretized $v$, i.e., the smallest discretization point larger than $v$. We show that, this non-standard discretization controls the error introduced in the payoff, and still supports optimization within Fu et al.'s framework.

## 1.1 Additional Related Works

In the context (and disguise) of channel probing in wireless systems, Guha et al. [14] gave a 0.8-approximation algorithm for PNOI, based on a structure theorem they showed for optimal policies. Since the work predated the rediscovery of the Pandora Box problem in the computer science community, and was not presented as part of this line of work, it has remained little known to this community. Doval [8] revived the problem in the economics literature, and observed the complex behaviors of optimal policies for PNOI. Beyhaghi and Kleinberg [4] reintroduced the algorithmic question to the econ-CS literature, and drew a connection to the adaptivity gap in stochastic submodular function maximization, which yields a simple $(1 - 1/e)$-approximation algorithm. Through personal communications, we learned that Beyhaghi and Cai [3] independently obtained a PTAS for PNOI; they also strengthened Guha et al.'s structure theorem to a

version equivalent to our Theorem 2.3.

Algorithms for natural variants of the Pandora box problem have received much attention lately. To name a few examples, Boodaghians et al. [5] studied the optimal search problem when there are constraints on the order in which the boxes may be inspected; Chawla et al. [6] studied the case when values in the boxes are correlated; Fu et al. [9] and Segev and Singla [17] used *Pandora box problem with commitment* as an application for their frameworks of designing PTAS and EPTAS, respectively, for stochastic combinatorial optimization problems. Singla [18] generalized the optimal search problem to settings known as *Price of Information*, where the set of options that may be selected is governed by combinatorial feasibility systems; Gamlath et al. [11] and Fu et al. [10] studied such settings when the feasibility systems are given by matchings in graphs.

Hardness for computing online optimal policies in Bayesian selection problems is relatively sparse in the literature, but has been gaining attention recently. Agrawal et al. [1] showed NP-hardness for choosing the optimal ordering in an online stopping problem. Papadimitriou et al. [16] showed PSPACE-hardness for the online stochastic bipartite matching problem.

## 2 Preliminaries

In an instance of the classical Pandora box problem, we are given $n$ sealed boxes, each box $i$ labeled with a search cost $c_i \geq 0$ and a distribution $F_i$. Box $i$ contains a value $v_i \geq 0$, initially hidden, drawn independently from $F_i$, and one may open the box at cost $c_i$ to reveal $v_i$. At any point, a policy may (adaptively) choose to open a sealed box, or to take the highest value revealed so far and quit. Upon quitting, the payoff is the value taken minus the costs incurred along the way. Our goal is to maximize the expected payoff, where the expectation is over the values and the possible randomness in the policy. In a problem of *Pandora box with non-obligatory inspection* (PNOI), it is allowed to take a box that has not been opened, in which case the payoff is the unseen value in that box minus the costs incurred before taking the box.

**Weitzman's Index-based Policy.** For box $i$, define its *index* $\tau_i$ to be the unique solution to the equation $\mathbf{E}_{v_i \sim F_i}[(v_i - \tau_i)_+] = c_i$, where $(v_i - \tau_i)_+$ denotes $\max\{0, v_i - \tau_i\}$. Let $\kappa_i$ be $\min\{v_i, \tau_i\}$.

Weitzman's index-based policy first writes on each box its index, then at each stage, if the largest positive number written on the boxes is an index, the policy opens that box and writes the value revealed in place of its index; or else the largest written number is a value, and the policy takes the box with that value and terminates.[1]

Kleinberg et al. [15] gave a new proof for the optimality of the index-based policy for the classical Pandora box problem. The proof has proved powerful and inspired multiple algorithmic works [e.g. 18, 11, 4]. Part of its power is to isolate the so-called *non-exposed* property (Definition 2.2), which we use in our reduction in Section 3. For completeness, we give the proof in Appendix A.

**Theorem 2.1.** *[19, 15] The index-based policy maximizes the expected payoff in the classical Pandora box problem. Its expected payoff is* $\mathbf{E}[\max_{i \in [n]} \kappa_i]$.

**Definition 2.2.** *A policy is non-exposed if, when it opens a box $i$ and finds $v_i > \tau_i$, it is guaranteed to take box $i$. More formally, a policy is non-exposed if $\mathbf{Pr}[(I_i - A_i)(v_i - \tau_i)_+] = 0$ with probability 1, for each $i$.*

**Decision Trees.** In various arguments, it is convenient to analyze policies as decision trees. A deterministic policy $\mathcal{A}$ on an instance $B$ of PNOI is fully described by a decision tree $T$. At each node $u$, $\mathcal{A}$ chooses an *action* $a_u$, which may be opening a box, taking a box without opening it, or taking a maximum

[1] In the degenerate case where all indices are negative to begin with, the policy does nothing and quits, getting a payoff of 0.

value seen so far. The latter two categories of actions lead to a leaf signifying the end of the process; upon opening a box, the system transits probabilistically to another node depending on the value revealed.

We use $\mathbb{P}(\mathcal{A}, B)$ to denote the expected profit of policy $\mathcal{A}$ on a PNOI instance $B$. When the instance is clear from the context, we omit the second argument and write $\mathbb{P}(\mathcal{A})$. For a given instance of PNOI, we let OPT be the maximum expected profit achievable by any policy.

**Strengthened Structure Theorem** Key to Guha et al. [14]'s approximation algorithm is their structure theorem, which states the existence of an optimal policy for which there is a unique box possibly taken without inspection. We strengthen the theorem and show the existence of a well-structured, succinctly describable optimal policy. Our proof largely follows the arguments in [14]. The theorem immediately shows the decision version of PNOI to be in NP. Our main technical results do not rely on this structure theorem, although the optimal policy's structure, which features an ordering of boxes, sheds some light on our reduction in Section 3.

**Theorem 2.3.** *For any PNOI instance, there is an optimal policy $\mathcal{A}$ described by a subset of boxes $T^*$, an ordering $\sigma$ on $T^*$, and a threshold value $V_i$ for each box $i \in T^*$ (except the last one according to $\sigma$). If $T^* = \emptyset$, $\mathcal{A}$ is the index-based policy. Otherwise $\mathcal{A}$ opens the boxes in $T^*$ according to the ordering $\sigma$ until either it sees a value at least $V_i$ from a box $i$, or only one box remains unopened in $T^*$. Once it sees a value $v_i \geq V_i$ from box $i \in T^*$, $\mathcal{A}$ switches to running an index-based policy on the remaining boxes, taking the highest value seen so far as a free outside option; if this does not happen till only one box in $T^*$ remains unopened, $\mathcal{A}$ takes that box without inspection.*

**Corollary 2.4.** *The following decision version of PNOI is in* NP*: given a PNOI instance and $P > 0$, decide whether there is a policy with an expected payoff at least $P$.*

# 3 Hardness of PNOI

In this section, we show that computing optimal policies for PNOI is NP-hard. Our reduction makes use of the following class of PNOI instances.

**Definition 3.1.** *An instance of PNOI is a* low-cost-low-return-support-3 (LCLRS3) *instance if the following conditions hold:*

1. *each value distribution $F_i$ is supported on $\{0, \frac{1}{2}, 1\}$, with probability masses $p_i \coloneqq \mathbf{Pr}[v_i = 1] > 0$, $q_i \coloneqq \mathbf{Pr}[v_i = \frac{1}{2}]$, $r_i \coloneqq 1 - p_i - q_i = \mathbf{Pr}[v_i = 0]$;*
2. *for each box $i$, the cost $c_i > 0$, and the expected value $\mathbf{E}[v_i] = p_i + \frac{q_i}{2} < \frac{1}{2}$;*
3. *for each box $i$, the index $\tau_i \geq \frac{1}{2}$, which implies $\tau_i = 1 - \frac{c_i}{p_i}$.*

In Section 3.1 we show that, optimal policies for LCLRS3 instances are particularly simple — in the language of Theorem 2.3, only the ordering $\sigma$ remains to be determined. In Section 3.2 we reduce the partition problem to computing optimal policies for LCLRS3 instances.

**Theorem 3.2.** *It is* NP*-hard to compute optimal policies for LCLRS3 instances of PNOI.*

Before proving the theorem, we quickly remark that the value $\frac{1}{2}$ in the support is necessary for a hardness result. This was already observed by Guha et al. [14]. We provide a proof for completeness in Appendix B.

**Proposition 3.3.** *[14] There is a polynomial-time computable optimal policy for PNOI instances where all value distributions are supported on $\{0, 1\}$.*

### 3.1 Normal Policies and Their Payoffs

In this section, we show that optimal policies for LCLRS3 instances are of a format that we call *normal* (Definition 3.4). We then make use of ideas from Kleinberg et al. [15]'s proof for the index-based policy, and derive an expression for normal policies' payoff (Lemma 3.6), which plays a crucial role in the reduction we present in Section 3.2.

**Definition 3.4.** *A policy $\mathcal{A}$ for a LCLRS3 instance is said to be* normal *if*

- *If $\mathcal{A}$ sees 0 in the first $n-1$ boxes it opens, then $\mathcal{A}$ takes the last box without inspection; this is the only situation in which $\mathcal{A}$ exercises the option to bypass inspection.*
- *Whenever $\mathcal{A}$ opens a box and sees value $1$ in it, it immediately takes the box and stops.*
- *Whenever $\mathcal{A}$ opens a box and sees value $\frac{1}{2}$ in it, it forsakes the option to take a box without inspection; on the remaining boxes, $\mathcal{A}$ runs the index-based policy, with an outside option of value $\frac{1}{2}$.*

In the language of Theorem 2.3, a policy is normal if $T^*$ is the set of all boxes, and $V_i = \frac{1}{2}$ for each box $i$. The following lemma is straightforward; we relegate its proof to Appendix B.

**Lemma 3.5.** *For any LCLRS3 instance, an optimal policy $\mathcal{A}$ is normal.*

Since both the set $T^*$ and the thresholds $V_i$'s for a normal policy are fixed, finding an optimal policy for an LCLRS3 instance amounts to finding an optimal permutation $\sigma$. For a given LCLRS3 instance and a policy $\mathcal{A}$ on it, recall that we denote the expected payoff of $\mathcal{A}$ as $\mathbb{P}(\mathcal{A})$. The next observation is that, for a normal policy $\mathcal{A}$, the difference between $\mathbb{P}(\mathcal{A})$ and the payoff of the classical index-based policy admits a relatively clean expression in terms ofthe ordering $\sigma$ of $\mathcal{A}$. This is by considering an intermediate policy $\mathcal{A}'$, whose payoff admits simplifications using ideas from Kleinberg et al. [15]'s proof (Theorem 2.1).

As in the proof of Theorem 2.1, define $\kappa_i \coloneqq \min\{v_i, \tau_i\}$.

**Lemma 3.6.** *Given an LCLRS3 instance and a normal policy $\mathcal{A}$ for it, let $\sigma$ be the corresponding ordering. For box $i$, let $T_\sigma(i)$ be the set of boxes ordered after box $i$ by $\sigma$, with Gittins indices strictly larger than $\tau_i$; that is, $T_\sigma(i) \coloneqq \{j \in [n] : \sigma^{-1}(j) > \sigma^{-1}(i), \tau_j > \tau_i\}$. For $i \in [n]$ and $T \subseteq [n]$, define $g(i,T) \coloneqq \mathbf{E}[(\max_{j\in T} \kappa_j - \tau_i)_+]$.[2] Let $\mathcal{A}_P$ be the index-based policy on the instance. Then*

$$\mathbb{P}(\mathcal{A}_P) = \mathbb{P}(\mathcal{A}) + \sum_i p_i g(i, T_\sigma(i)) \prod_{j=1}^{\sigma^{-1}(i)-1} r_{\sigma(j)} - c_{\sigma(n)} \prod_{i=1}^{n-1} r_{\sigma(i)}. \tag{1}$$

The proof can be found in Appendix B. Here we briefly explain the terms in (1). Recall that $\mathbb{P}(\mathcal{A}_P) = \mathbf{E}[\max_{j\in[n]} \kappa_j]$. For each box $i$, with probability $p_i \prod_{j=1}^{\sigma^{-1}(i)-1} r_{\sigma(j)}$, the boxes before $i$ all take value 0, and box $i$ takes value 1, with $\kappa_i = \tau_i$. Conditioning on this event, the expected gap between $\max_{j\in[n]} \kappa_j$ and $\kappa_i$ is $g(i, T_\sigma(i))$. The advantage of the normal policy $\mathcal{A}$ is that it can save the cost $c_{\sigma(n)}$ to open the last box with probability $\prod_{i=1}^{n-1} r_{\sigma(i)}$, and this leads to the last term $-c_{\sigma(n)} \prod_{i=1}^{n-1} r_{\sigma(i)}$.

### 3.2 Reduction

In this section we give a polynomial-time reduction from the classical partition problem to PNOI.

**Definition 3.7** (Partition Problem)**.** *Given a multiset $S$ of positive integers $s_1, \cdots, s_n$, decide whether $S$ can be partitioned into two subsets $S_1$ and $S_2$ such that the sum of the numbers in $S_1$ equals the sum of the numbers in $S_2$.*

[2] $g(i, \emptyset) \coloneqq 0$.

It is well-known that Partition problem is NP-complete [see, e.g. 12]. It is also not difficult to show that the problem is still NP-hard when $1 \le s_1 \le \cdots \le s_n \le 2^n$. We assume so in the following reduction. We first formally give the reduction, and explain the intuition below.

**Reduction from LCLRS3 to Partition.** Given the multiset $S = \{s_1, \ldots, s_n\}$ of integers between 1 and $2^n$, fix two constants $\Gamma = 2^{8n}$ and $\Delta = 2^{-7n}$. We construct an LCLRS3 instance with $n+2$ boxes, denoted as $B_1, \cdots, B_{n+1}, B_{n+2}$.

For box $B_{n+1}$, set $p_{n+1} = 1/\Gamma, q_{n+1} = 1 - 41/\Gamma$ and $c_{n+1} = p_{n+1}/2$. This makes $\tau_L := \tau_{n+1} = \frac{1}{2}$ and $r_{n+1} = 40/\Gamma$.

For box $B_{n+2}$, set $p_{n+2} = q_{n+2} = 1/8$, $c_{n+2} = 1/32$ and thus $\tau_{n+2} = 3/4$.

For each $i \in [n]$, set $p_i = q_i = s_i/\Gamma$. Set a constant $\tau_H = 3/4 - O(\Delta)$, whose precise value is to be determined later (see Claim 3.13). Set $c_i$ to make

$$\tau_i = \tau_H + \frac{p_i p_{n+1}(1-p_{n+2})(\tau_H - \tau_L)}{2p_{n+2}} = \tau_H + O(\Delta^2).$$

Note that $p_i \le \Delta$ for any $i \in [n+1]$, since we assumed $s_i \le 2^n$. The construction ensures $\tau_{n+2} > \tau_i > \tau_H > \tau_L = 1/2$ for any $i \in [n]$.

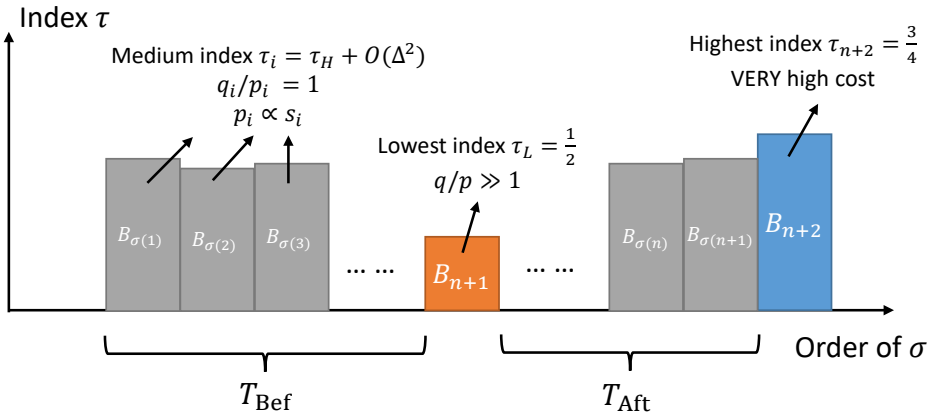


Figure 1: An overview of our reduction. Rectangles represent boxes, shown from left to right in the order of $\sigma$. The height of each rectangle represents the box's index.

Recall that the optimal solution to any LCLRS3 instance can be represented by a permutation $\sigma$. Given the permutation $\sigma$, the position of $B_{n+1}$ in the permutation plays a crucial role in the following analysis. Let the position of $B_{n+1}$ in $\sigma$ be $\xi$, i.e. $\sigma(\xi) = n+1$. Thus $B_1, \cdots, B_n$ are partitioned into two sets in $\sigma$: those before $B_{n+1}$ and those after, which we denote by $T_{\text{Bef}}$ and $T_{\text{Aft}}$, respectively. Formally, $T_{\text{Bef}} := \{i : 1 \le i \le n, \sigma^{-1}(i) < \xi\}$ and $T_{\text{Aft}} := \{i : 1 \le i \le n, \sigma^{-1}(i) > \xi\}$. The next key lemma builds the bridge between the partition problem and LCLRS3 instances:

**Lemma 3.8.** *The answer to the Partition problem with input $S$ is YES if and only if $\sum_{i \in T^*_{\text{Aft}}} p_i = \sum_{i \in T^*_{\text{Bef}}} p_i$ in the permutation $\sigma^*$ that corresponds to an optimal policy for the LCLRS3 instance.*

Theorem 3.2 follows immediately from Lemma 3.8 and the fact that Partition problem is NP-hard. It remains to prove Lemma 3.8.

**Intuition of the Reduction and Proof Overview.** By Lemma 3.5 and Lemma 3.6, giving an optimal policy for the LCLRS3 instance boils down to finding a permutation $\sigma$ that maximizes the objective value

$$\text{Utility}(\sigma) := c_{\sigma(n+2)} \prod_{i=1}^{n+1} r_{\sigma(i)} - \sum_i p_i g(i, T_\sigma(i)) \prod_{j=1}^{\sigma^{-1}(i)-1} r_{\sigma(j)}. \tag{2}$$

We would like to focus on the more complex second term (which we call the loss term (3)). The role of box $n+2$ is to fix the first term: $c_{n+2}$ is a constant whereas $c_1, \ldots, c_{n+1}$ are exponentially small. $c_{n+2}$ is so large compared with all other terms in (2) that any reasonable policy must leave box $n+2$ till the end:

**Claim 3.9.** *Let $\sigma^*$ be a permutation which maximizes* (2)*. Then $\sigma^*(n+2) = n+2$.*

From the discussion above, an optimal policy must leave box $n+2$ to the last in its permutation $\sigma$, and the ordering of the other boxes must minimize the loss term:

$$\text{Loss}(\sigma) \coloneqq \sum_i p_i g(i, T_\sigma(i)) \prod_{j=1}^{\sigma^{-1}(i)-1} r_{\sigma(j)}. \tag{3}$$

There is a non-trivial trade-off for deciding the position of $B_{n+1}$. On the one hand, it can be shown that if all the $n+1$ boxes before $B_{n+2}$ have the same index $\tau$, the ones with higher ratios of $q_i/p_i$ should be opened early to maximize the utility, i.e., the box with a higher ratio of $q_i/p_i$ should be put earlier in the permutation. On the other hand, if the ratio $q_i/p_i$ is the same for each of these $n+1$ boxes, the box with a higher index should be opened earlier. The special box $B_{n+1}$ has a smaller index than boxes $B_1, \ldots, B_n$, but a much higher ratio of $q_{n+1}/p_{n+1}$. Finding the non-trivial trade-off in deciding the position of $B_{n+1}$ in $\sigma$ can be shown NP-hard.

### 3.3 Correctness of Reduction: A Sketch

We demonstrate all the statements and prove the key Lemma 3.8. The omitted proofs can be found in Appendix B.1. We show the key lemma by using a function with $\sum_{i \in T_{\text{Aft}}} p_i + p_i^2$ as the single variable to approximate Equation (3). For ease of notation, define

$$y \coloneqq \sum_{i \in S} \frac{s_i}{\Gamma} + \left(\frac{s_i}{\Gamma}\right)^2 = \sum_{i \in T_{\text{Aft}} \cup T_{\text{Bef}}} p_i + p_i^2, \quad x \coloneqq \sum_{i \in T_{\text{Bef}}} p_i + p_i^2.$$

Note that $y$ is fixed once $S$ is given, whereas $x$ is a function of $T_{\text{Bef}}$ and hence of $\sigma$.

**Lemma 3.10.** *The parameters of the instance can be set up so that*

$$h(x) - O(n^2\Delta^4) \le \frac{\text{Loss}(\sigma) - C \pm O(n^2\Delta^4)}{k_1} \le h(x) + O(n\Delta^3), \tag{4}$$

$$\frac{k_2}{k_1} = 2e^{y/2} \pm O(\Delta^2),$$

*where*

$$h(x) := e^{-2x}\left(1 - \frac{k_2}{k_1} e^{-y+x}\right), \tag{5}$$

*with $C, k_1, k_2$ as constants independent of $\sigma$:*

$$k_1 := -\frac{1}{2} p_{n+2}(\tau_{n+2} - \tau_H)(p_{n+1} + q_{n+1}) + p_{n+1}[(1 - p_{n+2})(\tau_H - \tau_L) + p_{n+2}(\tau_{n+2} - \tau_L)], \tag{6}$$

$$k_2 := p_{n+1}(1-p_{n+2})(\tau_H - \tau_L), \tag{7}$$

$$C := \frac{1}{2}p_{n+2}(\tau_{n+2} - \tau_H)\left(1 - \prod_{i\in[n+1]} r_i\right) + \frac{1}{2}k_2\sum_{i=1}^{n} p_i^2. \tag{8}$$

We give a road map for the proof once we have Lemma 3.10. The minimum value of $h(x)$ is taken at $x^* = y - \ln(2k_1/k_2)$. When $k_1/k_2$ is near $2e^{y/2}$, $x^*$ is close to $y/2$. Our goal is to have the most even partition of $S$ be the $T_{\text{Bef}}$ and $T_{\text{Aft}}$ of an optimal policy, which in turn should have $x$ as close to $y/2$ as possible. Even with the approximation given in Lemma 3.10, a few obstacles still stand in the way: $x$ is not $\sum_{i\in T_{\text{Bef}}} p_i$, nor is $y$ equal to $\sum_{i\in[n]} p_i$; both of them have second-order terms, which cause further distortion in the objective through the fact that $x$ and $y$ appear in the exponents in $h$. We overcome these difficulties by carefully controlling the order of errors throughout our calculation: $p_i$'s are so small that the second-order terms in $x$ and $y$ are negligible; analytical properties of $h$ (Claim 3.16) guarantee that, around its optimum, $h$ is sensitive enough to perturbations, so that suboptimal solutions can be told from the optimal.

Much of the proof of Lemma 3.10, which is fairly technical, is relegated to Appendix B. We mention a tool instrumental in simplifying the calculations, which also explains our setting $p_i = q_i$ for all $i \in [n]$:

**Lemma 3.11.** *Given two sequences of positive real numbers $p_1, p_2, \cdots, p_n$ and $r_0, r_1, \cdots, r_n$. Let $r_0 = 1$. If there exists a constant $c > 0$ such that $p_i/(1-r_i) = c$ for each $1 \le i \le n$, then we have*

$$\sum_{i=1}^{n} p_i \prod_{j=0}^{i-1} r_j = c\left(1 - \prod_{i=1}^{n} r_i\right).$$

After much simplification, the main terms of $\text{Loss}(\sigma)$ are given in Claim 3.12, before we apply analytical tools and turn products to sums in the exponent (Fact 3.14, Claim 3.15), which leads to Lemma 3.10. Note that we have to appeal to second-order approximations of the exponential function for the required precision in the proof. The setup of the parameter $\tau_H$ is given in Claim 3.13.

**Claim 3.12.** *For a non-empty set $T \subseteq [n]$, let $f(T) := \prod_{i\in T} r_i = \prod_{i\in T}(1-2p_i)$ and $g(T) := \prod_{i\in T}(1-p_i)$. Also let $f(\emptyset) = g(\emptyset) = 1$. Then*

$$\text{Loss}(\sigma) = k_1 f(T_{\text{Bef}}) - k_2 f(T_{\text{Bef}})g(T_{\text{Aft}}) - k_2\sum_{i\in T_{\text{Aft}}} p_i^2/2 + C \pm O(n^2\Delta^4). \tag{9}$$

**Claim 3.13.** *If we choose $t$ so that $|t - 2e^{y/2}| \le O(\Delta^2)$, and set $\tau_H$ as follows, then $\frac{k_2}{k_1} = t$:*

$$\tau_H = \frac{-3t\Gamma + 28 + 94t}{-4t\Gamma + 56 + 104t}. \tag{10}$$

**Fact 3.14.** *For $0 \le x \le 1/2$, we have $1 - x \le e^{-x} \le 1 - x + x^2/2$, and $1 - x \le e^{-x-x^2/2} \le 1 - x + O(x^3)$.*

**Claim 3.15.** *For any subset $T$ of the first $n$ boxes, one has*

$$e^{-\sum_{i\in T} 2(p_i+p_i^2)} \ge f(T) \ge e^{-\sum_{i\in T} 2(p_i+p_i^2)} - O(n\Delta^3),$$
$$e^{-\sum_{i\in T}(p_i+p_i^2/2)} \ge g(T) \ge e^{-\sum_{i\in T}(p_i+p_i^2/2)} - O(n\Delta^3).$$

With the approximation in Lemma 3.10 in hand, we are almost ready to prove Lemma 3.8. The next lemma shows that the function $h$ is sensitive enough to perturbations around its minimum.

**Claim 3.16.** *If $|k_2/k_1 - 2e^{y/2}| \le O(\Delta^2)$, $\epsilon \in \mathbb{R}$ is such that $2^{-6n} \ge |\epsilon| \ge 1/\Gamma = 2^{-8n}$ , let $x^* \in [0, 1/2]$ be where $h(x)$ takes its minimum value, then $|x^* - \frac{y}{2}| \le O(\Delta^2)$,*

$$h(x^* + \epsilon) \ge h(x^*) + \epsilon^2/2.$$

*Proof of Lemma 3.8.* The "if" part is obvious: if the permutation $\sigma^*$ of a policy yields a partition $T^*_{\text{Bef}}$ and $T^*_{\text{Aft}}$ with $\sum_{i \in T^*_{\text{Bef}}} p_i = \sum_{i \in T^*_{\text{Aft}}} p_i$, then since $p_i = s_i/\Gamma$ for each $i \in [n]$, $(T^*_{\text{Bef}}, T^*_{\text{Aft}})$ certifies that $S$ is a YES instance of Partition.

For the "only if" part, suppose $S$ can be partitioned into disjoint subsets $S_1$ and $S_2$ with $\sum_{s \in S_1} s = \sum_{s \in S_2} s$, we show that any policy whose corresponding $T_{\text{Bef}}$ and $T_{\text{Aft}}$ is not an even partition of $S$ must be suboptimal. By our setting of parameters and Claim 3.13, we know $|x^* - \frac{y}{2}| \le O(\Delta^2)$. For any permutation $\sigma$ whose corresponding sets $T_{\text{Aft}}, T_{\text{Bef}}$ are such that $\sum_{i \in T_{\text{Aft}}} p_i \ne \sum_{i \in T_{\text{Bef}}} p_i$, we have $|x - y/2| \ge 1/\Gamma - \sum_{i \in T_{\text{Aft}} \cup T_{\text{Bef}}} p_i^2 \ge 1/\Gamma - n\Delta^2$, and hence $|x - x^*| \ge 1/\Gamma - n\Delta^2/2$. Hence, one has

$$\begin{aligned}\frac{\text{Loss}(\sigma) - C + O(n^2\Delta^4)}{k_1} &\ge h(x) - O(n^2\Delta^4) > h(x^*) + \Omega(1/\Gamma^2) \\ &\ge \frac{\text{Loss}(\sigma^*) - C + O(n^2\Delta^4)}{k_1} + \Omega(1/\Gamma^2),\end{aligned}$$

where the second inequality follows from Claim 3.16, the first and last inequality follow from Lemma 3.10. Hence $\text{Loss}(\sigma) > \text{Loss}(\sigma^*)$. □

# 4 Polynomial-Time Approximation Scheme

This section gives a polynomial time approximation scheme (PTAS) for PNOI. The PTAS is based on Fu et al. [9]'s framework which gives approximation schemes for a class of stochastic optimization problems with $O(1)$ sized state spaces. The natural instantiation of PNOI in this framework uses values as states, and must reduce their number to $O(1)$. Standard discretization, however, only works when all values are small. Our main technical contribution is a novel discretization method tailored for larger values (Section 4.2.2).

## 4.1 Fu et al.'s Framework and Its Application to PNOI

We first introduce Fu et al. [9]'s framework. We make some simplifications in this presentation, referring the reader to the original paper for full details. We then adapt the framework to PNOI and obtain a PTAS when the support of the value distributions is of $O(1)$ size; Section 4.2 deals with the general case.

**Fu et al.'s SSDP framework.** A *stochastic sequential decision process* (SSDP) is given by a 5-tuple $(V, A, f, G, I_0)$, where $V$ is the set of states of the process, $I_0 \in V$ the initial state, $A$ the set of actions, $f : V \times A \to V$ the stochastic state transition function, and $G : V \times A \to \mathbb{R}$ the marginal payoff function. In each step $t$, a policy chooses an action $a \in A$, with the restriction that each action can be taken at most once during the process; a policy may also choose to end the process at any time. Let $I_t$ denote the state at round $t$. Then when action $a_t \in A$ is taken at state $I_t \in V$ in round $t$, the state becomes $I_{t+1} = f(I_t, a_t)$, producing a marginal payoff of $G(I_t, a_t)$ whose expectation $\mathbf{E}[G(I_t, a_t)]$ is non-negative. Generally, both $f$ and $G$ are random, and may be correlated. In execution of the process, the total payoff is

the sum of the marginal payoffs incurred in all rounds; our goal is to design a policy that maximizes the expected total payoff. [3]

Given an SSDP, let OPT be the maximum expected payoff obtainable by any policy, and let MAX be the maximum expected payoff if a policy may (hypothetically) choose to start with any state $I'_0 \in V$, in place of $I_0$.

It is not difficult to see that, for any randomized policy $\mathcal{A}^R$, there is a deterministic policy whose expected payoff is no less than that of $\mathcal{A}^R$. Therefore we focus on deterministic policies.

**Theorem 4.1.** *[Essentially from [9]] If an SSDP problem $(V, A, f, G, I_0)$ satisfies the following conditions:*

1. *The number of possible states is a constant, i.e., $|V| = O(1)$.*
2. *The state space $V$ admits an ordering "$\geq$" such that $f(I, a) \geq I$ for any $I \in V, a \in A$, i.e., the state is non-decreasing with probability 1.*
3. *There exists an optimal policy that never takes an action with a negative expected marginal payoff in any round.*
4. $\mathrm{MAX} = O(\mathrm{OPT})$.

*Then, for any fixed $\epsilon > 0$, there is a policy $\mathcal{A}$ computable in time $n^{2^{O(\epsilon^{-3})}}$, with expected payoff at least $(1 - \epsilon) \cdot \mathrm{OPT}$.*

We sketch the main ideas behind Theorem 4.1 in Appendix C.1; readers interested in the details are referred to Fu et al. [9].

**PNOI as an SSDP.** We now present PNOI in the framework of SSDP and apply Theorem 4.1 to obtain a PTAS for PNOI when the value distributions have small supports.

**Proposition 4.2.** *If $|\bigcup_i \mathrm{supp}(F_i)| = O(1)$, then for any fixed $\epsilon > 0$, there is a polynomial-time algorithm that computes a policy with an expected payoff at least $(1 - \epsilon) \cdot \mathrm{OPT}$.*

We prove the proposition by casting the PNOI problem as an SSDP and then applying Theorem 4.1. In doing so, we must satisfy the conditions of Theorem 4.1.

Let $V$, the set of states, be $\bigcup_i \mathrm{supp}(F_i)$. The "$\geq$" ordering in condition 2. is simply the natural ordering on reals. Let MaxV denote the maximum value in $V$. The action space $A$ consists of three parts: $A_0 = \{a_1^0, \ldots, a_n^0\}$, where $a_i^0$ is the action of opening box $i$; $A_1 = \{a_1^1, \ldots, a_n^1\}$, where $a_i^1$ is the action of taking box $i$ without opening it (and ending the process); and $\mathsf{end}$, the action of taking the maximum value seen so far (and ending the process). Note that, for each $i$, at most one of the actions $a_i^0, a_i^1$ could be chosen throughout the process, and an action in $A_1$ precludes the action $\mathsf{end}$.[4] Set $I_0 = 0$ and let the state $I_t$ be the largest value seen in the boxes opened in the first $t$ rounds. For any $I \in V$ and $i \in [n]$, the state transition function $f$ is defined as

$$\begin{aligned} f(I, a_i^0) &= \max\{I, v_i\}, \\ f(I, a_i^1) &= \max\{I, \mathbf{E}[v_i]\}, \\ f(I, \mathsf{end}) &= I; \end{aligned}$$

[3] Fu et al.'s original framework allows the total payoff to also include a payoff that depends on the final state reached. Such final payoffs can be easily simulated by marginal payoffs, and our adoption of the framework for PNOI calls for no such final payoffs.

[4] In the application of Theorem 4.1, such constraints are easily taken care of by the dynamic programming. Similar constraints arose in *Committed ProbeTop-$k$ Problem* and *Committed Pandora Problem*, and were handled similarly by Fu et al. [9].

the marginal payoff function $G$ is

$$\begin{aligned} G(I, a_i^0) &= (\max\{I, v_i\} - I) - c_i, \\ G(I, a_i^1) &= \mathbf{E}[v_i] - I, \\ G(I, \mathsf{end}) &= 0. \end{aligned}$$

**Claim 4.3.** *$G(I, a)$ is non-increasing in $I$ for any $a$, and is Lipschitz in $I$, i.e., for any $I_1 < I_2$ and any action $a$, $G(I_2, a) - G(I_1, a) \le I_2 - I_1$.*

It is straightforward to see that, in any execution of a reasonable policy,[5] the sum of marginal payoffs from all rounds is exactly the payoff in the PNOI problem.

We now check the conditions of Theorem 4.1. The first two are satisfied immediately. For the last condition, note that, changing $I_0$ to any positive value only decreases the marginal payoffs (because $G$ is non-increasing in $I$), so $\mathrm{MAX} = \mathrm{OPT}$. Condition 3 is guaranteed by the lemma below.

**Lemma 4.4.** *An optimal PNOI policy never takes an action with a negative expected marginal payoff in any round.*

The argument is most readily seen using the language of decision trees and formally proved in Appendix C.1. We have thus shown that all four conditions in Theorem 4.1 are satisfied. Proposition 4.2 therefore follows.

## 4.2 Discretization and PTAS for General PNOI

The condition $|V| = O(1)$ is essential to Proposition 4.2. To obtain a PTAS for general PNOI instances, we look to discretize values to reduce the state space size. However, standard discretization turns out to work only for values small compared to OPT (Section 4.2.1). We develop in Section 4.2.2 a separate, novel technique to handle large values.

**Theorem 4.5.** *For any fixed constant $\epsilon > 0$, there is a polynomial-time algorithm for PNOI that computes a policy with an expected payoff at least $(1 - O(\epsilon)) \cdot \mathrm{OPT}$.*

Throughout this section, we fix a threshold $\theta \in [\mathrm{OPT}/\epsilon, 2\mathrm{OPT}/\epsilon]$, which can be obtained in polynomial time by running the simple approximation algorithm mentioned in the Introduction. *Small values* refer to values at most $\theta$, and *large values* are those above $\theta$. Proofs omitted in this section can be found in Appendix C.2.

### 4.2.1 Discretization of Small Values

Standard discretization techniques can handle small values. For fixed $\epsilon$ and $\theta$, define a discretization function $D^S : x \mapsto \lfloor \frac{x}{\theta \cdot \epsilon^2} \rfloor \cdot \theta \cdot \epsilon^2$. We say $D^S(v_i)$ is the *$S$-discretized* value of $v_i$, and a policy $\mathcal{A}^S$ is *$S$-discretized* if its decisions only depend on the $S$-discretized values $D^S(v_1), \ldots, D^S(v_n)$. Recall that $\mathbb{P}(\mathcal{A}, B)$ denotes the expected profit of a policy $\mathcal{A}$ on a PNOI instance $B$.

**Lemma 4.6.** *Let $B$ be an instance of PNOI and $B^S$ the $S$-discretized instance of $B$, in which each value $v_i$ is replaced by $D^S(v_i)$. Then,*

1. *for any policy $\mathcal{A}$, there is a $S$-discretized policy $\mathcal{A}^S$ such that*

$$\mathbb{P}(\mathcal{A}^S, B^S) \ge \mathbb{P}(\mathcal{A}, B) - O(\epsilon) \cdot \mathrm{OPT};$$

[5] Reasonable here means the policy never takes a box $i$ without inspection if $\mathbf{E}[v_i]$ is smaller than the current state $I$.

2. *for any $S$-discretized policy $\mathcal{A}^S$, $\mathbb{P}(\mathcal{A}^S, B) \geq \mathbb{P}(\mathcal{A}^S, B^S)$.*

Let MaxV be the largest value in all the supports of the value distributions. If MaxV $\leq \theta$, then the support of $D^S(v_1), \cdots, D^S(v_n)$ is $O(1)$. Combining Lemma 4.6 and Proposition 4.2 yields a PTAS for such instances. Nevertheless, when MaxV $\gg \theta$, the support size of *$S$-discretized* may be large, and we deal with this considerably more challenging case in Section 4.2.2.

### 4.2.2 Discretization of Large Values

Our discretization of large values is based on a few insights. First, any reasonable policy, after having seen a large value, should switch to the index policy. We observe that the additional expected payoff at this stage is upper bounded and approximated by an additive function (Lemma 4.8). We use this function to set $O(1/\epsilon)$ discretization points (Definition 4.9). Second, we find a non-standard way to make use of these discretization points. As we explain in more detail below, simply rounding values to the closest discretization points (as in Lemma 4.6) may introduce too much error.

**Lemma 4.7.** *If $\mathcal{A}^*$ is an optimal policy, after a probed box yields a value $v^* \geq \theta$, $\mathcal{A}^*$ only opens boxes with indices at least $v^* \geq \theta$. Moreover, with probability at most $\epsilon$, any box with index at least $\theta$ contains a value at least $\theta$.*

Let $W(S, v) \coloneqq \mathbf{E}[(\max_{i \in S} \kappa_i - v)_+]$ be the expected additional profit gained from the Weitzman's index-based policy on a set $S$ of unopened boxes, when the largest revealed value is $v$. Let $F(S, v)$ be $\sum_{i \in S} \mathbf{E}[(\kappa_i - v)_+]$. We show $F(S, v)$ upper bounds and approximates $W(S, v)$ for large $v$.

**Lemma 4.8.** *For any $S \subseteq [n]$ and $v \geq \theta$, we have $F(S, v) \geq W(S, v) \geq (1 - \epsilon) \cdot F(S, v)$.*

From Lemma 4.8, we know $F([n], \theta) < 2 \cdot \mathrm{OPT}$ when $\epsilon$ is small. We also have $F([n], \textsc{MaxV}) = 0$, and that $F([n], v)$ is non-increasing in $v$. The following discretization points can be found in polynomial time:

**Definition 4.9.** *Let $m = 2/\epsilon$. Let $\theta = \theta_1 < \theta_2 < \ldots < \theta_m = \textsc{MaxV}$ be such that for any $i < m$, $F([n], \theta_i) - F([n], \theta_{i+1}) < \epsilon \cdot \mathrm{OPT}$. The discretization function $D^L(x)$ is: for $x \in [\theta, \textsc{MaxV}]$, $D^L(x)$ is the smallest $\theta_i$ with $\theta_i \geq x$; for $x < \theta$, $D^L(x) = x$.*

Our choice of discretization points $\theta_1, \cdots, \theta_m$ is motivated by controlling the error in profit introduced when a *revealed, large* value $v$ is discretized to $D^L(v)$. No approximation is guaranteed if one discretizes all values, revealed and unrevealed. In particular, one may significantly overestimate the payoffs if one directly discretizes the value distribution using $\theta_1, \cdots, \theta_m$. To address this, we use raw values to calculate marginal payoffs and only discretize the revealed values. In the language of SSDP, we only discretize the state transition function, but not the marginal payoff function. A PTAS is made possible by the subtle fact that Theorem 4.1 only requires a small state space (and has no such requirement on the payoff function).

Now we formally give our discretization. A PNOI instance $B$ is discretized to an SSDP instance $B^L$. Compared with the SSDP instantiation we gave for PNOI in Section 4.1, $B^L$ modifies the state transition function by rounding up newly revealed values using $D^L(\cdot)$; however, given a current (rounded) state, the marginal payoff function of $B^L$ is given by the raw values. Formally:

1. For each state $I$ and action $a_i^0, a_i^1$, the state transition function $f^L$ in $B^L$ is

$$\begin{aligned} f^L(I, a_i^0) &= D^L(\max\{I, v_i\}); \\ f^L(I, a_i^1) &= \max\{I, \mathbf{E}[v_i]\}; \\ f^L(I, \mathsf{end}) &= I. \end{aligned}$$

2. For each state $I$ and action $a_i^0, a_i^1$, the marginal payoff function $G^L$ in $B^L$ is:

$$\begin{aligned} G^L(I, a_i^0) &= \max\{I, v_i\} - I - c_i; \\ G^L(I, a_i^1) &= \mathbf{E}[v_i] - I, \\ G^L(I, \mathsf{end}) &= 0. \end{aligned}$$

**Claim 4.10.** *Any policy $\mathcal{A}^L$ for the L-PNOI instance $B^L$ can be executed on the PNOI instance $B$, with an expected payoff no less than that of the execution on $B^L$.*

**Definition 4.11.** *In the L-PNOI instance $B^L$, when its state $I^L$ is at least $\theta$ and the set of unopened boxes is $S$, a policy is a* quasi-index policy *if it exhaustively opens (in any order) all boxes $i \in S$ with $\mathbf{E}[G^L(I^L, a_i^0)] > 0$, but terminates once its state transits to a higher one.*

**Lemma 4.12.** *For any $v \in \{\theta_1, \ldots, \theta_m\}$, $S \subseteq [n]$, let $W^L(S, v)$ be the expected marginal payoff of any quasi-index policy for $B^L$. We have $W^L(S, v) \geq (1 - \epsilon) \cdot F(S, v)$.*

**Lemma 4.13.** *Let $\mathrm{OPT}$ and $\mathrm{OPT}^L$ be the optimal expected payoff of the PNOI instance $B$ and the L-PNOI instance $B^L$, respectively. Then $\mathrm{OPT} \geq \mathrm{OPT}^L \geq (1 - O(\epsilon)) \cdot \mathrm{OPT}$.*

Finally, given any PNOI instance, one may first discretize its small values, followed by a discretization of the large values, and then apply Theorem 4.1. Theorem 4.5 is thus proved. The formal argument is given in Appendix C.

# 5 Conclusion and Open Problems

In this work, we proved the first computational hardness result for PNOI and improved the state-of-the-art approximation algorithm by a PTAS. There are other online decision problems where hardness results are missing, and approximation algorithms have been developed in the absence of a tractable optimal algorithm. The price of information setting for bipartite matching is one such example [18, 11].

Our PTAS yields policies for PNOI with arbitrarily good approximations, but its running time has an exponential dependence on $\epsilon$ which is inherited from Fu et al.'s framework. It is an attractive question whether there is an FPTAS for the problem.

# References


[1] Shipra Agrawal, Jay Sethuraman, and Xingyu Zhang. On optimal ordering in the optimal stopping problem. In Péter Biró, Jason D. Hartline, Michael Ostrovsky, and Ariel D. Procaccia, editors, *EC '20: The 21st ACM Conference on Economics and Computation, Virtual Event, Hungary, July 13-17, 2020*, pages 187–188. ACM, 2020.

[2] Mark Armstrong. Ordered Consumer Search. *Journal of the European Economic Association*, 15(5): 989–1024, 06 2017.

[3] Hedyeh Beyhaghi and Linda Cai. private communication, 2022.

[4] Hedyeh Beyhaghi and Robert Kleinberg. Pandora's problem with nonobligatory inspection. In Anna Karlin, Nicole Immorlica, and Ramesh Johari, editors, *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, pages 131–132. ACM, 2019. doi: 10.1145/3328526.3329626. URL https://doi.org/10.1145/3328526.3329626.

[5] Shant Boodaghians, Federico Fusco, Philip Lazos, and Stefano Leonardi. Pandora's box problem with order constraints. In Péter Biró, Jason D. Hartline, Michael Ostrovsky, and Ariel D. Procaccia, editors, *EC '20: The 21st ACM Conference on Economics and Computation, Virtual Event, Hungary, July 13-17, 2020*, pages 439–458. ACM, 2020.

[6] Shuchi Chawla, Evangelia Gergatsouli, Yifeng Teng, Christos Tzamos, and Ruimin Zhang. Pandora's box with correlations: Learning and approximation. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1214–1225. IEEE, 2020.

[7] Mahsa Derakhshan, Negin Golrezaei, Vahideh H. Manshadi, and Vahab S. Mirrokni. Product ranking on online platforms. In Péter Biró, Jason D. Hartline, Michael Ostrovsky, and Ariel D. Procaccia, editors, *EC '20: The 21st ACM Conference on Economics and Computation, Virtual Event, Hungary, July 13-17, 2020*, page 459. ACM, 2020.

[8] Laura Doval. Whether or not to open pandora's box. *J. Econ. Theory*, 175:127–158, 2018.

[9] Hao Fu, Jian Li, and Pan Xu. A PTAS for a class of stochastic dynamic programs. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 56:1–56:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi: 10.4230/LIPIcs.ICALP.2018.56. URL https://doi.org/10.4230/LIPIcs.ICALP.2018.56.

[10] Hu Fu, Zhihao Gavin Tang, Hongxun Wu, Jinzhao Wu, and Qianfan Zhang. Random order vertex arrival contention resolution schemes for matching, with applications. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 68:1–68:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[11] Buddhima Gamlath, Sagar Kale, and Ola Svensson. Beating greedy for stochastic bipartite matching. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2841–2854. SIAM, 2019.

[12] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences).* W. H. Freeman, 1979.

[13] J. C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society, Series B*, pages 148–177, 1979.

[14] Sudipto Guha, Kamesh Munagala, and Saswati Sarkar. Information acquisition and exploitation in multichannel wireless networks. *CoRR*, abs/0804.1724, 2008. URL http://arxiv.org/abs/0804.1724.

[15] Robert D. Kleinberg, Bo Waggoner, and E. Glen Weyl. Descending price optimally coordinates search. In Vincent Conitzer, Dirk Bergemann, and Yiling Chen, editors, *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016*, pages 23–24. ACM, 2016.

[16] Christos H. Papadimitriou, Tristan Pollner, Amin Saberi, and David Wajc. Online stochastic max-weight bipartite matching: Beyond prophet inequalities. In Péter Biró, Shuchi Chawla, and Federico Echenique, editors, *EC '21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021*, pages 763–764. ACM, 2021.

[17] Danny Segev and Sahil Singla. Efficient approximation schemes for stochastic probing and prophet problems. In Péter Biró, Shuchi Chawla, and Federico Echenique, editors, *EC '21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021*, pages 793–794. ACM, 2021.

[18] Sahil Singla. The price of information in combinatorial optimization. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2523–2532. SIAM, 2018.

[19] Martin L Weitzman. Optimal search for the best alternative. *Econometrica: Journal of the Econometric Society*, pages 641–654, 1979.

# A Omitted Proofs from Section 2

**Theorem 2.1.** *[19, 15] The index-based policy maximizes the expected payoff in the classical Pandora box problem. Its expected payoff is* $\mathbf{E}[\max_{i\in[n]} \kappa_i]$.

*Proof.* Consider any policy for an instance of the classical Pandora box problem. We first derive an upper bound on the policy's expected payoff, and then show that the index-based policy achieves the upper bound.

For each $i$, let random variables $I_i$ and $A_i$ be the indicator variables for the events that the policy opens box $i$ and takes box $i$, respectively. As the policy is not allowed to take a sealed box, we have $A_i \le I_i$ with probability 1. The policy's expected payoff is $\mathbf{E}[\sum_i (A_i v_i - I_i c_i)]$.

Note that the policy's decision to open box $i$ is independent from $v_i$, therefore $I_i$ and $v_i$ are independent. This allows us to rewrite the expected payoff using the definition of the indices:

$$\begin{aligned}\mathbf{E}\left[\sum_i (A_i v_i - I_i c_i)\right] &= \mathbf{E}\left[\sum_i [A_i v_i - I_i (v_i - \tau_i)_+]\right] \\ &\le \mathbf{E}\left[\sum_i A_i [v_i - (v_i - \tau_i)_+]\right] = \mathbf{E}\left[\sum_i A_i \kappa_i\right] \le \mathbf{E}\left[\max_i \kappa_i\right].\end{aligned}$$

where in the first inequality we used $A_i \le I_i$, in the ensuing equality we used the definition $\kappa_i = \min\{v_i, \tau_i\}$, and the last inequality follows from the constraint that $\sum_i A_i \le 1$ with probability 1.

Let us see that the index-based policy's payoff is precisely this upper bound. Consider properties of a policy that would turn the two inequalities in the chain into equalities:

1. $I_i(v_i - \tau_i)_+ = A_i(v_i - \tau_i)_+$ with probability 1 if, whenever the policy opens a box $i$ and finds $v_i > \tau_i$, the policy takes box $i$;
2. $\sum_i A_i \kappa_i = \max_i \kappa_i$ with probability 1 if the policy always takes the box with the maximum $\kappa_i$.

It is straightforward to verify that the index-based policy have both properties, and hence attains a payoff equal to the upper bound $\mathbf{E}[\max_i \kappa_i]$. The index-based policy therefore achieves maximum payoff among all policies. □

## A.1 Proof for the Structure Theorem 2.3

**Theorem 2.3.** *For any PNOI instance, there is an optimal policy $\mathcal{A}$ described by a subset of boxes $T^*$, an ordering $\sigma$ on $T^*$, and a threshold value $V_i$ for each box $i \in T^*$ (except the last one according to $\sigma$). If $T^* = \emptyset$,*

*$\mathcal{A}$ is the index-based policy. Otherwise $\mathcal{A}$ opens the boxes in $T^*$ according to the ordering $\sigma$ until either it sees a value at least $V_i$ from a box $i$, or only one box remains unopened in $T^*$. Once it sees a value $v_i \geq V_i$ from box $i \in T^*$, $\mathcal{A}$ switches to running an index-based policy on the remaining boxes, taking the highest value seen so far as a free outside option; if this does not happen till only one box in $T^*$ remains unopened, $\mathcal{A}$ takes that box without inspection.*

The following terminologies we inherit from Guha et al. [14].

**Definition A.1.** *We say a box is a* backup *in the execution of a policy, if the box is taken without inspection.*

**Definition A.2** ($\leq_V$ tree and $\leq_V$ path)**.** *For $V \geq 0$, a $\leq_V$* tree *is a decision tree that makes the same decisions irrespective of the values of the probed boxes, as long as these values are less than or equal to $V$. In a $\leq_V$ tree, decisions constitute a path if all observed values are no larger than $V$; such a path is called a $\leq_V$* path*.*

**Lemma A.3.** *Suppose an optimal policy $\mathcal{A}^*$ probes a box $B_j$ at a node $m$ in its decision tree, and suppose when $B_j$ is observed to take value $V$, $\mathcal{A}^*$ takes a backup box somewhere down the tree. Then there exists another optimal policy $\mathcal{A}'$ which has the same decision tree as $\mathcal{A}^*$ except possibly for the subtree rooted at $m$. In $\mathcal{A}'$, the subtree rooted at $m$ is a $\leq_V$ tree and takes a backup box at the end of its $\leq_V$ path.*

*Proof.* The proof proceeds via induction on the value $V$.

(Base:) If $V$ is the highest value in the support of all value distributions, the statement is true since it is optimal to end the game immediately and never take a backup (and so the condition cannot be satisfied).

(Inductive step:) Now suppose the statement is true for values larger than $V$. Let $u$ be the child node of $m$ where box $B_j$ yields value $V$, then the subtree rooted at $u$ is without loss of generality a $\leq_V$ tree. Suppose somewhere in this subtree, a backup is taken, we first show that it is without loss of generality to assume that the end of the $\leq_V$ path from $u$ is a backup. Suppose this is not the case, then some node along the $\leq_V$ path from $u$ has a child not on this path which has a backup descendant; let $w$ be such a node (on the $\leq_V$ path) closest to $u$. ($w$ may be $u$ itself.) Since the said child of $w$ with a backup descendant is not on the $\leq_V$ path, the box opened at $w$ must yield a value $V' > V$ to arrive at that child. By the induction hypothesis, we may assume that the subtree rooted at $w$ is a $\leq_{V'}$ tree, and the $\leq_{V'}$ path originating from $w$ ends in a backup. But the $\leq_{V'}$ path from $w$ is part of the $\leq_V$ path from $u$, and therefore the end of the $\leq_V$ path from $u$ is a backup.

For each value $h$ possibly taken by box $B_j$, let $T_h$ be the subtree rooted at the child of $w$ where $v_{B_j} = h$. (So $T_V$ is the subtree rooted at $u$.) The crucial observation is that, for every $h < V$, one may replace the subtree $T_h$ by $T_V$ without decreasing the expected payoff. (To see this, note first that box $B_j$ is never used in $T_V$: it is not used along the $\leq_V$ path because a backup is used in the end; $B_j$ is not used elsewhere either, because one must see a value larger than $V$ to leave the $\leq_V$ path in the first place, and that value is preferred to $v_{B_j} = V$. Therefore, it is feasible to replace $T_h$ by $T_V$. Such a replacement does not decrease the expected payoff, because the expected payoff of $T_h$ is no more than that of $T_V$ for $h < V$, by the optimality of $\mathcal{A}^*$.) After such replacements, the subtree rooted at $m$ becomes a $\leq_V$ tree, and its endpoint is a backup, the same backup as the end of the $\leq_V$ path from $u$. The resulting policy is the $\mathcal{A}^*$ stated by the lemma.

□

*Proof of Theorem 2.3.* Let $\mathcal{A}^*$ be an optimal policy. Consider the non-trivial case when $\mathcal{A}^*$ probes at least one box and takes a backup box somewhere. Suppose $\mathcal{A}^*$ probes box $i$ first. Let $V_i$ be the maximum value taken by box $i$ such that a backup is possibly taken later. Then, applying Lemma A.3 to the root of the decision tree, we may assume the decision tree is a $\leq_{V_i}$ tree. If box $i$ yields a value larger than $V_i$, then by the maximality of $V_i$, no backup is used downstream, and (without loss of generality) $\mathcal{A}^*$ follows the index policy on the remaining boxes, taking $v_i$ as a free outside option. Now apply the same argument to the

next node along the $\leq_{V_i}$ path. The theorem follows by a repeated application of this argument until the next node in line is a backup: as long as the next node probes a box, the threshold of that box is given by the largest value it can take to still see some backup somewhere downstream; the ordering $\sigma$ is given by the order in which boxes are probed by the nodes on which Lemma A.3 is applied.

□

# B Appendix for Section 3

**Proposition 3.3.** *[14] There is a polynomial-time computable optimal policy for PNOI instances where all value distributions are supported on* $\{0, 1\}$.

*Proof.* A few observations are in order.

(1) A box $i$ to be taken without inspection can be seen as a box with deterministic value $\mathbf{E}[v_i]$ with no search cost. Therefore, by the optimality of the index-based policy in the classic Pandora box problem, one should never take a box $i$ without inspection if some other unopened box has index larger than $\mathbf{E}[v_i]$.

(2) When a box yields value $1$ upon inspection, it is payoff-optimal to select the box immediately and quit.

(3) For any policy satisfying (2), the way in which it opens boxes is completely described by a subset $S \subseteq [n]$ and a permutation $\pi$ on $S$. The policy opens boxes in $S$ in the order specified by $\pi$: if a box yields value $1$, the box is taken immediately and the search terminates; otherwise, this goes till all boxes in $S$ are opened and yield value $0$, at which point the algorithm may terminate or take a box not in $S$ without inspection.

From these observations, it is without loss of generality to consider policies that: (i) commit to a certain box $i$ that is possibly selected without inspection; (ii) inspect boxes with indices at least $\mathbf{E}[v_i]$ in decreasing order of their indices, and if a value $1$ is found, select that box and quit; (iii) when all boxes in step (ii) yield value $0$, take box $i$ without inspection and quit.

There are altogether $n$ such policies (up to tie-breaking in step (ii), which does not matter). We can enumerate them and choose the best one in polynomial time.

□

**Lemma 3.5.** *For any LCLRS3 instance, an optimal policy $\mathcal{A}$ is normal.*

*Proof.* We prove the three properties of a normal policy in order.

- It is straightforward to see that $\mathcal{A}$ should take the last box without inspection if all previous boxes yield value 0. To see that this is the only situation $\mathcal{A}$ should bypass inspection, recall by observation (1) in the proof of Proposition 3.3 that an optimal policy should not take a box without inspection if there are other unopened boxes with higher indexes. Note that, for any two boxes $i$ and $j$, $\mathbf{E}[v_i] < \frac{1}{2} \leq \tau_j$ by definition of LCLRS3 instances.

- It is straightforward that $\mathcal{A}$ stops when it sees value $1$ — no other box can yield higher values in an LCLRS3 instance, and opening more boxes strictly diminishes the payoff.

- Since any unopened box has expected value strictly smaller than $\frac{1}{2}$, $\mathcal{A}$ should never take an unopened box without inspection if a value $\frac{1}{2}$ is already seen. In other words, with value $\frac{1}{2}$ seen, $\mathcal{A}$ ignores the option to bypass inspection, and the problem degenerates to the classical Pandora box problem for the remaining boxes. Therefore, after a value $\frac{1}{2}$ is seen, $\mathcal{A}$ runs the index-based policy on the remaining boxes.

□

**Lemma 3.6.** *Given an LCLRS3 instance and a normal policy $\mathcal{A}$ for it, let $\sigma$ be the corresponding ordering. For box $i$, let $T_\sigma(i)$ be the set of boxes ordered after box $i$ by $\sigma$, with Gittins indices strictly larger than $\tau_i$; that is, $T_\sigma(i) \coloneqq \{j \in [n] : \sigma^{-1}(j) > \sigma^{-1}(i), \tau_j > \tau_i\}$. For $i \in [n]$ and $T \subseteq [n]$, define $g(i,T) \coloneqq \mathbf{E}[(\max_{j\in T} \kappa_j - \tau_i)_+]$.[6] Let $\mathcal{A}_P$ be the index-based policy on the instance. Then*

$$\mathbb{P}(\mathcal{A}_P) = \mathbb{P}(\mathcal{A}) + \sum_i p_i g(i, T_\sigma(i)) \prod_{j=1}^{\sigma^{-1}(i)-1} r_{\sigma(j)} - c_{\sigma(n)} \prod_{i=1}^{n-1} r_{\sigma(i)}. \tag{1}$$

*Proof.* We first modify $\mathcal{A}$ to obtain another policy $\mathcal{A}'$. $\mathcal{A}'$ has the same behavior as $\mathcal{A}$ in all situations, except that when $\mathcal{A}$ takes the last box $\sigma(n)$ without opening it, $\mathcal{A}'$ opens box $\sigma(n)$, pays the cost $c_{\sigma(n)}$, and takes it. Since $\mathcal{A}$ is normal, it takes box $\sigma(n)$ only when all the other boxes have yielded value 0, which happens with probability $\prod_{i=1}^{n-1} r_{\sigma(i)}$. So we have

$$\mathbb{P}(\mathcal{A}) = \mathbb{P}(\mathcal{A}') + c_{\sigma(n)} \prod_{i=1}^{n-1} r_{\sigma(i)}. \tag{11}$$

Next, we derive a simple expression for $\mathbb{P}(\mathcal{A}')$ and then show that the difference between $\mathbb{P}(\mathcal{A}')$ and $\mathbb{P}(\mathcal{A}_P)$ gives rise to the second term in (1). We claim $\mathbb{P}(\mathcal{A}') = \sum_i \mathbf{E}[A'_i \kappa_i]$, where $A'_i$ is the indicator variable for the event that $\mathcal{A}'$ takes box $i$. To see this, we make two observations, which amounts to showing that $\mathcal{A}'$ is non-exposed (Definition 2.2). Let $I'_i$ be the indicator variable for the event that $\mathcal{A}'$ opens box $i$.

- $\mathcal{A}'$ never exercises the option to take a box without opening it, so $A'_i \le I'_i$ with probability 1, for all $i$.
- Whenever $\mathcal{A}'$ opens box $i$ and sees $v_i > \tau_i$, $\mathcal{A}'$ immediately takes box $i$. To see this, if $i = \sigma(n)$, by definition $\mathcal{A}'$ takes the box after opening it. For $i \neq \sigma(n)$, by definition of LCLRS3, $\tau_i \ge \frac{1}{2}$, so $v_i > \tau_i$ implies $v_i = 1$. Since $\mathcal{A}$ is normal, it immediately takes box $i$ when seeing $v_i = 1$; $\mathcal{A}'$ copies the behavior of $\mathcal{A}$ for $i \neq \sigma(n)$, and hence also immediately takes it.

From the second observation, we have $(v_i - \tau_i)_+ I'_i = (v_i - \tau)_+ A'_i$ with probability 1. Therefore

$$\begin{aligned}\mathbb{P}(\mathcal{A}') &= \sum_i \mathbf{E}\left[v_i A'_i - c_i I'_i\right] = \sum_i \mathbf{E}\left[v_i A'_i - (v_i - \tau_i)_+ I'_i\right] \\ &= \sum_i \mathbf{E}\left[v_i A'_i - (v_i - \tau_i)_+ A'_i\right] = \sum_i \mathbf{E}\left[A'_i \kappa_i\right].\end{aligned}$$

We now compare $\mathbb{P}(\mathcal{A}_P)$ and $\mathbb{P}(\mathcal{A}')$. By Theorem 2.1, $\mathbb{P}(\mathcal{A}_P) = \mathbf{E}[\max_i \kappa_i]$. For every realization of $\kappa_i$'s, $\max_i \kappa_i \ge \sum_i A'_i \kappa_i$. The inequality is strict only when $\mathcal{A}'$ takes some box $i$ with $\kappa_i < \max_j \kappa_j$. This can happen only when all boxes opened before $i$ yield value 0, and $v_i = 1$, in which case $\kappa_i = \tau_i$. This happens with probability $p_i \prod_{j=1}^{\sigma^{-1}(i)-1} r_{\sigma(j)}$; the expected contribution to the difference between $\max_i \kappa_i$ and $\tau_i$ conditioning on this happening is $\mathbf{E}[(\max_{j:\sigma^{-1}(j)>\sigma^{-1}(i)} \tau_j - \tau_i)_+]$, which is just $g(i, T_\sigma(i))$ as

[6] $g(i, \emptyset) \coloneqq 0$.

defined in the statement of the lemma. Therefore, overall, we have

$$\begin{aligned}\mathbb{P}(\mathcal{A}_P) - \mathbb{P}(\mathcal{A}') &= \mathbf{E}\left[\max_i \kappa_i\right] - \sum_i \mathbf{E}\left[A'_i \kappa_i\right] \\ &= \sum_i \Pr[A'_i = 1] \cdot \mathbf{E}\left[\max_j \kappa_j - \kappa_i \mid A'_i = 1\right] \\ &= \sum_i p_i g(i, T_\sigma(i)) \prod_{j=1}^{\sigma^{-1}(i)-1} r_{\sigma(j)}. \end{aligned} \tag{12}$$

Combining (11) and (12), we have

$$\mathbb{P}(\mathcal{A}_P) = \mathbb{P}(\mathcal{A}) + \sum_i p_i g(i, T_\sigma(i)) \prod_{j=1}^{\sigma^{-1}(i)-1} r_{\sigma(j)} - c_{\sigma(n)} \prod_{i=1}^{n-1} r_{\sigma(i)}.$$

□

**Claim 3.9.** *Let $\sigma^*$ be a permutation which maximizes* (2)*. Then $\sigma^*(n+2) = n+2$.*

*Proof.* By (2), it is easy to verify that when $n \geq 1$ and $\sigma^*(n+2) = n+2$ one has

$$\begin{aligned}&\text{Utility}(\sigma^*) \\ &\geq c_{n+2} r_{n+1} (1 - 2^{-6n})^n - n p_{n+2} \max_i p_i(\tau_H - \tau_i) - p_{n+1}\Big((\tau_H - \tau_L) p_{n+2} + n \max_{i \in [n]} p_i(\tau_i - \tau_L)\Big) \\ &\geq \frac{40}{32\Gamma}(1 - 2^{-6n})^n - \frac{1}{32\Gamma} - O(n\Delta^2) \\ &\geq \frac{38}{32\Gamma}.\end{aligned}$$

For any permutation $\sigma$ with $\sigma(n+2) \in [n]$, we have

$$\text{Utility}(\sigma) \leq r_{n+1} \max_{i \in [n]} c_i \leq O(\Delta^2)$$

as $c_i = \frac{p_i}{\tau_i + 1} \leq p_i$.

For any permutation $\sigma$ with $\sigma(n+2) = n+1$, we have

$$\text{Utility}(\sigma) \leq r_{n+2} c_{n+1} \leq \frac{3}{8\Gamma} < \frac{38}{32\Gamma}.$$

Hence if some permutation $\sigma^*$ maximize Equation (2), then $\sigma^*(n+2) = n+2$. □

## B.1 Omitted Proof from Appendix 3.3

For ease of presentation, we introduce the following notations.

**Definition B.1.**

$$\begin{aligned} g_H &\coloneqq p_{n+2}(\tau_{n+2} - \tau_H); \\ g_L &\coloneqq \left[1 - \prod_{i \in T_{\text{Aft}}} (1 - p_i)\right](1 - p_{n+2})(\tau_H - \tau_L) + p_{n+2}(\tau_{n+2} - \tau_L); \\ g_i &\coloneqq g(i, T_\sigma(i)), \quad \text{for } i = 1, \ldots, n+1. \end{aligned}$$

**Claim B.2.**

$$g_i = g_H - \frac{p_i p_{n+1}(1-p_{n+2})(\tau_H - \tau_L)}{2} \pm O(n\Delta^3), \quad \text{for } i = 1, \dots, n;$$
$$g_{n+1} = g_L \pm O(n\Delta^3).$$

*Proof.* Recall that $g(i,T) \coloneqq \mathbf{E}[(\max_{j\in T} \kappa_j - \tau_i)_+]$ and $\kappa_i = \min\{v_i, \tau_i\}$. Also, $\tau_{n+2} > \tau_i > \tau_H > \tau_L = 1/2$ for all $i \in [n]$ in our LCLRS3 instance. Therefore, for each $j \neq i$,

$$(\kappa_j - \tau_i)_+ = \begin{cases} 0, & \text{if } v_j \leq 1/2; \\ (\tau_j - \tau_i)_+, & \text{if } v_j = 1. \end{cases}$$

Since $\tau_{n+2}$ is by far the largest among all indices, and $\tau_{n+1} = \tau_L$ is the lowest index, we have for each $i \in [n]$,

$$\begin{aligned}
g_i &= p_{n+2}(\tau_{n+2} - \tau_i) + (1-p_{n+2})\,\mathbf{E}\left[\max_{j\in T_\sigma(i)\setminus\{n+2\}} (\kappa_j - \tau_i)_+\right] \\
&= p_{n+2}(\tau_{n+2} - \tau_H) - \frac{1}{2}p_i p_{n+1}(1-p_{n+2})(\tau_H - \tau_L) + (1-p_{n+2})\,\mathbf{E}\left[\max_{j\in T_\sigma(i)\setminus\{n+2\}} (\kappa_j - \tau_i)_+\right] \\
&\leq g_H - \frac{1}{2}p_i p_{n+1}(1-p_{n+2})(\tau_H - \tau_L) + (1-p_{n+2}) \sum_{j\in T_\sigma(i)\setminus\{n+2\}} p_j(\tau_j - \tau_i)_+ \\
&\leq g_H - \frac{1}{2}p_i p_{n+1}(1-p_{n+2})(\tau_H - \tau_L) \pm O(n\Delta^3).
\end{aligned}$$

The last inequality is from the fact that $\tau_i = \tau_H + O(\Delta^2)$ for any $i \in [n]$.

Similarly,

$$\begin{aligned}
g_{n+1} &= p_{n+2}(\tau_{n+2} - \tau_L) + (1-p_{n+2})\,\mathbf{E}\left[\max_{j\in T_{\text{Aft}}} (\kappa_j - \tau_L)_+\right] \\
&\leq p_{n+2}(\tau_{n+2} - \tau_L) + (1-p_{n+2})(1 - \Pi_{i\in T_{\text{Aft}}}(1-p_i)) \max_{j\in T_{\text{Aft}}} (\tau_j - \tau_L) \\
&\leq p_{n+2}(\tau_{n+2} - \tau_L) + (1-p_{n+2})(1 - \Pi_{i\in T_{\text{Aft}}}(1-p_i))(\tau_H \pm O(\Delta^2) - \tau_L) \\
&= g_L \pm O(n\Delta^3).
\end{aligned}$$

□

**Lemma 3.11.** *Given two sequences of positive real numbers $p_1, p_2, \cdots, p_n$ and $r_0, r_1, \cdots, r_n$. Let $r_0 = 1$. If there exists a constant $c > 0$ such that $p_i/(1-r_i) = c$ for each $1 \leq i \leq n$, then we have*

$$\sum_{i=1}^{n} p_i \prod_{j=0}^{i-1} r_j = c\left(1 - \prod_{i=1}^{n} r_i\right).$$

*Proof.* We prove this lemma by induction on $n$. When $n = 1$, we get $p_1 r_0 = p_1 = c(1-r_1)$ by the assumption.

Suppose one can have $\sum_{i=1}^{n} p_i \prod_{j=0}^{i-1} r_j = c(1 - \prod_{i=1}^{n} r_i)$, then

$$\begin{aligned}
\sum_{i=1}^{n+1} p_i \prod_{j=0}^{i-1} r_j =& \sum_{i=1}^{n} p_i \prod_{j=0}^{i-1} r_j + p_{n+1} \prod_{j=0}^{n} r_j \\
=& c(1 - \prod_{i=1}^{n} r_i) + p_{n+1} \prod_{j=0}^{n} r_j \\
=& c(1 - \prod_{i=1}^{n} r_i) + c(1 - r_{n+1}) \prod_{j=0}^{n} r_j \\
=& c(1 - \prod_{i=1}^{n+1} r_i).
\end{aligned}$$

□

**Claim 3.12.** *For a non-empty set $T \subseteq [n]$, let $f(T) := \prod_{i \in T} r_i = \prod_{i \in T}(1-2p_i)$ and $g(T) := \prod_{i \in T}(1-p_i)$. Also let $f(\emptyset) = g(\emptyset) = 1$. Then*

$$\mathrm{Loss}(\sigma) = k_1 f(T_{\text{Bef}}) - k_2 f(T_{\text{Bef}}) g(T_{\text{Aft}}) - k_2 \sum_{i \in T_{\text{Aft}}} p_i^2/2 + C \pm O(n^2\Delta^4). \tag{9}$$

*Proof.* First, we show

$$\begin{aligned}
\mathrm{Loss}(\sigma) =& \frac{g_H}{2} \left(1 - \prod_{i \in T_{\text{Bef}}} r_i\right) + g_L p_{n+1} \prod_{i \in T_{\text{Bef}}} r_i + \frac{g_H r_{n+1}}{2} \prod_{i \in T_{\text{Bef}}} r_i \left(1 - \prod_{i \in T_{\text{Aft}}} r_i\right) \\
&+ \frac{1}{2} \sum_{i \in T_{\text{Bef}}} p_i^2 p_{n+1}(1 - p_{n+2})(\tau_H - \tau_L) \pm O(n^2\Delta^4).
\end{aligned}$$

The claim follows by plugging in the parameters. One has

$$
\begin{aligned}
\mathrm{Loss}(\sigma) &= \sum_i p_i g(i, T_\sigma(i)) \prod_{j=1}^{\sigma^{-1}(i)-1} r_{\sigma(j)} \\
&= \sum_{i\in T_{\text{Bef}}} p_i g_i \prod_{j=1}^{\sigma^{-1}(i)-1} r_{\sigma(j)} + p_{n+1} g_{n+1} \prod_{i\in T_{\text{Bef}}} r_i + \sum_{i\in T_{\text{Aft}}} p_i g_i \prod_{j=1}^{\sigma^{-1}(i)-1} r_{\sigma(j)} \\
&= \sum_{i\in T_{\text{Bef}}} p_i \left( g_H - \frac{1}{2} p_i p_{n+1}(1-p_{n+2})(\tau_H - \tau_L) \pm O(n\Delta^3) \right) \prod_{j=1}^{\sigma^{-1}(i)-1} r_{\sigma(j)} \\
&\quad + p_{n+1}(g_L \pm O(n\Delta^3)) \prod_{i\in T_{\text{Bef}}} r_i + r_{n+1} \prod_{s\in T_{\text{Bef}}} r_s \sum_{i\in T_{\text{Aft}}} p_i g_i \prod_{j=|T_{\text{Bef}}+2|}^{\sigma^{-1}(i)-1} r_{\sigma(j)} \\
&= \sum_{i\in T_{\text{Bef}}} p_i g_H \prod_{j=1}^{\sigma^{-1}(i)-1} r_{\sigma(j)} - \sum_{i\in T_{\text{Bef}}} \left( \frac{1}{2} p_i^2 p_{n+1}(1-p_{n+2})(\tau_H - \tau_L) \right) \prod_{j=1}^{\sigma^{-1}(i)-1} r_{\sigma(j)} \\
&\quad + p_{n+1} g_L \prod_{i\in T_{\text{Bef}}} r_i + r_{n+1} \prod_{s\in T_{\text{Bef}}} r_s \sum_{i\in T_{\text{Aft}}} \prod_{j=|T_{\text{Bef}}+2|}^{\sigma^{-1}(i)-1} r_{\sigma(j)} p_i g_i \pm O(n^2\Delta^4) \\
&= \sum_{i\in T_{\text{Bef}}} p_i g_H \prod_{j=1}^{\sigma^{-1}(i)-1} r_{\sigma(j)} - \sum_{i\in T_{\text{Bef}}} \left( \frac{1}{2} p_i^2 p_{n+1}(1-p_{n+2})(\tau_H - \tau_L) \right) \prod_{j=1}^{\sigma^{-1}(i)-1} r_{\sigma(j)} \\
&\quad + r_{n+1} \prod_{s\in T_{\text{Bef}}} r_s \sum_{i\in T_{\text{Aft}}} \prod_{j=|T_{\text{Bef}}+2|}^{\sigma^{-1}(i)-1} r_{\sigma(j)} p_i \Big( g_H - \frac{p_i p_{n+1}(1-p_{n+2})(\tau_H - \tau_L)}{2} \pm O(n\Delta^3) \Big) \\
&\quad + p_{n+1} g_L \prod_{i\in T_{\text{Bef}}} r_i \pm O(n^2\Delta^4) \\
&= \sum_{i\in T_{\text{Bef}}} p_i g_H \prod_{j=1}^{\sigma^{-1}(i)-1} r_{\sigma(j)} - \sum_{i\in T_{\text{Bef}}} \left( \frac{1}{2} p_i^2 p_{n+1}(1-p_{n+2})(\tau_H - \tau_L) \right) \prod_{j=1}^{\sigma^{-1}(i)-1} r_{\sigma(j)} \\
&\quad + r_{n+1} \prod_{s\in T_{\text{Bef}}} r_s \sum_{i\in T_{\text{Aft}}} \prod_{j=|T_{\text{Bef}}+2|}^{\sigma^{-1}(i)-1} r_{\sigma(j)} p_i \left( g_H - \frac{1}{2} p_i p_{n+1}(1-p_{n+2})(\tau_H - \tau_L) \right) \\
&\quad + p_{n+1} g_L \prod_{i\in T_{\text{Bef}}} r_i \pm O(n^2\Delta^4),
\end{aligned}
$$

where in the last equality we used $p_{n+1} = 1/\Gamma = O(\Delta)$. Further analyzing the last term, we have

$$
\begin{aligned}
& r_{n+1} \prod_{s\in T_{\text{Bef}}} r_s \sum_{i\in T_{\text{Aft}}} \prod_{j=|T_{\text{Bef}}+2|}^{\sigma^{-1}(i)-1} r_{\sigma(j)} p_i \left( g_H - \frac{1}{2} p_i p_{n+1}(1-p_{n+2})(\tau_H - \tau_L) \right) \\
&= r_{n+1} \prod_{s\in T_{\text{Bef}}} r_s \sum_{i\in T_{\text{Aft}}} \prod_{j=|T_{\text{Bef}}+2|}^{\sigma^{-1}(i)-1} r_{\sigma(j)} p_i g_H - O(n\Delta^4).
\end{aligned}
$$

Note that $\prod_{j=1}^{\sigma^{-1}(i)-1} r_{\sigma(j)} = 1 - O(n\Delta)$ for each $i \in T_{\text{Bef}}$ and $r_{n+1} = 3/\Gamma = O(\Delta)$. Besides, notice that,

for each $i \in [n]$, we have $\frac{p_i}{1-r_i} = \frac{s_i/\Gamma}{2s_i/\Gamma} = \frac{1}{2}$. Therefore, Lemma 3.11 applies, and we have

$$\begin{aligned}
\mathrm{Loss}(\sigma) =& \frac{g_H}{2}\left(1 - \prod_{i\in T_{\text{Bef}}} r_i\right) + (1 - O(n\Delta)) \sum_{i\in T_{\text{Bef}}} \frac{1}{2} p_i^2 p_{n+1}(1-p_{n+2})(\tau_H - \tau_L) + p_{n+1} g_L \prod_{i\in T_{\text{Bef}}} r_i \\
&+ r_{n+1} \prod_{s\in T_{\text{Bef}}} r_s \sum_{i\in T_{\text{Aft}}} \prod_{j=|T_{\text{Bef}}+2|}^{\sigma^{-1}(i)-1} r_{\sigma(j)} p_i g_H \pm O(n^2\Delta^4) \\
=& \frac{g_H}{2}\left(1 - \prod_{i\in T_{\text{Bef}}} r_i\right) + \sum_{i\in T_{\text{Bef}}} \frac{1}{2} p_i^2 p_{n+1}(1-p_{n+2})(\tau_H - \tau_L) + p_{n+1} g_L \prod_{i\in T_{\text{Bef}}} r_i \\
&+ \frac{1}{2} g_H r_{n+1} \prod_{s\in T_{\text{Bef}}} r_s \left(1 - \prod_{i\in T_{\text{Aft}}} r_i\right) \pm O(n^2\Delta^4).
\end{aligned}$$

Rewrite the equation in terms of $f, g$ and $k_1, k_2$, and we have

$$\begin{aligned}
\mathrm{Loss}(\sigma) =& \Big(-\frac{g_H}{2} + g_L p_{n+1} + \frac{g_H r_{n+1}}{2}\Big) \prod_{i\in T_{\text{Bef}}} r_i + \frac{1}{2} p_{n+1}(1-p_{n+2})(\tau_H - \tau_L) \sum_{i\in T_{\text{Bef}}} p_i^2 \\
&+ \frac{g_H}{2} - \frac{1}{2} g_H \prod_{i\in[n+1]} r_i \pm O(n^2\Delta^4) \\
=& \Big(-\frac{g_H(p_{n+1}+q_{n+1})}{2} + p_{n+1}((1-p_{n+2})(\tau_H - \tau_L) + p_{n+2}(\tau_{n+2} - \tau_L))\Big) \prod_{i\in T_{\text{Bef}}} r_i \\
&- p_{n+1}(1-p_{n+2})(\tau_H - \tau_L) \prod_{i\in T_{\text{Bef}}} r_i \prod_{j\in T_{\text{Aft}}} (1-p_j) + \frac{1}{2} p_{n+1}(1-p_{n+2})(\tau_H - \tau_L) \sum_{i\in T_{\text{Bef}}} p_i^2 \\
&+ \frac{p_{n+2}(\tau_{n+2} - \tau_H)}{2}\left(1 - \prod_{i\in[n+1]} r_i\right) \pm O(n^2\Delta^4) \\
=& k_1 f(T_{\text{Bef}}) - k_2 f(T_{\text{Bef}}) g(T_{\text{Aft}}) - \frac{1}{2} k_2 \sum_{i\in T_{\text{Aft}}} p_i^2 + C \pm O(n^2\Delta^4).
\end{aligned}$$

□

**Claim 3.15.** *For any subset $T$ of the first $n$ boxes, one has*

$$\begin{aligned}
e^{-\sum_{i\in T} 2(p_i+p_i^2)} \geq f(T) \geq e^{-\sum_{i\in T} 2(p_i+p_i^2)} - O(n\Delta^3), \\
e^{-\sum_{i\in T} (p_i+p_i^2/2)} \geq g(T) \geq e^{-\sum_{i\in T} (p_i+p_i^2/2)} - O(n\Delta^3).
\end{aligned}$$

*Proof.* Consider $f(T)$ first. By Fact 3.14, as $p_i \leq 2^n/\Gamma = \Delta$, we know $e^{-2(p_i+p_i^2)} = 1 - 2p_i + O(p_i^3)$. Then $f(T) = \Pi_{i\in T}(1-2p_i) = \Pi_{i\in T}(e^{-2(p_i+p_i^2)} - O(p_i^3)) \geq \Pi_{i\in T} e^{-2(p_i+p_i^2)} - |T| \max_i O(p_i^3)$.

As for the other side, again by Fact 3.14 we have $1 - 2p_i \leq e^{-2(p_i+p_i^2)}$, which directly implies $e^{-\sum_{i\in T} 2(p_i+p_i^2)} \geq f(T)$.

The other inequality is based on $e^{-(p_i+p_i^2/2)} - O(p_i^3) \leq 1 - p_i \leq e^{-(p_i+p_i^2/2)}$, therefore it holds by the same argument. □

**Claim 3.13.** *If we choose $t$ so that $|t - 2e^{y/2}| \le O(\Delta^2)$, and set $\tau_H$ as follows, then $\frac{k_2}{k_1} = t$:*

$$\tau_H = \frac{-3t\Gamma + 28 + 94t}{-4t\Gamma + 56 + 104t}. \tag{10}$$

*Proof.* We would like

$$k_2 = tk_1 = tk_2 + t[-p_{n+2}(\tau_{n+2} - \tau_H)(p_{n+1} + q_{n+1})/2 + p_{n+1}p_{n+2}(\tau_{n+2} - \tau_L)],$$

which is equivalent to

$$(1-t)p_{n+1}(1-p_{n+2})(\tau_H - \tau_L) = t[-p_{n+2}(\tau_{n+2} - \tau_H)(p_{n+1} + q_{n+1})/2 + p_{n+1}p_{n+2}(\tau_{n+2} - \tau_L)].$$

Rearranging the terms, we get

$$\begin{aligned}&[(1-t)p_{n+1}(1-p_{n+2}) - tp_{n+2}(p_{n+1} + q_{n+1})/2]\tau_H \\ =&t[-p_{n+2}\tau_{n+2}(p_{n+1} + q_{n+1})/2 + p_{n+1}p_{n+2}(\tau_{n+2} - \tau_L)] + (1-t)p_{n+1}(1-p_{n+2})\tau_L.\end{aligned}$$

Recall that $p_{n+2} = 1/8, \tau_{n+2} = 3/4, p_{n+1} = 1/\Gamma, q_{n+1} = 1 - 41/\Gamma, \tau_L = 1/2$ and thus

$$\begin{aligned}\left[\frac{7(1-t)}{8\Gamma} - \frac{t}{16}\left(1 - \frac{40}{\Gamma}\right)\right]\tau_H &= \frac{7(1-t)}{16\Gamma} - \frac{3t}{64}\left(1 - \frac{40}{\Gamma}\right) + \frac{t}{32\Gamma} \\ \Rightarrow\ \tau_H = \frac{\frac{7(1-t)}{16\Gamma} - \frac{3t}{64}(1 - \frac{40}{\Gamma}) + \frac{t}{32\Gamma}}{\frac{7(1-t)}{8\Gamma} - \frac{t}{16}(1 - \frac{40}{\Gamma})} &= \frac{-3t\Gamma + 28 + 94t}{-4t\Gamma + 56 + 104t}.\end{aligned}$$

We need $1/2 < \tau_H < 3/4$, which means

$$\frac{1}{2} < \tau_H = \frac{-3t\Gamma + 28 + 94t}{-4t\Gamma + 56 + 104t} < \frac{3}{4}.$$

For the first inequality, noting that $|t - 2| \le O(y + \Delta^2) = O(\Delta^2)$ and $\Delta = 2^{-7n}$, we have

$$-4t\Gamma + 56 + 104t > -6t\Gamma + 2(28 + 94t),$$

which means the first inequality holds.

As for the second inequality, we need

$$\begin{aligned}-12t\Gamma + 4(28 + 94t) &> -12t\Gamma + 3(56 + 104)t, \\ \iff 64t &> 56,\end{aligned}$$

which holds by our choice of $t$. □

**Lemma 3.10.** *The parameters of the instance can be set up so that*

$$\begin{aligned}h(x) - O(n^2\Delta^4) \le \frac{\mathrm{Loss}(\sigma) - C \pm O(n^2\Delta^4)}{k_1} &\le h(x) + O(n\Delta^3), \\ \frac{k_2}{k_1} &= 2e^{y/2} \pm O(\Delta^2),\end{aligned} \tag{4}$$

*where*

$$h(x) := e^{-2x}\left(1 - \frac{k_2}{k_1}e^{-y+x}\right), \tag{5}$$

*with $C, k_1, k_2$ as constants independent of $\sigma$:*

$$k_1 := -\frac{1}{2}p_{n+2}(\tau_{n+2}-\tau_H)(p_{n+1}+q_{n+1}) + p_{n+1}[(1-p_{n+2})(\tau_H-\tau_L) + p_{n+2}(\tau_{n+2}-\tau_L)], \quad (6)$$

$$k_2 := p_{n+1}(1-p_{n+2})(\tau_H-\tau_L), \quad (7)$$

$$C := \frac{1}{2}p_{n+2}(\tau_{n+2}-\tau_H)\left(1-\prod_{i\in[n+1]} r_i\right) + \frac{1}{2}k_2\sum_{i=1}^{n} p_i^2. \quad (8)$$

*Proof.* By the setup of the parameters, we have $k_1, k_2 > 0$ and $|k_2/k_1 - 2e^{y/2}| \le \Delta^2$. Combining Claim 3.12 and Claim 3.15 gives

$$\begin{aligned}
&k_1\left(e^{-\sum_{i\in T_{\text{Bef}}}2(p_i+p_i^2)}\right)\left(1-\frac{k_2}{k_1}e^{-\sum_{i\in T_{\text{Aft}}}(p_i+p_i^2/2)}\right)\\
\le&\ \text{Loss}(\sigma) - C + \frac{1}{2}k_2\sum_{i\in T_{\text{Aft}}} p_i^2 \pm O(n^2\Delta^4),\\
\le&\ k_1\left(e^{-\sum_{i\in T_{\text{Bef}}}2(p_i+p_i^2)} - O(n\Delta^3)\right)\left(1-\frac{k_2}{k_1}\left(e^{-\sum_{i\in T_{\text{Aft}}}(p_i+p_i^2/2)} - O(n\Delta^3)\right)\right)\\
\le&\ k_1\left(e^{-\sum_{i\in T_{\text{Bef}}}2(p_i+p_i^2)}\right)\left(1-\frac{k_2}{k_1}e^{-\sum_{i\in T_{\text{Aft}}}(p_i+p_i^2/2)}\right) + O(|k_1|n\Delta^3).
\end{aligned}$$

Recall that $y := \sum_{i\in S} s_i/\Gamma + (s_i/\Gamma)^2 = \sum_{i\in T_{\text{Aft}}\cup T_{\text{Bef}}} p_i + p_i^2$, $x := \sum_{i\in T_{\text{Bef}}} p_i + p_i^2$ and we define $z := \sum_{i\in T_{\text{Aft}}} p_i^2/2$ for analysis. We know that $e^{-\sum_{i\in T_{\text{Bef}}}2(p_i+p_i^2)} = e^{-2x}$ and $e^{-y+x+z} = e^{-\sum_{i\in T_{\text{Aft}}}(p_i+p_i^2/2)}$. Note that $0 < k_1 = O(1)$ and we can rewrite the equations above as

$$\begin{aligned}
&e^{-2x}\left(1-\frac{k_2}{k_1}e^{-y+x+z}\right)\\
\le&\frac{\text{Loss}(\sigma) - C + k_2\sum_{i\in T_{\text{Aft}}} p_i^2/2 \pm O(n^2\Delta^4)}{k_1}\\
\le&e^{-2x}\left(1-\frac{k_2}{k_1}e^{-y+x+z}\right) + O(n\Delta^3).
\end{aligned}$$

Note that $1+z \le e^z \le 1+z+z^2$ as $0 \le z \le 1/4$ and $k_2/k_1 \approx 2$. On one hand, we know that

$$\begin{aligned}
e^{-2x}\left(1-\frac{k_2}{k_1}e^{-y+x+z}\right) \le& e^{-2x}\left(1-\frac{k_2}{k_1}e^{-y+x}(1+z)\right).\\
\le& e^{-2x}\left(1-\frac{k_2}{k_1}e^{-y+x}\right) - \frac{k_2}{k_1}z + O(xz).
\end{aligned}$$

On the other hand,

$$\begin{aligned}
e^{-2x}\left(1-\frac{k_2}{k_1}e^{-y+x+z}\right) \ge& e^{-2x}\left(1-\frac{k_2}{k_1}e^{-y+x}(1+z+z^2)\right)\\
\ge& e^{-2x}\left(1-\frac{k_2}{k_1}e^{-y+x}\right) - \frac{k_2}{k_1}z - O(z^2).
\end{aligned}$$

Noting that $O(xz) = O(n\Delta^3)$ and $O(z^2) \le O(n^2\Delta^4)$ completes the proof.

□

**Claim 3.16.** *If $|k_2/k_1 - 2e^{y/2}| \le O(\Delta^2)$, $\epsilon \in \mathbb{R}$ is such that $2^{-6n} \ge |\epsilon| \ge 1/\Gamma = 2^{-8n}$ , let $x^* \in [0, 1/2]$ be where $h(x)$ takes its minimum value, then $|x^* - \frac{y}{2}| \le O(\Delta^2)$,*

$$h(x^* + \epsilon) \ge h(x^*) + \epsilon^2/2.$$

*Proof.* Recall $h(x) = e^{-2x}(1 - \frac{k_2}{k_1}e^{-y+x})$. We have $\frac{\mathrm{d}h(x)}{\mathrm{d}x} = -2e^{-2x} + \frac{k_2}{k_1}e^{-y-x}$, and $\frac{\mathrm{d}^2h(x)}{\mathrm{d}x^2} = 4e^{-2x} - \frac{k_2}{k_1}(e^{-y-x}) \in [1, 4]$ for $-2^{-6n} \le x \le 1/2$. Therefore, $\frac{\mathrm{d}h(x)}{\mathrm{d}x} \mid_{x^*} = 0$. Hence by strong convexity, we know $h(x^* + \epsilon) \ge h(x^*) + \epsilon^2/2$.

Now we prove $|x^* - y/2| \le O(\Delta^2)$. We know the derivative $|h'(y/2)| = |-2e^{-y} + \frac{k_2}{k_1}e^{-3y/2}| \le O(\Delta^2)$, which means $|x^* - y/2| \le O(\Delta^2)$ by the strong convexity. □

# C Omitted Proofs from Section 4

## C.1 Omitted Proofs from Section 4.1

**Theorem 4.1.** *[Essentially from [9]] If an SSDP problem $(V, A, f, G, I_0)$ satisfies the following conditions:*

1. *The number of possible states is a constant, i.e., $|V| = O(1)$.*
2. *The state space $V$ admits an ordering "$\ge$" such that $f(I, a) \ge I$ for any $I \in V, a \in A$, i.e., the state is non-decreasing with probability 1.*
3. *There exists an optimal policy that never takes an action with a negative expected marginal payoff in any round.*
4. $\mathrm{MAX} = O(\mathrm{OPT})$.

*Then, for any fixed $\epsilon > 0$, there is a policy $\mathcal{A}$ computable in time $n^{2^{O(\epsilon^{-3})}}$, with expected payoff at least $(1 - \epsilon) \cdot \mathrm{OPT}$.*

*Proof Sketch.* A policy is described by a decision tree, where each node corresponds to an action to be taken (see Section 2). It can be shown that when $|V| = O(1)$, every policy may be approximated, with a loss of at most $O(\epsilon)\mathrm{MAX}$, by an ensemble of policies that is *block adaptive*. A block adaptive policy corresponds to a decision tree whose nodes may be grouped into a small number of *blocks*; within each block, the order of actions affects the expected payoff negligibly, and so the blocks may be seen as supernodes. When one condenses the nodes to such supernodes, the tree's depth and the maximum degree of each node are both bounded by constants; one may therefore enumerate the topologies of all such trees. For each topology, there are exponentially many ways to fill actions into each block, but each block may be approximated by a *signature vector* (signifying the state in a block and the discretized transition probabilities to the other blocks); there are only polynomially many possible signatures, so they can be efficiently enumerated as well. Finally, a dynamic programming is employed to check whether a given set of signatures, one on each node, in a given tree topology can be realized with actual actions. □

**Decision Tree** Given a policy $\mathcal{A}$, for each node $u$ in the decision tree, let $I_u$ be its state, let $a_u$ be the action to be taken by the policy, let $S_u$ be the set of boxes that haven't been opened yet, let $T(u)$ be the subtree rooted at $u$, let $G(u) = \mathbf{E}[G(I_u, a_u)]$ be the expected marginal payoff at node $u$, let $\Phi(u)$ be the probability with which $u$ is reached in an execution of the policy, and let $H(u)$ be the sum of marginal payoffs accumulated when the process reaches $u$. The expected payoff of $\mathcal{A}$ can then be written in two ways:

$$\mathbb{P}(\mathcal{A}) = \sum_u G(u) \cdot \Phi(u) = \sum_{u \text{ is leaf}} H(u) \cdot \Phi(u) \ . \tag{13}$$

**Lemma 4.4.** *An optimal PNOI policy never takes an action with a negative expected marginal payoff in any round.*

*Proof.* It suffices to prove that, in the decision tree $T^*$ of an optimal policy $\mathcal{A}^*$, for any node $u$, $G(u) \geq 0$. Assume, towards a contradiction, that $G(u) < 0$ for a node $u$. $a_u$ cannot be in $A^1$, as an optimal policy cannot take without inspection a box with an expected value smaller than the maximum value seen so far. If $a_u = a_i^0 \in A^0$ for some $i \in [n]$, we construct a modified policy $\mathcal{A}'$ and show it strictly outperforms $\mathcal{A}^*$. $\mathcal{A}'$ uses the same decision tree $T^*$, with the only difference being that when $\mathcal{A}'$ reaches node $u$, instead of probing box $i$ as $\mathcal{A}^*$ does, $\mathcal{A}'$ samples a value $v_i'$ from $F_i$ and simulates the rest of $\mathcal{A}^*$ *pretending* that $v_i$, which is not observed, is equal to $v_i'$. $\mathcal{A}^*$ and $\mathcal{A}'$ reach every subsequent node $w$ with the same probability. If $\mathcal{A}^*$ quits by calling end and taking box $i$, $\mathcal{A}'$ instead takes the maximum value seen so far.[7]

For each subsequent node $w$, let $I_w'$ be the minimum between $I_w$ and (the actual) $v_i$. $I_w'$ may be seen as the *true* state at $w$ in the execution of $\mathcal{A}'$. Then $I_w' \leq I_w$ with probability 1. Similarly, let $G'(w)$ be the *true* expected marginal payoff at node $w$ when $\mathcal{A}'$ is executed at node $w$; that is, $G'(u) = 0$ and $G'(w) = G(I_w', a_w)$ for $w \neq u$. Since $G$ is non-decreasing in its first parameter, $G'(w) \geq G(w)$ for any node $w \in T^*$; in particular, $0 = G'(u) > G(u)$ by assumption. From Equation (13), we conclude that $\mathbb{P}(\mathcal{A}') > \mathbb{P}(\mathcal{A}^*)$, a contradiction to the optimality of $\mathcal{A}^*$. □

## C.2 Omitted Proofs from Section 4.2

**Lemma 4.6.** *Let $B$ be an instance of PNOI and $B^S$ the $S$-discretized instance of $B$, in which each value $v_i$ is replaced by $D^S(v_i)$. Then,*

1. *for any policy $\mathcal{A}$, there is a $S$-discretized policy $\mathcal{A}^S$ such that*

$$\mathbb{P}(\mathcal{A}^S, B^S) \geq \mathbb{P}(\mathcal{A}, B) - O(\epsilon) \cdot \mathrm{OPT};$$

2. *for any $S$-discretized policy $\mathcal{A}^S$, $\mathbb{P}(\mathcal{A}^S, B) \geq \mathbb{P}(\mathcal{A}^S, B^S)$.*

*Proof.* Fix a deterministic optimal policy $\mathcal{A}$ on $B$ and let $T$ be its decision tree. We construct a randomized $S$-discretized policy $\mathcal{A}^S$ which simulates $\mathcal{A}$ when running on the $S$-discretized instance $B^S$. Whenever $\mathcal{A}^S$ probes a $S$-discretized variable $v_i^S$, $\mathcal{A}^S$ randomly samples a value $v_i$ from $F_{i|D^S(v_i)=v_i^S}$, the distribution of $v_i$ conditioning on that it discretizes to $v_i^S$. $\mathcal{A}^S$ then feeds $\mathcal{A}$ the value $v_i$ to simulate it.

It is straightforward that $\mathcal{A}^S$ can be represented by the same decision tree $T$ such that $\mathcal{A}^S$ reaches each node in $T$ with the same probability as $\mathcal{A}$ does. Let $G_S(u)$ be the marginal payoff at a node $u$ when $\mathcal{A}^S$ is executed on $B^S$, and let $G(u)$ be that of $\mathcal{A}$ on the same node $u$ when it is executed on $B$. The state of $\mathcal{A}$ and $\mathcal{A}^S$ differs by at most $\epsilon\theta$ on any node, so $G_S(u) \geq G(u) - \epsilon^2\theta$ since $G$ is Lipschitz. By Equation (13), we have

$$\mathbb{P}(\mathcal{A}^S, B^S) = \sum_{u \in T} G_S(u) \cdot \Phi(u) \geq \sum_{u \in T} (G(u) - \epsilon^2 \cdot \theta) \cdot \Phi(u) \geq \mathbb{P}(\mathcal{A}, B) - O(\epsilon^2) \cdot \theta\ .$$

This proves the first statement.

For the second statement, notice that for an $S$-discretized policy $\mathcal{A}^S$, by definition its decision tree is the same when it is executed on $B$ and on $B^S$; in particular, each node is reached with the same probability. Since $v_i \geq D^S(v_i)$, we have $\mathbb{P}(\mathcal{A}^S, B^S) \leq \mathbb{P}(\mathcal{A}^S, B)$. □

**Lemma 4.7.** *If $\mathcal{A}^*$ is an optimal policy, after a probed box yields a value $v^* \geq \theta$, $\mathcal{A}^*$ only opens boxes with indices at least $v^* \geq \theta$. Moreover, with probability at most $\epsilon$, any box with index at least $\theta$ contains a value at least $\theta$.*

[7] If there is no other opened box, $\mathcal{A}'$ may quit without taking anything, or take any box without inspection.

*Proof.* Since $\mathrm{OPT} \geq \max_i \mathbf{E}[v_i]$, after a box with value $v^* \geq \theta$ is opened, an optimal policy never exercises the option to take a box without inspection, and the (only) optimal policy is to follow the index-based strategy on the remaining boxes with indices at least $v^*$.

Recall from the proof of Theorem 2.1 that the profit of the index-based strategy is

$$\mathbf{E}\left[\max_i \kappa_i\right] \leq \mathrm{OPT},$$

where $\kappa_i = \min\{v_i, \tau_i\}$. Since $\mathrm{OPT} \leq \epsilon\theta$, by Markov inequality, with probability at $\epsilon$, $\max_i \kappa_i \geq \theta$. For a box $i$ with $\tau_i \geq v^* \geq \theta$, $v_i \geq \theta$ implies $\kappa_i \geq \theta$. Therefore, with probability at most $\epsilon$, any such box has value at least $\theta$. □

**Lemma 4.8.** *For any $S \subseteq [n]$ and $v \geq \theta$, we have $F(S, v) \geq W(S, v) \geq (1 - \epsilon) \cdot F(S, v)$.*

*Proof.* The first inequality is obvious. For the second, let $M_i$ be the event $i = \arg\max_{i \in S} \kappa_i - v$.[8] By definition,

$$\begin{aligned}
W(S, v) &= \mathbf{E}\left[(\max_{i \in S} \kappa_i - v)_+\right] \\
&= \sum_{i \in S} \mathbf{Pr}\left[M_i\right] \cdot \mathbf{E}\left[(\kappa_i - v)_+ \mid M_i\right] \\
&\geq \sum_{i \in S} \mathbf{Pr}\left[\forall j \neq i, \kappa_j \leq v \wedge \kappa_i > v\right] \cdot \mathbf{E}\left[(\kappa_i - v)_+ \mid M_i\right] \\
&= \sum_{i \in S} \mathbf{Pr}\left[\forall j \neq i, \kappa_j \leq v\right] \mathbf{Pr}\left[\kappa_i > v\right] \cdot \mathbf{E}\left[(\kappa_i - v)_+ \mid M_i\right] \\
&\geq \sum_{i \in S} (1 - \epsilon) \mathbf{Pr}\left[\kappa_i > v\right] \cdot \mathbf{E}\left[(\kappa_i - v)_+ \mid M_i\right] \\
&\geq \sum_{i \in S} (1 - \epsilon) \mathbf{Pr}\left[M_i\right] \cdot \mathbf{E}\left[(\kappa_i - v)_+ \mid M_i\right] \\
&= (1 - \epsilon) \sum_{i \in S} \mathbf{E}\left[(\kappa_i - v)_+\right] = (1 - \epsilon) \cdot F(S, v)
\end{aligned}$$

The second inequality follows from Lemma 4.7. □

**Claim 4.10.** *Any policy $\mathcal{A}^L$ for the L-PNOI instance $B^L$ can be executed on the PNOI instance $B$, with an expected payoff no less than that of the execution on $B^L$.*

*Proof.* Actions taken by $\mathcal{A}^L$ are valid actions in $B$: $a_i^0$ for opening box $i$, and $a_i^1$ for taking box $i$ without inspection. Since $D^L(v) \geq v$ for any $v$, a simple induction shows that, after taking the same sequence of actions, the state in $B^L$ is no less than the state in $B$, the latter being the largest value revealed so far. The marginal payoff of an action in $B$ is given by the value increase on top of its state, and therefore the same action yields weakly less marginal payoff in $B^L$.

□

**Lemma 4.12.** *For any $v \in \{\theta_1, \ldots, \theta_m\}$, $S \subseteq [n]$, let $W^L(S, v)$ be the expected marginal payoff of any quasi-index policy for $B^L$. We have $W^L(S, v) \geq (1 - \epsilon) \cdot F(S, v)$.*

[8] If there is a tie, break it lexicographically.

*Proof.* For any $i \in S$, if $\mathbf{E}[G^L(v, a_i^0)] > 0$, then $\tau_i > v \geq \theta$. Therefore, by Lemma 4.7, each such box $i$ is opened by the quasi-index policy with probability at least $1 - \epsilon$. The marginal payoff of the policy is therefore at least $(1-\epsilon)\sum_{i\in S}[\mathbf{E}[(v_i - v)_+] - c_i]_+ = (1-\epsilon)\sum_{i\in S}(\kappa_i - v)_+ = (1-\epsilon)F(S, v)$. □

**Lemma 4.13.** *Let* $\mathrm{OPT}$ *and* $\mathrm{OPT}^L$ *be the optimal expected payoff of the PNOI instance* $B$ *and the L-PNOI instance* $B^L$*, respectively. Then* $\mathrm{OPT} \geq \mathrm{OPT}^L \geq (1 - O(\epsilon)) \cdot \mathrm{OPT}$.

*Proof.* The first inequality results from Claim 4.10. We only need to show $\mathrm{OPT}^L \geq (1-\epsilon)\mathrm{OPT}$. Let $\mathcal{A}^*$ be an optimal policy for $B$, and let $T^*$ be its decision tree. Let $Q$ be the set of nodes in $T^*$ where the policy first sees a revealed value at least $\theta$. Formally, $Q \coloneqq \{u \in T^* : I_u \geq \theta, I_{\mathrm{Fa}(u)} < \theta\}$, where $\mathrm{Fa}(u)$ is the father node of $u$.

Recall that $\Phi(u)$ denotes the probability of $\mathcal{A}^*$ reaching a node $u$ in $T^*$. Let $S_u$ be the set of boxes not opened yet when node $u$ is reached. By equation (13), we have

$$\begin{aligned}\mathbb{P}(\mathcal{A}) &= \sum_{u\in T^*} G(u)\cdot\Phi(u) = \sum_{\substack{u\in T^*\\ I_u<\theta}} G(u)\cdot\Phi(u) + \sum_{u\in Q} W(S_u, I_u)\cdot\Phi(u)\\ &\leq \sum_{\substack{u\in T^*\\ I_u<\theta}} G(u)\cdot\Phi(u) + \sum_{u\in Q} F(S_u, I_u)\cdot\Phi(u). \end{aligned} \tag{14}$$

Let $\mathcal{A}^L$ be the following policy on $B^L$: $\mathcal{A}^L$ copies the behavior of $\mathcal{A}$ as long as the value seen so far is at most $\theta$; once its state reaches $\theta$, $\mathcal{A}^L$ implements a quasi-index policy.

Let $T^L$ be the decision tree of $\mathcal{A}^L$. To distinguish the notations from those in PNOI, we denote a typical node in $T^L$ as $u^L$, write $I^L_{u^L}$ as its state, $S^L_{u^L}$ the set of boxes not opened yet, $\Phi^L(u^L)$ the probability of $\mathcal{A}^L$ reaching $u^L$, and $\mathbb{P}^L(\mathcal{A}^L)$ the expected payoff of $\mathcal{A}^L$ on $B^L$. Similarly, let $Q^L$ be the set of nodes in $T^L$ with states at least $\theta$ and whose parents have states below $\theta$.

$T^L$ is idential to $T^*$ from the root down to the nodes in $Q^L$; each node $u \in T^*$ with $I_u \leq \theta$ has an image $R(u) \in T^L$, with $\mathcal{A}^L$ reaching $R(u)$ with probability $\Phi(u)$, taking the same action as $\mathcal{A}^*$ does on $u$, and making expected profit. For a node $u \in Q$, let $R(u)$ be the node in $Q^L$ such that when $\mathcal{A}^*$ reaches $u$, $\mathcal{A}^L$ reaches $R(u)$. Then $D^L(I_u) = I^L_{R(u)}$, and $S^L_{R(u)} = S_u$; for any $u^L \in Q^L$, $\Phi^L(u^L) = \sum_{u\in R^{-1}(u^L)}\Phi(u)$. Inheriting the notation from Lemma 4.12, the expected additional utility $\mathcal{A}^L$ makes after it reaches a node $u^L \in Q^L$ is $W^L(S^L_{u^L}, I^L_{u^L})$. By Lemma 4.12 and the definition of $D^L(\cdot)$, for any $u \in R^{-1}(u^L)$,

$$\begin{aligned} W^L(S^L_{u^L}, I^L_{u^L}) &\geq (1-\epsilon)\cdot F(S^L_{u^L}, I^L_{u^L}) = (1-\epsilon)\cdot F(S_u, D^L(I_u))\\ &\geq (1-\epsilon)\cdot(F(S_u, I_u) - \epsilon\cdot\mathrm{OPT})\\ &= F(S_u, I_u) - O(\epsilon)\cdot\mathrm{OPT}. \end{aligned}$$

Using inequality (14), we have

$$\begin{aligned}\mathbb{P}(\mathcal{A}) - \mathbb{P}^L(\mathcal{A}^L) &\leq \sum_{u\in Q}\Phi(u)F(S_u, I_u) - \sum_{u^L\in Q^L}\Phi^L(u^L)W^L(S^L_{u^L}, I^L_{u^L})\\ &= \sum_{u\in Q}\Phi(u)\cdot\Big(F(S_u, I_u) - W^L(S^L_{R(u)}, I^L_{R(u)})\Big) \leq O(\epsilon)\cdot\mathrm{OPT}\end{aligned}$$

Therefore $\mathrm{OPT}^L \geq \mathbb{P}^L(\mathcal{A}^L) \geq (1 - O(\epsilon))\cdot\mathrm{OPT}$. □

**Theorem 4.5.** *For any fixed constant* $\epsilon > 0$*, there is a polynomial-time algorithm for PNOI that computes a policy with an expected payoff at least* $(1 - O(\epsilon))\cdot\mathrm{OPT}$.

*Proof.* Given a PNOI instance $B$, denote the corresponding $S$-discretized instance as $B^S$. By Lemma 4.6, we know $\mathrm{OPT}^S \geq (1 - O(\epsilon)) \cdot \mathrm{OPT}$, where $\mathrm{OPT}^S$ is the optimal expected profit of $B^S$.

We further discretize the values above $\theta$ in $B^S$ using the discretization technique from Section 4.2.2. Let $B^{SL}$ be the resulting instance. From Lemma 4.13, we have $\mathrm{OPT}^{SL} \geq (1-O(\epsilon))\cdot\mathrm{OPT}^S = (1-O(\epsilon))\cdot\mathrm{OPT}$, where $\mathrm{OPT}^{SL}$ is the optimal expected profit of $B^{SL}$.

The number of states in $B^{SL}$ is $O(1/\epsilon)$. So we could apply Theorem 4.1 to $B^{SL}$ and get a policy $\mathcal{A}^L$ with $\mathbb{P}^L(\mathcal{A}^L, B^{SL}) \geq (1 - O(\epsilon)) \cdot \mathrm{OPT}^{SL}$.

Finally, using Lemma 4.6 and Claim 4.10, we prove that $\mathbb{P}(\mathcal{A}_L, B) \geq (1 - O(\epsilon)) \cdot \mathrm{OPT}$. □