THE COMPLEXITY OF CONTRACTS*

P. DÜTTING†, T. ROUGHGARDEN‡, AND I. TALGAM-COHEN§

Abstract. We initiate the study of computing (near-)optimal contracts in succinctly representable principal-agent settings. Here optimality means maximizing the principal's expected payoff over all incentive-compatible contracts—known in economics as "second-best" solutions. We also study a natural relaxation to *approximately* incentive-compatible contracts.

We focus on principal-agent settings with succinctly described (and exponentially large) outcome spaces. We show that the computational complexity of computing a near-optimal contract depends fundamentally on the number of agent actions. For settings with a constant number of actions, we present a fully polynomial-time approximation scheme (FPTAS) for the separation oracle of the dual of the problem of minimizing the principal's payment to the agent, and use this subroutine to efficiently compute a δ-incentive-compatible (δ-IC) contract whose expected payoff matches or surpasses that of the optimal IC contract.

With an arbitrary number of actions, we prove that the problem is hard to approximate within any constant c. This inapproximability result holds even for δ-IC contracts where δ is a sufficiently rapidly-decaying function of c. On the positive side, we show that simple linear δ-IC contracts with constant δ are sufficient to achieve a constant-factor approximation of the "first-best" (full-welfare-extracting) solution, and that such a contract can be computed in polynomial time.

Key words. Principal-agent problem, contract theory, moral hazard, computational complexity, hardness of approximation, FPTAS

AMS subject classifications. 68W40, 91B40

1. Introduction.

Economic theory distinguishes three fundamentally different problems involving asymmetric information and incentives. In the first—known as *mechanism design* (or *screening*)—the less informed party has to make a decision. A canonical example is Myerson's optimal auction design problem [42], in which a seller wants to maximize the revenue from selling an item, having only incomplete information about the buyers' willingness to pay. The second problem is known as *signalling* (or *Bayesian persuasion*). Here, as in the first case, information is hidden, but this time the more informed party is the active party. A canonical example is Akerlof's "market for lemons" [1]. In this example, sellers are better informed about the quality of the products they sell, and may benefit by sharing (some) of their information with the buyers.

Both of these basic incentive problems have been studied very successfully and extensively from a computational perspective, see, e.g., [9, 10, 11, 6, 12, 5, 28, 29] and [19, 21, 17, 22].

The third basic problem, *the agency problem* in *contract theory*, has received far less attention from the theoretical computer science community, despite being

*An extended abstract appeared in *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA'20)*, Salt Lake City, UT, USA, January 5–8, 2020.

Funding: Supported by BA/Leverhulme Small Research Grant SRG1819\191601, NSF Award CCF-1813188, ARO grant W911NF1910294, and the Israel Science Foundation (Grant No. 336/18). Part of the work of the first author was done while visiting Google Research. The third author is a Taub Fellow (supported by the Taub Family Foundation). The authors thank the anonymous referees for their helpful feedback.

†Department of Mathematics, London School of Economics, Houghton Street, London WC2A 2AE, UK (p.d.duetting@lse.ac.uk).

‡Department of Computer Science, Columbia University, 500 West 120th Street, New York, NY 10027, USA (tr@cs.columbia.edu).

§Department of Computer Science, Technion, Israel Institute of Technology, Haifa, Israel 3200003. (italgam@cs.technion.ac.il).

regarded as equally important in economic theory (see, e.g., the scientific background on the 2016 Nobel Prize for Hart and Holmström [48]). (A notable exception is [4], which we will discuss with further related work in more detail below.)

The basic scenario of contract theory is captured by the following *hidden-action principal-agent problem* [30]: There is one *principal* and one *agent*. The agent can take one of n actions $a_i \in A_n$. Each action a_i is associated with a distribution F_i over m outcomes $x_j \in \mathbb{R}_{\geq 0}$, and has a cost $c_i \in \mathbb{R}_{\geq 0}$. The principal designs a contract p that specifies a payment $p(x_j)$ for each outcome x_j. The agent chooses an action a_i that maximizes expected payment minus cost, i.e., $\sum_j F_{i,j} p(x_j) - c_i$. The principal seeks to set up the contract so that the chosen action maximizes expected outcome minus expected payment, i.e., $\sum_j F_{i,j} x_j - \sum_j F_{i,j} p(x_j)$.

The principal-agent problem is quite different from mechanism design and signalling, where the basic difficulty is the information asymmetry and that part of the information is hidden. In the principal-agent problem the issue is one of *moral hazard*: in and by itself the agent has no intrinsic interest in the expected outcome to the principal.

It is straightforward to see that the optimal contract can be found in time polynomial in n and m by solving n linear programs (LPs). For each action the corresponding LP gives the smallest expected payment at which this action can be implemented. The action that yields the highest expected reward minus payment gives the optimal payoff to the principal, and the LP for this action the optimal contract.

Succinct principal-agent problems. This linear programming-based algorithm for computing an optimal contract has several analogs in algorithmic game theory:

1. *Mechanism design.* For many basic mechanism design problems, the optimal (randomized) mechanism is the solution of a linear program with size polynomial in that of the players' joint type space.
2. *Signalling.* For many computational problems in signalling, the optimal signalling scheme is the solution to a linear program with size polynomial in the number of receiver actions and possible states of nature.
3. *Correlated equilibria.* In finite games, a correlated equilibrium can be computed using a linear program with size polynomial in the number of game outcomes.

These linear-programming-based solutions are unsatisfactory when their size is exponential in some parameter of interest. For example, in the mechanism design and correlated equilibria examples, the size of the LP is exponential in the number of players. A major contribution of theoretical computer science to game theory and economics has been the articulation of natural classes of succinctly representable settings and a thorough study of the computational complexity of optimal design problems in such settings. Examples include work on multi-dimensional mechanism design that has emphasized succinct type distributions [9, 10, 11, 12], succinct signalling schemes with an exponential number of states of nature [22], and the efficient computation of correlated equilibria in succinctly representable multi-player games [46, 36]. The goal of this paper is to initiate an analogous line of work for succinctly described agency problems in contract theory.

We focus on principal-agent settings with succinctly described (and exponentially large) outcome spaces, along with a reward function that supports value queries and a distribution for each action with polynomial description. While there are many such settings one can study, we focus on what is arguably the most natural one from a theoretical computer science perspective, where outcomes correspond to vertices of the hypercube, the reward function is additive, and the distributions are product distributions. (Cf., work on computing revenue-maximizing multi-item auctions with product distributions over additive valutions, e.g. [9, 10].)

For example, outcomes could correspond to sets of items, where items are sold separately using posted prices. Actions could correspond to different marketing strategies with different costs, which lead to different (independent) probabilities of sales of various items. Or, imagine that a firm (principal) uses a headhunter (agent) to hire an employee (action). Dimensions could correspond to tasks or skills. Actions correspond to types of employees, costs correspond to the difficulty of recruiting an employee of a given type, and for each employee type there is some likelihood that they will possess each skill (or be able to complete some task). The firm wants to motivate the headhunter to put in enough effort to recruit an employee who is likely to have useful skills for the firm, without actually running extensive interviews to find out the employee's type.

In our model, as in the classic model, there is a principal and an agent. The agent can take one of n actions $a_i \in A_n$, and each action has a cost $c_i \in \mathbb{R}_{\geq 0}$. Unlike in the original model, we are given a set of items M, with $|M| = m$. Outcomes correspond to subsets of items $S \in 2^M$. Each item has a reward r_j, and the reward of a set S of items is $\sum_{j \in S} r_j$. Every action a_i comes with probabilities $q_{i,j}$ for each item j. If action a_i is chosen, each item j is included in the outcome independently with probability $q_{i,j}$. A contract specifies a payment p_S for each outcome $S \in 2^M$. The goal is to compute a contract that maximizes (perhaps approximately) the principal's payoff in running time polynomial in n and m (which is logarithmic in the size $|2^M|$ of the outcome space).

A notion of approximate IC for contracts. The classic approach in contract theory is to require that the agent is incentivized exactly, i.e., he (weakly) prefers the chosen action over every other action. We refer to such contracts as incentive compatible or just IC contracts. Motivated in part by our hardness results for IC contracts (see the next section) and inspired by the success of notions of approximate incentive compatibility in mechanism design (as, for example, in [8, 51, 12], hereafter referred to as the *CDW framework*), we introduce a notion of approximate incentive compatibility that is suitable for contracts.

Our notion of δ-incentive compatibility (or δ-IC) is that the agent utility of the approximately incentivized action a_i is at least that of any other action $a_{i'}$, less δ. (See Section 2.4 for details, including how to turn δ-IC contracts into IC contracts with small multiplicative—and necessarily—additive loss.) This notion is natural for several reasons. First, it coincides with the usual notion of ϵ-IC in "normalized" mechanism design settings (with all valuations between 0 and 1), as in [8, 51]. A second reason is behaviorial. There is an increasing body of work in economics on behavioral biases in contract theory [39], including strong empirical evidence that such biases play an important role in practice—for example, that agents "gift" effort to the principals employing them [2]. The notion of δ-IC offers a mathematical formulation of an agent's bias. Along similar lines, [15] advocates generally for approximate IC constraints in settings where the designer can propose their "preferred action" to agents, in which case an agent may be biased against deviating due to the complexities involved in determining the agent-optimal action or the psychological costs of deviating. See also [25] for related discussion in the context of contract theory.

1.1. Our contribution and techniques.

We prove several positive and negative algorithmic results for computing near-optimal contracts in succinctly described principal-agent settings. Our work reveals a fundamental dichotomy between settings with a constant number of actions and those with an arbitrary number of actions.

Constant number of actions. For a constant number of actions, we prove in Section 3 that while it is NP-hard to compute an optimal IC contract, there is an FPTAS that computes a δ-IC contract with expected principal surplus at least that of the optimal IC contract; the running time is polynomial in m and $1/\delta$.

THEOREM 1.1 (See Theorem 3.1, Corollary 3.2). *For every constant $n \geq 1$ and $\delta > 0$, there is an algorithm that computes a δ-IC contract with expected principal surplus at least that of an optimal IC contract in time polynomial in m and $1/\delta$.*

The starting point of our algorithm is a linear programming formulation of the problem of incentivizing a given action with the lowest possible expected payment. Our formulation has a polynomial number of constraints (one per action other than the to-be-incentivized one) but an exponential number of variables (one per outcome). A natural idea is to then solve the dual linear program using the ellipsiod method. The dual separation oracle is: given a weighted mixture of $n-1$ product distributions (over the m items) and a reference product distribution q^*, minimize the ratio of the probability of outcome S in the mixture distribution and that in the reference distribution. Unfortunately, as we show, this is an NP-hard problem, even when there are only $n = 3$ actions. On the other hand, we provide an FPTAS for the separation oracle in the case of a constant number of actions, based on a delicate multi-dimensional bucketing approach. The standard method of translating an FPTAS for a separation oracle to an FPTAS for the corresponding linear program relies on a scale-invariance property that is absent in our problem. We proceed instead via a strengthened version of our dual linear program, to which our FPTAS separation oracle still applies, and show how to extract from an approximately optimal dual solution a δ-IC contract with objective function value at least that of the optimal solution to the original linear program.

Arbitrary number of actions. The restriction to a constant number of actions is essential for the positive results above (assuming $P \neq NP$). Specifically, we prove in Section 4 that computing the IC contract that maximizes the expected payoff to the principal is NP-hard, even to approximate to within any constant c. This hardness of approximation result persists even if we relax from exact IC to δ-IC contracts, provided δ is sufficiently small as a function of c.

THEOREM 1.2 (See Theorem 4.1, Corollary 4.2). *For every constant $c \in \mathbb{R}$, $c \geq 1$, it is NP-hard to find a IC contract that approximates the optimal expected payoff achievable by an IC contract to within a multiplicative factor of c.*

THEOREM 1.3 (See Theorem 4.1, Corollary 4.3). *For any constant $c \in \mathbb{R}, c \geq 5$ and $\delta \leq (\frac{1}{4c})^c$, it is NP-hard to find a δ-IC contract that guarantees $> \frac{2}{c} OPT$, where OPT is the optimal expected payoff achievable by an IC contract.*

We prove these hardness of approximation results by reduction from MAX-3SAT, using the fact that it is NP-hard to distinguish between a satisfiable MAX-3SAT instance and one in which there is no assignment satisfying more than a $7/8+\alpha$ fraction of the clauses, where α is some arbitrarily small constant [33]. Our reduction utilizes the gap between "first best" (full-welfare-extracting) and "second best" solutions in contract design settings, where satisfiable instances of MAX-3SAT map to instances where there is no gap between first and second best and instances of MAX-3SAT in which no more than $7/8 + \alpha$ clauses can be satisfied map to instances where there is a constant-factor multiplicative gap between the first-best and second-best solutions.

On the positive side, we prove that for every constant δ there is a simple (in fact, linear[1]) contract that achieves a c_δ-approximation, where c_δ is a constant that depends on δ. This approximation guarantee is with respect to the strongest possible benchmark, the first-best solution.[2]

THEOREM 1.4 (See Theorem 5.1). *For every constant $\delta > 0$ there exists a constant c_δ and a polynomial-time (in n and m) computable δ-IC contract that obtains a multiplicative c_δ-approximation to the optimal welfare.*

Our proof of this result, in Section 5, shows that the optimal social welfare can be upper bounded by a sum of (constantly many in δ) expected payoffs achievable by δ-IC contracts. The best such contract thus obtains a constant approximation to the optimal welfare.

Black-box distributions. Product distributions are a rich and natural class of succinctly representable distributions to study, but one could also consider other classes. Perhaps the strongest-imaginable positive result would be an efficient algorithm for computing a near-optimal contract that works with *no* assumptions about each action's probability distribution over outcomes, other than the ability to sample from them efficiently. (Positive examples of this sort in signalling include [22] and in mechanism design include [32] and its many follow-ups.) Interestingly, the principal-agent problem poses unique challenges to such "black-box" positive results. The moral reason for this is explained, for example, in [49]: Rewards play a dual role in contract settings, both defining the surplus from the joint project to be shared between the principal and agent *and* providing a signal to the principal of the agent's action. For this reason, in optimal contracts, the payment to the agent in a given outcome is governed both by the outcome's reward and on its "informativeness," and the latter is highly sensitive to the precise probabilities in the outcome distributions associated with each action. In Section 6 we translate this intuition into an information-theoretic impossibility result for the black-box model, showing that positive results are possible only under strong assumptions on the distributions (e.g., that the minimum non-zero probability is bounded away from 0).

1.2. Related work.

The study of computational aspects of contract theory was pioneered by Babaioff, Feldman and Nisan [4] (see also their subsequent works, notably [24] and [7]). This line of work studies a problem referred to as *combinatorial agency*, in which combinations of agents replace the single agent in the classic principal-agent model. The challenge in the new model stems from the need to incentivize multiple agents, while the action structure of each agent is kept simple (effort/no effort). The focus of this line of work is on complex combinations of agents' efforts influencing the outcomes, and how these determine the subsets of agents to contract with. The resulting computational problems are very different from the computational problems in our model.[3]

A second direction of highly related work is [3]. This work considers a principal-agent model in which the agent action space is exponentially sized but compactly

[1] A linear contract is defined by a single parameter $\alpha \in [0,1]$, and sets the payment p_S for any set $S \in 2^M$ to $p_S = \alpha \cdot \sum_{j \in S} r_j$. Linear contracts correspond to a simple percentage commission, and are arguably among the most frequently used contracts in practice. See [16] and [23] for recent work in economics and computer science in support of linear contracts.

[2] Note that the principal's objective function (reward minus payment to the agent) is a mixed-sign objective; such functions are generally challenging for relative approximation results.

[3] For example, several of the key computational questions in their problem turn out to be $\#P$-hard, while all of the problems we consider are in NP.

represented, and argue that in such settings indirect (interactive) mechanisms can be better than one-shot mechanisms. Our focus is more algorithmic, and instead of a compactly represented action space we consider a compactly represented outcome space.

A third direction of related work considers a bandit-style model for contract design [34]. In their model each arm corresponds to a contract, and they present a procedure that starts out with a discretization of the contract space, which is adaptively refined, and which achieves sublinear regret in the time horizon. Again the result is quite different from our work, where the complexity comes from the compactly represented outcome space, and our result on the black-box model sheds a more negative light on the learning approach.

Further related work comes from Kleinberg and Kleinberg [38] who consider the problem of delegating a task to an agent in a setting where (unlike in our model) monetary compensation is not an option. Although payments are not available, they show through an elegant reduction to the prophet-inequality problem that constant competitive solutions are possible.

A final related line of work was initiated by Carroll [16] who—working in the classic model (where computational complexity is not an issue)—shows a sense in which linear contracts are max-min optimal (see also the recent work of [50]). Dütting et al. [23] show an alternative such sense, and also provide tight approximation guarantees for linear contracts.

2. Preliminaries.

We start by defining succinct principal-agent settings and the contract design problem.

2.1. Succinct principal-agent settings.

Let n and m be parameters. A principal-agent setting is composed of the following: n *actions* A_n among which the agent can choose, and their *costs* $0 = c_1 \leq \cdots \leq c_n$ for the agent; *outcomes* which the actions can lead to, and their *rewards* for the principal; and a *mapping* from actions to distributions over outcomes. Crucially, the agent's choice of action is hidden from the principal, who observes only the action's realized outcome. Our goal is to study *succinct* principal-agent settings with description size polynomial in n and m; the (implicit) outcome space can have size exponential in m. Throughout, unless stated otherwise, all principal-agent settings we consider are succinct. We focus on arguably one of the most natural models of succinctly-described settings, namely those with *additive* rewards and *product* distributions.

In more detail, let $M = \{1, 2, .., m\}$, where M is referred to as the *item set*. Let the outcome space be $\{0, 1\}^M$, that is, every outcome is an item subset $S \subseteq M$. For every item $j \in M$, the principal gets an additive reward r_j if the realized outcome includes j, so the principal's reward for outcome S is $r_S = \sum_{j \in S} r_j$. Every action $a_i \in A_n$ is associated with probabilities $q_{i,1}, ..., q_{i,m} \in [0, 1]$ for the items. We denote the corresponding product distribution by q_i. When the agent takes action a_i, item j is included in the realized outcome independently with probability $q_{i,j}$. The probability of outcome S is thus $q_{i,S} = (\prod_{j \in S} q_{i,j})(\prod_{j \notin S}(1 - q_{i,j}))$. By linearity of expectation, the principal's expected reward given action a_i is $R_i = \sum_S q_{i,S} r_S = \sum_j q_{i,j} r_j$. Action a_i's expected *welfare* is $R_i - c_i$, and we assume $R_i - c_i \geq 0$ for every $i \in [n]$.

EXAMPLE 2.1 (Succinct principal-agent setting). *A company (principal) hires an agent to sell its m products. The agent may succeed in selling any subset of the m items, depending on his effort level, where the ith level leads to sale of item j with probability $q_{i,j}$. Reward r_j from selling item j is the profit-margin of product j for the* *company.*

Representation. A succinct principal-agent setting is described by an n-vector of costs c, an m-vector of rewards r, and an $n \times m$-matrix Q where entry (i, j) is equal to probability $q_{i,j}$ (and we assume for simplicity that the number of bits of precision for all values is $\text{poly}(n, m)$).

Assumptions. Our assumption that $c_1 = 0$ is a typical assumption in the contracts literature. It serves to make the individual rationality constraint a special case of the incentive compatibility constraint (also see Section 2.2 below).

Unless stated otherwise, we assume that all principal-agent settings are *normalized*, i.e., $R_i \leq 1$ for every $a_i \in A_n$ (and thus also $c_i \leq 1$). Normalization amounts to a simple change of "currency", i.e., of the units in which rewards and costs are measured. It is a standard assumption in the context of approximate incentive compatibility—see Section 2.3 (similar assumptions appear in both the CDW framework and in [15]).

2.2. Contracts and incentives. A *contract* p is a vector of payments from the principal to the agent. Payments are non-negative; this is known as *limited liability* of the agent.[4] The contractual payments are contingent on the outcomes and not actions, as the actions are not directly observable by the principal. A contract p can potentially specify a payment $p_S \geq 0$ for every outcome S, but by linear programming (LP) considerations detailed below, we can focus on contracts for which the support size of the vector p is polynomial in n. We sometimes denote by p_i the expected payment $\sum_{S \subseteq M} q_{i,S} p_S$ to the agent for choosing action a_i, and without loss of generality restrict attention to contracts for which $p_i \leq R_i$ for every $a_i \in A_n$ (in particular, $p_i \leq 1$ by normalization).

Given contract p, the agent's expected *utility* from choosing action a_i is $p_i - c_i$. The principal's expected *payoff* is then $R_i - p_i$. The agent wishes to maximize his expected utility over all actions and over an outside option with utility normalized to zero ("individual rationality" or *IR*). Since by assumption the cost c_1 of action a_1 is 0, the outside opportunity is always dominated by action a_1 and so we can omit the outside option from consideration. Therefore, the incentive constraints for the agent to choose action a_i are: $p_i - c_i \geq p_{i'} - c_{i'}$ for every $i' \neq i$. If these constraints hold we say a_i is *incentive compatible (IC)* (and as discussed, in our model IC implies IR). The standard tie-breaking assumption in the contract design literature is that among several IC actions the agent tie-breaks in favor of the principal, i.e. chooses the IC action that maximizes the principal's expected payoff.[5] We say contract p *implements* or *incentivizes* action a_i if given p the agent chooses a_i (namely a_i is IC and survives tie-breaking). If there exists such a contract for action a_i we say a_i is *implementable*, and slightly abusing notation we sometimes refer to the implementing contract as an *IC contract*.

Simple contracts. In a *linear* contract, the payment scheme is a linear function of the rewards, i.e., $p_S = \alpha r_S$ for every outcome S. We refer to $\alpha \in [0, 1]$ as the linear contract's *parameter*, and it serves as a succinct representation of the contract. Linear contracts have an alternative succinct representation by an m-vector of item payments $p_j = \alpha r_j$ for every $j \in M$, which induce additive payments $p_S = \sum_{j \in S} p_j$. A natural generalization is *separable* contracts, the payments of which can also be

[4]Limited liability plays a similar role in the contract literature as risk-averseness of the agent. Both reflect the typical situation in which the principal has "deeper pockets" than the agent and is thus the better bearer of expenses/risks.

[5]The idea is that one could perturb the payment schedule slightly to make the desired action uniquely optimal for the agent. For further discussion see [13, p. 8].

separated over the m items and represented by an m-vector of non-negative payments (not necessarily linear). The optimal linear (resp., separable) contract can be found in polynomial time (see Proposition A.1 in Appendix A). We return to linear contracts in Section 5 and to separable contracts in Appendix H.

2.3. Contract design and relaxations. The goal of contract design is to maximize the principal's expected payoff from the action chosen by the agent subject to IC constraints. A corresponding computational problem is OPT-CONTRACT: The input is a succinct principal-agent setting, and the output is the principal's expected payoff from the optimal IC contract. A related problem is MIN-PAYMENT: The input is a succinct principal-agent setting and an action a_i, and the output is the minimum expected payment p_i^* with which a_i can be implemented (up to tie-breaking). OPT-CONTRACT reduces to solving n instances of MIN-PAYMENT to find p_i^* for every action a_i, and returning the maximum expected payoff to the principal $\max_{i\in[n]}\{R_i - p_i^*\}$. Observe that MIN-PAYMENT can be formulated as an exponentially-sized LP with 2^m variables $\{p_S\}$ (one for each set $S \subseteq M$) and $n-1$ constraints:

$$\begin{aligned}
(2.1) \qquad & \min \sum_{S\subseteq M} q_{i,S}p_S \\
& \text{s.t.} \sum_{S\subseteq M} q_{i,S}p_S - c_i \geq \sum_{S\subseteq M} q_{i',S}p_S - c_{i'} && \forall i' \neq i, i' \in [n], \\
& p_S \geq 0 && \forall S \subseteq M.
\end{aligned}$$

While we can't use this LP formulation to compute an optimal contract, it implies that there is a succinct optimal contract: There exists an extreme point of the feasible region which is optimal. That extreme point must satisfy 2^m constraints with equality (one per variable). Only $n-1$ of those constraints aren't of the form $p_S = 0$, so the remaining constraints must all have $p_S = 0$.

The dual LP has $n-1$ nonnegative variables $\{\lambda_{i'}\}$ (one for every action i' other than i), and exponentially-many constraints:

$$\begin{aligned}
(2.2) \qquad & \max \sum_{i'\neq i} \lambda_{i'}(c_i - c_{i'}) \\
& \text{s.t.} \Big(\sum_{i'\neq i} \lambda_{i'}\Big) - 1 \leq \sum_{i'\neq i} \lambda_{i'} \frac{q_{i',S}}{q_{i,S}} && \forall S \subseteq E, q_{i,S} > 0, \\
& \lambda_{i'} \geq 0 && \forall i' \neq i, i' \in [n].
\end{aligned}$$

However, the ellipsoid method cannot be applied to solve the dual LP in polynomial time. The separation oracle, which is related to the concept of likelihood ratios from statistical inference, turns out to be NP-hard except for the $n = 2$ case—see Proposition B.1 in Appendix B.

We return to LP (2.1) and to its dual LP (2.2) in Section 3.

Relaxed IC. Contract design like auction design is ultimately an optimization problem subject to IC constraints. The state-of-the-art in optimal *auction* design requires a relaxation of IC constraints to ϵ-IC. In the CDW framework, the ϵ loss factor is additive and applies to normalized auction settings. The framework enables polytime computation of an ϵ-IC auction with expected revenue approximating that of the optimal IC auction.[6] Appropriate ϵ-IC relaxations are also studied in multiple

[6]To be precise, the CDW framework focuses on *Bayesian* IC (BIC) and ϵ-BIC auctions.

additional contexts—see [15] and references within for voting, matching and competitive equilibrium; and [45] for Nash equilibrium. We wish to achieve similar results in the context of optimal contracts. For completeness we include the definition of ϵ-IC cast in the language of contracts:

Definition 2.2 (δ-IC action). *Consider a (normalized) contract setting. For $\delta \geq 0$, an action a_i is* δ-IC *given a contract p if the agent loses no more than additive δ in expected utility by choosing a_i, i.e.: $p_i - c_i \geq p_{i'} - c_{i'} - \delta$ for every action $a_{i'} \neq a_i$.*

As in the IC case, we often slightly abuse notation and refer to the contract p itself as δ-IC. By this we mean a contract p with an (implicit) action a_i that is δ-IC given p (if there are several such δ-IC actions, by our tie-breaking assumption the agent chooses the one that maximizes the principal's expected payoff). We also say the contract δ-*implements* or δ-*incentivizes* action a_i. Finally if there exists such a contract for a_i then we say this action is δ-*implementable*. We denote by δ-OPT-CONTRACT and δ-MIN-PAYMENT the above computational problems with IC replaced by δ-IC (e.g., the input to δ-OPT-CONTRACT is a succinct principal-agent setting and a parameter δ, and the output is the principal's expected payoff from the optimal δ-IC contract).

2.4. Properties of approximately IC contracts. We conclude this section with a few observations concerning δ-IC contracts. Proofs appear in Appendix C.

Implementability. A first observation is that, by LP duality, any action can be δ-implemented up to tie-breaking even for arbitrarily small δ. Note that this result just talks about whether a given action can be δ-incentivized, it may be the case that the payments required for this are very high.

Proposition 2.3. *For every principal-agent setting and every $\delta > 0$, every action a_i can be δ-implemented up to tie-breaking.*

Relaxed vs. exact IC. Our next pair of results concerns the relation between IC contracts and δ-IC contracts.

Proposition 2.4 shows that for every δ-IC contract there is an IC contract with approximately the same expected payoff to the principal up to small—and necessary—multiplicative and additive losses. Thus relaxing IC to δ-IC increases the expected payoff of the principal only to a certain extent. More precisely, Proposition 2.4 shows that any δ-IC contract can be transformed into an IC contract that maintains at least $(1-\sqrt{\delta})$ of the principal's expected payoff up to an additive loss of $(\sqrt{\delta}-\delta)$. Similar results are known in the context of *auctions* (see [31, 20] for welfare maximization and [18] for revenue maximization).

To state Proposition 2.4, denote by $\ell_{\alpha=1}$ the linear contract with parameter $\alpha = 1$ (that transfers the full reward from principal to agent).

Proposition 2.4. *Fix a principal-agent setting and $\delta > 0$. Let p be a contract that δ-incentivizes action a_i. Then the IC contract p' defined as $(1-\sqrt{\delta})p + \sqrt{\delta}\ell_{\alpha=1}$ achieves for the principal expected payoff of at least $(1-\sqrt{\delta})(R_i - p_i) - (\sqrt{\delta} - \delta)$, where $R_i - p_i$ is the expected payoff of contract p.*

Proposition 2.5 shows that an additive loss is necessary, as even for tiny δ there can be a multiplicative constant-factor gap between the expected payoff of an IC contract and a δ-IC one.

Proposition 2.5. *For any $\delta \in (0, 1/2]$, there exists a principal-agent setting where the optimal contract extracts expected payoff OPT but a δ-IC contract extracts expected payoff $\geq \frac{4}{3}OPT$.*

Relaxed IC with exact IR. In our model, IC implies IR due to the existence of a zero-cost action a_1, but this is no longer the case for δ-IC. What if we are willing to relax IC to δ-IC due to the considerations above, but do not want to give up on IR? Suppose we enforce IR by assuming that the agent chooses a δ-IC action only if it has expected utility ≥ 0. The following lemma shows that this has only a small additive effect on the principal's expected payoff, allowing us from now on to focus on δ-IC contracts (IR can be later enforced by applying the lemma):

LEMMA 2.6. *For every δ-IC contract p that achieves expected payoff of Π for the principal, there exists a δ-IC* and IR *contract p' that achieves expected payoff of $\geq \Pi - \delta$.*

3. Constant number of actions. In this section we begin our exploration of the computational problems OPT-CONTRACT and MIN-PAYMENT by considering principal-agent settings with a constant number n of actions. For every constant $n \geq 3$ these problems are NP-hard, and this holds even if the IC requirement is relaxed to δ-IC (See Proposition D.1 and Corollary D.2 in Appendix D). As our main positive result, we establish the tractability of finding a δ-IC contract that matches the expected payoff of the optimal IC contract. In Section 4 we show this result is too strong to hold for non-constant values of n (under standard complexity assumptions), and in Section 5 we provide an approximation result for general settings.

To state our results more formally, fix a principal-agent setting and action a_i; let OPT_i be the solution to MIN-PAYMENT for a_i (or ∞ if a_i cannot be implemented up to tie-breaking without loss to the principal); and let OPT be the solution to OPT-CONTRACT. Observe that $OPT = \max_{i\in[n]}\{R_i - OPT_i\}$. Our main results in this section are the following:

THEOREM 3.1 (MIN-PAYMENT). *There exists an algorithm that receives as input a (succinct) principal-agent setting with a constant number of actions and m items, an action a_i, and a parameter $\delta > 0$, and returns in time poly$(m, \frac{1}{\delta})$ a contract that δ-incentivizes a_i with expected payment $\leq OPT_i$ to the agent.*

COROLLARY 3.2 (OPT-CONTRACT). *There exists an algorithm that receives as input a (succinct) principal-agent setting with a constant number of actions and m items, and a parameter $\delta > 0$, and returns in time poly$(m, \frac{1}{\delta})$ a δ-IC contract with expected payoff $\geq OPT$ to the principal.*

Proof. Apply the algorithm from Theorem 3.1 once per action a_i to get a contract that δ-incentivizes a_i with expected payoff at least $R_i - OPT_i$ to the principal. Maximizing over the actions we get a δ-IC contract with expected payoff $\geq OPT$ to the principal. ∎

Corollary 3.2 shows how to achieve OPT with a δ-IC contract rather than an IC one, in the same vein as the CDW results for auctions. A similar result does not hold for general n unless P=NP (Corollary 4.3). Note that the δ-IC contract can be transformed into an IR one with an additive δ loss by applying Lemma 2.6, and to a fully IC one with slightly more loss by Proposition 2.4, where δ can be an arbitrarily small inverse polynomial in m.

In the rest of the section we prove Theorem 3.1.

An FPTAS for the separation oracle. We begin by stating the separation oracle problem, and relating it to a problem called MIN-LR. LP (2.1) formulates

MIN-PAYMENT for action a_i. Its dual LP (2.2) has constraints of the form:

$$(\sum_{i'\neq i} \lambda_{i'}) - 1 \leq \sum_{i'\neq i} \lambda_{i'} \frac{q_{i',S}}{q_{i,S}}. \tag{3.1}$$

The separation oracle problem is thus: Given $n-1$ nonnegative values $\{\lambda_{i'}\}$ and n product distributions $q_i, \{q_{i'}\}$ over the m items, find an outcome S such that $(\sum_{i'\neq i} \lambda_{i'}) - 1 > \sum_{i'\neq i} \lambda_{i'} \frac{q_{i',S}}{q_{i,S}}$ (i.e., a violated constraint), or determine that no such S exists. Dividing by $\sum_{i'\neq i} \lambda_{i'}$ and letting $\alpha_{i'} = \lambda_{i'}/(\sum_{i'\neq i} \lambda_{i'})$ we can rewrite (3.1) as

$$1 - \frac{1}{\sum_{i'\neq i} \lambda_{i'}} \leq \sum_{i'\neq i} \left(\frac{\lambda_{i'}}{\sum_{i'\neq i} \lambda_{i'}} \cdot \frac{q_{i',S}}{q_{i,S}} \right) = \sum_{i'\neq i} \frac{\alpha_{i'} q_{i',S}}{q_{i,S}}.$$

Observe that the αs sum to 1, since $\sum_{i'\neq i} \alpha_{i'} = \sum_{i'\neq i} \lambda_{i'}/(\sum_{i'\neq i} \lambda_{i'}) = 1$. We conclude that the separation oracle problem for dual LP (2.2) is equivalent to searching for S such that $\sum_{i'} \frac{\alpha_{i'} q_{i',S}}{q_{i,S}}$ is strictly less than $1 - 1/(\sum_{i'\neq i} \lambda_{i'})$. Minimizing $\sum_{i'} \frac{\alpha_{i'} q_{i',S}}{q_{i,S}}$ over all S is sufficient to solve the problem.

We can restate this minimization problem over S in the language of likelihood ratios (*LR*). Let the *MIN-LR problem* be as follows: For constant n and parameter m, the input is $n-1$ nonnegative weights $\{\alpha_{i'}\}$ that sum to 1; $n-1$ product distributions $\{q_{i'}\}$; and a product distribution q_i; where all product distributions are over m items M. The goal is to minimize the likelihood ratio $\frac{\sum_{i'} \alpha_{i'} q_{i',S}}{q_{i,S}}$ over all outcomes $S \subseteq M$, where the numerator is the likelihood of S under the weighted combination distribution $\sum_{i'} \alpha_{i'} q_{i'}$, and the denominator is the likelihood of S under distribution q_i. Observe that a weighted combination distribution is *not* in general a product distribution itself, so the problem might be challenging. Denote the optimal solution to MIN-LR (the minimum likelihood ratio) by ρ^*.

Solving the separation oracle problem turns out to be NP-hard (see Proposition B.1 in Appendix C),[7] but we can give an FPTAS for the MIN-LR problem (Lemma 3.3, proof in Appendix E). Lemma 3.4 states the guarantee from applying this FPTAS to solve the separation oracle problem.

LEMMA 3.3 (FPTAS). *There is an algorithm for the MIN-LR problem that returns an outcome S with likelihood ratio $\leq (1+\delta)\rho^*$ in time polynomial in $m, \frac{1}{\delta}$.*

LEMMA 3.4. *If the FPTAS for the MIN-LR problem with parameter δ does not find a violated constraint of dual LP* (2.2) *(i.e., returns an outcome with likelihood ratio $\geq 1 - 1/(\sum_{i'\neq i} \lambda_{i'})$), then* for every S *the dual constraint* (3.1) *holds approximately up to $(1+\delta)$:*

$$(\sum_{i'\neq i} \lambda_{i'}) - 1 \leq (1+\delta) \sum_{i'\neq i} \lambda_{i'} \frac{q_{i',S}}{q_{i,S}}.$$

Proof. Assume there exists S such that $(\sum_{i'\neq i} \lambda_{i'}) - 1 > (1+\delta) \sum_{i'\neq i} \lambda_{i'} \frac{q_{i',S}}{q_{i,S}}$. Then dividing by $(\sum_{i'} \lambda_{i'})$ and using the definition of ρ^* as the minimum likelihood ratio we get $1 - \frac{1}{\sum_{i'} \lambda_{i'}} > (1+\delta)\rho^*$. Combining this with the guarantee of Lemma 3.3, the FPTAS returns S' with likelihood ratio $< 1 - \frac{1}{\sum_{i'} \lambda_{i'}}$, thus identifying a violated constraint. This completes the proof. ∎

[7] In fact the problem is strongly NP-hard; but because it involves products of the form $q_{i,S} = (\prod_{j\in S} q_{i,j})(\prod_{j\notin S}(1-q_{i,j}))$, the strong NP-hardness does not rule out an FPTAS [47, Theorem 17.12].

Applying the separation oracle FPTAS: The standard method. Given an FPTAS with parameter δ for the separation oracle of a dual LP, for many problems it is possible to find in polynomial time an approximately-optimal, feasible solution to the primal—see, e.g., [37, 14, 35, 44, 27, 26]. We first describe a fairly standard approach in the literature to utilizing a separation oracle FPTAS, which we refer to as the *standard method*, and explain where we must deviate from this approach. The proof of Theorem 3.1 then applies an appropriately modified approach.

The standard method works as follows: Let OPT_i be the optimal value of the primal (minimization) LP. For a benchmark value Γ, add to the (maximization) dual LP a constraint that requires its objective to be at least Γ, and attempt to solve the dual by running the ellipsoid algorithm with the separation oracle FPTAS.

Assume first that the ellipsoid algorithm returns a solution with value Γ. Since the separation oracle applies the FPTAS, it may wrongly conclude that some solution is feasible despite a slight violation of one or more of the constraints. For example, if we were to apply the FPTAS separation oracle from Lemma 3.3 to solve dual LP (2.2), we could possibly get a solution for which there exists S such that:

$$\sum_{i'\neq i}\lambda_{i'}\frac{q_{i',S}}{q_{i,S}} < (\sum_{i'\neq i}\lambda_{i'}) - 1 \leq (1+\delta)\sum_{i'\neq i}\lambda_{i'}\frac{q_{i',S}}{q_{i,S}}$$

where the second inequality is by Lemma 3.4. Clearly, the value Γ of an approximately-feasible solution may be higher than OPT_i. In the standard method, the approximately-feasible solution can be *scaled* by $\frac{1}{1+\delta}$ to regain feasibility while maintaining value of $\frac{\Gamma}{1+\delta}$. Scaling thus establishes that $\frac{\Gamma}{1+\delta} \leq OPT_i$. Now assume that for some (larger) value of Γ, the ellipsoid algorithm identifies that the dual LP is infeasible. In this case we can be certain that $OPT_i < \Gamma$, and we can also find in polynomial time a primal feasible solution with value $< \Gamma$ (more details in the proof of Theorem 3.1 below).

Using binary search (in our case over the range $[c_i, R_i] \subseteq [0,1]$ since R_i is the maximum the principal can pay without losing money), the standard method finds the smallest Γ^* for which the dual is identified to be infeasible, up to a negligible binary search error ϵ. This gives a primal feasible solution that achieves value $\Gamma^* + \epsilon$, and at the same time establishes that $\frac{(\Gamma^*)^-}{1+\delta} \leq OPT_i$ by the scaling argument, which is equivalent to $\frac{\Gamma^*}{1+\delta} \leq OPT_i$.[8] So the standard method has found an approximately-optimal, feasible solution to the primal.

Applying the separation oracle FPTAS: Our method. The issue with applying the standard method to solve MIN-PAYMENT is that the scaling argument does not hold. To see this, consider an approximately-feasible dual solution for which $(\sum_{i'\neq i}\lambda_{i'}) - 1 \leq (1+\delta)\sum_{i'\neq i}\lambda_{i'}\frac{q_{i',S}}{q_{i,S}}$ for every S, and notice that scaling the values $\{\lambda_{i'}\}$ does not achieve feasibility. We therefore turn to an alternative method to prove Theorem 3.1.

Proof of Theorem 3.1. We apply the standard method using the FPTAS with parameter δ (see Lemma 3.3) as separation oracle to the following *strengthened* version of dual LP (2.2),[9] where the extra $(1+\delta)$ multiplicative factor in the constraints makes

[8]The notation $(\Gamma^*)^-$ means any number smaller than Γ^*.
[9]Strengthened duals appear, e.g., in [44, 26].

them harder to satisfy:

$$
\begin{aligned}
(3.2)\qquad & \max \sum_{i'\neq i} \lambda_{i'}(c_i - c_{i'}) \\
& \text{s.t. } (1+\delta)\big((\sum_{i'\neq i}\lambda_{i'}) - 1\big) \le \sum_{i'\neq i}\lambda_{i'}\frac{q_{i',S}}{q_{i,S}} && \forall S\subseteq E, q_{i,S}>0 \\
& \lambda_{i'} \ge 0 && \forall i'\neq i, i'\in[n].
\end{aligned}
$$

Let Γ^* be the infimum value for which dual LP (3.2) would be identified as infeasible. The ellipsoid algorithm is thus able to find an approximately-feasible solution to dual LP (3.2) with objective $(\Gamma^*)^-$. The key observation is that this solution is *fully* feasible with respect to the original dual LP (2.2). This is because if the separation oracle FPTAS does not find a violated constraint of dual LP (3.2), then for every S it holds that $(\sum_{i'\neq i}\lambda_{i'}) - 1 \le \sum_{i'\neq i}\lambda_{i'}\frac{q_{i',S}}{q_{i,S}}$ (by the same argument as in the proof of Lemma 3.4). From the key observation it follows that

$$
(3.3)\qquad (\Gamma^*)^- \le OPT_i
$$

(despite the fact that the scaling argument does not hold).

Now let $\Gamma^*+\epsilon$ be the smallest value for which the binary search runs the ellipsoid algorithm for dual LP (3.2) and identifies its infeasibility. During its run for $\Gamma^*+\epsilon$, the ellipsoid algorithm identifies polynomially-many separating hyperplanes that constrain the objective to $<\Gamma^*+\epsilon$. Formulate a "small" primal LP with variables corresponding exactly to these hyperplanes. By duality, the small primal LP has a solution with objective $<\Gamma^*+\epsilon$, and moreover since the number of variables and constraints is polynomial we can find such a solution p^* in polynomial time. Observe that p^* is also a feasible solution to the primal LP corresponding to dual (3.2) (the only difference from the small LP is more variables):

$$
\begin{aligned}
(3.4)\qquad & \min\ (1+\delta)\sum_{S\subseteq E} q_{i,S}p_S \\
& \text{s.t. } (1+\delta)\big(\sum_{S\subseteq E} q_{i,S}p_S\big) - c_i \ge \sum_{S\subseteq E} q_{i',S}p_S - c_{i'} && \forall i'\neq i, i'\in[n] \\
& p_S \ge 0 && \forall S\subseteq E.
\end{aligned}
$$

We have thus obtained a contract p^* that is a feasible solution to LP (3.4) with objective $(1+\delta)\sum_{S\subseteq E} q_{i,S}p_S < \Gamma^*+\epsilon$. For action a_i, this contract pays the agent an expected transfer of $\sum_{S\subseteq E} q_{i,S}p_S < \frac{\Gamma^*+\epsilon}{1+\delta}$. We have the following chain of inequalities: $\sum_{S\subseteq E} q_{i,S}p_S \le \frac{(\Gamma^*)^-+\epsilon}{1+\delta} \le \frac{OPT_i+\epsilon}{1+\delta} \le OPT_i$, where the second inequality is by (3.3), and the last inequality is by taking the binary search error to be sufficiently small.[10] To complete the proof we must show that p^* is δ-IC. This holds since the constraints of LP (3.4) ensure that for every action $a_{i'}\neq a_i$, using the notation $p_i = \sum_{S\subseteq E} q_{i,S}p_S$, we have $p_{i'} - c_{i'} \le (1+\delta)p_i - c_i \le p_i - c_i + \delta p_i \le p_i - c_i + \delta$ (the last inequality uses that $p_i \le R_i \le 1$ by normalization). □

4. Hardness of approximation. In this section unlike the previous one, the number of actions is no longer assumed to be constant. We show a hardness of

[10] We use here that $OPT_i \ge c_i$ and that the number of bits of precision is polynomial.

approximation result for optimal contracts, based on the known hardness of approximation for MAX-3SAT. In his landmark paper, Håstad [33] shows that it is NP-hard to distinguish between a satisfiable MAX-3SAT instance, and one in which there is no assignment satisfying more than $7/8+\alpha$ of the clauses, where α is an arbitrarily-small constant (Theorems 5.6 and 8.3 in [33]). We build upon this to prove our main technical contribution stated in Theorem 4.1, which immediately leads to our main results for this section in Corollaries 4.2-4.3.

Theorem 4.1. *Let $c \in \mathbb{Z}, c \geq 3$ be an (arbitrarily large) constant integer. Let $\epsilon, \Delta \in \mathbb{R}, \epsilon > 0, \Delta \in [0, \frac{1}{20^c}]$ be such that $\frac{\epsilon - 2\Delta^{1/c}}{3} \in (0, \frac{1}{20}]$ and $(\frac{\epsilon-2\Delta^{1/c}}{3})^c$ is an (arbitrarily small) constant. Then it is NP-hard to determine whether a principal-agent setting has an IC contract extracting full expected welfare, or whether there is no Δ-IC contract extracting $> \frac{1}{c} + \epsilon$ of the expected welfare.*

We present two direct implications of Theorem 4.1. First, Corollary 4.2 applies to the OPT-CONTRACT problem, and states hardness of approximation within any constant of the optimal expected payoff by an IC contract. (A similar result can be shown for MIN-PAYMENT; see Appendix F.)

Corollary 4.2. *For any constant $c \in \mathbb{R}, c \geq 1$, it is NP-hard to approximate the optimal expected payoff achievable by an IC contract to within a multiplicative factor c.*

Corollary 4.2 suggests that in order to achieve positive results, we may want to follow the approach of the CDW framework and relax IC to Δ-IC. That is, instead of trying to compute in polynomial time an approximately-optimal IC contract, we should try to compute in polynomial time a Δ-IC contract with expected payoff that is guaranteed to approximately exceed that of the optimal IC contract. The next corollary establishes a computational limitation on this approach: Corollary 4.3 fixes a constant approximation factor c, and derives Δ for which a c-approximation by a Δ-IC contract is NP-hard to find. (It is also possible to reverse the roles—fix Δ and derive a constant approximation factor for which NP-hardness holds.) We shall complement this limitation with a positive result in Section 5.

Corollary 4.3. *For any constant $c \in \mathbb{R}, c \geq 5$ and $\Delta \leq (\frac{1}{4c})^c$, it is NP-hard to find a Δ-IC contract that guarantees $> \frac{2}{c}OPT$, where OPT is the optimal expected payoff achievable by an IC contract.*[11]

Proof. The corollary follows from Theorem 4.1 by setting $\epsilon = \frac{1}{c}$. ∎

It also follows from Theorem 4.1 and Corollary 4.3 that for every c, Δ as specified, it is NP-hard to approximate the optimal expected payoff achievable by a Δ-IC contract to within a multiplicative factor $c/2$. That is, hardness of approximation also holds for δ-OPT-CONTRACT.

In the remainder of the section we prove Theorem 4.1. After a brief overview, Section 4.2 sets up some tools for the proof, in Section 4.3 we focus on the special case of $c = 2$, and in Section 4.4 we prove the more general statement for any constant c.

4.1. Proof overview. It will be instructive to consider first a version of Theorem 4.1 for the case of $c = 2$:

Theorem 4.4. *Let $\epsilon, \Delta \in \mathbb{R}, \epsilon > 0, \Delta \in [0, \frac{1}{20^2}]$ be such that $\frac{\epsilon-2\Delta^{1/2}}{3} \in (0, \frac{1}{20}]$ and $(\frac{\epsilon-2\Delta^{1/2}}{3})^2$ is an (arbitrarily small) constant. Then it is NP-hard to determine*

[11] The relevant hardness notion is more accurately FNP-hardness.

	SAT item 1	…	SAT item m	Gap item
SAT action 1, gap action 1	SAT setting probabilities			ϵ
…				…
SAT action n, gap action 1				ϵ
Gap action 2	0.5	…	0.5	1

Fig. 1: Outline of a product setting for $c = 2$.

whether a principal-agent setting has an IC contract extracting full expected welfare, or whether there is no Δ-IC contract extracting $> \frac{1}{2} + \epsilon$ of the expected welfare.

This theorem is already interesting as it shows that even relaxing IC to Δ-IC where $\Delta \gg 0$, approximating the optimal expected payoff within 65% is computationally hard:

Corollary 4.5. *For any $\Delta \leq \frac{1}{20^2}$, it is NP-hard to find a Δ-IC contract that guarantees $> 0.65 \cdot OPT$, where OPT is the optimal expected payoff achievable by an IC contract.*

Proof. The corollary follows from Theorem 4.4 by setting $\epsilon = \frac{3}{20}$. ∎

To establish Theorem 4.4 we present a gap-preserving reduction from any MAX-3SAT instance φ to a principal-agent setting that we call the "product setting" (the reduction appears in Algorithm 4.2 and is analyzed in Proposition 4.15). The product setting encompasses a 2-action, 1-item principal-agent "gap setting", in which any δ-IC contract for sufficiently small δ cannot extract much more than $\frac{1}{2}$ of the expected welfare (Proposition 4.8). The "gap setting" is coupled with a useful gadget we call the "SAT setting", which is a principal-agent setting with n actions and m items whose probabilities depend on the 3SAT instance φ. See Figure 1 to see how the gap and SAT settings are combined to form the product setting.

The important property of the SAT setting is the following: if assigning TRUE to exactly the variable subset S satisfies the 3SAT formula, then item subset S occurs in the SAT setting with probability zero for every action. This property becomes useful once the gap actions are added to this gadget (see Figure 1). In particular, "gap action 2" achieves set S with non-zero probability, and so a contract paying only for set S can incentivize this action by just covering its cost, thus extracting the full welfare. If on the other hand, the 3SAT formula is unsatisfiable, then the "gap" in the gap setting kicks in and prevents any contract from extracting more than $\frac{1}{2}$ of the expected welfare.

Constant $\mathbf{c > 2}$. The special case of $c = 2$ captures most ideas behind the proof of the more general Theorem 4.1, but the analysis is simplified by the fact that to extract more than roughly $\frac{1}{2}$ of the expected welfare in the 2-action gap setting, there is a single action that the contract could potentially incentivize. The more general case involves gap settings with more actions (the reduction appears in Algorithm 4.3 and is analyzed in Proposition 4.17). To extract more than $\approx \frac{1}{c}$ of the expected welfare, the contract could potentially incentivize almost any one of these actions (Proposition 4.9).

Barrier to going beyond constant c. Our techniques for establishing Theorem 4.1 do not generalize beyond constant values of c (the approximation factor). The reason for this is that we do not know of (c, ϵ, f)-gap settings (Definition 4.6) where

$f(c,\epsilon) = o(\epsilon^c)$. As long as $f(c,\epsilon)$ is of order ϵ^c, the gap in the MAX-3SAT instance we reduce from must be between $7/8 + \epsilon^c$ and 1, and this gap problem is known to be NP-hard only for constant c. As [33] notes, significantly stronger complexity assumptions may lead to hardness for slightly (but not significantly) larger values of c.

4.2. Main tools used in the proof. In this section we formalize the notions of "gap" and "SAT" principal-agent settings as well as the notion of an "average action", which will be useful in proving Theorems 4.1 and 4.4. The term "gap setting" reflects the gap between the first-best solution (i.e., the expected welfare), and the second-best solution (i.e., the expected payoff to the principal from the optimal contract). It will be convenient *not* to normalize gap settings (and thus also the product settings encompassing them). This makes our negative results only stronger, as we show next.

Unnormalized settings and a stronger δ-IC notion. Before proceeding we must define what we mean by a δ-IC contract in an unnormalized setting. Moreover we show that if Theorems 4.1 or 4.4 hold for unnormalized settings with the new δ-IC notion, then they also hold for normalized settings with the standard δ-IC notion.

Recall that in a *normalized* setting, action a_i that is δ-incentivized by the contract must satisfy δ-IC constraints of the form $p_i - c_i + \delta \geq p_{i'} - c_{i'}$ for every $i' \neq i$. In an *unnormalized* setting, an additive δ-deviation from optimality is too weak of a requirement; we require instead that a_i satisfy δ-IC constraints of the form

$$(1+\delta)p_i - c_i \geq p_{i'} - c_{i'} \quad \forall i' \neq i. \tag{4.1}$$

Two key observations are: (i) The constraints in (4.1) imply the standard δ-IC constraints if $p_i \leq 1$, as is the case if the setting is normalized; (ii) The constraints in (4.1) are invariant to scaling of the setting and contract (i.e., to a change of currency of the rewards, costs and payments). By these observations, a δ-IC contract according to the new notion in an unnormalized setting becomes a standard δ-IC contract after normalization of the setting and payments, with the same fraction of optimal expected welfare extracted as payoff to the principal.

Assume a negative result holds for unnormalized settings, i.e., it is NP-hard to determine between the two cases stated in Theorem 4.1 (or Theorem 4.4). Assume for contradiction this does not hold for normalized settings. Then given an unnormalized setting, we can simply scale the expected rewards and costs to normalize it, and then determine whether or not there is an IC contract extracting full expected welfare. If such a contract exists, it is also IC and full-welfare-extracting in the unnormalized setting after scaling back the payments. On the other hand, by the discussion above, if there is no standard-notion Δ-IC contract extracting a given fraction of the expected welfare in the normalized setting, there can also be no such contract with the new Δ-IC notion in any scaling of the setting. We have thus reached a contradiction to NP-hardness. We conclude that proving our negative results for unnormalized settings only strengthens these results.

Gap settings and their construction. We now turn to the definition of gap settings.

DEFINITION 4.6 (Unstructured gap setting). *Let $f(c,\epsilon) \in \mathbb{R}_{\geq 0}$ be an increasing function where $c \in \mathbb{Z}_{>0}$ and $\epsilon \in \mathbb{R}_{>0}$. An* unstructured (c,ϵ,f)-gap setting *is a principal-agent setting such that for every $0 \leq \delta \leq f(c,\epsilon)$, the optimal δ-IC contract can extract no more than $\frac{1}{c} + \epsilon$ of the expected welfare as the principal's expected payoff.*

For convenience we focus on (structured) gap settings as follows.

Definition 4.7 (Gap setting). *A* (c, ϵ, f)-gap setting *is a setting as in Definition 4.6 with the following structure: there is a single item and c actions; the first action has zero cost; the last action has probability 1 for the item and maximum expected welfare among all actions.*

To construct a gap setting, we construct a principal-agent setting with a single item, c actions and parameter $\gamma \in \mathbb{R}_{>0}, \gamma < 1$. The construction is similar to [23], but requires a different analysis. For every $i \in [c]$, set the probability of action a_i for the item to γ^{c-i}, and set a_i's cost to $c_i = (1/\gamma^{i-1}) - i + (i-1)\gamma$. Set the reward for the item to be $1/\gamma^{c-1}$. Observe that the expected welfare of action a_i is $i - (i-1)\gamma$, so the last action has the maximum expected welfare $c - (c-1)\gamma$. This establishes the structural requirements from a gap setting (Definition 4.7). Propositions 4.8 and 4.9 establish the gap requirements from a gap setting (Definition 4.6) for $c = 2$ and $c \geq 3$, respectively—the separation between these cases is for clarity of presentation. We use the former in Section 4.3, in which we show hardness for the $c = 2$ case; the latter is a generalization to arbitrary-large constant c. See Appendix G for proofs.

Proposition 4.8 (2-action gap settings). *For every* $\epsilon \in (0, \frac{1}{4}]$*, there exists a* $(2, \epsilon, \epsilon^2)$*-gap setting.*

Proposition 4.9 (c-action gap settings). *For every* $c \geq 3$ *and* $\epsilon \in (0, \frac{1}{4}]$*, there exists a* $(c, \epsilon, \epsilon^c)$*-gap setting.*

For concreteness we describe the 2-action gap setting: The agent has $c = 2$ actions, which can be thought of as "effort" and "no effort". Effort has cost $\frac{1}{\epsilon} - 2 + \epsilon$, and no effort has cost 0. Without effort the item has probability ϵ, and with effort the probability is 1. The reward associated with the item is $\frac{1}{\epsilon}$. It is immediate to see that the maximum expected welfare (first-best) is $2 - \epsilon$. In the proof of Proposition 4.8 we show that the best an ϵ^2-IC contract can extract is ≈ 1.

Average actions and SAT settings. The motivation for the next definition is that given a contract, for an action to be IC or δ-IC it must yield higher expected utility for the agent in comparison to the "average action". Average actions are thus a useful tool for analyzing contracts.

Definition 4.10 (Average action). *Given a principal-agent setting and a subset of actions, by the* average action *we refer to a hypothetical action with the average of the subset's distributions, and average cost. (If a particular subset is not specified, the average is taken over all actions in the setting.)*

Another useful ingredient will be SAT settings defined as follows.

Definition 4.11 (SAT setting). *A* SAT *principal-agent setting corresponds to a MAX-3SAT instance* φ*. If* φ *has* n *clauses and* m *variables then the SAT setting has* n *actions and* m *items. Two conditions hold: (1)* φ *is satisfiable if and only if there is an item set in the SAT setting that the average action leads to with zero probability; (2) If every assignment to* φ *satisfies at most* $7/8 + \alpha$ *of the clauses, then for every item set* S *the average action leads to* S *with probability at least* $\frac{1-8\alpha}{2^m}$*.*

The following proposition provides a reduction from MAX-3SAT instances to SAT settings.

Proposition 4.12. *For every* φ *the reduction in Algorithm 4.1 runs in polynomial time on input* φ *and returns a SAT setting corresponding to* φ*.*

Proof of Proposition 4.12. We first argue that there is a satisfying assignment to the MAX-3SAT instance if and only if there is a set S with 0-probability in every one

Algorithm 4.1 SAT setting construction in polytime

Input : A MAX-3SAT instance φ with n clauses and m variables.
Output: A principal-agent SAT setting (Definition 4.11) corresponding to φ.
begin
Given φ, construct a principal-agent setting in which every clause corresponds to an action with a product distribution, and for every variable there is a corresponding item. If variable j appears in clause i of φ as a positive literal, then let item j's probability in the ith product distribution be 0, and if it appears as a negative literal then let item j's probability be 1. Set all other probabilities to be $\frac{1}{2}$. We set the costs of all actions and the rewards for all items to be 0.
end

of the product distributions. First note that there is a natural 1-to-1 correspondence between subsets $\{S\}$ of items and truth assignments to the variables: for every variable j, if item $j \in S$ then assign TRUE and otherwise FALSE. Now consider a set S and its corresponding assignment. S has 0-probability in the ith product distribution iff either an item in S has probability 0 or an item in $\overline{S}$ has probability 1 according to this distribution. Therefore, in clause i, either one of the TRUE variables appears as a positive literal or one of the FALSE variables appears as a negative literal. And this is a necessary and sufficient condition for the clause to be satisfied. We conclude that S has 0-probability in every product distribution if and only if the corresponding assignment satisfies every clause, establishing condition (1) of Definition 4.11. To show condition (2), assume that at most $\frac{7}{8} + \alpha$ of the clauses can be satisfied. Consider the average action whose distribution results from averaging over all actions. This distribution has for every S a probability at least $(\frac{1}{8} - \alpha) \cdot \frac{8}{2^m} = \frac{1-8\alpha}{2^m}$, since the probability of S is $\frac{8}{2^m}$ in every distribution corresponding to a clause which the assignment corresponding to S does not satisfy. This completes the proof. ∎

4.3. The $\mathbf{c = 2}$ case: Proof of Theorem 4.4. In this section we present a polynomial-time reduction from MAX-3SAT to a product setting, which combines gap and SAT settings. The reduction appears in Algorithm 4.2. We then analyze the guarantees of the reduction and use them to prove Theorem 4.4. Most of the analysis appears in Proposition 4.15, which shows that the reduction in Algorithm 4.2 is gap-preserving. Some of the results are formulated in general terms so they can be reused in the next section (Section 4.4).

Before turning to Proposition 4.15, we begin with two simple observations about the product setting resulting from the reduction.

Observation 4.13. *Partition all actions of the product setting but the last one into blocks of n actions each.*[12] *Every action in the ith block has the same expected reward for the principal as action a_i in the gap setting, and the last action in the product setting has the same expected reward as the last action in the gap setting.*

Corollary 4.14. *The optimal expected welfares of the product and gap settings are the same, and are determined by their respective last actions.*

Proposition 4.15 (Gap preservation by Algorithm 4.2). *Let φ be a MAX-3SAT instance for which either there is a satisfying assignment, or every assignment satisfies at most $7/8 + \alpha$ of the clauses for $\alpha \leq (0.05)^2$. Let $\Delta \leq (0.05)^2$. Consider*

[12] If the number of actions in the gap setting is 2, there is a single such block.

Algorithm 4.2 Polytime reduction from MAX-3SAT to principal-agent

Input : A MAX-3SAT instance φ with n clauses and m variables; a parameter $\epsilon \in \mathbb{R}_{\geq 0}$.

Output: A principal-agent *product setting* combining a *SAT setting* and a *gap setting*.

begin

Combine the SAT setting corresponding to φ (attainable in polytime by Proposition 4.12) with a poly-sized $(2, \epsilon, \epsilon^2)$-gap setting (exists by Proposition 4.8) to get the product setting, as follows:

- The product setting has $n+1$ actions and $m+1$ items: m "SAT items" correspond to the SAT setting items, and the last "gap item" corresponds to the gap setting item.
- The upper-left block of the product setting's $(n+1) \times (m+1)$ matrix of probabilities is the SAT setting's $n \times m$ matrix of probabilities. The entire lower-left $1 \times m$ block is set to $\frac{1}{2}$. The entire upper-right $n \times 1$ block is set to the probability that action a_1 in the gap setting results in the item. The remaining lower-right 1×1 block is set to the probability that the last action (i.e., action a_2) in the gap setting results in the item (recall that this probability is 1).
- In the product setting, the rewards for the m SAT items are set to 0, and the reward for the gap item is set as in the gap setting.
- The costs of the first n actions in the product setting are the cost of action a_1 in the gap setting; the cost of the last action in the product setting is the cost of the last action (i.e., action a_2) in the gap setting.

end

the product setting resulting from the reduction in Algorithm 4.2 run on input $\varphi, \epsilon = 3\alpha^{1/2} + 2\Delta^{1/2} \leq \frac{1}{4}$. Then:

1. *If φ has a satisfying assignment, the product setting has an IC contract that extracts full expected welfare;*
2. *If every assignment to φ satisfies at most $7/8+\alpha$ of the clauses, the optimal Δ-IC contract can extract no more than $\frac{1}{2} + \epsilon$ of the expected welfare.*

Proof. First, if φ has a satisfying assignment, then there is a subset of SAT items that has zero probability according to every one of the first n actions. Consider the outcome S^* combining this subset together with the gap item. We construct a full-welfare extracting contract: the contract's payment for S^* is the cost of the last action in the product setting multiplied by 2^m (since the probability of S^* according to the last action is $1/2^m$), and all other payments are set to zero. It is not hard to see that the resulting contract makes the agent indifferent among all actions, so by tie-breaking in favor of the principal, the principal receives the full expected welfare as her payoff.

Now consider the case that every assignment to φ satisfies at most $7/8 + \alpha$ of the clauses, and assume for contradiction that there is a Δ-IC contract p for the product setting that extracts more than $\frac{1}{2} + \epsilon$ of the expected welfare. We derive from p a δ-IC contract p' for the $(2, \epsilon, \epsilon^2)$-gap setting where $\delta \leq \epsilon^2$, which extracts more than $\frac{1}{2} + \epsilon$ of the expected welfare. This is a contradiction to the properties of the gap setting (Definition 4.6).

It remains to specify and analyze contract p' : For brevity we denote the singleton containing the gap item by M', and define

$$p'(S') = \frac{1-8\alpha}{2^m} \sum_{S \subseteq [m]} p(S \cup S') \forall S' \subseteq M', \tag{4.2}$$

where S' is either the singleton containing the gap item or the empty set. The starting point of the analysis is the observation that to extract $> \frac{1}{2} + \epsilon$ of the expected welfare in the product setting, contract p must Δ-incentivize the last action (this follows since the expected rewards and costs of the actions are as in the gap setting by Observation 4.13, and so the same argument as in the proof of Proposition 4.8 holds).

Claim 4.16 below establishes that if contract p Δ-incentivizes the last action in the product setting, then contract p' δ-incentivizes the last action in the gap setting for $\delta = \frac{8\alpha+\Delta}{1-8\alpha}$. So indeed

$$\begin{aligned}
\delta &= \frac{8\alpha}{1-8\alpha} + \frac{\Delta}{1-8\alpha} \\
&\leq 9\alpha + 4\Delta \\
&= (3\alpha^{1/2})^2 + (2\Delta^{1/2})^2 \\
&\leq (3\alpha^{1/2} + 2\Delta^{1/2})^2 \quad = \epsilon^2,
\end{aligned}$$

using that $\alpha, \Delta \leq (0.05)^2$ for the first inequality.

Now observe that the expected payoff to the principal from contract p' that δ-incentivizes the last gap setting action is at least that of contract p that Δ-incentivizes the last product setting action: the payments of p' as defined in (4.2) are the average payments of p lowered by a factor of $(1 - 8\epsilon)$, and the expected rewards in the two settings are the same (Observation 4.13). The expected welfares in the two settings are also equal (Corollary 4.14). We conclude that like contract p in the product setting, contract p' guarantees extraction of $> \frac{1}{2} + \epsilon$ of the expected welfare in the gap setting. This leads to a contradiction and completes the proof of Proposition 4.15 (up to Claim 4.16 proved below). ∎

The next claim is formulated in general terms so that it can also be used in Section 4.4. It references the contract p' defined in (4.2).

Claim 4.16. *Assume every assignment to the MAX-3SAT instance φ satisfies at most $7/8 + \alpha$ of its clauses where $\alpha < \frac{1}{8}$, and consider the product and gap settings returned by the reduction in Algorithm 4.2 (resp., Algorithm 4.3). If in the product setting the last action is Δ-incentivized by contract p, then in the gap setting the last action is δ-incentivized by contract p' for $\delta = \frac{8\alpha+\Delta}{1-8\alpha}$.*

Proof. Let g_i denote the distribution of action a_i in the gap setting and let c be the number of actions in this setting. In the product setting, by construction its last action assigns probability $\frac{g_c(S')}{2^m}$ to every set $S \cup S'$ such that S contains SAT items and $S' \subseteq M'$. Thus the expected payment for the last action given contract p is

$$\sum_{S \subseteq [m]} \sum_{S' \subseteq M'} \frac{g_c(S')}{2^m} p(S \cup S') = \frac{1}{1-8\alpha} \sum_{S' \subseteq M'} g_c(S') p'(S'), \tag{4.3}$$

where the equality follows from the definition of p' in (4.2). Note that the resulting expression in (4.3) is precisely the expected payment for the last action in the gap setting given contract p', multiplied by factor $1/(1-8\alpha)$.

Similarly, for every $i \in c$ consider the average action over the ith block of n actions in the product setting.[13] Again by construction, the probability this ith average action assigns to $S \cup S'$ is $\geq \frac{g_i(S')(1-8\alpha)}{2^m}$, where we use that the average action of the SAT setting has probability $\geq \frac{1-8\alpha}{2^m}$ for S (Definition 4.11). Thus the expected payment for the ith average action given contract p is at least

$$\sum_{S \subseteq [m]} \sum_{S' \subseteq M'} \frac{g_i(S')(1-8\alpha)}{2^m} p(S \cup S') = \sum_{S' \subseteq M'} g_i(S') p'(S') \qquad \forall i \in [c], \tag{4.4}$$

where again the equality follows from (4.2). Note that the resulting expression in (4.4) is precisely the expected payment for action a_i in the gap setting given contract p'.

We now use the assumption that in the product setting, contract p Δ-incentivizes the last action. This means the agent Δ-prefers the last action to the ith average action, which has cost zero. Combining (4.3) and (4.4) we get

$$\frac{1+\Delta}{1-8\alpha} \sum_{S' \subseteq M'} g_c(S') p'(S') - \mathcal{C} \geq \sum_{S' \subseteq M'} g_i(S') p'(S') \qquad \forall i \in [c], \tag{4.5}$$

where $\mathcal{C}$ denotes the cost of the last action in the product and gap settings. By definition of δ-IC, Inequality (4.5) immediately implies that in the gap setting, the last action is δ-IC given contract p' where $\delta = \frac{8\alpha+\Delta}{1-8\alpha}$, thus completing the proof of Claim 4.16. ◻

We can now use Proposition 4.15 to prove Theorem 4.4.

Proof of Theorem 4.4. Recall that $\frac{(\epsilon-2\Delta^{1/2})^2}{9}$ is a constant $\leq (0.05)^2$. Assume a polynomial-time algorithm for determining whether a principal-agent setting has a (fully-IC) contract that extracts the full expected welfare, or whether no Δ-IC contract can extract more than $\frac{1}{2}+\epsilon$. Then given a MAX-3SAT instance φ for which either there is a satisfying assignment or every assignment satisfies at most $\frac{7}{8} + \frac{(\epsilon-2\Delta^{1/2})^2}{9}$ of the clauses, by Proposition 4.15 the product setting (constructed in polynomial time) either has a full-welfare extracting contract or has no Δ-IC contract that can extract more than $\frac{1}{2}+\epsilon$. Since the algorithm can determine among these two cases, it can solve the MAX-3SAT instance φ. But by [33] and since $\frac{(\epsilon-2\Delta^{1/2})^2}{9}$ is a constant, we know that there is no polynomial-time algorithm for solving such MAX-3SAT instances unless $P = NP$. This completes the proof of Theorem 4.4. ◻

4.4. The general case: Proof of Theorem 4.1. In this section we formulate and analyze the guarantees of the reduction in Algorithm 4.3.

PROPOSITION 4.17 (Gap preservation by Algorithm 4.3). *Let $c \in \mathbb{Z}, c \geq 3$. Let φ be a MAX-3SAT instance for which either there is a satisfying assignment, or every assignment satisfies at most $7/8 + \alpha$ of the clauses for $\alpha \leq (0.05)^c$. Let $\Delta \leq (0.05)^c$. Consider the product setting resulting from the reduction in Algorithm 4.3 run on input $\varphi, c, \epsilon = 3\alpha^{1/c} + 2\Delta^{1/c} \leq \frac{1}{4}$. Then:*

1. *If φ has a satisfying assignment, the product setting has an IC contract that extracts full expected welfare;*
2. *If every assignment to φ satisfies at most $7/8+\alpha$ of the clauses, the optimal Δ-IC contract can extract no more than $\frac{1}{c}+\epsilon$ of the expected welfare.*

[13] If $c = 2$ there is a single such block.

Algorithm 4.3 Generalized polytime reduction from MAX-3SAT to principal-agent

Input : A MAX-3SAT instance φ with n clauses and m variables; parameters $\epsilon \in \mathbb{R}_{\geq 0}$ and $c \in \mathbb{Z}_{>0}$ where $c \geq 3$.

Output: A principal-agent *product setting* combining copies of a *SAT setting* and a *gap setting*.

begin

Combine multiple copies of the SAT setting corresponding to φ (attainable in polytime by Proposition 4.12) with a poly-sized $(c, \epsilon, \epsilon^c)$-gap setting (exists by Proposition 4.9) to get the product setting, as follows:

- The product setting has $cn + 1$ actions and $m + 1$ items: m "SAT items" correspond to the SAT setting items, and the last "gap item" corresponds to the gap setting item.
- For every $i \in [c]$, consider the ith block of n rows of the product setting's $(cn + 1) \times (m + 1)$ matrix of probabilities. The ith block consists of row $(i - 1) \cdot n + 1$ to row $i \cdot n$ and forms a submatrix of size $n \times (m + 1)$. The first m columns of the sub-matrix are set to a copy of the SAT setting's $n \times m$ matrix of probabilities, and the entire last column is set to the probability that action a_i in the gap setting results in the item. Finally, the first m entries of the last row of the product setting's matrix (i.e., row $cn + 1$) are set to $\frac{1}{2}$, and the last entry (the lower-right corner of the matrix) is set to the probability that the last action in the gap setting results in the item.
- In the product setting, the rewards for the m SAT items are set to 0, and the reward for the gap item is set as in the gap setting.
- For every $i \in [c]$, the costs of the n actions in block i are the cost of action a_i in the gap setting; the cost of the last action in the product setting is the cost of the last action in the gap setting.

end

Proof. First, if φ has a satisfying assignment, then there is a subset of SAT items that has zero probability according to every one of the actions in the product setting except for the last action, and so we can construct a full-welfare extracting contract as in the proof of Proposition 4.15. From now on consider the case that every assignment to φ satisfies at most $7/8 + \alpha$ of the clauses, and assume for contradiction there is a Δ-IC contract p for the product setting that extracts more than $\frac{1}{c} + \epsilon$ of the expected welfare.

Consider the case that p Δ-incentivizes the last action in the product setting. Then we can derive from it a δ-IC contract p' for the $(c, \epsilon, \epsilon^c)$-gap setting where $\delta \leq \epsilon^c$, which extracts more than $\frac{1}{c} + \epsilon$ of the expected welfare. This is a contradiction to the properties of the gap setting (Definition 4.6). The construction of p' and its analysis are as in the proof of Proposition 4.15 (where Equation (4.2) defines p'), and so are omitted here except for the following verification: we must verify that indeed $\delta \leq \epsilon^c$. We know from Claim 4.16 that $\delta = \frac{8\alpha+\Delta}{1-8\alpha}$. As in the proof of Proposition 4.15 this is $\leq 9\alpha + 4\Delta$, and it is not hard to see that

$$9\alpha + 4\Delta \leq (3\alpha^{1/c})^c + (2\Delta^{1/c})^c \leq (3\alpha^{1/c} + 2\Delta^{1/c})^c = \epsilon^c,$$

where the first inequality uses that $c \geq 3$.

In the remaining case, p Δ-incentivizes an action a_{i^*k} in the product setting which is the kth action in block $i^* \in [c]$ (recall each block has n actions). We derive from p a contract p'_k (depending on k) for the gap setting that Δ-incentivizes a_{i^*} at the same expected payment. As in the proof of Proposition 4.17, this means that p'_k extracts $> \frac{1}{c} + \epsilon$ of the expected welfare in the gap setting. Since $\Delta \leq \delta = \frac{8\alpha+\Delta}{1-8\alpha}$ it follows from the argument above that $\Delta \leq \epsilon^c$, and so we have reached a contradiction to the properties of the gap setting (Definition 4.6).

We define p'_k as follows: Let s_k denote the distribution of action a_k in the SAT setting. For every subset $S' \subseteq M'$ of gap items,

$$p'_k(S') = \sum_{S \subseteq [m]} p(S \cup S') s_k(S) \qquad \forall S' \subseteq M', \tag{4.6}$$

where S' is either the singleton containing the gap item or the empty set.

For the analysis, let g_i denote the distribution of action a_i in the gap setting. In the product setting, for every $i \in [c], k \leq n$ the expected payment for action a_{ik} by contract p is

$$\sum_{S \in [m]} \sum_{S' \subseteq M'} s_k(S) g_i(S') p(S \cup S'). \tag{4.7}$$

In the gap setting, the expected payment for a_i by contract p'_k is $\sum_{S' \subseteq M'} g_i(S') p'(S')$, and by definition of p'_k in (4.6) this coincides with the expected payment in (4.7). We know that contract p Δ-incentivizes a_{i^*k} in the product setting, in particular against any action a_{ik} where $i \in [c] \setminus \{i^*\}$ (i.e., against actions in the same position k but in different blocks). This implies that contract p'_k Δ-incentivizes a_{i^*} in the gap setting against any action a_i, completing the proof. ∎

We can now use Proposition 4.17 to prove Theorem 4.1. The proof is identical to that of Theorem 4.4 and so is omitted here.

5. Approximation guarantees.

In this section we show that for any constant δ there is a simple, namely linear, δ-IC contract that extracts as expected payoff for the principal a c_δ-fraction of the optimal welfare, where c_δ is a constant that depends only on δ. Recall that a linear contract is defined by a parameter $\alpha \in [0, 1]$, and pays the agent $p_S = \alpha \sum_{j \in S} r_j$ for every outcome $S \subseteq M$.

THEOREM 5.1. *Consider a principal-agent setting with n actions. For every $\delta > 0$ let $c_\delta = \max_{\gamma \in (0,1)} (1-\gamma)(\lceil \log_{1+\delta}(\frac{1}{\gamma}) \rceil + 1)^{-1}$. Then there is a δ-IC linear contract with expected payoff ALG where*

$$ALG \geq c_\delta \cdot \max_{i \in [n]} \{R_i - c_i\}.$$

An immediate corollary of Theorem 5.1 is that we can compute a δ-IC linear contract that achieves a constant-factor approximation in polynomial time. By Corollary 4.2 we cannot achieve a similar result for IC (rather than δ-IC) contracts unless $P = NP$. In fact, an even stronger lower bound holds for the class of exactly IC linear (or, more generally, separable) contracts. These contracts cannot achieve an approximation ratio better than n (see [23] and Appendix H for details).

5.1. Geometric understanding of linear contracts.

To prove Theorem 5.1 we will rely on the following geometric understanding of linear contracts developed in [23]. Fix a principal-agent setting. For a linear contract with parameter $\alpha \in [0, 1]$ and an action a_i, the expected reward $R_i = \sum_S q_{i,S} r_S$ is split between the principal and the agent, leaving the principal with $(1-\alpha)R_i$ in expected utility and the agent with $\alpha R_i - c_i$ (the sum of the players' expected utilities is action a_i's expected welfare).

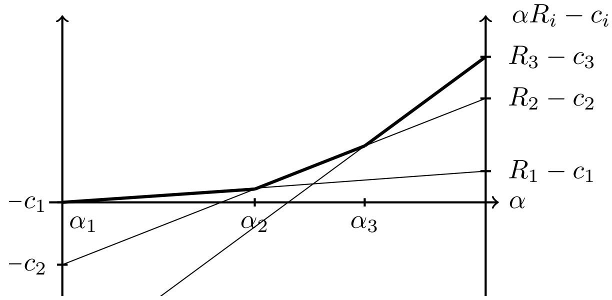

Fig. 2: Upper envelope diagram for linear contracts.

The agent's expected utility for choosing action a_i *as a function of* α is thus a line from $-c_i$ (for $\alpha = 0$) to $R_i - c_i$ (for $\alpha = 1$). Drawing these lines for each of the n actions, we trace the maximum the agent's utility for his best action as α goes from 0 to 1. This gives us the *upper envelope* diagram for linear contracts in the given principal-agent setting.

Figure 2 illustrates the construction and enables a few key observations that hold in general. A first observation is that only actions that appear on the upper envelope can be incentivized, and for each action that can be incentivized the smallest α for which this action is part of the upper envelope is the one that yields the highest expected payoff for the principal. Moreover, if we index actions from left to right as they appear on the upper envelope, then they will be sorted by increasing welfare $R_i - c_i$, increasing expected reward R_i, and increasing cost c_i as these correspond to the intercept of $\alpha R_i - c_i$ with the y-axis at $\alpha = 1$, the slope of $\alpha R_i - c_i$, and the intercept of $\alpha R_i - c_i$ with the y-axis at $\alpha = 0$.

In the remainder of this section, we will use I_N for the subset of $N \leq n$ actions that are implementable by some linear contract, and we will index them in the order in which they appear on the upper envelope. Note that then $i < i'$ implies that $c_i < c_{i'}$, $R_i < R_{i'}$, and $R_i - c_i < R_{i'} - c_{i'}$. Moreover, $\max_i\{R_i - c_i\} = R_N - c_N$ as the action with the highest welfare must appear on the upper envelope.

For every action $a_i \in I_N$, we denote by α_i the smallest parameter α of a linear contract that incentivizes a_i. Note that because of our assumption that the minimum cost of any action is 0, we have that $\alpha_1 = 0$.

5.2. Bucketing construction. Our proof of Theorem 5.1 relies on a bucketing construction that is parametrized by $\delta > 0$ and $\gamma \in (0, 1)$. We describe this construction below, and visualize it in Figure 3.

For a fixed $\delta > 0$ and fixed $\gamma \in (0, 1)$ we subdivide the range $[0, 1]$ of α-parameters into $\kappa + 1 = \lceil \log_{1+\delta}(\frac{1}{\gamma}) \rceil + 1$ buckets as follows:

$$\begin{aligned}
B_1 &= [0, \gamma(1+\delta)^0), \\
B_k &= [\gamma(1+\delta)^{k-2}, \gamma(1+\delta)^{k-1}) \qquad\qquad \text{for } k \in \{2, \ldots, \kappa\}, \\
B_{\kappa+1} &= [\gamma(1+\delta)^{\kappa-1}, 1].
\end{aligned}$$

For each bucket B_k with $k \in [\kappa + 1]$ we now specify an action $a_{h(k)}$. If bucket B_k has a single action a_i that is implementable with an $\alpha \in B_k$, then we let $a_{h(k)} = a_i$. Otherwise, if bucket B_k has more than one action a_i that is implementable with an $\alpha \in B_k$, then we let $a_{h(k)}$ be the action a_i with the highest expected reward that is

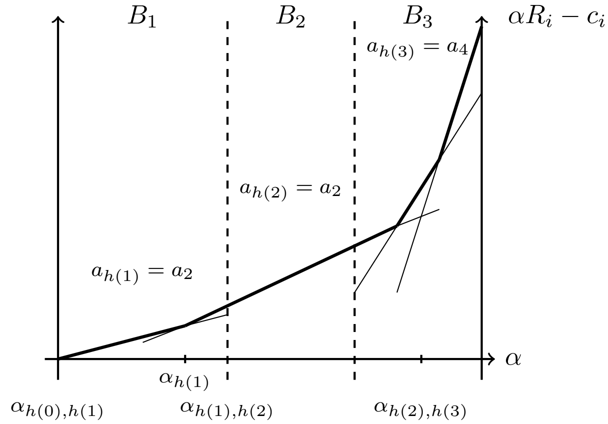

Fig. 3: Bucketing construction.

implementable with an $\alpha \in B_k$.

Next for each bucket B_k and associated action $a_{h(k)}$ we define a value of α, which we will denote by $\alpha_{h(k-1),h(k)}$. For $k = 1$ we set $\alpha_{h(k-1),h(k)} = 0$. For $k \geq 2$ we distinguish between the case where B_k has exactly one implementable action, and the case where it has more than one. If it has exactly one implementable action we set $\alpha_{h(k-1),h(k)} = \gamma(1+\delta)^{k-2}$, i.e., we define $\alpha_{h(k-1),h(k)}$ to be the left endpoint of B_k. Note that in this case $h(k) = h(k-1)$ and so

$$R_{h(k)} - c_{h(k)} = R_{h(k-1)} - c_{h(k-1)}.$$

Otherwise, if B_k has more than one implementable action, then we have $h(k) > h(k-1)$ and therefore also $R_{h(k)} > R_{h(k-1)}$, and we set

$$\alpha_{h(k-1),h(k)} = \frac{c_{h(k)} - c_{h(k-1)}}{R_{h(k)} - R_{h(k-1)}},$$

i.e., in this case $\alpha_{h(k-1),h(k)}$ is the α that makes the agent indifferent between actions $a_{h(k-1)}$ and $a_{h(k)}$.

5.3. Upper bound on the optimal welfare. The first key ingredient in our proof of Theorem 5.1 will be the following upper bound on the optimal welfare $\max_{i\in[n]}(R_i - c_i) = R_N - c_N$ in terms of the parameters of the bucketing construction in Section 5.2 for any $\delta > 0$ and $\gamma \in (0,1)$.

LEMMA 5.2. *Fix $\delta > 0$ and $\gamma \in (0,1)$ and consider the bucketing construction from Section 5.2. Then,*

$$\max_{i\in[n]}(R_i - c_i) = R_N - c_N \leq \sum_{k=1}^{\kappa+1}(1 - \alpha_{h(k-1),h(k)})R_{h(k)}.$$

To prove Lemma 5.2 we rely on the following observation from [23].

OBSERVATION 5.3. *Consider two actions $a_i, a_{i'}$ such that a_i has higher expected reward and higher welfare than $a_{i'}$, i.e., $R_i > R_{i'}$ and $R_i - c_i > R_{i'} - c_{i'}$, and let $\alpha_{i',i} = (c_i - c_{i'})/(R_i - R_{i'})$. Then*

$$(R_i - c_i) - (R_{i'} - c_{i'}) \leq (1 - \alpha_{i',i})R_i.$$

Proof of Lemma 5.2. We argue by induction that for all $k \geq 1$, $R_{h(k)} - c_{h(k)} \leq \sum_{i=1}^{k}(1-\alpha_{h(i-1),h(i)})R_{h(i)}$. For $k=1$, recall that $\alpha_{h(0),h(1)} = 0$ by definition, and it trivially holds that $R_{h(1)} - c_{h(1)} \leq R_{h(1)}$. Now assume that the inequality holds for $k-1$, i.e.,

$$R_{h(k-1)} - c_{h(k-1)} \leq \sum_{i=1}^{k-1}(1-\alpha_{h(i-1),h(i)})R_{h(i)}. \tag{5.1}$$

If B_k is a bucket that contains only one implementable action, then $h(k) = h(k-1)$ and thus $(R_{h(k)} - c_{h(k)}) - (R_{h(k-1)} - c_{h(k-1)}) = 0$. So, in particular, $(R_{h(k)} - c_{h(k)}) - (R_{h(k-1)} - c_{h(k-1)}) \leq (1-\alpha_{h(k-1),h(k)})R_{h(k)}$.

Otherwise, if B_k is a bucket that contains more than one implementable action, then $h(k) > h(k-1)$ and thus $R_{h(k)} > R_{h(k-1)}$ and $R_{h(k)} - c_{h(k)} > R_{h(k-1)} - c_{h(k-1)}$. So we can apply Observation 5.3 to actions $a_{h(k)}$ and $a_{h(k-1)}$. This shows $(R_{h(k)} - c_{h(k)}) - (R_{h(k-1)} - c_{h(k-1)}) \leq (1-\alpha_{h(k-1),h(k)})R_{h(k)}$.

We conclude that in both cases $(R_{h(k)} - c_{h(k)}) - (R_{h(k-1)} - c_{h(k-1)}) \leq (1-\alpha_{h(k-1),h(k)})R_{h(k)}$. Adding this inequality to inequality (5.1) we obtain

$$R_{h(k)} - c_{h(k)} \leq \sum_{i=1}^{k}(1-\alpha_{h(i-1),h(i)})R_{h(i)},$$

as claimed. ∎

5.4. Approximate implementability.

5.4. Approximate implementability. The second crucial observation concerning the bucketing construction in Section 5.2 for any fixed $\delta > 0$ and $\gamma \in (0,1)$ concerns the (approximate) implementability of the actions $a_{h(k)}$ for $k \in [\kappa+1]$.

For $k = 1$, action $a_{h(1)}$ is incentivized exactly at α_1. For $k \geq 2$ and buckets B_k that contain only one implementable action, action $a_{h(k)}$ is incentivized exactly at $\alpha_{h(k-1),h(k)}$. For $k \geq 2$ and buckets B_k that contain more than one implementable action, action $a_{h(k)}$ is not incentivized exactly at $\alpha_{h(k-1),h(k)}$, but—as the following lemma shows—it is δ-incentivized.

LEMMA 5.4. *Fix $\delta > 0$ and $\gamma \in (0,1)$ and consider the bucketing construction from Section 5.2. For any $k \in \{2, \ldots, \kappa+1\}$ such that B_k contains more than one implementable action, the linear contract with $\alpha = \alpha_{h(k-1),h(k)}$ ensures that*

$$\alpha R_{h(k)} - c_{h(k)} + \delta \geq \alpha R_i - c_i \qquad \text{for every } i \in [n].$$

Proof. The lines $R_{h(k)} - c_{h(k)}$ and $R_{h(k-1)} - c_{h(k-1)}$ intersect at $\alpha_{h(k-1),h(k)}$. By construction, their intersection must fall between, on the one hand, the left endpoint $\gamma(1+\delta)^{k-2}$ of the bucket in which $\alpha_{h(k)}$ falls, and $\alpha_{h(k)}$ on the other hand. This shows that $(1+\delta)\alpha_{h(k-1),h(k)} \geq (1+\delta)\gamma(1+\delta)^{k-2} = \gamma(1-\delta)^{k-1} \geq \alpha_{h(k)}$. Combining this with the fact that $a_{h(k)}$ is incentivized exactly at $\alpha_{h(k)}$, we obtain that $\alpha_{h(k-1),h(k)}R_{h(k)} - c_{h(k)} + \delta \geq (1+\delta)\alpha_{h(k-1),h(k)}R_{h(k)} - c_{h(k)} \geq \alpha_{h(k)}R_{h(k)} - c_{h(k)} \geq \alpha_{h(k)}R_i - c_i$ for all $i \in [n]$, where the first inequality holds since $R_{h(k)} \leq 1$ by normalization. ∎

5.5. Proof of the approximation guarantee. We are now ready to prove Theorem 5.1. We will use the bucketing construction from Section 5.2, and we will use Lemma 5.2 to derive an upper bound on the optimal welfare and Lemma 5.4 to derive a lower bound on what a δ-IC linear contract can achieve.

Proof of Theorem 5.1. Fix some $\delta > 0$ and some $\gamma \in (0,1)$, and consider the bucketing construction from Section 5.2 for these parameters. Write ALG for the payoff achievable with a δ-IC linear contract, and OPT for the maximum welfare of any action. For the linear contract we consider choosing the best α among $\alpha_{h(1)}$ and $\alpha_{h(k-1),h(k)}$ for $k \geq 2$. We then have,

$$\begin{aligned} ALG &\geq \max\{(1-\alpha_{h(1)})R_{h(1)}, (1-\alpha_{h(1),h(2)})R_{h(2)}, \ldots, (1-\alpha_{h(\kappa),h(\kappa+1)})R_{h(\kappa+1)}\} \\ &\geq (1-\gamma)\max\{(1-\alpha_{h(0),h(1)})R_{h(1)}, (1-\alpha_{h(1),h(2)})R_{h(2)}, \\ &\qquad\qquad \ldots, (1-\alpha_{h(\kappa),h(\kappa+1)})R_{h(\kappa+1)}\} \\ &\geq (1-\gamma)\frac{1}{\kappa+1}\sum_{i=1}^{\kappa+1}(1-\alpha_{h(k-1),h(k)})R_{h(k)} \\ &\geq (1-\gamma)\frac{1}{\kappa+1}OPT, \end{aligned}$$

where for the first inequality we use Lemma 5.4, for the second inequality we use that $\alpha_{h(1)} \leq \gamma$ and that $\alpha_{h(0),h(1)} \geq 0$, for the third inequality we lower bound the maximum with the average, and for the final inequality we use Lemma 5.2.

The proof is completed by observing that for a fixed $\delta > 0$ the above argument applies for all $\gamma \in (0,1)$. We can thus conclude that

$$ALG \geq \max_{\gamma\in(0,1)}(1-\gamma)\frac{1}{\lceil \log_{1+\delta}(\frac{1}{\gamma})\rceil + 1}OPT,$$

as claimed. ∎

6. Black-box model.

We conclude by considering a *black-box model* which concerns non-necessarily succinct principal-agent settings. In this model, the principal knows the set of actions A_n, the cost c_i of each action $a_i \in A_n$, the set of items M and the rewards r_j for each item $j \in M$, but does not know the probabilities $q_{i,S}$ that action a_i assigns to outcome $S \subseteq M$. Instead, the principal has *query access* to the distributions $\{q_i\}$. Upon querying distribution q_i of action a_i, a (random) set is returned where S is selected with probability $q_{i,S}$. Our goal is to study how well a δ-IC contract in this model can approximate the optimal IC contract if limited to a polynomial number of queries (where the guarantees should hold with high probability over the random samples). Black-box models have been studied in other algorithmic game theory contexts such as signaling—see [22] for a successful example.

Let $\eta = \min\{q_{i,S} \mid i \in [n], S \subseteq M, q_{i,S} \neq 0\}$ be the minimum non-zero probability of any set of items under any of the actions. Note that then either $q_{i,S} = 0$ or $q_{i,S} \geq \eta$ for every S. In Section 6.1 we address the case in which η is inverse super-polynomial and obtain a negative result; in Section 6.2 we show a positive result for the case of inverse polynomial η.

6.1. Inverse super-polynomial probabilities.

We show a negative result for the case where the minimum probability η is inverse super-polynomial, by proving that $\text{poly}(1/\sqrt{\eta})$ samples are required to obtain a constant factor multiplicative approximation better than ≈ 1.15. The negative result holds even for succinct settings, in which the unknown distributions are product distributions.

The basic idea is to construct two nearby instances, which, with high probability, cannot be distinguished with polynomially many samples, and for which no single contract can simultaneously be good for both settings.

THEOREM 6.1. *Assume $\eta \le \eta_0 = 1/625$ and $\delta \le \delta_0 = 1/100$. Even with $n = 2$ actions and $m = 2$ items, achieving a multiplicative ≤ 1.15 approximation to the optimal IC contract through a δ-IC contract, where the approximation guarantee is required to hold with probability at least $1-\gamma$, may require at least $s \ge -\log(\gamma)/(9\sqrt{\eta})$ queries.*

Proof. We consider a scenario with two settings, both of which have $n = 2$ actions and $m = 2$ items, and which differ only in the probabilities of the items given the second action. Let τ be some constant > 2 (to be fixed later), and let $\mu = \frac{\sqrt{\eta}}{\tau}$. Let $\beta = (1 + \frac{1}{\tau^2})^{-1}$ and note that $\beta < 1$.

Setting I:

	$r_1 = \frac{\beta}{\tau^2\mu}$	$r_2 = \frac{\beta}{\tau^2\mu}$	
a_1 :	$\tau\mu$	$\tau\mu$	$c_1 = 0$
a_2 :	$\tau^2\mu$	μ	$c_2 = \frac{\tau-1}{\tau^3}\frac{1}{1-\mu}\beta$

Setting II:

	$r_1 = \frac{\beta}{\tau^2\mu}$	$r_2 = \frac{\beta}{\tau^2\mu}$	
a_1 :	$\tau\mu$	$\tau\mu$	$c_1 = 0$
a_2 :	μ	$\tau^2\mu$	$c_2 = \frac{\tau-1}{\tau^3}\frac{1}{1-\mu}\beta$

Note further that the minimum probability of any set of items in both settings is $q_{2,\{1,2\}} = \tau^2\mu^2 = \eta$, as required by definition of η.

The expected reward achieved by the two actions in the two settings is $R_1 = 2\beta/\tau < 1$ and $R_2 = (1 + 1/\tau^2)\beta = 1$. Moreover, the cost of action 2 is $c_2 \le \beta/\tau^2$. So the welfare achieved by the two actions is $R_1 - c_1 < \beta$ and $R_2 - c_2 \ge \beta$.

In both settings the optimal IC contract incentivizes action 2, by paying only for the set of items that maximizes the likelihood ratio. In Setting 1 this is $\{1\}$, in Setting 2 it is $\{2\}$. The payment for this set in both cases is $c_2/(\tau^2\mu(1-\mu) - \tau\mu(1-\tau\mu)) = c_2/(\tau^2\mu - \tau\mu)$. This leads to an expected payment of $\tau^2\mu(1-\mu) \cdot c_2/(\tau^2\mu - \tau\mu) = \beta/\tau^2$. The resulting payoff (and our benchmark) is therefore $R_2 - \beta/\tau^2 = \beta$.

We now argue that if we cannot distinguish between the two settings, then we can only achieve a ≈ 1.1568 approximation. Of course, we can always pay nothing and incentivize action 1, but this only yields a payoff of $2\beta/\tau$. We can also try to δ-incentivize action 2 in both settings, by paying for outcome $\{1\}$ and $\{2\}$. But (as we show below) the payoff that we can achieve this way is (for $\delta \to 0$ and $\mu \to 0$) at most $(1 + 1/\tau^2 - (\tau^2+1)/((\tau-1)\tau^3)\beta$. Now $\max\{2/\tau, 1 + 1/\tau^2 - (\tau^2+1)/((\tau-1)\tau^3\}$ is minimized at $\tau = 1 + \sqrt{2}$ where it is $2/(1+\sqrt{2}) \approx 0.8284$. The upper bound on the payoff from action 2 for this choice of τ is actually increasing in both μ and δ and equal to $\approx 0.8644 \cdot \beta$ at the upper bounds $\mu_0 = \sqrt{\eta_0}/(2^2) = 1/100$ and $\delta_0 = 1/100$, implying that the best we can achieve without knowing the setting is a $\approx 1/0.8644 \approx 1.1568$ approximation.

So if we want to achieve at least a ≤ 1.15 approximation with probability at least $1 - \gamma$, then we need to be able to distinguish between the two settings with at least this probability. A necessary condition for being able to distinguish between the two settings is that we see at least some item in one of our queries to action 2. So,

$$1 - \gamma \le 1 - (1 - \tau^2\mu)^{2s},$$

which implies that $s \ge \log(\gamma)/(2\log(1-\tau^2\mu) \ge -\log(\gamma)/(2 \cdot \mu \cdot \tau^2) \ge -\log(\gamma)/(18\mu)$. Plugging in μ we get $s \ge -\log(\gamma)/(18\frac{\sqrt{\mu}}{\tau}) > -\log(\gamma)/(9\sqrt{\mu})$.

We still need to prove our claims regarding the payoff that we can achieve if we want to δ-incentivize action 2 in both settings. To this end consider the IC constraints for δ-incentivizing action 2 over action 1 in Setting I and Setting II, respectively:

$$\tau^2\mu(1-\mu)p_{\{1\}} + (1-\tau^2\mu)\mu p_{\{2\}} - c_2 \geq \tau\mu(1-\tau\mu)p_{\{1\}} + (1-\tau\mu)\tau\mu p_{\{2\}} - \delta, \quad \text{and}$$
$$(1-\tau^2\mu)\mu p_{\{1\}} + \tau^2\mu(1-\mu)p_{\{2\}} - c_2 \geq \tau\mu(1-\tau\mu)p_{\{1\}} + (1-\tau\mu)\tau\mu p_{\{2\}} - \delta.$$

Adding up these constraints yields

$$(\tau^2\mu(1-\mu) + (1-\tau^2\mu)\mu - 2\tau\mu(1-\tau\mu)) \cdot (p_{\{1\}} + p_{\{2\}}) \geq 2c_2 - 2\delta.$$

We maximize the minimum performance across the two settings by choosing $p_{\{1\}} = p_{\{2\}}$. Letting $p = p_{\{1\}} = p_{\{2\}}$ we thus obtain

$$(\tau^2\mu(1-\mu) + (1-\tau^2\mu)\mu - 2\tau\mu(1-\tau\mu))p \geq c_2 - \delta.$$

It follows that

$$p \geq \frac{c_2 - \delta}{\tau^2\mu + \mu - 2\tau\mu}.$$

The performance of the optimal contract that δ-incentivizes action 2 in both settings thus achieves an expected payoff of

$$R_2 - (\tau^2\mu(1-\mu) + (1-\tau^2\mu)\mu)\frac{c_2 - \delta}{\tau^2\mu + \mu - 2\tau\mu} = R_2 - \frac{\tau^2(1-2\mu)+1}{(\tau-1)^2}(c_2 - \delta).$$

Plugging in R_2 and c_2 and letting $\delta \to 0$ and $\mu \to 0$ we obtain the aforementioned $1 + 1/\tau^2 - (\tau^2+1)/((\tau-1)\tau^3)\beta$. Finally, to see that the expected payoff evaluated at $\tau = 1 + \sqrt{2} > 2$ is increasing in both δ and μ observe that the derivative in δ is simply the probability term $(\tau^2(1-2\mu)+1)/(\tau-1)^2$ which is positive and that both this probability term and the cost c_2 are decreasing in μ implying that as μ increases we subtract less. ∎

6.2. Inverse polynomial probabilities. We show a positive result for the case where the minimum probability η is inverse polynomial. Namely, let OPT denote the expected payoff of the optimal IC contract; then with $\text{poly}(n, m, \frac{1}{\eta}, \frac{1}{\epsilon}, \frac{1}{\gamma})$ queries it is possible to find, with probability at least $(1-\gamma)$, a 4ϵ-IC contract with expected payoff at least $OPT - 5\epsilon$. Formally:

THEOREM 6.2. *Fix $\epsilon > 0$, and assume $\epsilon \leq 1/2$. Fix distributions Q such that $q_{i,S} \geq \eta$ for all $i \in [n]$ and $S \subseteq M$. Denote the expected payoff of the optimal IC contract for distributions Q by OPT. Then there is an algorithm that with $s = (3\log(\frac{2n}{\eta\gamma}))/(\eta\epsilon^2)$ queries to each action and probability at least $1-\gamma$, computes a contract $\tilde{p}$ which (i) is 4ϵ-IC on the actual distributions Q; and (ii) has expected payoff Π on the actual distributions satisfying $\Pi \geq OPT - 5\epsilon$.*

We will show that the optimal 2ϵ-IC contract for the empirical distributions obtained from $s = (3\log(\frac{2n}{\eta\gamma}))/(\eta\epsilon^2)$ queries to each action has the desired properties.[14]

[14]Note that this contract can be computed in polynomial time by solving $n-1$ LPs similar to the MIN-PAYMENT LP, with an appropriately relaxed IC constraint, because there will be at most ns outcomes with a non-zero probability.

Our proof goes through a series of technical lemmas (Lemmas 6.3 to 6.7), which we describe and state below, and whose proofs appear in Appendix I.

The first lemma (Lemma 6.3) establishes that $s = (3\log(\frac{2n}{\eta\gamma}))/(\eta\epsilon^2)$ queries to each action suffice to ensure that with probability at least $1-\gamma$ all empirical probabilities are within an error of at most ϵ of the actual probabilities.

LEMMA 6.3. *Consider the algorithm that issues s queries to each action $i \in N$, and sets $\tilde{q}_{i,S}$ to be the empirical probability of set S under action i. With $s = (3\log(\frac{2n}{\eta\gamma}))/(\eta\epsilon^2)$ queries to each action, with probability at least $1-\gamma$, for all $i \in [n]$ and $S \subseteq M$,*

$$(1-\epsilon)q_{i,S} \leq \tilde{q}_{i,S} \leq (1+\epsilon)q_{i,S}.$$

The remaining lemmas (Lemma 6.4 to Lemma 6.7) all operate on the assumption that the empirical probabilities are close to the actual probabilities.

The first two of these lemmas—Lemma 6.4 and Lemma 6.5—show that IC and δ-IC are approximately preserved when switching from the actual distributions to the empirical distributions, and vice versa.

We will use Lemma 6.4 to relate the performance of the optimal 2ϵ-IC contract for the empirical distributions to that of the optimal IC contract for the actual distributions. We will use Lemma 6.5 to show that the optimal 2ϵ-IC contract for the empirical distributions is 4ϵ-IC under the actual distributions.

LEMMA 6.4. *Suppose that $(1-\epsilon)q_{i,S} \leq \tilde{q}_{i,S} \leq (1+\epsilon)q_{i,S}$ for all $i \in [n]$ and $S \subseteq M$. Consider contract p. If a_i is the action that is incentivized by this contract under the actual probabilities Q, then the payoff of a_i under the empirical distributions $\tilde{Q}$ is at least as high as that of any other action up to an additive term of 2ϵ.*

LEMMA 6.5. *Suppose that $(1-\epsilon)q_{i,S} \leq \tilde{q}_{i,S} \leq (1+\epsilon)q_{i,S}$ for all $i \in [n]$ and $S \subseteq M$. Consider contract $\tilde{p}$. If a_i is the action that is δ-incentivized by this contract under the empricial probabilities $\tilde{Q}$, then the payoff of a_i under the actual distributions is at least as high as that of any other action up to an additive term of $\delta + 2\epsilon$. $(\delta + 2\epsilon)$-IC for the actual probabilities Q.*

The final two lemmas (Lemma 6.6 and Lemma 6.7) relate the payoff of an action on the actual distributions to that on the empirical distributions, and vice versa.

We will use these lemmas to connect the performance of the two aforementioned contracts under the empirical and actual distributions.

LEMMA 6.6. *Suppose that $(1-\epsilon)q_{i,S} \leq \tilde{q}_{i,S} \leq (1+\epsilon)q_{i,S}$ for all $i \in [n]$ and $S \subseteq M$. If action a_i achieves payoff $\tilde{\Pi}$ under contract $\tilde{p}$ when evaluated on the empirical distributions $\tilde{Q}$, then it achieves payoff $\Pi \geq \tilde{\Pi} - 2\epsilon$ when evaluated on the actual distributions Q.*

LEMMA 6.7. *Assume $\epsilon \leq 1/2$. Suppose that $(1-\epsilon)q_{i,S} \leq \tilde{q}_{i,S} \leq (1+\epsilon)q_{i,S}$ for all $i \in [n]$ and $S \subseteq M$. If action a_i achieves payoff P under contract p when evaluated on the actual distributions Q, then it achieves payoff $\tilde{P} \geq P - 3\epsilon$ when evaluated on the empirical distributions Q.*

We are now ready to prove the theorem.

Proof of Theorem 6.2. Let $\tilde{Q}$ denote the empirical distributions that result from querying each action s times. By Lemma 6.3, with probability at least $1-\gamma$, the empirical probabilities obtained in this way will satisfy $(1-\epsilon)q_{i,S} \leq \tilde{q}_{i,S} \leq (1+\epsilon)q_{i,S}$ for all $i \in [n]$ and $S \subseteq M$.

Denote the optimal 2ϵ-IC contract for the empirical distributions $\tilde{Q}$ by $\tilde{p}$. We will use $\tilde{\Pi}$ for the expected payoff that this contract achieves under the empirical distributions $\tilde{Q}$, and Π for the expected payoff that it achieves under the actual distributions Q. Likewise, denote by p the optimal IC contract for the actual distributions Q. We will write P for the expected payoff that it achieves under the actual distributions Q, and $\tilde{P}$ for its expected payoff under the empirical distributions $\tilde{Q}$.

By Lemma 6.5, contract $\tilde{p}$ which is 2ϵ-IC on $\tilde{Q}$ is 4ϵ-IC on Q, as claimed. Furthermore, by Lemma 6.4, contract p which is IC on Q is 2ϵ-IC on $\tilde{Q}$. Since $\tilde{p}$ is the optimal such contract, this implies that $\tilde{\Pi} \geq \tilde{P}$. Together with Lemma 6.6 and Lemma 6.7 we thus obtain

$$\Pi \geq \tilde{\Pi} - 2\epsilon \geq \tilde{P} - 2\epsilon \geq P - 5\epsilon,$$

which completes the proof. ◻

REFERENCES

[1] G. A. Akerlof, *The market for "lemons": Quality uncertainty and the market mechanism*, The Quarterly Journal of Economics, 84 (1970), pp. 488–500.

[2] G. A. Akerlof, *Labor contracts as partial gift-exchange*, The Quarterly Journal of Economics, 97 (1982), pp. 543–569.

[3] P. D. Azar and S. Micali, *Computational principal-agent problems*, Theoretical Economics, 13 (2018), pp. 553–578.

[4] M. Babaioff, M. Feldman, N. Nisan, and E. Winter, *Combinatorial agency*, Journal of Economic Theory, 147 (2012), pp. 999–1034.

[5] M. Babaioff, Y. A. Gonczarowski, and N. Nisan, *The menu-size complexity of revenue approximation*, in STOC'17, 2017, pp. 869–877.

[6] M. Babaioff, N. Immorlica, B. Lucier, and S. M. Weinberg, *A simple and approximately optimal mechanism for an additive buyer*, in FOCS'14, 2014, pp. 21–30.

[7] M. Babaioff and E. Winter, *Contract complexity*, in EC'14, 2014, p. 911.

[8] Y. Cai, *Mechanism design: A new algorithmic framework*, PhD thesis, Massachusetts Institute of Technology (MIT), 2013.

[9] Y. Cai, C. Daskalakis, and S. M. Weinberg, *An algorithmic characterization of multi-dimensional mechanisms*, in STOC'12, 2012, pp. 459–478.

[10] Y. Cai, C. Daskalakis, and S. M. Weinberg, *Optimal multi-dimensional mechanism design: Reducing revenue to welfare maximization*, in FOCS'12, 2012, pp. 130–139.

[11] Y. Cai, C. Daskalakis, and S. M. Weinberg, *Understanding incentives: Mechanism design becomes algorithm design*, in FOCS'13, 2013, pp. 618–627.

[12] Y. Cai, N. R. Devanur, and S. M. Weinberg, *A duality based unified approach to bayesian mechanism design*, in STOC'16, 2016, pp. 926–939.

[13] B. Caillaud and B. E. Hermalin, *Hidden-information agency*. Lecture notes available from http://faculty.haas.berkeley.edu/hermalin/mechread.pdf, 2000.

[14] R. D. Carr and S. Vempala, *Randomized metarounding*, Random Structures and Algorithms, 20 (2002), pp. 343–352.

[15] G. Carroll, *A quantitative approach to incentives: Application to voting rules*. Working paper, 2013.

[16] G. Carroll, *Robustness and linear contracts*, American Economic Review, 105 (2015), pp. 536–563.

[17] Y. Cheng, H. Y. Cheung, S. Dughmi, E. Emamjomeh-Zadeh, L. Han, and S. Teng, *Mixture selection, mechanism design, and signaling*, in FOCS'15, 2015, pp. 1426–1445.

[18] C. Daskalakis and S. M. Weinberg, *Symmetries and optimal multi-dimensional mechanism design*, in EC'12, 2012, pp. 370–387.

[19] S. Dughmi, *On the hardness of signaling*, in FOCS'14, 2014, pp. 354–363.

[20] S. Dughmi, J. D. Hartline, R. Kleinberg, and R. Niazadeh, *Bernoulli factories and black-box reductions in mechanism design*, in STOC'17, 2017, pp. 158–169.

[21] S. Dughmi, N. Immorlica, and A. Roth, *Constrained signaling in auction design*, in SODA'14, 2014, pp. 1341–1357.

[22] S. Dughmi and H. Xu, *Algorithmic Bayesian persuasion*, in STOC'16, 2016, pp. 412–425.

[23] P. Dütting, T. Roughgarden, and I. Talgam-Cohen, *Simple versus optimal contracts*, in EC'19, 2019, pp. 369–387.

[24] Y. Emek and M. Feldman, *Computing optimal contracts in combinatorial agencies*, Theoretical Computer Science, 452 (2012), pp. 56–74.

[25] F. Englmaier and S. Leider, *Contractual and organizational structure with reciprocal agents*, American Economic Journal: Microeconomics, 4 (2012), pp. 146–183.

[26] M. Feldman, G. Kortsarz, and Z. Nutov, *Improved approximation algorithms for directed Steiner forest*, Journal of Computer and System Sciences, 78 (2012), pp. 279–292.

[27] L. Fleischer, M. X. Goemans, V. S. Mirrokni, and M. Sviridenko, *Tight approximation algorithms for maximum separable assignment problems*, Mathematics of Operations Research, 36 (2011), pp. 416–431.

[28] Y. A. Gonczarowski, *Bounding the menu-size of approximately optimal auctions via optimal-transport duality*, in STOC'18, 2018, pp. 123–131.

[29] Y. A. Gonczarowski and S. M. Weinberg, *The sample complexity of up-to-ϵ multi-dimensional revenue maximization*, in FOCS'18, 2018, pp. 416–426.

[30] S. J. Grossman and O. D. Hart, *An analysis of the principal-agent problem*, Econometrica, 51 (1983), pp. 7–45.

[31] J. D. Hartline, R. Kleinberg, and A. Malekian, *Bayesian incentive compatibility via matchings*, Games and Economic Behavior, 92 (2015), pp. 401–429.

[32] J. D. Hartline and B. Lucier, *Non-optimal mechanism design*, American Economic Review, 105 (2015), pp. 3102–3124.

[33] J. Håstad, *Some optimal inapproximability results*, Journal of the ACM, 48 (2001), pp. 798–859.

[34] C. Ho, A. Slivkins, and J. W. Vaughan, *Adaptive contract design for crowdsourcing markets: Bandit algorithms for repeated principal-agent problems*, Journal of Artificial Intelligence Research, 55 (2016), pp. 317–359.

[35] K. Jain, M. Mahdian, and M. R. Salavatipour, *Packing Steiner trees*, in SODA'03, 2003, pp. 266–274.

[36] A. X. Jiang and K. Leyton-Brown, *Polynomial-time computation of exact correlated equilibrium in compact games*, Games and Economic Behavior, 91 (2015), pp. 347–359.

[37] N. Karmarkar and R. M. Karp, *An efficient approximation scheme for the one-dimensional bin-packing problem*, in FOCS'82, 1982, pp. 312–320.

[38] J. M. Kleinberg and R. Kleinberg, *Delegated search approximates efficient search*, in EC'18, 2018, pp. 287–302.

[39] B. Koszegi, *Behavioral contract theory*, Journal of Economic Literature, 52 (2014), pp. 1075–1118.

[40] M. T. Kovalyov and E. Pesch, *A generic approach to proving NP-hardness of partition type problems*, Discrete Applied Mathematics, 158 (2010), pp. 1908–1912.

[41] S. Moran, *General approximation algorithms for some arithmetical combinatorial problems*, Theorertical Computer Science, 14 (1981), pp. 289–303.

[42] R. B. Myerson, *Optimal auction design*, Mathematics of Operations Research, 6 (1981), pp. 58–73.

[43] C. T. Ng, M. S. Barketau, T. C. E. Cheng, and M. T. Kovalyov, *"Product partition" and related problems of scheduling and systems reliability: Computational complexity and approximation*, European Journal of Operational Research, 207 (2010), pp. 601–604.

[44] Z. Nutov, I. Beniaminy, and R. Yuster, *A (1-1/e)-approximation algorithm for the generalized assignment problem*, Operations Research Letters, 34 (2006), pp. 283–288.

[45] C. H. Papadimitriou, *The complexity of finding Nash equilibria*, in Algorithmic Game Theory, N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, eds., Cambridge University Press, 2006, ch. 2, pp. 29–51.

[46] C. H. Papadimitriou and T. Roughgarden, *Computing correlated equilibria in multi-player games*, Journal of the ACM, 55 (2008), pp. 14:1–14:29.

[47] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: Algorithms and complexity*, Prentice-Hall, 1982.

[48] Royal Swedish Academy of Sciences, *Scientific background on the 2016 Nobel Prize in Economic Sciences*, 2016.

[49] B. Salanié, *The Economics of Contracts: A Primer*, MIT Press, 2005.

[50] D. Walton and G. Carroll, *When are robust contracts linear?* Working paper, 2019.

[51] S. M. Weinberg, *Algorithms for strategic agents*, PhD thesis, Massachusetts Institute of Technology (MIT), 2014.

Appendix A. Tractability of linear and separable contracts. Proposition A.1 establishes that the problem of finding an optimal IC or δ-IC linear resp. separable contract is tractable.

PROPOSITION A.1. *Let $\delta \geq 0$. Given a principal-agent setting, an optimal linear (resp., separable) δ-IC contract can be found in polynomial time.*

Proof. The problem of finding an optimal linear (resp., separable) δ-IC contract for incentivizing any action a_i can be formulated as a polynomial-sized LP with 1 variable (resp., m variables) representing the contract's parameter α (resp., the item payments $\{p_j\}$), and $n-1$ δ-IC constraints. ∎

Appendix B. Intractability of the ellipsoid method. In this appendix we establish the intractability of the ellipsoid method for MIN-PAYMENT, except for the special case of $n = 2$. Recall LP (2.1) for the MIN-PAYMENT problem. Its dual is as follows, where $\{\lambda_{i'}\}$ are $n-1$ nonnegative variables (one for every action other than i):

$$\max \sum_{i' \neq i} \lambda_{i'}(c_i - c_{i'})$$

$$\text{s.t. } \Big(\sum_{i' \neq i} \lambda_{i'}\Big) - 1 \leq \sum_{i' \neq i} \lambda_{i'} \frac{q_{i',S}}{q_{i,S}} \qquad \forall S \subseteq E, q_{i,S} > 0,$$

$$\lambda_{i'} \geq 0 \qquad \forall i' \neq i, i' \in [n].$$

Consider applying the ellipsoid method to solve LP (2.1) for action a_i. The separation oracle problem is: Given an instantiation of the dual variables $\{\lambda_{i'}\}$, consider the *combination distribution* $\sum_{i' \neq i} \lambda_{i'} q_{i'}$, which is a convex combination of the product distributions $\{q_{i'}\}$. To find a violated constraint of the dual LP we need to find a set S for which the likelihood ratio between the combination distribution and the product distribution q_i is sufficiently small.

Note that a combination distribution is *not* itself a product distribution.[15] Therefore solving the separation oracle is not easy and in fact it is an NP-hard problem even for $n = 3$, as formalized in Proposition B.1. In the special case of $n = 2$, the combination distribution *is* a product distribution. By taking S to be all items that are more likely according to q_i than according to the combination distribution, we minimize the likelihood ratio and solve the separation oracle. (This is one way to conclude that OPT-CONTRACT with $n = 2$ is tractable.)

PROPOSITION B.1. *Solving the separation oracle of dual LP (2.2) is NP-hard for $n \geq 3$.*

Proof. Rather than prove Proposition B.1 directly, it is enough to point the reader to Corollary D.2, which establishes the NP-hardness of MIN-PAYMENT. ∎

Remark B.2. Proposition B.1 immediately holds for δ-IC as well, i.e., for the separation oracle of dual LP (3.2). This dual corresponds to primal LP (3.4) solving MIN-PAYMENT for δ-IC contracts. This is simply because the separation oracle problem of dual LP (3.2) is identical to that of dual LP (2.2).

[15] For example, consider a fifty-fifty mix between the following two product distributions over two items: a point mass on the empty set, and a point mass on the grand bundle. This combination distribution has probability $\frac{1}{2}$ for the empty set and probability $\frac{1}{2}$ for the grand bundle, and the item marginals are $\frac{1}{2}$. A product distribution with item marginals of $\frac{1}{2}$ has probability $\frac{1}{4}$ for every set.

Appendix C. Properties of δ-IC contracts. In this appendix we give the proofs that were omitted from Section 2.4.

Proof of Proposition 2.3. Action a_i can be δ-implemented if and only if LP C.1 has a feasible solution.

$$\text{(C.1)} \qquad \begin{aligned} \min \quad & 0 \\ \text{s.t.} (1+\delta)\left(\sum_{S\subseteq E} q_{i,S}p_S\right) - c_i \geq \sum_{S\subseteq E} q_{i',S}p_S - c_{i'} & \quad \forall i' \neq i, i' \in [n] \\ p_S \geq 0 & \quad \forall S \subseteq E. \end{aligned}$$

Consider the dual:

$$\text{(C.2)} \qquad \begin{aligned} \max \quad & \sum_{i'\neq i} \lambda_{i'}(c_i - c_{i'}) \\ \text{s.t.} (1+\delta) q_{i,S} \sum_{i'\neq i} \lambda_{i'} \leq \sum_{i' \neq i} \lambda_{i'} q_{i',S} & \quad \forall S \subseteq E, q_{i,S} > 0 \\ \lambda_{i'} \geq 0 & \quad \forall i' \neq i, i' \in [n]. \end{aligned}$$

Since q_i and $\{q_{i'}\}$ are distributions and $\delta > 0$, the only feasible solution to the dual LP (C.2) is $\lambda_{i'} = 0$ for every $i' \neq i$. The dual is feasible and bounded, hence the primal must be feasible, completing the proof. ∎

Proof of Proposition 2.4. The expected payoff of action a_i under the interpolation contract p' is

$$R_i - [(1-\sqrt{\delta})p_i + \sqrt{\delta}R_i] = (1-\sqrt{\delta})(R_i - p_i).$$

We will argue that for every action $a_{i'}$ with $i' \neq i$, either i' is not incentivized by p' (Case 1) or its expected payoff is sufficiently high (Case 2).

Case 1: Assume $R_i - (1+\sqrt{\delta})p_i > R_{i'} - p_{i'}$. We claim that in this case a_i is preferred over $a_{i'}$ under contract p'. Namely,

$$\begin{aligned} (1-\sqrt{\delta})p_i + \sqrt{\delta}R_i - c_i &= (1+\delta)p_i - c_i + \sqrt{\delta}(R_i - (1+\sqrt{\delta})p_i) \\ &\geq p_{i'} - c_{i'} + \sqrt{\delta}(R_i - (1+\sqrt{\delta})p_i) \\ &> p_{i'} - c_{i'} + \sqrt{\delta}(R_{i'} - p_{i'}) \\ &= (1-\sqrt{\delta})p_{i'} + \sqrt{\delta}R_{i'} - c_{i'}, \end{aligned}$$

where we used that action a_i is δ-incentivized under p for the first inequality, and the second inequality holds by assumption because we are in Case 1.

Case 2: Assume now that $R_i - (1+\sqrt{\delta})p_i \leq R_{i'} - p_{i'}$. In this case the expected payoff achieved by action $a_{i'}$ is high. Namely,

$$\begin{aligned} R_{i'} - (1-\sqrt{\delta})p_{i'} - \sqrt{\delta}R_{i'} &= (1-\sqrt{\delta})(R_{a'_i} - p_{a'_i}) \\ &\geq (1-\sqrt{\delta})(R_i - (1+\sqrt{\delta})p_i) \\ &= (1-\sqrt{\delta})(R_i - p_i) - (1-\sqrt{\delta})\sqrt{\delta}p_i, \end{aligned}$$

where the inequality holds by assumption because we are in Case 2. ∎

Proof of Proposition 2.5. Consider the following principal-agent setting parameterized by δ and $\epsilon > 0$. Let $\mathcal{M} = \epsilon/\delta$. There are $n = 2$ actions and $m = 2$ items. The probabilities of the items given the actions is described by the following matrix

$$\begin{pmatrix} \frac{1}{4} & \frac{2\epsilon}{3(\mathcal{M}+\epsilon)} \\ 0 & 1 \end{pmatrix},$$

where the first column corresponds to item 1 and the second column to item 2. Set the rewards to be $r_1 = \frac{4\epsilon}{3}$ for item 1 and $r_2 = \mathcal{M} + \epsilon$ for item 2 (notice $r_1 < r_2$), and the costs to be $c_1 = 0$ and $c_2 = \mathcal{M} - \frac{\mathcal{M}\epsilon}{2(\mathcal{M}+\epsilon)} > 0$. Observe that the expected rewards are $R_1 = \epsilon$ and $R_2 = \mathcal{M} + \epsilon$.

Claim C.1. *$OPT = \epsilon$.*

Proof of Claim C.1. The expected payoff from letting the agent chose the zero-cost action a_1 is $R_1 = \epsilon$. Can we get any better by incentivizing a_2? The optimal contract for incentivizing the costly action in a 2-action setting is well-understood (see e.g. [23]): The only positive payment should be for the single subset of items maximizing the likelihood that the agent has chosen action a_2; in our case this is the subset $\{2\}$ containing item 2 only. Observe that its probability given action 1 is $\frac{\epsilon}{2(\mathcal{M}+\epsilon)}$. The 2-action characterization also specifies the payment for this outcome, setting it at $p_{\{2\}} = c_2/\left(1 - \frac{\epsilon}{2(\mathcal{M}+\epsilon)}\right) = \mathcal{M}$. Subtracted from R_2 we get expected payoff of ϵ from optimally incentivizing a_2. ∎

Claim C.2. *Contract p that pays $\mathcal{M} - \frac{\epsilon}{3}$ for outcome $S = \{2\}$ and 0 otherwise δ-incentivizes action a_2 with expected payoff $R_2 - p_2 = \frac{4}{3}\epsilon$.*

Proof of Claim C.2. We show action a_2 is δ-IC: The agent's expected utility from a_1 is $\frac{\epsilon}{2(\mathcal{M}+\epsilon)}p_2 = \frac{\epsilon(3\mathcal{M}-\epsilon)}{6(\mathcal{M}+\epsilon)}$, and from a_2 given contract $(1+\delta)p$ it is $(1+\delta)p_2 - c_2 = (1 + \frac{\epsilon}{\mathcal{M}})(\mathcal{M} - \frac{\epsilon}{3}) - \mathcal{M} + \frac{\mathcal{M}\epsilon}{2(\mathcal{M}+\epsilon)} = \frac{\epsilon(2\mathcal{M}-\epsilon)}{3\mathcal{M}} + \frac{\mathcal{M}\epsilon}{2(\mathcal{M}+\epsilon)}$. It can be verified that the former is less than the latter for $\delta \leq \frac{1}{2}$. ∎

Putting these claims together completes the proof of Proposition 2.5. ∎

Proof of Lemma 2.6. Fix a principal-agent setting. Let a_i be the action that is δ-incentivized by contract p and assume a_i is not IR. Observe that the agent's expected utility from a_i is $\geq -\delta$ (otherwise a_i would not be δ-IC with respect to a_1, which has expected utility ≥ 0 for the agent). First, if $\Pi > \delta$, then let p' be identical to p except for an additional δ payment for every outcome. Contract p' still δ-incentivizes action a_i, but now the agent's expected utility from a_i is ≥ 0, as required. Otherwise if $\Pi \leq \delta$, let p' be the contract with all-zero payments. The expected payoff to the principal is zero, which is at most an additive δ loss compared to Π. ∎

Appendix D. Hardness with a constant number of actions. In this appendix we show NP-hardness of the two computational problems related to optimal contracts when the number of actions n is constant. Appendices D.1 and D.2 prove hardness of δ-OPT-CONTRACT (Proposition D.1), from which hardness of δ-MIN-PAYMENT follows by the reduction in Section 2 (Corollary D.2).

Proposition D.1. δ-OPT-CONTRACT *is NP-hard even for $n = 3$ actions.*

Corollary D.2. δ-MIN-PAYMENT *is NP-hard even for $n = 3$ actions.*

D.1. The computational problem MIN-MAX-PROB. It will be convenient to reduce to δ-OPT-CONTRACT from a computational problem we call MIN-MAX-PROB, which is a variant of MIN-MAX PRODUCT PARTITION [40] and thus NP-hard.

- Input: A product distribution q over m items such that for every item j, its probability q_j is equal to $\frac{1}{a_j+1}$ where a_j is an integer $\in [3, a_{\max}]$ ($\log a_{\max}$ is polynomial in m).
- Output: YES iff there exists a subset of items S^* such that $q_{S^*} = \ell A$, where $A = \sqrt{\prod_j a_j}$ and $\ell = \prod_j q_j$.

We now take a closer look at MIN-MAX-PROB. Denote $a_S = \prod_{j \in S} a_j$.

Observation D.3. *The probability of subset S is $q_S = \ell a_{\overline{S}}$.*

Proof. For every item j, the probability it is excluded is

$$1 - q_j = 1 - \frac{1}{a_j + 1} = \frac{a_j}{a_j + 1} = q_j a_j.$$

So the probability of the outcome being precisely S is

$$\begin{aligned} q_S &= \left(\prod_{j \in S} q_j\right)\left(\prod_{j \notin S}(1 - q_j)\right) \\ &= \left(\prod_{j \in S} q_j\right)\left(\prod_{j \notin S} q_j a_j\right) \\ &= \left(\prod_{j=1}^{m} q_j\right)\left(\prod_{j \notin S} a_j\right) = \ell a_{\overline{S}}, \end{aligned}$$

as claimed. ∎

Observation D.3 immediately implies:

Observation D.4. *For every subset S, $a_S + a_{\overline{S}} = a_S + \frac{A^2}{a_S} \geq 2A$, where equality holds iff $a_S = a_{\overline{S}} = A$. Equivalently, $q_S + q_{\overline{S}} \geq 2\ell A$, where equality holds iff $q_S = q_{\overline{S}} = \ell A$.*

Proof. The inequality in the observation holds by the inequality of arithmetic and geometric means (AM-GM inequality), which states that for any two non-negative numbers w, z, $(w + z)/2 \geq \sqrt{wz}$. Namely, for $z = a_S$, $w = A^2/a_S$, and $A = \sqrt{zw}$ the AM-GM inequality states that $a_S + A^2/a_S = z + w \geq 2\sqrt{wz} = 2\sqrt{a_S \cdot A^2/a_S} = 2A$ as claimed. ∎

Observation D.4 shows the connection between MIN-MAX-PROB and the NP-hard problem MIN-MAX PRODUCT PARTITION: q is a YES instance (there exists a subset of items S such that $q_S = \ell A$) iff $a_S = A$.

The following observation will be useful in the reduction to δ-OPT-CONTRACT.

Observation D.5. *Let $\Delta = 1 - \ell A 2^{m-1}$, then $0 < \Delta < 1$.*

Proof. By definition,

$$\ell A = \frac{\sqrt{\prod a_j}}{\prod (a_j + 1)} \leq \frac{\prod \sqrt{a_j + 1}}{\prod (a_j + 1)} = \frac{1}{\prod \sqrt{a_j + 1}} \leq \frac{1}{2^m} < \frac{1}{2^{m-1}},$$

where the second-to-last inequality follows since $a_j \geq 3$ and so $\sqrt{a_j+1} \geq 2$. We conclude that $\ell A 2^{m-1} < 1$, completing the proof. ∎

D.2. Proof of Proposition D.1. We now use hardness of MIN-MAX-PROB to establish hardness of δ-OPT-CONTRACT.

Proof of Proposition D.1. The proof is by reduction from MIN-MAX-PROB, as follows.

Reduction. Given an instance q of MIN-MAX-PROB, construct a principal-agent setting with $n = 3$ actions.

- For action a_1, set its product distribution q_1 to be q.
- For action a_2, set its product distribution q_2 to be $1 - q$ (i.e., $q_{1,j} + q_{2,j} = 1$ for every item j).
- For action a_3, set its product distribution q_3 to be such that $q_{3,1} = 1$ (i.e., this action's outcome always includes item 1), and $q_{3,j} = \frac{1}{2}$ for every other item $j > 1$.

Set costs c_1, c_2 to zero and set c_3 to be $c = (a_{\max} + 1)^{-1}$. The only nonzero reward is $r = r_1$ for item 1; set r to be any number strictly greater than Δ^{-1}.

Analysis. First notice that the reduction is polynomial in m; in particular, the number of bits of precision required to describe the probabilities, cost c and reward r is polynomial.

The analysis will show that the expected payoff the principal can extract by a δ-IC contract if q is a YES instance is strictly larger than if q is a NO instance. We introduce some notation: Let $\mathcal{S}^1 = \{S \subseteq [m] \mid 1 \in S\}$, i.e., $\mathcal{S}^1$ is the collection of all item subsets containing item 1. Given a contract p, let $P = \sum_{S \in \mathcal{S}^1} p_S$ (the total payment for subsets in $\mathcal{S}^1$). Observe that the expected payment to the agent if he chooses action a_3 is $\frac{P}{2^{m-1}}$.

Claim D.6. *Action a_3 can be weakly δ-incentivized with expected payment $\frac{c}{\Delta(1+\delta)}$ if and only if q is a YES instance of MIN-MAX-PROB.*

Proof of Claim D.6. Fix a δ-IC contract p that weakly δ-incentivizes action a_3. By Observation D.3, the agent's expected utility from action a_1 is $\ell \sum_S p_S a_{\overline{S}}$ and from action a_2 is $\ell \sum_S p_S a_S$. The agent's expected utility from action a_3 (after boosting by $(1+\delta)$) is $\frac{P(1+\delta)}{2^{m-1}} - c$.

Assume first that q is a NO instance. If p weakly incentivizes action a_3 then

$$\frac{P(1+\delta)}{2^{m-1}} - c \geq \ell \cdot \max\left\{\sum_S p_S a_S, \sum_S p_S a_{\overline{S}}\right\}$$
$$\geq \frac{\ell}{2}\left(\sum_S p_S a_S + \sum_S p_S a_{\overline{S}}\right)$$
$$= \frac{\ell}{2}\sum_S p_S (a_S + a_{\overline{S}}) > \ell A \sum_S p_S \geq \ell A P,$$

where the second-to-last inequality is by Observation D.4, and is strict by our assumption that q is a NO instance. Rearranging $\frac{P(1+\delta)}{2^{m-1}} - c > \ell A P$ we get

$$c < \frac{P(1+\delta)}{2^{m-1}} - \ell A P(1+\delta) = \frac{P(1+\delta)}{2^{m-1}}\left(1 - \ell A 2^{m-1}\right) = \frac{P\Delta(1+\delta)}{2^{m-1}}.$$

By Observation D.5 we can divide both sides by $\Delta(1+\delta) > 0$ to establish $\frac{P}{2^{m-1}} > \frac{c}{\Delta(1+\delta)}$, completing the proof of the first direction.

Assume now that q is a YES instance. Then there exists S^* such that $a_{S^*} = a_{\overline{S^*}} = A$, and without loss of generality $S^* \in \mathcal{S}^1$ (otherwise take its complement). Consider the following contract: Let $p_{S^*} = \frac{c2^{m-1}}{\Delta(1+\delta)}$ and set all other payments to 0. The expected payment to the agent for action a_3 is $\frac{p_{S^*}}{2^{m-1}} = \frac{c}{\Delta(1+\delta)}$ as required, and the agent's expected utility (after boosting by $(1+\delta)$) is $\frac{p_{S^*}(1+\delta)}{2^{m-1}} - c = \frac{c}{\Delta} - c = \frac{c(1-\Delta)}{\Delta}$. Plugging in $\Delta = 1 - \ell A 2^{m-1}$, we get that the expected utility from action a_3 is $\ell A \frac{c2^{m-1}}{\Delta} = \ell A p_{S^*}$. This is equal to the expected utility from action a_1, since $\ell \sum_S p_S a_{\overline{S}} = \ell p_{S^*} a_{\overline{S^*}} = \ell A p_{S^*}$ Similarly, the expected utility from action a_2 is also $\ell A p_{S^*}$. We conclude that p weakly δ-incentivizes a_3, completing the proof of Claim D.6. ∎

We now use Claim D.6 to complete the hardness proof by showing that the expected payoff the principal can extract if q is a YES instance is strictly larger than if q is a NO instance.

For a YES instance, by Claim D.6 action a_3 can be weakly δ-incentivized with expected payment $\frac{c}{\Delta(1+\delta)}$. We argue that the values chosen in the reduction for c and r guarantee that action a_3 has the (strictly) highest expected payoff for the principal, so the agent breaks ties in favor of a_3: Since the only positive reward is $r_1 = r$ and since $q_{3,1} = 1$, the expected payoff from a_3 is $q_{3,1} r_1 - \frac{c}{\Delta(1+\delta)} = r - \frac{c}{\Delta(1+\delta)}$. The expected reward (and thus also payoff) from a_1 is at most $q_{1,1} r_1 \leq \frac{r}{4}$ (using that $a_1 + 1 \geq 4$), and the expected reward from a_2 is at most $q_{2,1} r_1 \leq (1 - \frac{1}{a_{\max}+1}) r$. Since $\frac{r}{4} \leq (1 - \frac{1}{a_{\max}+1}) r$ (using that $a_{\max} \geq 3$), it suffices to show $r - \frac{c}{\Delta(1+\delta)} \geq r - \frac{c}{\Delta} > (1 - \frac{1}{a_{\max}+1}) r$, or simplifying, $r > \frac{c(a_{\max}+1)}{\Delta}$. Since the reduction sets $c = (a_{\max}+1)^{-1}$ and $r > \Delta^{-1}$, the argument is complete.

For a NO instance, by Claim D.6 the expected payoff from a_3 is strictly lower than $r - \frac{c}{\Delta(1+\delta)}$. By the analysis of the YES case we know that the expected rewards from a_1, a_2 are strictly lower than $r - \frac{c}{\Delta}$ (and by limited liability the principal's expected payoff is bounded by the expected reward). This completes the proof of Proposition D.1. ∎

Appendix E. An FPTAS for the separation oracle. In this appendix we establish the FPTAS for MIN-LR stated in Lemma 3.3. Recall from the discussion leading to Lemma 3.3 that the separation oracle problem reduces to MIN-LR.

Proof of Lemma 3.3. We adapt an FPTAS of Moran [41] (see also subsequent papers such as [43]). Let

$$\Delta = (1+\epsilon)^{1/2m}.$$

FPTAS algorithm. The algorithm proceeds in iterations from 0 to m. In iteration j, the partial solutions in that iteration are subsets of the first j items. For a partial solution $S \subseteq \{1, \ldots, j\}$, recall that $q_{\ell,S}$ is the marginal probability to draw S among the first k items if the sample is distributed according to q_ℓ.

The partial solutions in iteration j are partitioned into families $Y_{j,1}, \ldots, Y_{j,r_j}$. The partition is such that for every family $r \in [r_j]$ and partial solutions $S, S' \in Y_{j,r}$, for every distribution $\ell \in [k] \cup \{i\}$, the ratio between $q_{\ell,S}$ and $q_{\ell,S'}$ is at most Δ.

In the first iteration $j = 0$, the only solution is the empty set. The solutions in iteration $j+1$ are generated from the families in iteration j as follows: One arbitrary partial solution S is chosen from every family $Y_{j,r}$ to "represent" it, and for each such S two partial solutions $S \cup \{j+1\}$ and S are added to the solutions of iteration $j+1$ (i.e., with and without the $(j+1)$st item).

The algorithm outputs the minimum objective $\frac{1}{q_{i,S}}\sum_k \alpha_k q_{k,S}$ among the solutions S in iteration m.

Analysis. We first argue that $ALG \leq (1+\epsilon)OPT$. Let S^* be the optimal solution, and denote the subset of S^* containing only items among the first j by S^*_j. By induction, in iteration j there is a partial solution S'_j such that $\Delta^{-j}\cdot q_{\ell,S^*_j} \leq q_{\ell,S'_j} \leq \Delta^j \cdot q_{\ell,S^*_j}$ for every distribution $\ell \in [k]\cup\{i\}$. Denote $S' = S'_m$. Then $ALG \leq \frac{1}{q_{i,S'}}\sum_k \alpha_k q_{k,S'} \leq \Delta^{2m}\cdot \frac{1}{q_{i,S^*}}\sum_k \alpha_k q_{k,S^*} = (1+\epsilon)OPT$.

It remains to show that the FPTAS runs in polynomial time. The running time is $O(\sum_j r_j)$. In the input distributions $\{q_k\}, q_i$, denote the range of every *nonzero* probability by $[q_{\min}, 1]$ ($q_{\min}$ can be exponentially small). For every distribution $\ell \in [k]\cup\{i\}$, the probabilities that are not 0 are at least $q^m_{\min}$. So a partition "in jumps of Δ" requires $O(t)$ parts, where t is the smallest integer satisfying $q^m_{\min}\cdot\Delta^t \geq 1$. So

$$t = \left\lceil \frac{m\log(q^{-1}_{\min})}{\log\Delta}\right\rceil = \left\lceil \frac{2m^2\log(q^{-1}_{\min})}{\log(1+\epsilon)}\right\rceil \leq \left\lceil \frac{2m^2\log(q^{-1}_{\min})}{\epsilon}\right\rceil,$$

where the last inequality uses $\log(1+\epsilon)\geq\epsilon$ for $\epsilon\in(0,1]$. Since the partition to r_j families maintains "jumps of Δ" for n distributions, $r_k = O(t^n)$. We invoke the assumption that n is constant to complete the analysis and the proof of Lemma 3.3. ∎

Appendix F. Hardness of MIN-PAYMENT. In this appendix we show the following counterpart to Corollary 4.2.

Proposition F.1. *For any constant $c\in\mathbb{R}, c\geq 1$, it is NP-hard to approximate the minimum expected payment for implementing a given action to within a multiplicative factor c.*

Proof. The proof is by reduction from MAX-3SAT. Given an instance of MAX-3SAT, the goal is to determine whether the instance is satisfiable or whether at most $\frac{7}{8}+\epsilon$ of the clauses can be satisfied, where ϵ is an arbitrarily small constant.

Reduction. Given φ, we obtain the SAT principal-agent setting corresponding to φ (Proposition 4.12), but we set the reward for every item to be 1 rather than 0. We add an action a_{n+1} with cost $\mathcal{C}$ and product distribution q_{n+1} with probability $\frac{1}{2}$ for every item.

Analysis. As in the analysis in the proof of Proposition 4.15, if φ has a satisfying assignment then we can implement a_{n+1} at cost $\mathcal{C}$. Otherwise recall that by Definition 4.11, the average action over the first n actions leads to every item set S with probability at least $\frac{1-8\epsilon}{2^m}$. Consider a contract p and let $P=\sum_S p_S$. The expected utility of the agent for choosing a_{n+1} is $P/2^m - \mathcal{C}$. Consider again the average action over the first n actions. The expected payment to the agent for "choosing" this action (i.e., the expected payment over the average distribution) is at least $\frac{1-8\epsilon}{2^m}P = \frac{P}{2^m} - \frac{8\epsilon P}{2^m}$, and there is some action a_i (with cost 0) for which the expected payment is as high. To incentivize a_{n+1} over a_i it must hold that $\frac{P}{2^m} - \mathcal{C} \geq \frac{P}{2^m} - \frac{8\epsilon P}{2^m}$, i.e., $\frac{P}{2^m}\geq\frac{\mathcal{C}}{8\epsilon}$. We conclude that if there is no assignment satisfying more than $\frac{7}{8}+\epsilon$ of the clauses, the expected payment for implementing a_{n+1} is $\frac{\mathcal{C}}{8\epsilon}$ rather than $\mathcal{C}$. Approximating the expected payment within a multiplicative factor $\frac{1}{8\epsilon}$ would thus solve the MAX-3SAT instance we started with, and we can make ϵ as small a constant as we want. ∎

Appendix G. Proofs omitted from Section 4. In this appendix we provide proofs for Propositions 4.8 and 4.9. In particular, we establish the existence of gap settings for 2 actions (Proposition 4.8) and c actions (Proposition 4.9).

Proof of Proposition 4.8. For the gap setting constructed above with $c = 2$ actions and $\gamma = \epsilon$, consider a δ-IC contract. Since the expected reward of the first action a_1 is 1, and the maximum expected welfare is $2 - \gamma \geq 2 - \frac{4\epsilon}{1+2\epsilon}$, if a contract is to extract more than $\frac{1}{2-4\epsilon/(1+2\epsilon)} = \frac{1}{2} + \epsilon$ of the expected welfare then it must δ-incentivize the last action a_c (a limited liability contract cannot extract more than the expected reward from an agent choosing a_1, since a_1 is zero-cost). Let p be the payment for the item and let p_0 be the payment for the empty set. For any action a_{i^*} that the contract δ-incentivizes, the following inequality must hold for every $i \in [c]$:

$$(1+\delta)\left(\gamma^{c-i^*}p + (1-\gamma^{c-i^*})p_0\right) - \frac{1}{\gamma^{i^*-1}} + i^* - (i^*-1)\gamma \;\geq$$

$$\left(\gamma^{c-i}p + (1-\gamma^{c-i})p_0\right) - \frac{1}{\gamma^{i-1}} + i - (i-1)\gamma. \tag{G.1}$$

Observe that for the contract to δ-incentivize a_c at minimum expected payment, it must hold that $p_0 = 0$. We can now plug $p_0 = 0$ into inequality (G.1) and choose $i^* = c, i = i^* - 1$. We get a lower bound on the expected payment for δ-incentivizing a_c:

$$p \geq \frac{(1-\gamma)^2}{\gamma(1+\delta-\gamma)}.$$

The principal's expected payoff is thus $\leq \frac{1}{\gamma} - \frac{(1-\gamma)^2}{\gamma(1+\delta-\gamma)} \leq \frac{1}{1+\gamma^2-\gamma}$, where the last inequality uses $\delta \leq f(\epsilon) = \gamma^2$. We get an upper bound of $\frac{1}{1+\gamma^2-\gamma}$ on what the best δ-IC contract can extract out of $2-\gamma$ for the principal. The ratio is thus at most $\frac{1}{2} + \epsilon$ (using $\gamma \leq \frac{1}{4}$), and this completes the proof of Proposition 4.8. ∎

Proof of Proposition 4.9. For the gap setting constructed above with c actions and $\gamma = \epsilon$, consider a δ-IC contract. As in the proof of Proposition 4.8, this contract cannot extract more than $\frac{1}{c} + \epsilon$ of the expected welfare by δ-incentivizing action a_1. Assume from now on that the contract δ-incentivizes action a_{i^*} for $i^* \geq 2$ at minimum expected payment. As in the proof of Proposition 4.8, Inequality (G.1) must hold for i^* and every $i \in [c]$.

Assume first that the contract's payment p_0 for the empty set is zero. (This assumption is without loss of generality for the case of $c = 2$ actions, as well as for $c \geq 3$ and fully-IC optimal contracts by Proposition 6 in [23].) Plugging $p_0 = 0$ into Inequality (G.1) and choosing $i = i^* - 1$, we get a lower bound on the expected payment for δ-incentivizing a_{i^*} (in particular making it preferable to a_{i^*-1}):

$$\gamma^{c-i^*}p \geq \frac{(1-\gamma^{i^*-1})(1-\gamma)}{\gamma^{i^*-1}(1+\delta-\gamma)}. \tag{G.2}$$

The principal's expected payoff is thus $\leq \frac{1}{\gamma^{i^*-1}} - \frac{(1-\gamma^{i^*-1})(1-\gamma)}{\gamma^{i^*-1}(1+\delta-\gamma)} \leq \frac{\gamma^c+\gamma^{i^*-1}(1-\gamma)}{\gamma^{i^*-1}(1+\gamma^c-\gamma)} = \frac{\gamma^c}{\gamma^{i^*-1}(1+\gamma^c-\gamma)} + \frac{1-\gamma}{1+\gamma^c-\gamma}$, where the last inequality uses $\delta \leq f(\epsilon) = \gamma^c$. Maximizing this expression by plugging in $i^* = c$, we get an upper bound of $\frac{1}{1+\gamma^c-\gamma}$ on what the best δ-IC contract can extract out of $c-(c-1)\gamma$ for the principal. The ratio can thus be shown to be at most $\frac{1}{c} + \epsilon$, as required (using that $c \geq 3$ and $\gamma \leq \frac{1}{4}$; see Claim G.1).

Now consider the case that $p_0 > 0$. We argue that in this case, plugging $i = i^* - 1$ into Inequality (G.1) gives a lower-bound on $\gamma^{c-i^*}p$ that is only higher than that in Inequality (G.2). To see this, consider the contribution of $p_0 > 0$ to the left-hand side of Inequality (G.1), which is $(1+\delta)(1-\gamma^{c-i^*})p_0$. Compare this to its contribution to the right-hand side of Inequality (G.1), which is $(1-\gamma^{c-i})p_0$. For $\delta \le \gamma^c$, $\gamma \le \frac{1}{4}$ and $i = i^*-1$ it holds that $(1+\delta)(1-\gamma^{c-i^*}) \le 1-\gamma^{c-i}$. This completes the proof of Proposition 4.9 up to Claim G.1. ∎

Claim G.1. *For every $\gamma \in (0, \frac{1}{4}]$ and $c \in \mathbb{Z}, c \ge 3$,*

$$\frac{1}{1+\gamma^c-\gamma} \cdot \frac{1}{c-(c-1)\gamma} \le \frac{1}{c} + \gamma.$$

Proof. We first establish the claim for $c = 3$. We need to show $\frac{1}{1+\gamma^3-\gamma} \cdot \frac{1}{3-2\gamma} \le \frac{1}{3} + \gamma$. Simplifying, we need to show $13\gamma + 6\gamma^4 \le 4 + 9\gamma^2 + 7\gamma^3$, which holds for every $\gamma \le \frac{1}{4}$.

We now consider $c \ge 4$: It is sufficient to show $\frac{1}{1-\gamma} \cdot \frac{1}{c-c\gamma} \le \frac{1}{c} + \gamma$. Multiplying by c we get $\frac{1}{(1-\gamma)^2} \le 1 + c\gamma$. This holds if and only if $c \ge \frac{2-\gamma}{(1-\gamma)^2}$. The right-hand side is an increasing function in the range $0 < \gamma \le \frac{1}{4}$ and so we can plug in $\gamma = \frac{1}{4}$ and verify. Since $c \ge 4 \ge \frac{28}{9}$, the proof is complete. ∎

Appendix H. Approximation by separable contracts. In this appendix we examine the gap between separable and optimal contracts.

Recall that a contract p is *separable* if there are payments $p_1, ..., p_m$ such that $p(S) = \sum_{j \in S} p_j$ for every $S \subseteq M$. By linearity of expectation, the expected payment for action a_i given a separable contract p is $\sum_j q_{i,j} p_j$.

As we have shown in Proposition A.1 the optimal separable contract can be computed in polynomial time via linear programming. Thus we know that separable (and other simple computationally-tractable) contracts cannot achieve a constant approximation to OPT unless $P = NP$ (Corollary 4.2).

In fact, an even stronger lower bound holds—they cannot achieve an approximation better than n, unless we relax the IC requirement to δ-IC. We provide a proof of this general lower bound for the case of $n = 2$.

Proposition H.1. *For every $\epsilon > 0$ there is a principal-agent instance with $n = 2$ actions and $m = 2$ items, in which the best separable contract only provides a $2 - \epsilon$ approximation to OPT.*

Proof. For $\delta \in (0, 1)$ consider the following $n = 2$ actions and $m = 2$ items instance. The probabilities $q_{i,j}$ for the two actions $i \in \{1, 2\}$ and items $j \in \{1, 2\}$ are

$$q_{1,1} = \frac{\delta}{2}, \quad q_{1,2} = 1 - \frac{\delta}{2} \quad \text{and} \quad q_{2,1} = \frac{1}{2}, \quad q_{2,2} = \frac{1}{2}.$$

The rewards r_j for the two items $j \in \{1, 2\}$ are

$$r_1 = \frac{1-(1-\frac{\delta}{2})\delta}{\frac{\delta}{2}} \quad \text{and} \quad r_2 = \delta.$$

The resulting expected rewards R_i for the two actions $i \in \{1, 2\}$ are

$$R_1 = q_{1,1}r_1 + q_{1,2}r_2 = \frac{\delta}{2}\frac{1-(1-\frac{\delta}{2})\delta}{\frac{\delta}{2}} + (1-\frac{\delta}{2})\delta = 1, \quad \text{and}$$

$$R_2 = q_{2,1}r_1 + q_{2,2}r_2 = \frac{1}{2}\frac{1-(1-\frac{\delta}{2})\delta}{\frac{\delta}{2}} + \frac{1}{2}\delta = \frac{1}{\delta} - 1 + \delta,$$

so that $R_2 > 1$ for all $\delta \in (0,1)$ and $R_2 \to \infty$ as $\delta \to 0$. The costs c_i for the two actions $i \in \{1,2\}$ are

$$c_1 = 0 \quad \text{and} \quad c_2 = (1-\delta)(R_2 - R_1) = (1-\delta)(\frac{1}{\delta} - 2 + \delta).$$

Note that on this instance

$$R_1 - c_1 = 1 \quad \text{and} \quad R_2 - c_2 = 2 - 2\delta + \delta^2.$$

We claim that: (1) The optimal contract can incentivize action 2 with an expected payment of $c_2/(1-\delta^2)$, so that the expected payoff to the principal is $R_2 - c_2/(1-\delta^2) = (1/\delta - 1 + \delta) - (1/\delta - 2 + \delta)/(1+\delta)$. (2) The optimal separable contract can either incentivize action 1 by paying nothing or it can incentivize action 2 by setting $p_1 = 2c_2/(1-\delta)$ and $p_2 = 0$. Since

$$R_2 - q_{2,1}p_1 = (\frac{1}{\delta} - 1 + \delta) - \frac{1}{2}\frac{2c_2}{(1-\delta)} = 1$$

the expected payoff to the principal in both cases is 1.

Using (1) and (2) and setting $\delta = \frac{1}{2}(3 - \epsilon - \sqrt{\epsilon^2 - 10\epsilon + 9})$ we have

$$\frac{OPT}{ALG} = (\frac{1}{\delta} - 1 + \delta) - \frac{\frac{1}{\delta} - 2 + \delta}{1+\delta} = 2 - \epsilon.$$

It remains to show (1) and (2). For (1) denote the payments in the optimal contract for outcomes (1,0), (0,1), and (1,1) by $p_1, p_2, p_{1,2}$. The optimal contract can incentivize action 2 via $p_1 > 0$ and $p_2 = p_{1,2} = 0$ as long as

$$\begin{aligned} & q_{2,1}(1-q_{2,2})p_1 - c_2 \geq q_{1,1}(1-q_{1,2})p_1 \\ \Leftrightarrow \quad & p_1 \geq \frac{c_2}{q_{2,1}(1-q_{2,2}) - q_{1,1}(1-q_{1,2})} = \frac{4c_2}{1-\delta^2} \end{aligned}$$

Setting $p_1 = 4c_2/(1-\delta^2)$ leads to an expected payment of $q_{2,1}(1-q_{2,2})p_1 = c_2/(1-\delta^2)$.

For (2) denote the payments of the optimal separable contract by p_1 and p_2 and note that the optimal separable contract either has $p_1 > 0$ and $p_2 = 0$ or it has $p_1 = 0$ and $p_2 > 0$. In the former case the incentive constraint is

$$q_{2,1}p_1 - c_2 \geq q_{1,1}p_1$$

and in the latter it is

$$q_{2,2}p_2 - c_2 \geq q_{1,2}p_2.$$

Note that since $q_{1,2} = 1 - \delta/2 > 1/2 = q_{1,2}$ it is impossible to incentivize action 2 by having only $p_2 > 0$. In the other case, where only $p_1 > 0$, the smallest p_1 that satisfies the incentive constraint is $p_1 = c_2/(q_{2,1} - q_{1,1}) = 2c_2/(1-\delta)$. ∎

Appendix I. Proofs of technical lemmas in Section 6. In this appendix we provide proofs for Lemma 6.3, Lemma 6.4, Lemma 6.5, and Lemma 6.7.

Proof of Lemma 6.3. Note that with $s = (3\log(\frac{2n}{\eta\gamma}))/(\eta\epsilon^2)$ we have $\gamma = \frac{n}{\eta} \cdot 2\exp(-\eta s\epsilon^2/3)$. Further note that since $q_{i,S} \geq \eta$ for all $i \in [n]$ and $S \subseteq M$ each action can assign positive probability to at most $1/\eta$ sets S. Finally, for all $i \in [n], S \subseteq M$

such that $q_{i,S} = 0$ we have $\tilde{q}_{i,S} = 0$. So, by the union bound, it suffices to show that for each of the at most n/η pairs i, S with $q_{i,S} > 0$ the probability with which $\tilde{q}_{i,S}$ does not fall into $[(1-\epsilon)q_{i,S}, (1+\epsilon)q_{i,S}]$ is at most $2\exp(-\eta s \epsilon^2/3)$.

Consider any such pair i, S. Let $X_{i,S}$ denote the random variable that counts the number of times set S was returned in the s queries to action i. Then $\tilde{q}_{i,S} = X_{i,S}/s$ and $\mathbb{E}[X] = sq_{i,S}$. So, using Chernoff's bound,

$$\begin{aligned}\Pr[\tilde{q}_{i,S} \notin [(1-\epsilon)q_{i,S}, (1+\epsilon)q_{i,S}]] &= \Pr[|X_{i,S} - \mathbb{E}[X_{i,S}]| \geq \epsilon] \\ &\leq 2\exp(-\eta s \epsilon^2/3),\end{aligned}$$

as claimed. ∎

Proof of Lemma 6.4. Let a_i be the action that is incentivized by p under the actual probabilities Q, and let $a_{i'}$ be any other action. Then,

$$\begin{aligned}\sum_{S\subseteq M} \tilde{q}_{i,S} p_{i,S} - c_i + 2\epsilon &\geq (1-\epsilon) \sum_{S\subseteq M} q_{i,S} p_{i,S} - c_i + 2\epsilon \\ &\geq \sum_{S\subseteq M} q_{i,S} p_{i,S} - c_i + \epsilon \\ &\geq \sum_{S\subseteq M} q_{i',S} p_{i',S} - c_{i'} + \epsilon \\ &\geq (1+\epsilon) \sum_{S\subseteq M} q_{i',S} p_{i',S} - c_{i'} \\ &\geq \sum_{S\subseteq M} \tilde{q}_{i',S} p_{i',S} - c_{i'},\end{aligned}$$

where we used the bounds on the probabilities in the first and last step, that we are considering normalized settings in the second and fourth step, and the IC constraint in the third step. ∎

Proof of Lemma 6.5. Let a_i be the action that is incentivized by $\tilde{p}$ under the empirical probabilities $\tilde{Q}$, and let $a_{i'}$ be any other action. Then,

$$\begin{aligned}\sum_{S\subseteq M} q_{i,S} p_{i,S} - c_i + \delta + 2\epsilon &\geq (1+\epsilon) \sum_{S\subseteq M} q_{i,S} p_{i,S} - c_i + \delta + \epsilon \\ &\geq \sum_{S\subseteq M} \tilde{q}_{i,S} p_{i,S} - c_i + \delta + \epsilon \\ &\geq \sum_{S\subseteq M} \tilde{q}_{i',S} p_{i',S} - c_{i'} + \epsilon \\ &\geq (1-\epsilon) \sum_{S\subseteq M} q_{i',S} p_{i',S} - c_{i'} + \epsilon \\ &\geq \sum_{S\subseteq M} q_{i',S} p_{i',S} - c_{i'},\end{aligned}$$

where we used that we are considering normalized settings in the first and the last step, the bounds on the probabilities in the second and fourth step, and the δ-IC constraint in the third step. ∎

Proof of Lemma 6.6. We have,

$$\begin{aligned}\tilde{\Pi} &= \sum_{S\subseteq M} \tilde{q}_{i,S} r_S - \sum_{S\subseteq M} \tilde{q}_{i,S} p_{i,S} \\ &\leq (1+\epsilon) \sum_{S\subseteq M} q_{i,S} r_S - (1-\epsilon) \sum_{S\subseteq M} q_{i,S} p_{i,S} \\ &\leq \sum_{S\subseteq M} q_{i,S} r_S - \sum_{S\subseteq M} q_{i,S} p_{i,S} + 2\epsilon \\ &= \Pi + 2\epsilon,\end{aligned}$$

where we used the bounds on the payments in the first step and that we are considering normalized settings in the second. ∎

Proof of Lemma 6.7. We have,

$$\begin{aligned}P &= \sum_{S\subseteq M} q_{i,S} r_S - \sum_{S\subseteq M} q_{i,S} p_{i,S} \\ &\leq \frac{1}{1-\epsilon} \sum_{S\subseteq M} \tilde{q}_{i,S} r_S - \frac{1}{1+\epsilon} \sum_{S\subseteq M} \tilde{q}_{i,S} p_{i,S} \\ &\leq (1+2\epsilon) \sum_{S\subseteq M} \tilde{q}_{i,S} r_S - (1-\epsilon) \sum_{S\subseteq M} q_{i,S} p_{i,S} \\ &= \Pi + 3\epsilon,\end{aligned}$$

where we used the bounds on the probability in the first step, and that $1/(1-\epsilon) \leq 1+2\epsilon$ and $1/(1+\epsilon) \geq 1-\epsilon$ for all $\epsilon \leq 1/2$. ∎