

Designing Exploration Contracts

Martin Hoefer   

Department of Computer Science, RWTH Aachen University, Germany

Conrad Schecker  

Institute for Computer Science, Goethe University Frankfurt, Germany

Kevin Schewior   

Department of Mathematics and Computer Science, University of Cologne, Germany
University of Southern Denmark, Odense, Denmark

Abstract

We study a natural application of contract design in the context of sequential exploration problems. In our principal–agent setting, a search task is delegated to an agent. The agent performs a sequential exploration of n boxes, suffers the exploration cost for each inspected box, and selects the content (called the *prize*) of one inspected box as outcome. Agent and principal obtain an individual value based on the selected prize. To influence the search, the principal a-priori designs a contract with a non-negative payment to the agent for each potential prize. The goal of the principal is to maximize her expected reward, i.e., value minus payment. Interestingly, this natural contract scenario shares close relations with the *Pandora’s Box* problem.

We show how to compute optimal contracts for the principal in several scenarios. A popular and important subclass is that of *linear* contracts, and we show how to compute optimal linear contracts in polynomial time. For general contracts, we obtain optimal contracts under the standard assumption that the agent suffers cost but obtains value only from the transfers by the principal. More generally, for general contracts with non-zero agent values for outcomes we show how to compute an optimal contract in two cases: (1) when each box has only one prize with non-zero value for principal and agent, (2) for i.i.d. boxes with a single prize with positive value for the principal.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory and mechanism design

Keywords and phrases Exploration, Contract Design, Pandora’s Box Problem

Digital Object Identifier 10.4230/LIPIcs.STACS.2025.50

Related Version *Full Version*: <https://arxiv.org/abs/2403.02317>

Funding *Martin Hoefer*: Supported by DFG Research Unit ADYN (project number 411362735) and DFG grant Ho 3831/9-1 (project number 514505843).

Kevin Schewior: Supported by the Independent Research Fund Denmark, Natural Sciences, grant DFF-0135-00018B.

1 Introduction

In many real-world situations, e.g., the search for a job candidate or house, a decision maker is faced with the choice between different alternatives of a-priori unknown value, which can be explored at some cost. Due to missing qualifications or time constraints, in many markets such exploration tasks are not executed directly by the decision maker. Rather, a decision maker (called *principal* \mathcal{P} in the following) can delegate the exploration to an *agent* \mathcal{A} , whose incentives are potentially misaligned with that of \mathcal{P} .

Indeed, suppose the principal intends to buy a house. They might know an inspection cost and have some prior knowledge about the value of each house. However, they might not be qualified to determine the exact market value by inspecting it. They decide to delegate the search to a real-estate agent. The agent can have an individual valuation for each house



© Martin Hoefer, Conrad Schecker, and Kevin Schewior;
licensed under Creative Commons License CC-BY 4.0

42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025).

Editors: Olaf Beyersdorff, Michał Pilipczuk, Elaine Pimentel, and Nguyễn Kim Thăng;

Article No. 50; pp. 50:1–50:19



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



(e.g., a provision that is paid internally for selling the house). This individual value might also depend on the condition of the house, which is revealed only after inspection. Similar situations arise, e.g., when the principal wants to invest in a financial product and delegates this search to a financial agent.

In these settings, as a consequence of delegating the search, \mathcal{P} can merely observe the outcome of the search, not the other actions taken by \mathcal{A} . This leads to a possibly undesirable outcome for \mathcal{P} , even when they have the power to commit to accepting or rejecting certain outcomes [5].

A natural approach to align the incentives of an agent with the goals of the principal in such hidden-action settings are *contracts*. A contract allows for utility transfers to be made from the principal to the agent as a function of the outcome of the delegated task. While contract theory is a well-established [9] and celebrated [35] area in economics, the interest in *algorithmic* contract design has surged only relatively recently, see e.g., [14, 17, 20, 21, 27].

In this work, we initiate the study of algorithmic contract design for exploration problems. Specifically, we consider the following problem (for a fully formal definition, see Section 2): There are n boxes, the i -th of which contains a prize, drawn independently from a known probability distribution, and can be opened by paying a known exploration cost c_i . The j -th prize from box i has some utilities a_{ij} and b_{ij} for \mathcal{A} and \mathcal{P} , respectively. Based on that information, \mathcal{P} commits to a contract, specifying a transfer t_{ij} for each prize, where we assume non-negative transfers (limited liability). Given such a contract, \mathcal{A} picks an adaptive search strategy that allows to optimize the trade-off between value of the selected prize (for \mathcal{A}) and total exploration cost. \mathcal{A} uses the strategy to sequentially open boxes by paying the exploration cost. Depending on the observed prize, it determines the next box to open or to stop by selecting a prize from an open box. Assuming \mathcal{A} selects prize j from box i , the resulting final utility of \mathcal{P} is $b_{ij} - t_{ij}$. The goal is to efficiently compute a contract that maximizes the expectation of the latter quantity. Here we assume that ties in the optimization of \mathcal{A} are broken in favor of \mathcal{P} , which is a standard assumption that makes the maximization problem well-defined.

For the problem of computing optimal contracts, there is a solution by linear programming that runs in time polynomial in the number of outcomes and the number of actions of the agent (cf., e.g., [20]). Note that this result does not yield a polynomial-time algorithm for our problem since the action space is extremely succinctly represented – its size is lower-bounded by the number of opening orders, $n!$.

Our Contribution. We show that, in a variety of natural cases, optimal contracts can be computed in polynomial time (such as, e.g., optimal *linear* contracts, or optimal contracts when the agent has no intrinsic value). While our results leave open the question whether computing optimal contracts efficiently is always possible in general, our insights suggest that doing so is a highly non-trivial task and represents one (of several) very interesting avenue for future research.

Notably, \mathcal{A} faces an exploration problem known in the literature as the *Pandora's Box Problem*. It was famously proposed and analyzed by Weitzman [39], who characterized optimal search strategies by a simple greedy rule. Starting from [33], this problem has recently received a lot of attention at the intersection of economics and computer science (see, e.g., [6, 8, 10, 25, 26, 38] and a recent survey [7]). We reformulate Weitzman's characterization for \mathcal{A} 's task: The *fair cap* of box i is a value such that the expected amount of $a_{ij} + t_{ij}$ exceeding the fair cap is precisely c_i . The boxes are considered in non-increasing order of fair caps. If the best $a_{ij} + t_{ij}$ (initially 0) observed so far is larger than the next fair cap, the next box is opened; if it is smaller, a corresponding prize is accepted. If neither is the case, \mathcal{A} is indifferent.

The first observation we make is that Weitzman’s policy does *not* solve \mathcal{A} ’s problem entirely. The reason is that \mathcal{A} breaks ties in favor of \mathcal{P} — recall that this is the standard assumption that makes the maximization problem well-defined. Note that the number of boxes that share some given fair cap φ may be linear in n , leading to a number of possible ways of choosing the order that is exponential in n . Indeed, the order of such boxes may matter to \mathcal{P} since different orders may lead to stopping with prizes that lead to vastly different utilities for \mathcal{P} . A complicating factor is that ties whether to select a prize with agent utility equal to φ have to be broken simultaneously. Interestingly, we show that the problem of breaking ties can be reduced – in polynomial time but in a non-obvious way – to solving *different instances* of the classic Pandora’s box problem.

With a solution of \mathcal{A} ’s problem at hand, the main trade-off inherent to an optimal contract is as follows: Transfers manipulate the fair caps, the resulting exploration order, and the acceptance decisions of \mathcal{A} in such a way that leads to a more favorable outcome for \mathcal{P} – but transfers are costly for \mathcal{P} which is unfavorable.

We start by considering *linear contracts* [20, 30], a class of simple and very intuitive contracts that has received a lot of attention due to its widespread use in applications. The agent receives a constant fraction of the principal’s revenue as commission. Formally, a linear contract is represented by a number $\alpha \in [0, 1]$, which defines transfers $t_{ij} = \alpha \cdot b_{ij}$.

To compute linear exploration contracts, we identify a polynomial-time computable set of *critical* values of α at which the set of \mathcal{A} -optimal policies (i.e., the set of policies that satisfy the above description) changes. By reasoning about the behavior of \mathcal{P} ’s utility between any two such values, we can argue that the optimal contract needs to be a critical value. This type of argument is reminiscent of arguments from the recent literature (e.g., [16, 17, 22]), but here it requires special care. Due to the tie breaking in favor of \mathcal{P} , which – as discussed above – is a key property here, we need to keep track of the *set* of \mathcal{A} -optimal policies.

For standard contract design, the authors in [20] describe a class of instances in which the restriction to linear contracts leads to a multiplicative loss of n in \mathcal{P} ’s achievable utility, where n is the number of actions. In this class of instances, action i with cost c_i deterministically leads to an individual outcome with value R_i . We can create one box for each action i , with opening cost c_i and a single outcome with agent utility 0 and principal utility R_i . This leads to a straightforward one-to-one correspondence between contracts in the two instances that preserves utilities and linearity. Thus, the multiplicative loss of n transfers to our model, which motivates our search for optimal *general* exploration contracts.

We first consider general contracts under the standard assumption that $a_{ij} = 0$, i.e., that the agent has no intrinsic motivation to explore any boxes and is only motivated by payments from \mathcal{P} . In contract design, this assumption is usually without loss of generality, since arbitrary intrinsic agent values $a_{ij} \neq 0$ can be accounted for with action costs to obtain an equivalent instance with agent values $a_{ij} = 0$. However, for the exploration contracts we consider, such an adjustment changes the costs of *exploration strategies* (rather than individual boxes). This substantially changes the structure of the problem. Indeed, as we see below, optimal policies for the problem with non-zero agent value have substantially different properties than the ones for zero value.

Clearly, the utility that \mathcal{P} can extract by solving the exploration problem themselves (i.e., ignoring the agent and paying the cost themselves) is a straightforward baseline; no policy can extract more value for \mathcal{P} . Solving the problem from \mathcal{P} ’s perspective yields fair caps $\varphi_i^{\mathcal{P}}$. If all $a_{ij} = 0$, then by transferring any value exceeding $\varphi_i^{\mathcal{P}}$ to \mathcal{A} , the \mathcal{P} -optimal policy is adopted by \mathcal{A} , without a loss in utility for \mathcal{P} . This implies, in particular, that delegating the exploration to \mathcal{A} using an optimal contract yields the same utility for \mathcal{P} as if she performed the exploration on her own. Formally, the idea is somewhat inspired by [33] for the original Pandora’s Box problem but different.

An orthogonal special case that has been considered in the literature (see, e.g., [5]) is that of *binary boxes*. That is, each box contains one of only two prizes, and one of them has 0 value for both \mathcal{P} and \mathcal{A} while the other one is arbitrary. Without any payments, the fair cap (from \mathcal{A} 's perspective again) of each box is its *basic fair cap*. The principal now faces the question of whether to lift (by transfers) the basic fair caps of favorable boxes above (or up to) the fair caps of less favorable boxes. While the latter is true in general, the special structure of binary boxes allows us to compute an optimal exploration contract by considering boxes in non-increasing order of their basic fair caps and moving their fair cap *greedily*. Note that this algorithm alone does not suffice to tackle more general cases. Indeed, when boxes are not binary, it may be profitable for the principal to choose different payments for two boxes with the same distribution and the same fair caps – a condition we encounter in the next and final case.

The final case we consider has a more general value structure, but we compromise on the boxes having different distributions. Specifically, we consider instances with only a single positive prize for \mathcal{P} , and all distributions are identical. Still, transfers can be made depending on the box from which the prize originates. Again, the complexity is reduced by the fact that only a single payment per box has to be decided. Through an intricate sequence of exchange arguments and the help of continuization, we establish that there is an optimal contract with a simple structure: (i) In a first phase, the prize with positive principal value gets immediately accepted, and all transfers are identical; (ii) all transfers for boxes in the second phase are also identical. These conditions allow to find the optimal contract by enumerating a polynomial-sized set of contracts.

In this final case, we also give a family of instances in which both the above phases exist and have substantial lengths in any optimal solution. Hence, perhaps surprisingly, such a relatively simple case already has optimal contracts with a fairly complicated structure.

Overview. After a review of related work in the subsequent section, we formally introduce our problem in Section 2 along with preliminary observations. Section 3 treats linear contracts. Section 4 deals with general contracts in several special cases (no agent value in Section 4.1 and binary boxes in Section 4.2). General contracts for i.i.d. boxes with a single positive prize are discussed in Section 4.3. Missing proofs are provided in the full version of the paper.

1.1 Related Work

Contract design has recently gained a lot of attention from a computational point of view. Duetting et al. [20] advocated the study of linear contracts as an alternative to the more complicated (and sometimes unintuitive) general contracts. They show robustness results and give parameterized approximation guarantees w.r.t. optimal (general) contracts. For settings in which the outcome space is succinctly represented, hardness results are shown in [21]. In a similar but different approach, *combinatorial contracts* [17] let the agent choose a subset of actions which stochastically determines the (binary) outcome. Important recent work in the area of combinatorial contracts includes [22], [16]. Contracts were also studied with multiple agents [16, 18], ambiguity [19], and private types [2] with connections to classic mechanism design [1]. More generally, contracts have been of interest in specific application domains, such as classification in machine learning [36].

Most closely related is prior work by Postl [34], who analyzes our contract-design problem for two boxes without intrinsic agent value. The paper provides an explicit characterization of the chosen box in the optimal contract, but neither considers algorithmic aspects nor the reduction to the standard Pandora's Box problem we discover in Sec. 4.1. Concurrent

to our work, Ezra et al. [23] study the special case without agent value. In this special case, they also give a polynomial-time algorithm for linear contracts (without discussing the issues of tie-breaking in favor of \mathcal{P}). For general contracts, the paper studies a technical approach when there is a constant number of prizes. Finally, the authors show NP-hardness for correlated distributions in the boxes.

Delegation is a related approach in the principal–agent framework and also received a lot of attention in the economics literature starting from seminal work of Holstrom [31]. Computational aspects of delegation are receiving interest recently, initiated by Kleinberg and Kleinberg [32]. In these models, the principal \mathcal{P} delegates a (search) task to an agent \mathcal{A} , again with potentially different interests, who has to inspect alternatives and *propose* an observed prize to \mathcal{P} . Instead of committing to a contract with payments, \mathcal{P} limits the set of prizes she is willing to accept upon proposal by \mathcal{A} . A prominent objective is to find good acceptance policy for \mathcal{P} and bound their performance by the *delegation gap*, which measures the multiplicative loss in utility for \mathcal{P} in comparison to the case when \mathcal{P} performs the (undelegated) search problem themselves. On a technical level, there are close connections to prophet inequalities [32] and online contention resolution schemes, which were established in different variants, such as multiple agents [37], stochastic probing [4, 5], as an online problem [11], or with limited information about agent utilities [13]. Perhaps most related is a version of the problem studied in [5]. Here, a principal delegates an instance of the Pandora’s Box problem by committing to a set of acceptable outcomes. An agent has to perform the costly inspection of alternatives. This standard delegation setting has some obvious limitations, e.g., the agent would never perform an inspection if his expected intrinsic value does not cover the inspection cost. A model variant with transfers is also studied in [5] in which the principal can reimburse the agent for exploration costs. Crucially, this is impossible in *hidden-action principal-agent* settings that we consider, where \mathcal{P} simply cannot directly reimburse \mathcal{A} for his expenses, since she cannot observe which inspections have been performed by \mathcal{A} .

In a more general context, online optimization in principal–agent settings has generated significant interest recently, e.g., in game-theoretic versions of the Pandora box problem [15], general optimal stopping problems [28, 29], or with unknown agent utilities [3, 12, 24, 40].

2 Preliminaries

There are two parties, a principal \mathcal{P} and an agent \mathcal{A} , and n boxes. Each box $i \in [n]$ has an opening cost $c_i \geq 0$ and contains one of m possible prizes. Prize $j \in [m]$ in box i yields a value-pair (a_{ij}, b_{ij}) , where $a_{ij} \geq 0$ is the value for \mathcal{A} and $b_{ij} \geq 0$ the value for \mathcal{P} . For each box $i \in [n]$, the prize in the box is distributed independently according to a known prior. We denote by $p_{ij} \geq 0$ the probability that prize j is in box i .

\mathcal{P} would like to motivate \mathcal{A} to open and inspect boxes in order to find and select a good prize (for \mathcal{P}). \mathcal{P} cannot observe the actions taken by \mathcal{A} (i.e., which boxes were opened in which order), only the final selected box with the prize inside is revealed. Instead, \mathcal{P} can commit to an *exploration contract* (or simply, *contract*) T : For each selected prize (i, j) , she specifies an amount $t_{ij} \in [0, b_{ij}]$ ¹ that she pays to \mathcal{A} . Then the final utility of \mathcal{P} is $b_{ij} - t_{ij}$.

¹ We make the natural assumption that $t_{ij} \leq b_{ij}$, i.e., every transfer in the contract is bounded by the prize value of \mathcal{P} . For all scenarios we study here (linear contracts, no agent value, binary boxes, i.i.d. with single positive prize for \mathcal{P}) it is straightforward to see that this assumption is w.l.o.g. – there is an optimal contract that satisfies $t_{ij} \leq b_{ij}$ for all $(i, j) \in [n] \times [m]$. It is a very interesting open problem to prove that this holds beyond the cases we study here.

Our model is an instance of the standard principal-agent model, where the hidden action of the agent \mathcal{A} is the policy to inspect and select from the boxes. Specifically, the task of \mathcal{A} is to open and inspect boxes to find and select at most one prize from an opened box. Formally, a *policy* specifies in every situation, based on the previously seen prizes, which box to open next, if any, or otherwise which prize to select from an opened box. Given a contract T , she is facing the classic Pandora's Box problem [39]: She obtains a utility of $a_{ij} + t_{ij}$ for the selected prize (0 if no prize is selected) minus the opening cost c_i for all opened boxes. Weitzman [39] shows that every optimal policy of \mathcal{A} that maximizes his expected total utility (called \mathcal{A} -optimal policy in the following) is in the form of the following index policy: For each box $i \in [n]$ compute an index or *fair cap* φ_i such that

$$\sum_{j \in [m]} p_{ij} \max\{0, a_{ij} + t_{ij} - \varphi_i\} = c_i.$$

The boxes are considered in non-increasing order of fair caps. Suppose box i is the next unopened box in this order, and the best prize \mathcal{A} has found thus far is v . If no box has been opened, then $v = 0$, and otherwise $v = a_{i'j'} + t_{i'j'}$ for some (i', j') where i' is a box that has already been opened, and prize j' has been found in it. There are three cases: (1) $v < \varphi_i$: \mathcal{A} opens box i ; (2) $v = \varphi_i$: \mathcal{A} is indifferent between opening box i and stopping opening boxes; and (3) $v > \varphi_i$: \mathcal{A} stops opening boxes. Note that any box i with $\varphi_i < 0$ is never opened. For each $(i, j) \in [n] \times [m]$ we define the capped value $\kappa_i = \min\{a_{ij} + t_{ij}, \varphi_i\}$. A policy as described above always selects the prize in box $i^* \in \arg \max_i \kappa_i$, i.e., the prize with the largest capped agent value.

Even under these conditions, there may still be several \mathcal{A} -optimal policies. Specifically,

- (i) the choice of φ_i is not unique when $c_i = 0$,
- (ii) the non-decreasing order of fair caps may not be unique,
- (iii) in the case $v = \varphi_i$ above, the choice of whether to stop or continue is not unique,
- (iv) among the observed prizes, the prize (i, j) maximizing $a_{ij} + t_{ij}$ may not be unique.

We assume that \mathcal{A} breaks ties in favor of \mathcal{P}^2 . Among the set of \mathcal{A} -optimal policies, she selects a \mathcal{P} -optimal one, i.e., a policy that maximizes the expected utility for \mathcal{P} . (Note that such a policy may be vastly different from an \mathcal{P} -optimal policy among *all* policies.) We call a policy that is selected in this way an *optimal policy (under contract T)*.

In case (iv), it is clear how to resolve the ambiguity in favor of \mathcal{P} : Simply select (i, j) maximizing $b_{ij} - t_{ij}$. The other cases, however, can give rise to a very large number of potential policies. Interestingly, in the full version of the paper we show that it is always possible to implement the tie-breaking in polynomial time and find an optimal policy under contract T . We prove this result through a reduction to the original Pandora's Box problem.

► **Theorem 1.** *Given a contract T , an optimal policy can be computed in polynomial time.*

We study contracts for \mathcal{P} that steer the index policy executed by \mathcal{A} towards good outcomes for \mathcal{P} . We explore both linear and general contracts. A *linear* contract is given by a single number $\alpha \in [0, 1]$, and $t_{ij} = \alpha \cdot b_{ij}$. Linear contracts are popular because of their simplicity, but they often suffer from substantial limitations of the achievable revenue. As such, we also explore *general* contracts, in which we only require every payment to be non-negative $t_{ij} \geq 0$ for every $i \in [n]$ and $j \in [m]$. A (linear) contract T that, among all possible (linear) contracts, achieves maximum expected utility for \mathcal{P} when \mathcal{A} executes an optimal policy under T , is called *optimal (linear) contract*.

² This is a standard assumption in bi-level problems. On a technical level, it ensures that the optimization problem of finding an optimal contract for \mathcal{P} is well-defined.

3 Optimal Linear Contracts

In this section we consider linear contracts. Recall that these are contracts characterized by a single $\alpha \in [0, 1]$ such that $t_{ij} = \alpha \cdot b_{ij}$ for all $(i, j) \in [n] \times [m]$. Our result is the following.

► **Theorem 2.** *An optimal linear exploration contract can be computed in polynomial time.*

Let $\alpha_1, \alpha_2 \in [0, 1]$ such that the set of \mathcal{A} -optimal policies (i.e., index policies) is the same for all $\alpha \in [\alpha_1, \alpha_2]$. Consider the expected utility of \mathcal{P} according to an optimal policy (i.e., an \mathcal{P} -optimal policy among this set of \mathcal{A} -optimal policies) as a function of α within the interval $[\alpha_1, \alpha_2]$. The following straightforward observation implies that this function is a linear and non-increasing function of α .

► **Observation 3.** *For every policy, there exists a constant c such that the expected utility of \mathcal{P} as a function of α is $(1 - \alpha) \cdot c$.*

Hence, within $[\alpha_1, \alpha_2]$, an optimal contract is α_1 . To find a polynomial-time algorithm for computing the global optimum α , it therefore suffices to show that, in polynomial time, one can find a partition of the interval $[0, 1]$ into (polynomially many) subintervals such that, in each subinterval, the set of \mathcal{A} -optimal policies is constant. (Strictly speaking, we will encounter a technicality at the endpoints of the subintervals.) In the following, we will define a (polynomial-time computable) set of *critical values* in $[0, 1]$ such that, in an interval (strictly) between any two consecutive critical values, the set of \mathcal{A} -optimal policies is constant.

Towards this definition, for any given $\alpha \in [0, 1]$ and every $(i, j) \in [n] \times [m]$, let $v_{ij}(\alpha) := a_{ij} + \alpha \cdot b_{ij}$ denote the value of prize j in box i for \mathcal{A} . Furthermore, let $\varphi_i(\alpha)$ denote the (unique) fair cap of box $i \in [n]$ with $c_i > 0$ as a function of α , i.e., it holds that

$$\sum_{j \in [m]} p_{ij} \max \{0, v_{ij}(\alpha) - \varphi_i(\alpha)\} = c_i$$

for all $\alpha \in [0, 1]$. Note that $\varphi_i(\alpha)$ is a continuous function of α in $[0, 1]$.

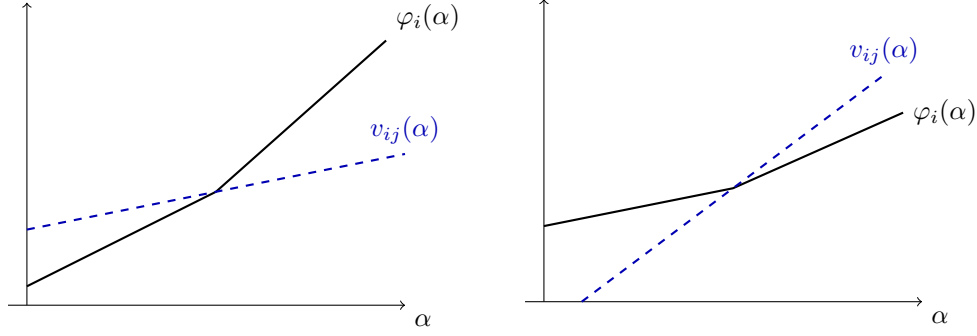
We call $\alpha \in [0, 1]$ a critical value if $\alpha \in \{0, 1\}$ or one of the following properties holds:

- (a) There exist boxes $i, i' \in [n]$ with $c_i > 0, c_{i'} > 0$ such that the order of their fair caps changes at α . Formally, $\varphi_i(\alpha) = \varphi_{i'}(\alpha)$, and there exists $\varepsilon > 0$ such that $\varphi_i(\alpha') \neq \varphi_{i'}(\alpha')$ for all $\alpha' \in (\alpha - \varepsilon, \alpha)$ or for all $\alpha' \in (\alpha, \alpha + \varepsilon)$.
- (b) There exist $i, i' \in [n]$ with $c_i > 0, c_{i'} > 0$ and $j \in [m]$ such that the order between the value of prize (i, j) for \mathcal{A} and the fair cap of box i' changes. Formally, $v_{ij}(\alpha) = \varphi_{i'}(\alpha)$, and there exists $\varepsilon > 0$ such that $v_{ij}(\alpha') \neq \varphi_{i'}(\alpha')$ for all $\alpha' \in (\alpha - \varepsilon, \alpha)$ or for all $\alpha' \in (\alpha, \alpha + \varepsilon)$.
- (c) There exists $i \in [n]$ with $c_i > 0$ such that $\varphi_i(\alpha) = 0$.

Observe that the critical values indeed have the property that, for any two such values α_1^c, α_2^c , the set of \mathcal{A} -optimal policies is constant within (α_1^c, α_2^c) . Note that any \mathcal{A} -optimal policy is also \mathcal{A} -optimal under both α_1^c and α_2^c , but there are potentially additional \mathcal{A} -optimal policies under α_1^c or under α_2^c . Together with Observation 3, this implies that the optimal contract is a critical value.

The following auxiliary lemma supports the analysis of the number of critical values.

► **Lemma 4.** *For each $i \in [n]$ with $c_i > 0$, the fair cap $\varphi_i : [0, 1] \rightarrow \mathbb{R}$ is a monotone convex piece-wise linear function with at most $2m + 1$ linear segments.*



■ **Figure 1** Two cases for an intersection between v_{ij} and φ_i . Left: The slope of v_{ij} was less than slope of φ_i before the intersection. By definition, it contributed to the weighted average slope that defines the slope of φ_i . After the intersection, it does not contribute anymore, and the weighted average only increases. Right: The slope of v_{ij} was greater than slope of φ_i before the intersection. Symmetrical arguments.

Proof. For any $\alpha \in [0, 1]$, it holds $\sum_{j \in [m]} p_{ij} \max\{0, v_{ij}(\alpha) - \varphi_i(\alpha)\} = c_i$ by definition. An equivalent formulation would be

$$\varphi_i(\alpha) = \frac{\sum_{j \in S_i(\alpha)} p_{ij} v_{ij}(\alpha) - c_i}{\sum_{j \in S_i(\alpha)} p_{ij}},$$

where $S_i(\alpha) := \{j \in [m] : v_{ij}(\alpha) \geq \varphi_i(\alpha)\}$. Hence, $\varphi_i(\alpha)$ is the weighted average (minus offset c_i) of all $v_{ij}(\alpha)$ that are in $S_i(\alpha)$, i.e., above $\varphi_i(\alpha)$. For a fixed set $S_i(\alpha)$, this means that the fair cap φ_i is linear in α , as every v_{ij} is linear in α . Therefore, the slope of φ_i only changes at intersections with some v_{ij} , where the set $S_i(\alpha)$ changes. We argue now that the fair cap can only increase at those intersections (cf. Fig. 1).

For every $(i, j) \in [n] \times [m]$ and $\alpha \in (0, 1]$ with $v_{ij}(\alpha) = \varphi_i(\alpha)$, but $v_{ij}(\alpha - \varepsilon) \neq \varphi_i(\alpha - \varepsilon)$ for some $\varepsilon > 0$, there are two cases:

1. $v_{ij}(\alpha - \varepsilon) > \varphi_i(\alpha - \varepsilon)$. Then $j \in S_i(\alpha - \varepsilon)$, but $j \notin S_i(\alpha + \varepsilon')$ for some $\varepsilon' > 0$. Hence the slope of φ_i was greater than the slope of v_{ij} before the intersection at α . The new weighted average of affine functions that are above the fair cap can only increase when v_{ij} is not contributing to that average anymore.
2. $v_{ij}(\alpha - \varepsilon) < \varphi_i(\alpha - \varepsilon)$. Then $j \notin S_i(\alpha - \varepsilon)$, but $j \in S_i(\alpha)$. Hence the slope of φ_i was less than the slope of v_{ij} before the intersection at α . The new weighted average of affine functions that are above the fair cap can only increase when v_{ij} is starting to contribute to that average.

Thus the slope of $\varphi_i(\alpha)$ is never decreasing when α increases, which makes φ_i convex. As v_{ij} is an affine function, every v_{ij} can intersect at most twice with φ_i . Therefore, there are at most $2m + 1$ linear segments of φ_i on the interval $[0, 1]$. ◀

According to Lemma 4, there are at most $\mathcal{O}(n)$ critical values induced by case (c). By the same lemma, there are at most $\mathcal{O}(nm)$ critical values for every box $i \in [n]$ with $c_i > 0$ induced by case (b), as φ_i is convex and there are nm affine functions v_{ij} , each of which intersects φ_i at most twice. Thus, $\mathcal{O}(n^2m)$ critical values in total are due to cases (b) and (c). Similarly, each linear segment of φ_i can intersect with another convex function $\varphi_{i'}$ at most twice. As there are at $\mathcal{O}(m)$ linear segments for $\alpha \in [0, 1]$ and n possible functions $\varphi_{i'}$, there are at most $\mathcal{O}(nm)$ such intersections for φ_i . Thus, case (a) induces at most $\mathcal{O}(n^2m)$ critical values. Overall, the number of critical values is polynomial in n and m , and the set of them can be computed in polynomial time.

By Theorem 1, we can compute an optimal contract for any given critical value in polynomial time. The expected utility of \mathcal{P} under this contract can also be computed in polynomial time. Thus, we can enumerate all critical values and take the best contract with respect to the expected utility for \mathcal{P} , implying Theorem 2.

4 Optimal General Contracts

4.1 No Intrinsic Agent Value

In the consideration of contract problems, it is often assumed that the agent has no intrinsic value (only cost) and receives benefit only via transfers from the principal. In this case, we have $a_{ij} = 0$ for all $(i, j) \in [n] \times [m]$. We show that under this assumption, we can compute an optimal contract in polynomial time.

► **Theorem 5.** *Suppose that $a_{ij} = 0$ holds for all $(i, j) \in [n] \times [m]$. Then an optimal exploration contract can be computed in polynomial time.*

Without any transfers from \mathcal{P} , \mathcal{A} has no intrinsic motivation to open any box – except the ones with inspection cost 0. Specifically, if the agent opens box $i \in [n]$, the (expected) payments from \mathcal{P} have to cover the inspection costs for \mathcal{A} , i.e., it holds that $\sum_{j \in [m]} p_{ij} t_{ij} \geq c_i$. As a direct consequence, \mathcal{P} cannot obtain more utility from a contract with \mathcal{A} over the one obtained by exploring boxes and paying inspection costs by herself (she can simply imitate \mathcal{A} 's behavior under the contract).

► **Definition 6.** *A given contract T implements a policy π if (1) π is optimal under T , and (2) $\sum_{j \in [m]} p_{ij} t_{ij} = c_i$ for all $i \in [n]$.*

As observed above, an optimal contract cannot yield more utility for \mathcal{P} than an optimal policy π^* that she can apply by herself (paying the costs herself). Hence, the next lemma implies Theorem 5.

► **Lemma 7.** *There exists a contract T^* that implements π^* .*

Proof. Recall that the following is an optimal policy π^* for \mathcal{P} (when she pays the costs herself). Define fair caps $\varphi_i^{\mathcal{P}}$ for each box $i \in [n]$ such that $\sum_{j \in [m]} p_{ij} \max\{0, b_{ij} - \varphi_i^{\mathcal{P}}\} = c_i$. Then \mathcal{P} opens boxes in non-increasing order of their fair caps as long as the best prize observed so far does not exceed the fair cap of the next box.

A contract T^* that implements π^* is given by payments $t_{ij} := \max\{0, b_{ij} - \varphi_i^{\mathcal{P}}\}$ for all $(i, j) \in [n] \times [m]$. Clearly, we have

$$\sum_{j \in [m]} p_{ij} t_{ij} = c_i, \quad \text{for all } i \in [n]. \quad (1)$$

Consequently, \mathcal{A} is faced with an instance of the Pandora's Box problem where her expected prize in each box i equals the opening cost. Thus, if \mathcal{A} applies the (\mathcal{A} -optimal) index policy for the emerging instance, the fair caps for \mathcal{A} are given by $\varphi_i^{\mathcal{A}} = 0$ for all $i \in [n]$.

This means that \mathcal{A} is entirely indifferent about the opening order. As long as there are only prizes with $t_{ij} = 0$, \mathcal{A} is also indifferent about stopping to open further boxes and selecting any previously observed prize at any point in time. Thus, \mathcal{A} 's behavior under these conditions can be assumed to be consistent with the one imposed by π^* .

Restrictions only arise from the fact that \mathcal{A} *must stop immediately* whenever a prize j with $t_{ij} > 0$ is drawn from box i , according to \mathcal{A} 's index policy. However, this is consistent with the behavior of the index policy π^* for \mathcal{P} : Note that $t_{ij} > 0$ if and only if $b_{ij} > \varphi_i^{\mathcal{P}}$. Hence, \mathcal{P} also stops when prize j is drawn from box i , because b_{ij} exceeds the fair cap of box i and, thus, the fair cap of the next box in the order.

Therefore, π^* is an optimal policy under contract T^* . Together with (1) this shows that T^* implements π^* . ◀

4.2 Binary Boxes

We study a subclass of the problem where every box $i \in [n]$ contains two prizes with positive probability. One of those prizes has value 0 for both \mathcal{P} and \mathcal{A} , and is called *0-prize*. The other prize (the *positive prize*) is denoted by the value pair (a_i, b_i) with $a_i, b_i \geq 0$ and has probability $p_i \in (0, 1]$. Consequently, the 0-prize is drawn from box i with probability $1 - p_i$. Note that there cannot be a positive payment for the 0-prize. Thus, payments for box i are given by a single $t_i \geq 0$, the payment for the positive prize.

Depending on a payment of $t \geq 0$, we define the fair cap $\varphi_i(t)$ of box i for \mathcal{A} by

$$\varphi_i(t) = t + a_i - \frac{c_i}{p_i}. \quad (2)$$

► **Lemma 8.** *Given a contract T , there is an optimal policy that has the following properties. \mathcal{A} opens boxes in order of the fair cap defined by (2). Boxes with the same fair cap are ordered according to the value $b_i - t_i$. \mathcal{A} accepts the first box with a positive prize (if any).*

Proof. As discussed in Section 2, the fair cap for boxes is unique whenever $c_i > 0$. For boxes with $c_i = 0$ we also assume that the fair cap is given as in (2). This is the smallest fair cap for this box and defers the inspection of box i to the latest point. Clearly, the agent is indifferent regarding this choice. Consequently, it is \mathcal{A} -optimal to stop immediately and accept as soon as \mathcal{A} opens a box with a positive prize in it, since the value is $a_i + t_i \geq \varphi_i(t_i) \geq \varphi_{i+1}(t_{i+1})$.

To argue that this behavior is also \mathcal{P} -optimal, suppose some box i^* with $c_{i^*} = 0$ gets a higher fair cap $\varphi_{i^*}(t_{i^*}) > a_{i^*} + t_{i^*}$ and gets opened earlier (in accordance with the resulting index policy of \mathcal{A}). With such fair cap for i^* , \mathcal{A} does not stop until all subsequent boxes i with fair cap $\varphi_i(t_i) > a_{i^*} + t_{i^*}$ are opened. As such, we can defer the opening of box i^* until after these boxes are opened. Now if there are several boxes with the same fair caps, it is optimal for \mathcal{P} if the opening is done in non-increasing order of $b_i - t_i$. This guarantees that there is no subsequent box with the same fair cap and a prize that has a better value for \mathcal{P} . Conversely, any box with the same fair cap and potentially better value for \mathcal{P} has been inspected before. As such, using minimal fair caps for boxes with $c_i = 0$, breaking ties w.r.t. $b_i - t_i$, and stopping as soon as a positive prize is observed is also \mathcal{P} -optimal. ◀

Let us define a *basic* fair cap of box i to be $\varphi_i(0)$. A box with negative basic fair cap would not be considered by \mathcal{A} unless the payment t_i is at least $c_i/p_i - a_i$, in which case the fair cap would be precisely 0. Note that boxes with prohibitively high cost $p_i(a_i + b_i) \leq c_i$ are w.l.o.g. never opened by \mathcal{A} under any contract T . In what follows, we exclude such boxes from consideration. For the remaining boxes $p_i(a_i + b_i) > c_i$, and, hence, $b_i > c_i/p_i - a_i =: \tilde{t}_i$. Whenever $\tilde{t}_i > 0$, the exploration cost exceeds the expected value for \mathcal{A} in box i . As such, a transfer of \tilde{t}_i is clearly necessary to motivate \mathcal{A} to inspect box i at all. We renormalize the box by adjusting the value of the positive prize to $(a_i + \tilde{t}_i, b_i - \tilde{t}_i)$. Then the basic fair cap becomes $\varphi_i(0) = 0$. Indeed, assuming $t_i \geq \tilde{t}_i$ is w.l.o.g.: Consider any contract T in which $t_i < \tilde{t}_i$ for all i in some non-empty set of boxes B , and the optimal policy π under T . Note

that choosing a contract T' where t_i is increased to \tilde{t}_i for such boxes does not hurt: This will increase the fair caps in B to 0. An \mathcal{A} -optimal policy π' for T' can be obtained from π by opening the boxes B in the end if no prize has been found yet, in any order. The resulting utility for \mathcal{P} in such cases is non-negative rather than 0 (as $\tilde{t}_i \leq b_i$); the utility in all other cases does not change.

In the remainder of the section, we consider the instance with renormalized boxes, i.e., each box i has a basic fair cap $\varphi_i(0) \geq 0$. Furthermore, we re-number the boxes and assume that the indices are assigned in non-increasing order of basic fair caps, i.e. $i < j \implies \varphi_i(0) \geq \varphi_j(0)$ for all $i, j \in [n]$. We break ties in the ordering w.r.t. non-increasing value of b_i .

Our next observation will allow us to restrict attention to the permutation and the fair caps in the policy of \mathcal{A} . By (2), fair caps are linear in payments. Clearly, for an optimal contract we strive to set smallest payments (and, hence, smallest fair caps) to induce a behavior of \mathcal{A} . For any given contract T , we say T implements an ordering σ of boxes if there is an optimal policy under T that considers boxes in the order of σ . For the reverse direction, we need a slightly more technical definition.

► **Definition 9.** For any given ordering σ of boxes, we define a contract $T(\sigma)$ as follows: For each position i (occupied with box $\sigma(i)$), let $j = \max(\arg \max_{j' \in \{i, i+1, \dots, n\}} \{\varphi_{\sigma(j')}(0)\})$ be the latest position of any subsequent box (including $\sigma(i)$ itself) with the highest basic fair cap of the subsequent boxes. In $T(\sigma)$, we assign payments to the unique positive prizes for the boxes on positions $i, i+1, \dots, j$ such that they all have a fair cap of exactly $\varphi_{\sigma(j)}(0)$. If $T(\sigma)$ has feasible payments and implements σ , we call $T(\sigma)$ the canonical contract for σ .

We have the following lemma concerning whether $T(\sigma)$ is the canonical contract for σ .

► **Lemma 10.** For any ordering σ , we can decide in polynomial time if $T(\sigma)$ is the canonical contract for σ (and compute it in this case). If $T(\sigma)$ is not the canonical contract for σ , then every contract T that implements σ is suboptimal for the given instance. Otherwise, the canonical contract has point-wise minimal payments among all contracts that implement σ .

Proof. The construction of $T(\sigma)$ and the subsequent feasibility check described in Definition 9 can be done in polynomial time. Note that if $T(\sigma)$ is the canonical contract for σ , it uses point-wise minimal payments for implementing σ : Having less payments for any box would directly contradict implementation of σ , since every optimal policy considers boxes in the order of their fair caps, which are lower bounded by the respective basic fair caps. Thus, if $T(\sigma)$ is not the canonical contract for σ due to infeasible payments, then implementing σ is impossible with any contract, because even higher payments would be necessary. If $T(\sigma)$ is not the canonical contract for σ , but the construction in Definition 9 defines feasible payments, it defines a canonical contract $T(\sigma')$ of some other ordering $\sigma' \neq \sigma$ using point-wise minimal payments. Note that σ' instead of σ only emerges because of tie-breaking in favor of \mathcal{P} , which implies that boxes with identical fair caps have to be considered in non-increasing order of $b_i - t_i$. Thus it could be possible to implement σ with another contract T , but additional payments would be required to remove (some of the) ties. However, since tie-breaking was already done in favor of \mathcal{P} , this contract T clearly yields less utility than $T(\sigma')$ and hence is not an optimal contract for the given instance. ◀

In the following, we will discuss how to compute the fair caps for the policy of \mathcal{A} in an optimal contract. By Lemma 10, the order σ that we compute in this way has to be implemented by $T(\sigma)$.

Let $e_{\mathcal{P}}(\varphi_1, \dots, \varphi_n)$ denote the expected utility for \mathcal{P} under a contract given by fair caps $\varphi_1, \dots, \varphi_n$. Algorithm 1 computes optimal fair caps for all boxes. Initially, the fair cap of each box is given by its basic fair cap. Then we calculate the optimal fair cap of boxes

■ **Algorithm 1** Optimal Contract for Binary Boxes.

```

Data: Binary boxes  $1, \dots, n$  (non-increasing  $\varphi_i(0)$ , tie-break: non-increasing  $b_i$ )
Result: Contract  $(t_1, \dots, t_n)$ 
 $x_k \leftarrow \varphi_k(0)$  for all  $k \in [n]$ ; // Initialize with basic fair caps.
for  $k = 1, \dots, n$  do
  for  $j = 1, \dots, k$  do
    /* Compute expected utility for  $\mathcal{P}$  when box  $k$  had fair cap  $x_j$  */
     $\rho \leftarrow x_j$ ;
     $e_j \leftarrow e_{\mathcal{P}}(x_1, \dots, x_{k-1}, \rho, x_{k+1}, \dots, x_n)$ ;
  end
   $j^* \leftarrow \arg \max_{j \in \{1, \dots, k\}} e_j$ ; // Box  $k$  receives optimal fair cap  $x_{j^*}$ .
   $t_k \leftarrow x_{j^*} - \varphi_k(0)$ ;
end

```

$1, \dots, n$ iteratively. In iteration k of the outer for-loop, we find the optimal fair cap for box k among the fair caps of previous boxes. Crucially, it is sufficient to test which of those fair caps yield maximum utility, even if the final fair caps of subsequent boxes are not determined yet. Note that by (2), to raise the fair cap of any box $i \in [n]$ to $\varphi > \varphi_i(0)$, the required payment t_i is a constant independent of the probabilities and values in box i – more precisely, it is exactly $t_i = \varphi - \varphi_i(0)$.

► **Theorem 11.** *Algorithm 1 computes an optimal exploration contract for binary boxes in polynomial time.*

Proof. For the proof, we show inductively that the decision of the algorithm in each iteration of the outer for-loop regarding box k is optimal. Formally, there is an optimal ordering of all boxes that, when restricted to boxes $1, \dots, k$, is identical to the ordering computed by the algorithm after iteration k . By Lemma 10, this implies that the selected fair cap for box k is optimal (and, thus, the payment in T), as boxes are numbered and considered in a non-increasing order of their basic fair caps.

For the rest of the proof we show the following statement. For every $k \in [n]$, let σ_k denote the ordering of boxes after iteration k of the outer for-loop in Algorithm 1. There is an optimal ordering σ^* such that the relative order of boxes $1, \dots, k$ is identical in σ^* and σ_k .

We proceed by induction. The statement is trivially true for $k = 1$. Now suppose the statement holds for every of the first $k - 1$ iterations, and consider iteration k . Let i^* denote the position of box k w.r.t. boxes $1, \dots, k$ in σ^* , and let i denote its position σ_k . Note that position i^* corresponds to a position $i_n^* \geq i^*$ in the overall ordering σ^* (since for definition of i^* we ignore boxes $k + 1, \dots, n$). Clearly, the statement holds when $i^* = i$.

For the two remaining cases ($i^* > i$ and $i^* < i$) we show that the optimal ordering can be changed sequentially into one that has $i^* = i$ without decreasing the expected value for \mathcal{P} . We discuss the proof for case 1: $i^* > i$. The proof of the other case is very similar (as seen in the full version of the paper).

Consider $i^* > i$. Now suppose box k was brought to position i (w.r.t. first k boxes) in σ^* . The fair cap of k must be strictly increased, since otherwise k would receive the same payment and the same position in σ_k and σ^* (due to optimal tie-breaking by the agent in non-increasing order of principal value). Note that position i corresponds to a position $i_n < i_n^*$ in σ^* .

Consider the set L of boxes located between positions i_n and i_n^* in σ^* . Consider the first box $r \in L$ with $r > k$ and the position i_r in σ^* . Let L_1 be the boxes located before box r in positions $i_n, \dots, i_r - 1$. We use \tilde{p}_1 for the combined probability of selection in L_1 and \tilde{b}_1 for the combined expected value of a selected box in L_1 .

Let b_r^k be the remaining value of box r after a payment that lifts the fair cap of r to the basic fair cap of k . We split the analysis into two cases: $b_r^k \geq b_k$ and $b_r^k < b_k$. If $b_r^k \geq b_k$, consider the move of r to position i_n . Since there are only boxes $j < k$ in L_1 , we could place box k on position $i + |L_1| + 1$ in σ_k – right after boxes from L_1 , similar to the position i_r of box r in σ^* . We use Δ_{i_r} to denote the payment required to lift the fair cap of box k from its basic fair cap to the one required at position i_r . Moreover, Δ_{i_n, i_r} is the additional payment required to lift the fair cap of box k from the one at position i_r to the one at i_n . Since the algorithm places k at i_n instead of i_r , we know that

$$p_k(b_k - \Delta_{i_r} - \Delta_{i_n, i_r}) + (1 - p_k)\tilde{p}_1\tilde{b}_1 \geq \tilde{p}_1\tilde{b}_1 + (1 - \tilde{p}_1)p_k(b_k - \Delta_{i_r}),$$

which implies

$$b_k \geq \tilde{b}_1 + \Delta_{i_r} + \Delta_{i_n, i_r}/\tilde{p}_1.$$

Suppose now we move r from i_r to i_n , then we have the same terms with b_r^k instead of b_k . Clearly, since $b_r^k \geq b_k$ we know that the move is also profitable. This means that we can move r to position i_n without deteriorating the value of σ^* . Thereby we reduce the number of boxes $j \geq k$ between the positions of k in σ^* and σ_k .

For the second case consider $b_r^k < b_k$. We here consider two moves in σ^* – either swap box r right after box k , or swap box k right before box r . Neither of these moves shall maintain the value of σ^* , and we show that this is impossible.

Let L_2 be the boxes located between r and k in σ^* . We use \tilde{p}_2 and \tilde{b}_2 to denote selection probability and expected value of selected box, respectively. Now if we swap box r after box k , a strict decrease in value yields

$$\begin{aligned} & p_r(b_r^k - \Delta_{i_n^*} - \Delta_{i_n^*, i_r}) + (1 - p_r)\tilde{p}_2\tilde{b}_2 + (1 - p_r)(1 - \tilde{p}_2)p_k(b_k - \Delta_{i_n^*}) \\ & > \tilde{p}_2\tilde{b}_2 + (1 - \tilde{p}_2)p_k(b_k - \Delta_{i_n^*}) + (1 - \tilde{p}_2)(1 - p_k)p_r(b_r^k - \Delta_{i_n^*}). \end{aligned}$$

Note that ensuring the same fair cap as box k at position i_n^* is enough, since $b_r^k < b_k$ and box r is then sorted after box k . We obtain

$$\tilde{p}_2 b_r^k + p_k(1 - \tilde{p}_2)b_r^k > \tilde{p}_2\tilde{b}_2 + p_k(1 - \tilde{p}_2)b_k + \tilde{p}_2\Delta_{i_n^*} + \Delta_{i_n^*, i_r}$$

and, since $b_r^k < b_k$,

$$b_r^k > \tilde{b}_2 + \Delta_{i_n^*} + \Delta_{i_n^*, i_r}/\tilde{p}_2. \tag{3}$$

For the other move, we see that a strict decrease in value yields

$$\begin{aligned} & p_r(b_r^k - \Delta_{i_n^*} - \Delta_{i_n^*, i_r}) + (1 - p_r)\tilde{p}_2\tilde{b}_2 + (1 - p_r)(1 - \tilde{p}_2)p_k(b_k - \Delta_{i_n^*}) \\ & > p_k(b_k - \Delta_{i_n^*} - \Delta_{i_n^*, i_r}) + (1 - p_k)p_r(b_r^k - \Delta_{i_n^*} - \Delta_{i_n^*, i_r}) + (1 - p_k)(1 - p_r)\tilde{p}_2\tilde{b}_2. \end{aligned}$$

Note that ensuring the same fair cap as box r at position i_r is enough, since $b_r^k < b_k$ and box k is then sorted before box r . We obtain

$$(-p_r - \tilde{p}_2 + p_r\tilde{p}_2)b_k + (1 - p_r)\tilde{p}_2\Delta_{i_n^*} > -(1 - p_r)\Delta_{i_n^*, i_r} - p_r b_r^k - \tilde{p}_2\tilde{b}_2 + p_r\tilde{p}_2\tilde{b}_2,$$

so

$$(1 - p_r)\tilde{p}_2\tilde{b}_2 + (1 - p_r)\tilde{p}_2\Delta_{i_n^*} + (1 - p_r)\Delta_{i_n^*, i_r} > \tilde{p}_2(1 - p_r)b_k + p_r(b_k - b_r^k).$$

This implies $p_r < 1$, since otherwise we derive $b_k - b_r^k < 0$, a contradiction. Now, if $p_r \in (0, 1)$, $\tilde{p}_2 \in (0, 1]$, and $b_r^k < b_k$, the above implies

$$\tilde{b}_2 + \Delta_{i_n^*} + \Delta_2/\tilde{p}_2 > b_k. \quad (4)$$

Equations (3) and (4) imply a contradiction with $b_r^k < b_k$.

At least one swap (either moving r right after k , or moving k right before r) must be non-deteriorating in terms of solution value. Using this swap, we can change σ^* and decrease the set of agents $j \geq k$ between i_n^* and i_n without loss in value.

Now using these swaps (either the one for $b_r^k \geq b_k$ or one of the two for $b_r^k < b_k$) iteratively, we can turn σ^* into an ordering such that there are only boxes $j < k$ in L . Then moving box k from i_n^* to i_n is not harmful. We let p_ℓ denote the overall probability that a box in L is chosen and b_ℓ the expected value. Let Δ_{i, i^*} be the additional payment required to raise the fair cap from the one of position i^* to the one of position i .

Indeed, our algorithm could have placed box k at position i^* , but preferred to place k in position $i < i^*$. This means that

$$p_k(b_k - \Delta_{i, i^*}) + (1 - p_k)p_\ell b_\ell \geq p_\ell b_\ell + (1 - p_\ell)p_k b_k,$$

which also reflects the change in value if the move is executed in σ^* . This proves the inductive step when $i^* > i$. \blacktriangleleft

4.3 I.I.D. with Single Positive Prize for Principal

Finally, we study the subclass of the problem where all distributions are identical, with the further restriction that there is only a single positive prize for the principal. Even this restriction has a perhaps surprisingly complicated solution. We show the following result.

► **Theorem 12.** *There exists an algorithm that computes an optimal exploration contract for the i.i.d. case with a single positive prize for the principal in polynomial time.*

We introduce some simplified notation for this subclass. All of the n boxes have an identical distribution over prizes $\{0\} \cup [m] = \{0, 1, \dots, m\}$ and the same opening cost $c \geq 0$. For each prize $j \in \{0, 1, \dots, m\}$, the utilities for \mathcal{A} and \mathcal{P} are a_j and b_j , respectively. Without loss of generality, prize 0 is the only prize for which \mathcal{P} has a positive value, i.e., $b_j = 0$ for all $j \geq 1$. The probability that prize j is drawn when a box is opened is denoted by $q_j \in [0, 1]$. For the ease of notation, we define $p := q_0$ and $v := b_0$. For each box $i \in [n]$, \mathcal{P} defines a (non-negative) payment of $t_i \leq v$ for prize 0. We may assume $v > 0$ and $p > 0$ as otherwise $t_1 = \dots = t_n = 0$ would be the unique transfers.

We make two further assumptions that are without loss of generality. First, the values a_j for $j \geq 1$ are unique: If there are $j_1, j_2 \in [m]$ with $a_{j_1} = a_{j_2}$, replace both prizes by a new one with probability $q_{j_1} + q_{j_2}$. Second, we assume $a_1 \leq a_2 \leq \dots \leq a_m$.

Similarly as in Section 4.2, let $\varphi(t)$ be the fair cap of a box for \mathcal{A} when payment $t \geq 0$ is made for prize 0 of that box. We call $\varphi(0)$ the *basic fair cap*. Note that there is an optimal contract in which each fair cap is either $\varphi(0)$ or some value a_j , where $j \in [m]$, with $a_j > \varphi(0)$. The reason is that, if that were not the case, we could decrease the payment for each box, until that property is satisfied while keeping the policy optimal. Importantly, the agent value

of outcome 0 (which is given by $a_0 + t_i$) need not be considered here: Whenever outcome 0 is drawn from a box i with $\varphi_i > \varphi(0)$, the process stops immediately since, by definition of the fair cap and the index policy, it must hold $a_0 + t_i > \varphi_i \geq \varphi_{i+1}$, where φ_{i+1} is the fair cap of the next box.

Observe that $\varphi(0)$ can be achieved with different payments t , as long as $a_0 + t \leq \varphi(0)$. This may influence the stopping behavior of \mathcal{A} as the payment may determine the order of $a_0 + t$ and a_j for some prize j . To achieve some fair cap larger than $\varphi(0)$, however, there is a unique payment that achieves this fair cap because changing a values above the fair cap that occurs with non-zero probability always changes the fair cap by definition.

Now consider an optimal contract. Index the boxes in the order in which they are opened, and let $\varphi_1 \geq \varphi_2 \geq \dots \geq \varphi_n$ be the corresponding fair caps. We first concentrate on a potential set of *basic* boxes with $\varphi(0)$ in the final rounds of the process.

If $a_0 \geq \varphi(0)$, then every positive payment of \mathcal{P} would increase the fair cap to above $\varphi(0)$. Therefore, if basic boxes with fair cap $\varphi(0)$ exist in an optimal contract, $t_i = 0$ must hold for all these boxes.

Otherwise, if $a_0 < \varphi(0)$, suppose $j^* \in [m]$ is the best other prize for \mathcal{A} that is not exceeding the basic fair cap, i.e., $j^* := \arg \max_{j \in [m]} \{a_j : a_j \leq \varphi(0)\}$. Observe that for every basic box i , an optimal payment t_i either fulfills $a_0 + t_i = a_j$ for some $j \in [j^*]$, or $a_0 + t_i = \varphi(0)$. W.l.o.g., when prize 0 is drawn in a basic box i with $a_0 + t_i = \varphi(0)$, the process stops (by tie-breaking in favor of \mathcal{P} , who cannot improve), and when an outcome $j \geq 1$ with $a_j = \varphi(0)$ is drawn in any round before n , the process continues (again by tie-breaking in favor of \mathcal{P} , who gets 0 if a_j is accepted).

We next show that all basic boxes i with $a_0 + t_i = \varphi(0)$ can be assumed to be opened first among the set of basic boxes. The argument is an exchange argument and given in the full version of the paper.

► **Lemma 13.** *There is an optimal contract where all basic boxes ℓ with $a_0 + t_\ell = \varphi(0)$ are opened first among the set of basic boxes.*

We now divide the process into two phases. In every round i of Phase 1, $\varphi_i \geq \varphi(0)$ and prize 0 gets accepted by \mathcal{A} immediately if it is found. In Phase 2, $\varphi_i = \varphi(0)$ and prize 0 is not accepted immediately (only if in the last round a prize 0 represents the best one for \mathcal{A}). Suppose Phase 1 contains boxes $1, \dots, k$ and Phase 2 boxes $k + 1, \dots, n$. Note that either phase might be empty.

Before analyzing the process, we show an example that shows the perhaps counter-intuitive fact that the optimal contract might indeed involve two such phases, each of substantial length. Moreover, even though all boxes have the same distribution and the same fair cap, in the optimal contract they are assigned one of two different payments depending on their position in the inspection sequence.

► **Example 14.** Consider an instance with $m = 2$. Prize 0 has value $a_0 = 0$ and $b_0 = 1 + \alpha$ for some $\alpha > 0$ determined below, as well as $q_0 = 1/n$. The other prizes have $a_1 = 2$, $q_1 = 1/n$ and $a_2 = 0$, $q_2 = 1 - 2/n$. The inspection cost is $c = 1/n$, and the basic fair cap is $\varphi(0) = 1$.

Clearly, if prize 1 is found in a box, then \mathcal{A} always accepts. If prize 2 is found, \mathcal{A} continues to inspect further boxes. Suppose \mathcal{A} draws a prize 0 in the first round. It is rather unlikely that \mathcal{A} will run through all remaining $n - 1$ boxes without finding a prize 1 and eventually accept prize 0 from round 1. Specifically, this probability is $(1 - 1/n)^{n-1}$. Now \mathcal{P} can set $t_1 = 1 = \varphi(0)$, leading to direct acceptance by \mathcal{A} and immediate profit of α . Alternatively, \mathcal{P} can decide to set $t_1 = 0$ (and, in an optimal contract, then $t_i = 0$ throughout) gambling for a high profit of $1 + \alpha$ in the end. The gambling strategy is unprofitable if

50:16 Designing Exploration Contracts

$$\alpha \geq (1 + \alpha) \cdot (1 - 1/n)^{n-1} \quad \text{or, equivalently,} \quad \alpha \geq \frac{(1 - 1/n)^{n-1}}{1 - (1 - 1/n)^{n-1}} .$$

Suppose α satisfies this inequality, then round 1 is part of Phase 1 in an optimal contract. Via the same calculation, we can determine further rounds i of Phase 1. \mathcal{A} accepts prizes 0 and 1 in all earlier rounds $1, \dots, i-1$, so reaching round i is only possible if prize 2 has been found in all these rounds.

Now suppose for some $k \in \{1, \dots, n-1\}$

$$\alpha = \frac{(1 - 1/n)^k}{1 - (1 - 1/n)^k} .$$

Then \mathcal{P} wants to set $t_i = 1$ for all $i \leq k$, i.e., this becomes Phase 1 with direct acceptance of prize 0. For all rounds $i \geq k+1$, it is more profitable to gamble that no prize 1 will arrive in the remaining rounds and \mathcal{P} will secure a value of $1 + \alpha$. This represents Phase 2. \square

Turning to the analysis, we start by examining Phase 1. Using a relatively straightforward exchange argument given in the full version of the paper, we show the following.

► **Lemma 15.** *There is an optimal contract in which all boxes of Phase 1 have the same fair cap.*

In Phase 2, all boxes have fair cap $\varphi(0)$, but this does not exclude the possibility of different payments. If the first box of Phase 2, box $k+1$, is about to be opened, the highest agent value observed so far (if any) is at most $\varphi(0)$. In particular, prize 0 was never observed at that point, since this would have led to acceptance of \mathcal{A} in an earlier round.

During Phase 2, the process only stops early when an outcome $j > j^*$ is drawn. In this case, the utility for \mathcal{P} is 0. If the process does not stop early, \mathcal{P} receives a utility only if outcome 0 is the best outcome for the agent among all n outcomes.

\mathcal{P} could have different payments for different boxes as long as, for every box i , there is some $j \in [j^*]$ such that $a_0 + t_i = a_j$. If an outcome from a box with payment $t = a_j - a_0 \leq \varphi(0) - a_0$ (for some $j \in [j^*]$) is the maximum after n rounds, then

- outcome 0 must have been drawn from any number of boxes that have this payment t , and
- in all boxes with a payment of at most t , some outcome $0 \leq j' < j$ was drawn, and
- in all remaining boxes, some outcome $1 \leq j' < j$ was drawn.

Let $n_j := |\{i \in [n] : a_0 + t_i = a_j\}|$ denote the number of boxes of Phase 2 with such payment, for every $j \in [j^*]$. Furthermore, for every $j \in [j^*]$, let $Q_j := \sum_{j'=1}^j q_{j'}$ denote the combined probability of all outcomes (other than 0) that are not better for \mathcal{A} , and $N_j = \sum_{j'=1}^j n_{j'}$ denote the number of boxes with the according payment. The probability that outcome 0 with a payment of $t = a_j - a_0$ is the best outcome after n rounds is then given by

$$\begin{aligned} & \sum_{i=1}^{n_j} \binom{n_j}{i} \cdot p^i \cdot (p + Q_j)^{N_j-1} \cdot Q_j^{n-N_j-1-i} \\ &= \sum_{i=1}^{n_j} \binom{n_j}{i} \cdot p^i \cdot \left(\frac{p + Q_j}{Q_j}\right)^{N_j-1} \cdot Q_j^{n+n_j-n_j-i} \\ &= \sum_{i=1}^{n_j} \binom{n_j}{i} \cdot p^i \cdot \left(\frac{p}{Q_j} + 1\right)^{N_j-1} \cdot Q_j^{n-n_j} \cdot Q_j^{n_j-i} \end{aligned}$$

$$\begin{aligned}
&= \left(\frac{p}{Q_j} + 1\right)^{N_{j-1}} \cdot Q_j^{n-n_j} \cdot \left(\sum_{i=0}^{n_j} \binom{n_j}{i} \cdot p^i \cdot Q_j^{n_j-i} - Q^{n_j}\right) \\
&= \left(\frac{p}{Q_j} + 1\right)^{N_{j-1}} \cdot Q_j^{n-n_j} \cdot ((p + Q_j)^{n_j} - Q^{n_j}) \\
&= \left(\frac{p}{Q_j} + 1\right)^{N_{j-1}} \cdot Q_j^n \cdot \left(\left(\frac{p}{Q_j} + 1\right)^{n_j} - 1\right) \\
&= Q_j^n \cdot \left(\left(\frac{p}{Q_j} + 1\right)^{N_j} - \left(\frac{p}{Q_j} + 1\right)^{N_{j-1}}\right).
\end{aligned}$$

Thus, the expected utility for \mathcal{P} for Phase 2 is

$$\sum_{j=1}^{j^*} v_j \cdot Q_j^n \cdot \left(\left(\frac{p}{Q_j} + 1\right)^{N_j} - \left(\frac{p}{Q_j} + 1\right)^{N_{j-1}}\right).$$

With this at hand, we show the following lemma in the full version of the paper.

► **Lemma 16.** *There is an optimal contract in which all boxes of Phase 2 have the same payment for prize 0. There is $j^+ \in [j^*]$ such that $n_{j^+} = n - k$ (and $n_j = 0$ for all $j \neq j^+$).*

By the previous lemmas, it suffices to enumerate all combinations of the length of Phase 1, the fair cap in Phase 1, and the unique payment of prize 0 for Phase 2 (determined by a prize j to be targeted, a payment of 0, or a payment resulting in value $\varphi(0)$ for \mathcal{A} for prize 0). These are polynomially many combinations, and for each combination the expected utility of \mathcal{P} for the resulting contract can be computed in polynomial time. Theorem 12 follows.

References

- 1 Tal Alon, Paul Duetting, Yingkai Li, and Inbal Talgam-Cohen. Bayesian analysis of linear contracts. In *Proc. 24th Conf. Econ. Comput. (EC)*, page 66, 2023. doi:10.1145/3580507.3597795.
- 2 Tal Alon, Paul Dütting, and Inbal Talgam-Cohen. Contracts with private cost per unit-of-effort. In *Proc. 22nd Conf. Econ. Comput. (EC)*, pages 52–69, 2021. doi:10.1145/3465456.3467651.
- 3 Yakov Babichenko, Inbal Talgam-Cohen, Haifeng Xu, and Konstantin Zabarnyi. Regret-minimizing bayesian persuasion. *Games Econ. Behav.*, 136:226–248, 2022. doi:10.1016/J.GEB.2022.09.001.
- 4 Curtis Bechtel and Shaddin Dughmi. Delegated Stochastic Probing. In *Proc. Symp. Innov. Theoret. Comput. Sci. (ITCS)*, pages 37:1–37:19, 2021. doi:10.4230/LIPICS.ITCS.2021.37.
- 5 Curtis Bechtel, Shaddin Dughmi, and Neel Patel. Delegated Pandora’s box. In *Proc. 23rd Conf. Econ. Comput. (EC)*, pages 666–693, 2022. doi:10.1145/3490486.3538267.
- 6 Hedyeh Beyhaghi and Linda Cai. Pandora’s problem with nonobligatory inspection: Optimal structure and a PTAS. In *Proc. 55th Symp. Theory Comput. (STOC)*, pages 803–816, 2023. doi:10.1145/3564246.3585217.
- 7 Hedyeh Beyhaghi and Linda Cai. Recent developments in Pandora’s Box problem: Variants and applications. *ACM SIGecom Exchanges*, 21(1):20–34, 2023. doi:10.1145/3699814.3699817.
- 8 Hedyeh Beyhaghi and Robert Kleinberg. Pandora’s problem with nonobligatory inspection. In *Proc. 20th Conf. Econ. Comput. (EC)*, pages 131–132, 2019. doi:10.1145/3328526.3329626.
- 9 Patrick Bolton and Mathias Dewatripont. *Contract Theory*. MIT Press, 2005.
- 10 Shant Boodaghians, Federico Fusco, Philip Lazos, and Stefano Leonardi. Pandora’s box problem with order constraints. *Math. Oper. Res.*, 48(1):498–519, 2023. doi:10.1287/MOOR.2022.1271.

- 11 Pirmin Braun, Niklas Hahn, Martin Hoefer, and Conrad Schecker. Delegated online search. In *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, pages 2528–2536, 2023. doi:10.24963/IJCAI.2023/281.
- 12 Matteo Castiglioni, Andrea Celli, Alberto Marchesi, and Nicola Gatti. Online Bayesian persuasion. In *Proc. Conf. Adv. Neural Inf. Processing Syst. (NeurIPS)*, 2020.
- 13 Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Bayesian agency: Linear versus tractable contracts. In *Proc. 22nd Conf. Econ. Comput. (EC)*, pages 285–286, 2021. doi:10.1145/3465456.3467602.
- 14 Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Designing menus of contracts efficiently: The power of randomization. In *Proc. 23rd Conf. Econ. Comput. (EC)*, pages 705–735, 2022. doi:10.1145/3490486.3538270.
- 15 Bolin Ding, Yiding Feng, Chien-Ju Ho, Wei Tang, and Haifeng Xu. Competitive information design for pandora’s box. In *Proc. Symp. Discret. Algorithms (SODA)*, pages 353–381, 2023. doi:10.1137/1.9781611977554.CH15.
- 16 Shaddin Dughmi, Neel Bharatkumar Patel, Aditya Prasad, and Ram Deo-Campo Vuong. On supermodular contracts and dense subgraphs. In *Proc. Symp. Discret. Algorithms (SODA)*, pages 109–132, 2024. doi:10.1137/1.9781611977912.6.
- 17 Paul Dütting, Tomer Ezra, Michal Feldman, and Thomas Kesselheim. Combinatorial contracts. In *Proc. Symp. Found. Comput. Sci. (FOCS)*, pages 815–826, 2021. doi:10.1109/FOCS52979.2021.00084.
- 18 Paul Dütting, Tomer Ezra, Michal Feldman, and Thomas Kesselheim. Multi-agent contracts. In *Proc. Symp. Theory Comput. (STOC)*, pages 1311–1324, 2023. doi:10.1145/3564246.3585193.
- 19 Paul Dütting, Michal Feldman, and Daniel Peretz. Ambiguous contracts. In *Proc. 24th Conf. Econ. Comput. (EC)*, page 539, 2023.
- 20 Paul Dütting, Tim Roughgarden, and Inbal Talgam-Cohen. Simple versus optimal contracts. In *Proc. 20th Conf. Econ. Comput. (EC)*, pages 369–387, 2019. doi:10.1145/3328526.3329591.
- 21 Paul Dütting, Tim Roughgarden, and Inbal Talgam-Cohen. The complexity of contracts. *SIAM J. Comput.*, 50(1):211–254, 2021. doi:10.1137/20M132153X.
- 22 Paul Dütting, Michal Feldman, and Yoav Gal Tzur. Combinatorial contracts beyond gross substitutes. In *Proc. Symp. Discret. Algorithms (SODA)*, pages 92–108, 2024. doi:10.1137/1.9781611977912.5.
- 23 Tomer Ezra, Michal Feldman, and Maya Schlesinger. Sequential contracts. *CoRR*, abs/2403.09545, 2024. doi:10.48550/arXiv.2403.09545.
- 24 Yiding Feng, Wei Tang, and Haifeng Xu. Online Bayesian recommendation with no regret. In *Proc. 23rd Conf. Econ. Comput. (EC)*, pages 818–819, 2022. doi:10.1145/3490486.3538327.
- 25 Hu Fu, Jiawei Li, and Daogao Liu. Pandora box problem with nonobligatory inspection: Hardness and approximation scheme. In *Proc. 55th Symp. Theory Comput. (STOC)*, pages 789–802, 2023. doi:10.1145/3564246.3585229.
- 26 Anupam Gupta, Haotian Jiang, Ziv Scully, and Sahil Singla. The markovian price of information. In *Proc. Int. Conf. Integer Prog. and Comb. Opt. (IPCO)*, pages 233–246, 2019. doi:10.1007/978-3-030-17953-3_18.
- 27 Guru Guruganesh, Jon Schneider, Joshua R. Wang, and Junyao Zhao. The power of menus in contract design. In *Proc. 24th Conf. Econ. Comput. (EC)*, pages 818–848, 2023. doi:10.1145/3580507.3597735.
- 28 Niklas Hahn, Martin Hoefer, and Rann Smorodinsky. Prophet inequalities for Bayesian persuasion. In *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, pages 175–181, 2020. doi:10.24963/IJCAI.2020/25.
- 29 Niklas Hahn, Martin Hoefer, and Rann Smorodinsky. The secretary recommendation problem. *Games Econ. Behav.*, 134:199–228, 2022. doi:10.1016/J.GEB.2022.05.002.
- 30 Bengt Holmstrom and Paul Milgrom. Aggregation and linearity in the provision of intertemporal incentives. *Econometrica*, 55(2):303–328, 1987.

- 31 Bengt Holmström. Moral hazard and observability. *Bell J. Econ.*, 10(1):74–91, 1979.
- 32 Jon M. Kleinberg and Robert D. Kleinberg. Delegated search approximates efficient search. In *Proc. 19th Conf. Econ. Comput. (EC)*, pages 287–302, 2018. doi:10.1145/3219166.3219205.
- 33 Robert D. Kleinberg, Bo Waggoner, and E. Glen Weyl. Descending price optimally coordinates search. In *Proc. 17th Conf. Econ. Comput. (EC)*, pages 23–24, 2016. doi:10.1145/2940716.2940760.
- 34 P. Postl. Delegated search: Procedure matters. Discussion Paper 04-17, Department of Economics, University of Birmingham, 2004.
- 35 Royal Swedish Academy of Sciences. Scientific background on the 2016 nobel prize in economic sciences, 2016.
- 36 Eden Saig, Inbal Talgam-Cohen, and Nir Rosenfeld. Delegated classification. In *Proc. Conf. Adv. Neural Inf. Processing Syst. (NeurIPS)*, 2023.
- 37 Suho Shin, Keivan Rezaei, and Mohammadtaghi Hajiaghayi. Delegating to multiple agents. In *Proc. 24th Conf. Econ. Comput. (EC)*, pages 1081–1126, 2023. doi:10.1145/3580507.3597669.
- 38 Sahil Singla. The price of information in combinatorial optimization. In *Proc. Symp. Discret. Algorithms (SODA)*, pages 2523–2532, 2018. doi:10.1137/1.9781611975031.161.
- 39 Martin L. Weitzman. Optimal Search for the Best Alternative. *Econometrica*, May, 47(3):641–654, 1979.
- 40 You Zu, Krishnamurthy Iyer, and Haifeng Xu. Learning to persuade on the fly: Robustness against ignorance. In *Proc. 22nd Conf. Econ. Comput. (EC)*, pages 927–928, 2021. doi:10.1145/3465456.3467593.