

# Contracts with Private Cost per Unit-of-Effort

TAL ALON, Technion – Israel Institute of Technology

PAUL DÜTTING, Google Research

INBAL TALGAM-COHEN, Technion – Israel Institute of Technology

Economic theory distinguishes between principal-agent settings in which the agent has a private type and settings in which the agent takes a hidden action. Many practical problems, however, involve aspects of both. For example, brand X may seek to hire an influencer Y to create sponsored content to be posted on social media platform Z. This problem has a hidden action component (the brand may not be able or willing to observe the amount of effort exerted by the influencer), but also a private type component (influencers may have different costs per unit-of-effort).

This “effort” and “cost per unit-of-effort” perspective naturally leads to a principal-agent problem with hidden action and single-dimensional private type, which generalizes both the classic principal-agent hidden action model of contract theory à la Grossmann and Hart [1986] and the (procurement version) of single-dimensional mechanism design à la Myerson [1983]. A natural goal in this model is to design an incentive-compatible contract, which consist of an allocation rule that maps types to actions, and a payment rule that maps types to payments for the stochastic outcomes of the chosen action.

Our main contribution is an LP-duality based characterization of implementable allocation rules for this model, which applies to both discrete and continuous types. This characterization shares important features of Myerson’s celebrated characterization result, but also departs from it in significant ways. We present several applications, including a polynomial-time algorithm for finding the optimal contract with a constant number of actions. This in sharp contrast to recent work on hidden action problems with multi-dimensional private information, which has shown that the problem of computing an optimal contract for constant numbers of actions is APX-hard.

CCS Concepts: • **Theory of computation** → **Algorithmic game theory**.

Additional Key Words and Phrases: Myerson’s theory; moral hazard; adverse selection; single-parameter; duality characterization

## ACM Reference Format:

Tal Alon, Paul Dütting, and Inbal Talgam-Cohen. 2021. Contracts with Private Cost per Unit-of-Effort. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC '21), July 18–23, 2020, Budapest, Hungary*. ACM, New York, NY, USA, 18 pages. <https://doi.org/10.1145/3465456.3467651>

## 1 INTRODUCTION

Mechanism design (MD) [e.g. 21] is a corner stone of economic theory, with major successes in both theory and practice. MD studies mechanisms for resource allocation among agents with private information, known as *hidden types*, with the goal of maximizing social welfare or revenue. The past two decades have seen a surge of interest in the study of mechanism design through an algorithmic and computational lens. Major success stories range from computational advertising and spectrum auctions to applications in internet routing and loadbalancing. An equally important and central

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*EC '21, July 18–23, 2020, Budapest, Hungary.*

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8554-1/21/07...\$15.00.

<https://doi.org/10.1145/3465456.3467651>

branch in economics is contract theory (CT) [e.g. 14]. While the natural focus of mechanism design is on the allocation of goods, contract theory has a natural focus on the *allocation of effort*. Contracts are a main tool for effort allocation since they use payments (monetary or other) to determine which actions agents will take. While the computer science community has largely ignored contract theory, driven by increased practical demand, there is a growing momentum and a recent set of papers have started to explore applications of contract theory from a computational viewpoint [e.g., 1, 8, 9, 15].

The increased practical demand for a computational and algorithmic approach to contracts is caused by an accelerating movement of contract-based markets from the analog/pen-and-paper world to the digital/electronic world. This includes online markets for crowdsourcing, sponsored content creation, affiliate marketing, freelancing and more. The economic value of these markets is substantial.<sup>1</sup> In such applications, platforms serve as a bridge between the two sides of the market (e.g., advertising brands and content creators), and are thus well-situated to play the role of market makers, applying their position and data to design better contracts.

**The Need for CT  $\cap$  MD.** Our work is motivated by the fact that in many of the applications that motivate the surge of interest in contracts, we actually see features of both — contract theory and mechanism design; and while there is some work on this in economics (which we survey in Section 1.2) — with the exception of [15] — we are not aware of any concurrent or prior work on the combination of the two from the computer science perspective.

The starting point of our work is that many of the applications at the intersection of CT and MD naturally involve an agent who can exert different levels of effort (his actions), and as in classic contract theory this leads to a stochastic outcome/reward to the principal, but the cost per unit-of-effort may differ between agents and is naturally modelled as private information. For example, a brand (the principal) may approach an influencer on a social media platform (the agent) to create branded or sponsored content on their behalf; and the opportunity cost of different influencers for different amounts of effort and outcome levels may differ.

The point is that this naturally leads to a model that combines the classic-principal agent problem with hidden action with features of *single-dimensional* mechanism design that we propose and study in this paper.

**Single vs. Multi-Parameter.** To concretely discuss agent types and explain where our contribution diverges from [15], we briefly introduce the classic model for a principal-agent contractual relation [14]. A basic principal-agent setting is described by  $n$  (hidden) actions the agent can choose among,  $m$  (observable) possible outcomes the actions can lead to, and an  $n \times m$  matrix  $F$  whose  $i$ th row maps action  $i$  to a distribution over the outcomes. In addition, there is a cost vector with  $n$  costs, one per action, specifying how much costly effort the agent must invest to take that action. Finally there is a reward vector with  $m$  rewards, one per outcome, specifying how much the principal gains when the agent achieves that outcome. Given a contract (payment per outcome), the agent picks the utility-maximizing action (which maximizes her expected payment from the contract minus her cost of effort), and the principal gets as revenue the expected reward minus payment.

In our proposed model, there are  $n$  different effort levels  $i$  (actions). Action  $i$  costs  $\gamma_i$  units of effort. As before taking an action triggers a stochastic outcome according to some probability distribution  $F_i$  (the  $i$ -th row of  $F$ ). A new feature relative to [15] is that agent's have a *single-dimensional* private type  $c$  — their cost per unit-of-effort. So their cost for taking action  $i$  is  $c \cdot \gamma_i$ .

<sup>1</sup>For example, according to Statista, influencer marketing on Instagram was worth 5.67 billion U.S. dollars in 2018. See <https://www.statista.com/statistics/950920/global-instagram-influencer-marketing-spending/>.

	Known action	Hidden action
Known type	Trivial	Classic contract theory [14]
Hidden type (single-dim)	Myerson's theory [21]	Our model
Hidden type (multi-dim)	E.g., Cai et al. [5]	Guruganesh et al. [15]

Table 1. Our model's relations to other settings in the literature

The agent's hidden type in [15], in contrast, is her mapping from actions to outcomes, as modeled by the matrix  $F$  (and is therefore naturally *multi-dimensional*). The cost vector is assumed to be public knowledge. The (discrete) distribution of types is also publicly known. The authors of [15] establish in this model the hardness of computing an optimal truthful contract menu (where optimal refers to the principal's revenue). We study a complementary model, where  $F$  is known but costs are hidden. The fundamental difference from a computational viewpoint is that in the new model, the type of an agent can be represented by a *single parameter* – her cost per unit of effort. It is well-known from auction design that single-parameter types may allow positive results even when hardness results hold for multi-parameter types.

Table 1 positions our model with respect to the classic work in both mechanism design and contract theory, and on the intersection of the two.

## 1.1 Our Results

One of the most useful tools that Myerson's theory [21] provides for single-parameter mechanism design is a characterization of implementable allocation rules. Known as Myerson's lemma, this characterization is useful since it defines the design space – the designer need consider only monotone allocation rules,<sup>2</sup> and any such rule is guaranteed to have a corresponding payment rule such that the resulting mechanism is truthful.

Our main result is a “Myerson's lemma” for the more general model of contracts with private costs. Such a lemma should incorporate both the original characterization of Myerson, and the characterization of implementable actions from principal-agent theory. These two previously-known characterizations seem quite different: The first characterizes implementable allocation rules as monotone, which means (in the procurement variant) that agents (service providers) with high reported costs are not assigned by the mechanism to provide service.<sup>3</sup> The second characterizes implementable actions – those for which the principal has some contractual payment scheme over the outcomes that incentivizes the agent to choose this action. This characterization says that for an action  $i$  to be implementable, there can be no convex combination of the other actions that achieves the same distribution as  $i$  over outcomes, at a lower combined cost.

To contrast with our unified characterization, it's useful to restate monotonicity (as in Myerson's theory) as follows: Assume for simplicity a discrete type space, and consider all agent types and whether they're assigned by the allocation rule to provide service or not. Consider the aggregate service by the allocated agents and its overall cost. An allocation rule is monotonic (and thus

<sup>2</sup>An allocation rule maps an agent's (reported) type to his allocation under the mechanism; a payment rule maps his type to what he pays (or in procurement is paid).

<sup>3</sup>Or, if there are multiple levels of service, agents with high costs are assigned to provide lower, less costly such levels.

implementable) if and only if there is no combination of agent types that would together provide the same aggregate service at a lower combined cost.

Our unified characterization result is:

**THEOREM (INFORMAL CHARACTERIZATION – SEE THEOREM 3.4).** *Consider an allocation rule mapping agent types to assigned actions. The rule is implementable if and only if there is no weighted combination of agent types and actions that together achieve the same distributions over outcomes at a lower combined cost.*

Our characterization provides a computationally tractable way of checking (in time polynomial in the number of actions, outcomes and agent types) whether a given allocation rule is implementable (see Corollary 3.5). For a continuous type space, the characterization is similar in spirit but slightly more involved and appears in Theorem 3.8.

One implication of our characterization is that monotonicity of the allocation rule is no longer sufficient for implementability (as we demonstrate in Proposition 3.7). Intuitively this happens because the overall cost can now be improved not only by assigning the same actions to agents with lower costs, but also by taking an altogether different combination of actions.

We show two additional applications of our characterization results. The first stands in stark contrast to the APX-hardness of optimal contract design for multi-parameter types (which holds even with only constantly-many actions):

**THEOREM (TRACTABILITY FOR CONSTANTLY-MANY ACTIONS – SEE THEOREM 4.1).** *Finding the optimal contract for single-parameter types is solvable in polynomial time for a constant number of actions.*

Our second application (Theorem 4.5) uses the simple but powerful observation that if an allocation rule is implementable, the rest of Myerson’s theory holds for it despite the general setting of contracts with private types. In particular, Myerson’s payment identity holds and the expected revenue is equal to the expected virtual welfare. It is thus tempting to consider mechanisms that maximize virtual welfare – would they turn out to be implementable in our general setting, just like virtual welfare maximizers turn out to be monotone in Myerson’s original setting? We establish that this is indeed the case when the agent’s private cost is distributed uniformly:

**THEOREM (OPTIMAL CONTRACT FOR UNIFORM COSTS – SEE THEOREM 4.5).** *For uniform costs, the virtual welfare maximizing allocation rule is implementable.*

We leave as our main open question whether implementability holds for the candidate rule (the optimal monotone rule à la Myerson) beyond uniform distributions, and if not what should the mechanism be.

## 1.2 Related Work

Contract theory is an important and well-studied sub-field of microeconomics with many practical implications; for leading textbooks see [4, 20, 24]. Many works on contract design study it entirely separately from mechanism design (screening) – the former deals with hidden actions of the agent (moral hazard), and the latter with private types of the agent (adverse selection). An analysis of the basic single principal, single agent setting (with no types) is found in the seminal work of Grossman and Hart [14], and Holmström [17] summarizes some of the classic foundations (see also [23] for the scientific background on the Nobel prize shared by Hart and Holmström). One of the main take-aways is that the optimal contract can be found in this setting by solving (polynomially-many) linear programs.

A much smaller collection of works studies the combination of moral hazard and adverse selection. Closest to our work are the recent papers of Chade and Swinkels [7] and Guruganesh et al. [15]. Chade and Swinkels [7] study a similar setup in which there is a single-parameter type for the agent and the goal is to design the optimal (menu of) contracts. They make different assumptions than us, such as continuous action space or the simplifying MLRP assumption (a strengthening of first-order stochastic dominance among any pair of distributions associated with the actions). They obtain two different sufficient conditions under which they are able to optimally solve the design problem. Their solution involves minimizing the cost of implementing any given action at any given surplus for any given type in a pure moral hazard problem. Our focus is on a necessary and sufficient condition for implementability, and on computationally efficient solutions.

Guruganesh et al. [15] are the first to consider contract design with typed agents from a computational point of view. Importantly, their agent types are multi-dimensional, namely, an agent’s type determines the outcome distributions corresponding to every available effort level. They establish computational hardness of optimal contract design under moral hazard and adverse selection. Their hardness results motivate their exploration of the approximation power of simple classes of contracts (such as linear, fixed-provision contracts).

Other works on moral hazard and adverse selection also differ from ours in their focus on multi-dimensional types. One classic such work is by Myerson [22], who studies “generalized” principal-agent problems where agents have both private information and “private decision domains” (hidden action). The key insight is that the principal may, without loss of generality, restrict herself to incentive compatible direct mechanisms. This extends the revelation principle to situations where there are moral hazard factors. We apply this insight in our results. Gottlieb and Moreira [12] study a simple setting with an effort/no effort binary choice for the agent and two possible outcomes, so their types are two-dimensional vectors. Gottlieb and Moreira [13] study a more general setting where types and efforts are multi-dimensional (possibly infinite-dimensional). Their work identifies assumptions under which “optimal contracts are simple” in the sense that the optimal (menu of) contracts consists of a single contract for all types. Our work focuses on solving for optimal “menus”, i.e., contracts with separate payments per type.

Further afield, some works consider hidden types of principals rather than agents [e.g., 3].

Additional works on contract design through the computational lens (but without private types) include, e.g., [1, 9, 16, 19].

## 2 MODEL AND DEFINITIONS

In this section we introduce our model of contracts with hidden action and a single-parameter private type. For a comparison (and positioning) relative to other models of contracts and/or private types see Section 2.4.

*Notation.* Throughout,  $[n] = \{0, \dots, n\}$  (i.e., zero is included).

### 2.1 Model Instance

An instance of our model consists of two players, a *principal* and an *agent*. An action set  $[n]$  is available to the agent. An agent’s action leads to one of  $m$  outcomes with non-decreasing rewards  $r_1 \leq \dots \leq r_m$  where  $r_j \in \mathbb{R}_{\geq 0} \forall j$  (we do not distinguish between the outcomes and their rewards). Action  $i \in [n]$  requires  $\gamma_i \in \mathbb{R}_{\geq 0}$  units of effort from the agent, and induces a distribution  $F_i$  over the  $m$  rewards. We assume without loss of generality that actions are ordered by the amount of effort units they require, i.e.,  $\gamma_0 \leq \dots \leq \gamma_n$ , and that the first action requires no effort from the agent, i.e.,  $\gamma_0 = 0$ .

Let  $R_i = \mathbb{E}_{j \sim F_i}[r_j] = \sum_{j \in [m]} F_{i,j} \cdot r_j$  be the expected reward of action  $i \in [n]$ . As in [8] we assume there are no “dominated” actions: every two actions  $i < i'$  have distinct expected rewards  $R_i \neq R_{i'}$ , and the action that requires more units of effort has the higher expected reward, i.e.,  $R_i < R_{i'}$ . So actions are also ordered by expected reward  $R_0 \leq \dots \leq R_n$ . Without loss we normalize  $R_0 = 0$ .

The agent has a type  $c \in \mathbb{R}_{\geq 0}$  drawn from a distribution  $G$  with density  $g$ . The support of the distribution  $G$  is the set of all possible types  $C \subseteq \mathbb{R}_{\geq 0}$ . Type  $c$  captures the agent’s cost per unit-of-effort, i.e., his *marginal cost* for effort. When an agent with type  $c$  takes action  $i$ , then a reward  $r_j$  is drawn according to distribution  $F_i$ . The principal gets  $r_j$  and the agent loses  $\gamma_i \cdot c$ , which is the number of effort units action  $i$  requires multiplied by the agent’s cost per unit.

To summarize, an instance is described by: distributions  $F = (F_0, \dots, F_n)$  and required efforts  $\gamma = (\gamma_0, \dots, \gamma_n)$  for the actions, reward vector  $r = (r_1, \dots, r_m)$  for the outcomes, and a type distribution  $G$  over support  $C$  for the agent. We distinguish between instances (settings) with a *discrete* type space  $C$  and those with a *continuous* one. We denote the former by  $\text{Dis}(F, \gamma, r, G, C)$  and the latter by  $\text{Con}(F, \gamma, r, G, C)$ . We omit from the notation components of the setting that are clear from the context.

**Who knows what.** The instance itself is publicly known. The action  $i$  which the agent actually takes is *hidden* from the principal, who only observes its stochastic reward  $r_j$ . The agent’s realized type  $c$  is also *privately-known* only to the agent himself.

## 2.2 Contracts

Our notion of a contract is a *generalization* of the standard one [see, e.g., 8], since in our model we need to accommodate for agent *types*. What we refer to as a contract in this work is in fact the direct revelation version of the notion of “menu of contracts” as it appears in [15].

A *contract*  $(x, t)$  is composed of an *allocation rule*  $x$  and a *payment rule*  $t$ . It solicits a type report  $c' \in C$  from the agent (where  $c'$  need not necessarily be the true type  $c$ ). The allocation rule  $x : C \rightarrow [n]$  maps the agent’s reported type  $c'$  to an action  $x(c')$  recommended to the agent by the contract. We assume that  $x(\infty) = 0$ , i.e., when  $c = \infty$  the no-effort action is implemented and the payment is zero. The payment rule  $t : C \rightarrow \mathbb{R}_{\geq 0}^m$  maps the agent’s reported type  $c'$  to a vector of  $m$  non-negative payments or *transfers*  $(t_1^{c'}, \dots, t_m^{c'})$ , one for each outcome. Notice that by requiring that transfers are always non-negative, we are enforcing here the standard *limited liability* property of contract theory.<sup>4</sup>

*Remark.* It is possible for the allocation rule  $x$  to be randomized in the sense of mapping type  $c'$  to a distribution over actions and to expected payments. In the settings we consider, such randomized contracts have no extra power relative to deterministic ones. Thus we restrict attention to deterministic contracts unless stated otherwise. We discuss this and more general notions of randomization in the full version.

**The game.** A contract  $(x, t)$  induces the following 2-stage game:

Stage 1: The agent submits a type report  $c'$ , fixing the contractual payment vector  $t(c') = (t_1^{c'}, \dots, t_m^{c'})$ .

Stage 2: The agent chooses an action  $i$  (which is not necessarily the action  $x(c')$  prescribed to it by the contract), and incurs a cost of  $\gamma_i \cdot c$  (where  $c$  is his true type). An outcome  $j$  is realized according to distribution  $F_i$ . The principal is rewarded  $r_j$  and pays  $t_j^{c'}$  to the agent.

<sup>4</sup>Limited liability is assumed in many works on contract design, for example [6, 11, 18]. Without it we would need a risk-aversion assumption on the agent. Indeed, with neither of these assumptions, the principal-agent model loses its bite – there is no economic justification for having a contract to begin with.

*Remark.* Once the game reaches Stage 2, we are back in a standard principal-agent setting with no types, in which the agent faces a standard contract (simply a vector of  $m$  payments), and chooses accordingly a costly action that rewards the principal.

**Utilities.** Let  $c'$  be the reported type and  $i$  be the action chosen by the agent. The utilities of the players are as follows:  $t_j^{c'} - \gamma_i \cdot c$  for the agent, and  $r_j - t_j^{c'}$  for the principal, where outcome  $j$  is drawn from distribution  $F_i$  corresponding to action  $i$ . In expectation over the random outcome these are  $T_i^{c'} - \gamma_i \cdot c$  for the agent and  $R_i - T_i^{c'}$  for the principal, where  $T_i^{c'} = \mathbb{E}_{j \in F_i}[t_j^{c'}]$  is the expected payment from principal to agent for action  $i$ . Note that the sum of the players' expected utilities is the expected *welfare*  $R_i - \gamma_i \cdot c$ .

Since the players are rational, when facing payment vector  $t(c')$  in Stage 2, the agent whose true type is  $c$  will choose the action  $i^*(c', c)$  that maximizes his expected utility. I.e.,

$$i^*(c', c) = \arg \max_{i \in [m]} \{T_i^{c'} - \gamma_i \cdot c\}.$$

As is standard in contract design [see, e.g., 15], if there are several actions with the same maximum expected utility, we assume tie-breaking in favor of the principal.

When reporting his type in Stage 1, the agent will report  $c'$  such that his expected utility  $T_{i^*(c', c)}^{c'} - \gamma_{i^*(c', c)} \cdot c$  is maximized.

**Incentive compatibility.** We say a contract  $(x, t)$  is *incentive compatible (IC)* if for every type  $c \in C$ , it is in the best interest of an agent of type  $c$  to both truthfully report his type  $c$  and take the prescribed action  $x(c)$ . Formally (using the above notation):

DEFINITION 2.1. A contract  $(x, t)$  is IC if  $\forall c \in C$ ,

- (1)  $x(c) = i^*(c, c)$ ; and
- (2)  $c \in \arg \max_{c'} \{T_{i^*(c', c)}^{c'} - \gamma_{i^*(c', c)} \cdot c\}$ .

Condition (1) in the above definition ensures that the prescribed action for type  $c$  maximizes the agent's expected utility when he truthfully reports his type  $c$ .<sup>5</sup> Condition (2) ensures that reporting truthfully is a (weakly) dominant strategy.

Note that focusing on IC contracts is without loss of generality by the revelation principal [22]. Also, incentive compatibility usually goes hand in hand with individual rationality (IR), which in our context requires that the utility of an agent who reports truthfully is non-negative. Here the assumption that  $\gamma_0 = 0$  comes in handy, as it means that the agent can always choose an action that requires zero effort. Together with limited liability this ensures non-negative utility for a truthful agent.

**Objective.** The principal's goal is to design an IC contract  $(x, t)$  that maximizes her expected utility, where the expectation is over the agent's random type  $c$  drawn from  $G$ , as well as over the random outcome of the agent's prescribed action  $x(c)$ :

$$\mathbb{E}_{c \sim G} [R_{x(c)} - T_{x(c)}^c] = \mathbb{E}_{c \sim G} [\mathbb{E}_{j \sim F_{x(c)}} [r_j - t_j^c]].$$

### 2.3 An Example

The following principal-agent instance will serve as the running example of this paper.

EXAMPLE 2.2. There are two types, four actions with required effort  $\gamma_0 = 0, \gamma_1 = 1, \gamma_2 = 3, \gamma_3 = 10$ , and three outcomes with rewards  $r_1 = 0, r_2 = 10, r_3 = 30$ . The probabilities over outcomes are

<sup>5</sup>The contracts we consider are designed in favor of the principal and so the tie-breaking rule is compatible with the recommended action.

$F_0 = (1, 0, 0)$ ,  $F_1 = (0, 1, 0)$ ,  $F_2 = (0, 0.5, 0.5)$ , and  $F_3 = (0, 0, 1)$ . The two types are  $c = 1$  or  $c = 4$  and occur with equal probability

Consider Example 2.2. A possible contract (as depicted in Figure 1 below) is  $x(1) = 4$  with payments  $t(1) = (0, 0, 14)$  and  $x(4) = 1$  with payments  $t(4) = (0, 4, 0)$ .

$t_1(1) = 0$	$t_2(1) = 0$	$t_3(1) = 14$		$t_1(4) = 0$	$t_2(4) = 4$	$t_3(4) = 0$	
$r_1 = 0$	$r_2 = 10$	$r_3 = 30$		$r_1 = 0$	$r_2 = 10$	$r_3 = 30$	
1	0	0	$\gamma_0 = 0$	1	0	0	$\gamma_0 = 0$
0	1	0	$\gamma_1 = 1$	0	1	0	$\gamma_1 = 1$
0	0.5	0.5	$\gamma_2 = 3$	0	0.5	0.5	$\gamma_2 = 3$
0	0	1	$\gamma_3 = 10$	0	0	1	$\gamma_3 = 10$

Fig. 1. Contract example. The left tableau shows the payments for reported type  $c' = 1$ , and the right tableau those for reported type  $c' = 4$ .

Now for the contract to be incentive compatible we would like the agent to choose the tableau (= payments) that correspond to his type, and the recommended action for that type. For the agent with type  $c = 1$  this means choosing the left tableau, and action 4. This gives him a utility of  $1 \cdot 14 - 10 \cdot 1 = 4$ . The agent with type  $c = 4$  should choose the right tableau, and action 1. This choice yields a utility of  $1 \cdot 4 - 1 \cdot 4 = 0$ .

Moreover, for neither type the agent should want to pretend that he is of a different type and/or choose a different action. So, for example, the agent with type  $c = 1$  should not want to pretend to be of type  $c' = 4$  and take action 1. Indeed, in this case his utility for this would be  $1 \cdot 4 - 1 \cdot 1 = 3$ , which is smaller than the utility that he is currently getting.

This is in fact true for all types and possible deviations, and so this contract is incentive compatible. The principal's expected utility is  $0.5 \cdot (30 - 14) + 0.5 \cdot (10 - 4) = 11$ .

## 2.4 Relation to Classic Settings

Consider a simple instance of our model in which there is only a single type in  $C$ , so that the type of the agent is publicly known. For such instances our model reduces to the classic principal-agent model of Grossman and Hart [14]. The goal in that model is to design a *standard* contract, including a single payment vector and action recommendation, such that the agent takes the recommended action and the expected utility of the principal is maximized.

At the other extreme, consider an instance of our model in which every action  $i$  deterministically leads to a distinct reward  $r_i$  (i.e., the distributions  $F$  are point mass and when viewed as a matrix constitute the identity matrix), and so the agent's action is not hidden. For such instances our model reduces to a reverse (procurement) variant of the single-parameter mechanism design setting of Myerson [21]. In this variant, the agent acts as a seller of a service whose cost for providing the service is private. The service can be supplied at different (known) levels  $\gamma_0, \dots, \gamma_n$ , and the values  $\{r_i\}_i$  of the principal/buyer for these levels are also publicly known. The goal is to design a truthful procurement auction that maximizes the designer's expected revenue.

## 3 CHARACTERIZATION OF IMPLEMENTABLE CONTRACTS

A main driver of mechanism design with single-dimensional types has been Myerson's theory [21]. It characterizes the types of allocation rules that can be turned into incentive-compatible contracts as those that are monotone, and for each such allocation rule it provides an essentially unique

payment rule that turns this allocation rule into an incentive-compatible mechanism.<sup>6</sup> In this section we develop such a theory for our principal-agent model with hidden action and single-dimensional private cost. The key question is to find necessary and sufficient conditions for an allocation to be incentive compatible implementable.

**DEFINITION 3.1.** *An allocation rule  $x : C \rightarrow [n]$  is implementable if there exists a payment rule  $t : C \rightarrow \mathbb{R}_{\geq 0}^m$  such that contract  $(x, t)$  is IC.*

In Section 3.1 we define properties of allocation rules. Section 3.2 gives our characterization for discrete types, and Section 3.3 gives our characterization for continuous types.

### 3.1 Properties of Allocation Rules

Two key properties of allocation rules that will play a role in our characterization results are *monotonicity* and being *piecewise constant*.

**Monotone Allocation Rule.** A *monotone* allocation rule recommends actions that are (weakly) more costly in terms of effort and hence also more rewarding in expectation for the principal to agents whose cost per unit-of-effort is lower. Intuitively, such agents are better-suited to take on effort-intensive tasks.

**DEFINITION 3.2.** *An allocation rule  $x : C \rightarrow [n]$  is monotone if for every  $c, c' \in C$ ,*

$$c < c' \implies x(c) \geq x(c').$$

**Piecewise Constant Allocation Rule.** A special class of monotone allocation rules are *piecewise constant* allocation rules. Informally, these are allocation rules such that: (i) the allocation function  $x(\cdot)$  is locally constant in distinct intervals of  $[0, \infty)$ , and (ii) action  $n - i$  is allocated for types with costs in the  $i$ th interval (counting from 0). Formally,

**DEFINITION 3.3.** *An allocation rule  $x : C \rightarrow [n]$  is monotone piecewise constant if and only if there exist and  $0 = z_0 \leq \dots \leq z_n \leq z_{n+1} = \infty$ , such that for every  $c \in C$ , and  $i \in [n]$*

$$c \in [z_i, z_{i+1}) \implies x(c) = n - i.$$

$Z = \{z_0, \dots, z_n\}$  is referred to as the set of  $x$ 's breakpoints.<sup>7</sup>

### 3.2 Characterization for Discrete Types

We now present our characterization of implementable allocation rules with *discrete types*. The approach is reminiscent of the standard LP-based argument in contract theory for establishing whether or not a given action is implementable (see, e.g., Appendix A.2 of [8]), but it also generalizes Myerson's characterization of implementable allocation rules in single-parameter mechanism design problems (for discrete types). Our characterization will share features of Myerson's theory, but also depart from it in significant ways.

**THEOREM 3.4 (DISCRETE CHARACTERIZATION).** *An allocation rule  $x : C \rightarrow [n]$  is implementable if and only if there exist no weights  $\lambda_{(c, c', k)} \geq 0$  for all  $c, c' \in C$  and  $k \in [n]$  that satisfy both of the following:*

<sup>6</sup>Technically, Myerson [21] only considered the continuous types case. A similar characterization, however, applies also in the discrete types case [e.g. 2, 10]. Allocation rules still have to be monotone, and while payment identity no longer applies, it can be shown that payments have to be in a certain range, and hence there is still a revenue-optimal choice for any given allocation rule.

<sup>7</sup>Consistently including and excluding the left and right endpoint (respectively) of  $[z_i, z_{i+1})$  is for simplicity and without loss of generality.

(1) WEAKLY DOMINANT DISTRIBUTIONS:

$$\sum_{c' \in C} \sum_{k \in [n]} \lambda_{(c', c, k)} \cdot F_{k, j} \geq F_{x(c), j} \cdot (\sum_{c' \in C} \sum_{k \in [n]} \lambda_{(c, c', k)}) \quad \forall c \in C, j \in [m],$$

(2) STRICTLY LOWER JOINT COST:

$$\sum_{c \in C} \sum_{c' \in C} \sum_{k \in [n]} \lambda_{(c, c', k)} \cdot \gamma_k \cdot c < \sum_{c \in C} \sum_{c' \in C} \sum_{k \in [n]} \lambda_{(c, c', k)} \cdot \gamma_{x(c)}.$$

One can think of the weights  $\lambda_{(c, c', k)}$  in the previous theorem as a “joint deviation plan,” which says that whenever the true type of the agent is  $c$  with probability proportional to  $\lambda_{(c, c', k)}$  he could pretend to be of type  $c'$  and take action  $k$  instead. Then the first condition says that on aggregate this deviation plan shouldn't lead to weakly higher expected payoff to the principal, and the second condition states that it shouldn't have strictly lower cost.

PROOF OF THEOREM 3.4. We suggest the following LP-based approach to determine whether or not  $x$  is implementable. The LP for finding any IC menu of contracts that implements  $x$  has  $C \cdot m$  payment variables  $\{t_j^c\}$ , which must be nonnegative by limited liability, and  $C^2 n$  constraints ensuring that type  $c$ 's expected payoff from action  $x(c)$  and the contract  $t^c$  is at least as her expected payoff from any other action and any other contract. The LP is:

$$\begin{aligned} & \text{maximize} && 0 \\ & \text{subject to} && \sum_{j \in [m]} F_{x(c), j} \cdot t_j^c - \gamma_{x(c)} \cdot c \geq \\ & && \sum_{j \in [m]} F_{k, j} \cdot t_j^{c'} - \gamma_k \cdot c \quad \forall c, c' \in C, k \in [n], \\ & && t_j^c \geq 0 \quad \forall c \in C, j \in [m]. \end{aligned} \tag{LP1}$$

The dual of the above linear program has  $C^2 m$  nonnegative variables, indexed by  $\lambda_{(c, c', k)}$  for  $c, c' \in C$  and  $k \in [n]$ . The dual is:

$$\begin{aligned} & \text{minimize} && \sum_{c \in C} \sum_{c' \in C} \sum_{k \in [n]} \lambda_{(c, c', k)} \cdot (\gamma_k - \gamma_{x(c)}) \cdot c \\ & \text{subject to} && \sum_{c' \in C} \sum_{k \in [n]} F_{k, j} \cdot \lambda_{(c', c, k)} \geq \\ & && F_{x(c), j} \cdot (\sum_{c' \in C} \sum_{k \in [n]} \lambda_{(c, c', k)}) \quad \forall c \in C, j \in [m], \\ & && \lambda_{(c, c', k)} \geq 0 \quad \forall c, c' \in C, k \in [n]. \end{aligned}$$

The claim thus follows from strong duality.  $\square$

An immediate corollary of the LP-based proof is that we can efficiently (i) check whether a given allocation is implementable, and (ii) compute optimal payments for this allocation rule if it is. For (i) we can check whether LP1 has a feasible solution, for (ii) we can minimize  $\sum_j F_{x(c), j} t_j^c$  over the same feasibility region.

COROLLARY 3.5. *The problem of determining whether or not an allocation rule is implementable is solvable in time  $O(\text{poly}(n, m, |C|^2))$ . Moreover, if an allocation rule is implementable, then we can find optimal payments for this allocation rule in time  $O(\text{poly}(n, m, |C|^2))$ .*

We next observe that the conditions in Theorem 3.4 imply that any implementable allocation rule has to be monotone as defined in Definition 3.2.

PROPOSITION 3.6. *Every implementable allocation rule is monotone.*

The proof idea is simple. It goes by contradiction. The idea is to consider a “swap” between the two types  $\ell$  and  $h$  that certify a violation of monotonicity and use this as a “deviation plan” that violates the conditions of Theorem 3.4.

PROOF OF PROPOSITION 3.6. Suppose towards a contradiction that there exists an implementable allocation rule  $x : C \rightarrow [n]$  that is non-monotone. By definition, there are two types  $\ell < h$ , such that  $\gamma_{x(\ell)} < \gamma_{x(h)}$ . We specify weights that satisfy the conditions in Theorem 3.4, contradicting

our hypothesis that  $x$  is implementable. Take  $\lambda_{(\ell, h, x(h))}, \lambda_{(h, \ell, x(\ell))} = 1$  and  $\lambda_{(c, c', k)} = 0$  for all other entries. Simply put,  $\lambda$  “swaps” the allocations of  $\ell$  and  $h$ . We first show that  $\lambda$  gives dominant distributions:  $\sum_{c' \in C} \sum_{k \in [n]} \lambda_{(c', c, k)} \cdot F_{k, j} \geq F_{x(c), j} \cdot (\sum_{c' \in C} \sum_{k \in [n]} \lambda_{(c, c', k)}) \forall c \in C, j \in [m]$  (Theorem 3.4.1). For  $c \neq \ell, h$ , the inequality holds vacuously since  $\lambda_{(c, c', k)}, \lambda_{(c', c, k)} = 0 \forall k, c'$ . For  $c = \ell$ , since the only non-zero  $\lambda$ -entry in the left-hand side is  $\lambda_{(h, \ell, x(\ell))} = 1$ , the left-hand side is equal to  $F_{x(\ell), j}$ . Similarly, the only non-zero  $\lambda$ -entry in the right-hand side is  $\lambda_{(\ell, h, x(h))} = 1$ , the right-hand side is equal to  $F_{x(\ell), j}$ . Thus, the inequality holds tightly for  $c = \ell$ . The same arguments apply for  $c = h$ . Next, we show strictly lower joint cost  $\sum_{c \in C} \sum_{c' \in C} \sum_{k \in [n]} \lambda_{(c, c', k)} \cdot \gamma_k \cdot c < \sum_{c \in C} \sum_{c' \in C} \sum_{k \in [n]} \lambda_{(c, c', k)} \cdot \gamma_{x(c)}$  (Theorem 3.4.2). By the definition of  $\lambda$ , the last inequality simplifies to  $\gamma_{x(\ell)} \cdot h + \gamma_{x(h)} \cdot \ell < \gamma_{x(\ell)} \cdot \ell + \gamma_{x(h)} \cdot h$ . Equivalently, to  $\gamma_{x(\ell)} \cdot (h - \ell) < \gamma_{x(h)} \cdot (h - \ell)$ , which holds since  $0 < (h - \ell)$  and since  $\gamma_{x(\ell)} < \gamma_{x(h)}$ . This completes the proof.  $\square$

Interestingly, however, as we show in Proposition 3.7 below, monotonicity is *not* a sufficient condition in our model – a clear distinction from Myerson’s theory [21].

In fact, and perhaps surprisingly, this observation is rather persistent and not just caused by the fact that in contract theory it sometimes so happens that certain actions are not implementable at all even with known types.<sup>8</sup> In fact, the example that we use to establish Proposition 3.7 satisfies a condition presented in [8, Appendix A.2] for the implementability (in the contract sense) of any actions. That is, for all possible costs, and all actions, there is a contract that incentivizes that action. However, as we demonstrate, there is no payment rule that incentives truthful reporting of the hidden cost, while also ensuring that for each possible cost the agent prefers to take the prescribed action. We give here a proof sketch capturing the essence of the argument (for a formal proof see the full version).

**PROPOSITION 3.7.** *Consider Example 2.2. The monotone allocation rule  $x(1) = 2$ , and  $x(4) = 2$  is unimplementable.*

**PROOF SKETCH.** The proof relies on Theorem 3.4 with weights  $\lambda_{(1, c, 3)}, \lambda_{(4, c, 1)} = 0.5$  for  $c = 1$  and  $c = 4$ , and  $\lambda_{(c, c', k)} = 0$  for all other entries. One may interpret  $\lambda$  as the deviation where type 1 takes action 3, type 4 takes action 1, and both types report either  $c' = 1$  or  $c' = 4$  with equal probability. With this in mind, when the agents’ report is  $c' = 4$ , the generated distribution over outcomes is as follows:  $0.5 \cdot F_1 + 0.5 \cdot F_3$ , (from  $\lambda_{(4, 4, 1)}, \lambda_{(1, 4, 1)}$  respectively). According to the definition of  $F$ , this is exactly  $F_2 = F_{x(4)}$ . The same argument applies for  $F_{x(1)}$ . As for lower joint cost, the initial joint cost is  $\gamma_2 \cdot (1 + 4) = 15$ , while the joint cost after the deviation is  $\gamma_3 \cdot 1 + \gamma_1 \cdot 4 = 14 < 15$ .  $\square$

We note that while Proposition 3.7 seems to hint at a deeper incompatibility between monotonicity and implementability, it could still be possible that the *optimal* monotone allocation rule is always implementable. We will return to this question in Section 4.

### 3.3 Continuous Types

We next tackle the case of continuous types. We will show our characterization result (Theorem 3.8) by reduction to the discrete case. The key idea is that we can argue independently of the LP-approach that any implementable allocation rule has to be monotone. Then because the image of  $x$  is finite,  $x$  must be monotone piecewise constant. The characterization can then be phrased for the  $O(n)$  breakpoints  $z_i$  of  $x$ .

We state the theorem below. Intuitively, just like Theorem 3.4, it states that there should be no “deviation plan”, where this time deviations are from breakpoints  $z_i$  to other breakpoints  $z'_i$  and

<sup>8</sup>At least not in a way that ensures the principal non-negative utility, and the monotone contract just winds up choosing some of these actions.

actions  $k$ . For a technical reason, the actual deviation plan involves two deviations away from  $z_i$ , which we refer to as  $R$  and  $L$ .

**THEOREM 3.8 (CONTINUOUS CHARACTERIZATION).** *A allocation rule  $x : \mathbb{R} \rightarrow [n]$  is implementable if and only if it is monotone piecewise constant and there exist no weights  $\lambda_{(R,i,i',k)}, \lambda_{(L,i,i',k)} \geq 0$  for all  $i, i', k \in [n]$  that satisfy both of the following:*

(1) WEAKLY DOMINANT DISTRIBUTIONS.

$$\sum_{i'=0}^n \sum_{k \in [n]} \lambda_{(R,i',i,k)} \cdot F_{k,j} + \sum_{i'=1}^n \sum_{k \in [n]} \lambda_{(L,i',i,k)} \cdot F_{k,j} \geq F_{n-i,j} \cdot (\sum_{i'=0}^n \sum_{k \in [n]} \lambda_{(R,i,i',k)} + \lambda_{(L,i+1,i',k)}) \quad \forall j \in [m], i \in [n],$$

(2) STRICTLY LOWER JOINT COST.

$$\sum_{i=0}^n \sum_{i'=0}^n \sum_{k \in [n]} (\lambda_{(R,i,i',k)} + \lambda_{(L,i,i',k)}) \cdot \gamma_k \cdot z_i < \sum_{i=0}^n \sum_{i'=0}^n \sum_{k \in [n]} (\lambda_{(L,i,i',k)} \cdot \gamma_{n+1-i} + \lambda_{(R,i,i',k)} \cdot \gamma_{n-i}) \cdot z_i,$$

where  $\lambda_{(L,0,i',k)} = 0 \forall i', k \in [n]$ .

The reason why we need R/L is the continuous-to-discrete mapping. Arguably, the trivial thing to do is to map the set of breakpoint  $z_0, \dots, z_n$  to the set of types, ensuring that breakpoint  $z_i$  will not deviate from  $x(z_i)$ . However, this approach eliminates the IC-requirement for types in  $(z_i, z_{i+1})$ . A key observation is that ensuring IC for the types at the endpoints of  $[z_i, z_{i+1})$  suffices to guarantee IC for all types in this interval (since types there share the same allocation.) So we use the following types:  $(i+1, L)$  with cost  $c \downarrow z_{i+1}$  ( $L$  for tending to  $z_{i+1}$  from below/left) this type corresponds to the open endpoint of  $[z_i, z_{i+1})$ , and  $(i, R)$  with cost  $z_i$  ( $R$  for right) corresponding to the closed endpoint of this interval. Both  $(i+1, L)$  and  $(i, R)$  are allocated to action  $x(z_i)$ . For example, in Figure 2, types  $(2, L)$  and  $(1, R)$  both take action 2 (allocated to the interval  $[z_1, z_2)$ ).

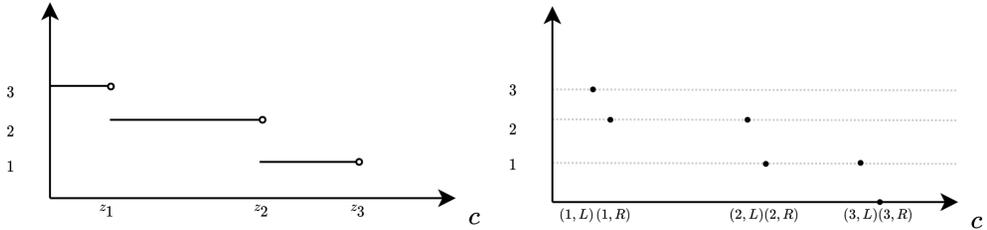


Fig. 2. Left: A monotone piecewise constant allocation rule  $x(\cdot)$ , for 4 actions (the  $x$ -axis is  $c$ , and the  $y$ -axis is  $x(c)$ ). Right: The discretization of  $x$  (note that points  $(i, L)$ ,  $(i, R)$  are plotted as if separated by a gap but actually  $(i, L) \rightarrow (i, R) \forall i \in [3]$ ).

**Proof of Theorem 3.8.** The proof of Theorem 3.8 relies on the following proposition (Proposition 3.9) and on Lemma 3.10.

**PROPOSITION 3.9.** *Every implementable allocation rule is monotone piecewise constant.*

**PROOF.** Using Myerson-like arguments, one can show that every implementable allocation rule is monotone (for completeness we show this in the full version). Thus, and since there is a finite number of actions, the allocation function  $x(\cdot)$  is piecewise constant.  $\square$

The following Lemma shows that the implementability problem of a monotone piecewise constant allocation rule reduces to the implementability problem of the same rule for a *discrete* case.

LEMMA 3.10. For a continuous instance  $\text{Con}(F, C, \gamma)$ , a monotone piecewise constant allocation rule  $x : \mathbb{R} \rightarrow [n]$  with breakpoints set  $Z$ , and a contract  $t : \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}^m$ . The following are equivalent:

- (1) There is a contract that implements  $x$  with respect to  $\text{Con}(F, C, \gamma)$ .
- (2) There is a contract  $t$  that implements  $x$  with respect to  $\text{Dis}(F, Z, \gamma)$ , where for every  $0 < i \in [n]$

$$T_{x(z_i)}^{z_i} - \gamma_{x(z_i)} \cdot z_i = T_{x(z_{i-1})}^{z_{i-1}} - \gamma_{x(z_{i-1})} \cdot z_i.$$

The equality implies that type  $z_i$  is equally incentivized to take action  $x(z_i)$ , as allocated to types in the interval on its right  $[z_i, z_{i+1})$ , and to take action  $x(z_{i-1})$ , as allocated to types in the interval on its left  $[z_{i-1}, z_i)$ . Note that this is equivalent to having that type  $(i, L)$  is incentivized to take action  $x(z_{i-1})$ , and that type  $(i, R)$  is incentivized to take action  $x(z_{i-1})$  (in the limit as their type  $c \rightarrow z_i$ .)

PROOF OF LEMMA 3.10. For the forward direction, suppose that there exists a contract  $t$  that implements  $x$  with respect to  $\text{Con}(F, C, \gamma)$ . We show that  $t$  implements  $x$  with respect to  $\text{Dis}(F, Z, \gamma)$ , and under the constraint in Lemma 3.10 (2). Consider  $c \in C'$ , note that type  $c$  cannot increase his payoff by reporting  $c' \neq c$ , or taking an action other than  $x(c)$ , since  $t$  implements  $x$  with respect to  $\text{Con}(F, C, \gamma)$ , and  $Z \subseteq C$ . Suppose towards a contradiction that for some type  $z_i$ , for which  $T_{x(z_i)}^{z_i} - \gamma_{x(z_i)} \cdot z_i \neq T_{x(z_{i-1})}^{z_{i-1}} - \gamma_{x(z_{i-1})} \cdot z_i$ . That is, without loss of generality,  $(\gamma_{x(z_{i-1})} - \gamma_{x(z_i)}) \cdot z_i < T_{x(z_{i-1})}^{z_{i-1}} - T_{x(z_i)}^{z_i}$ . It follows that for a small enough value of  $\epsilon$ ,  $(\gamma_{x(z_{i-1})} - \gamma_{x(z_i)}) \cdot (z_i + \epsilon) < T_{x(z_{i-1})}^{z_{i-1}} - T_{x(z_i)}^{z_i}$ . This contradicts to assumption that  $t$  implements  $x$ : since type  $z_i + \epsilon \in C$  could increase his payoff by reporting  $c' = z_{i-1}$  and by taking action  $x(z_{i-1})$ .

For the backwards direction, suppose that  $t$  implements  $x$  with respect to  $\text{Dis}(F, C, \gamma)$  under the constraint in 2. Define  $\hat{t} = \{t^{z_i} \mid c \in [z_i, z_{i+1})\}$ . Since  $t^{z_i}$  and action  $x(z_i)$  maximize type  $z_i$ 's payoff,

$$(\gamma_k - \gamma_{x(z_i)}) \cdot z_i \geq T_k^{z_{i'}} - T_{x(z_i)}^{z_i} \quad \forall i, k \in [n], z_{i'} \in C'. \quad (1)$$

Further, according to (2), and the fact that  $t^{z_{i+1}}$  and action  $x(z_{i+1})$  maximize type  $z_{i+1}$ 's payoff, we have that  $t^{z_i}$  and action  $x(z_i)$  maximize type  $z_{i+1}$ 's payoff. Thus,

$$T_{x(z_i)}^{z_i} - T_k^{z_{i'}} \geq (\gamma_{x(z_i)} - \gamma_k) \cdot z_{i+1} \quad \forall i, k \in [n], z_{i'} \in C'. \quad (2)$$

After having both inequalities established, we show that type  $c$ 's payoff is maximized by  $\hat{t}^c$  and  $x(c)$   $\forall c \in [z_i, z_{i+1})$ ,  $i \in [n]$ . If  $\gamma_{x(z_i)} > \gamma_k$ , using (2), the fact that  $\hat{t}^c = t^{z_i}$ ,  $x(c) = x(z_i)$ , and that  $z_{i+1} > c$  we have  $\hat{T}_{x(c)}^c - \hat{T}_k^c \geq (\gamma_{x(c)} - \gamma_k) \cdot c \forall k \in [n], c' \in C$ . Otherwise, if  $\gamma_i \leq \gamma_k$ , using (1), the fact that  $\hat{t}^c = t^{z_i}$ ,  $x(c) = x(z_i)$ , and that  $c \geq z_i$  we have that  $(\gamma_k - \gamma_{x(c)}) \cdot c \geq \hat{T}_k^c - \hat{T}_{x(c)}^c \forall k \in [n], c' \in C$ . This completes the proof.  $\square$

PROOF OF THEOREM 3.8. According to Proposition 3.9 and Lemma 3.10, in order to determine whether or not  $x$  is implementable, it suffices to determine whether or not there exists a contract  $t$  that implements  $x$  for  $\text{Dis}(F, Z, \gamma)$  under the constraint that  $T_{x(z_i)}^{z_i} - \gamma_{x(z_i)} \cdot z_i = T_{x(z_{i-1})}^{z_{i-1}} - \gamma_{x(z_{i-1})} \cdot z_i$  for every  $0 < i \in [n]$ . As in the proof of Theorem 3.4, we suggest the following LP-based approach.

The LP for finding any IC menu of contracts that implements  $x$  (under the added constraint) has  $Z \cdot m = O(n \cdot m)$  payment variables  $\{t_j^{z_i}\}$ , which must be nonnegative by limited liability, and two sets of  $O(Z^2 n) = O(n^3)$  constraints; referred to as the Right set (the first one), and the Left set. The Right set of constraints ensures that reporting  $c' = z_i$  and taking action  $x(z_i)$  maximizes the type  $z_i$ 's payoff among all types and actions. Note that this set of constraints is equivalent to the constraints in (LP1), ensuring that  $t$  implements  $x$ .<sup>9</sup> The Left set of constraints ensures that

<sup>9</sup>As previously explained, we call it the Right set, since it incentivizes type  $z_i$  to take action  $x(z_i)$ , as allocated by  $x$  to types in the interval on the right of breakpoint  $z_i$ , i.e.,  $[z_i, z_{i+1})$ .

reporting  $c' = z_{i-1}$  and taking action  $x(z_{i-1})$  also maximizes type  $z_i$ 's payoff among all types and actions. Thus, ensuring that the equality constraint in (2) holds. Similarly, we call it the Left set since it incentivizes type  $z_i$  to take action  $x(z_{i-1})$ , as allocated to types in the interval on the left of  $z_i$ , i.e.,  $[z_i, z_{i+1})$ . Conclusively, the LP is:

$$\begin{aligned}
 & \text{minimize} && 0 \\
 & \text{subject to} && \sum_{j \in [m]} F_{x(z_i), j} \cdot t_j^{z_i} - \gamma_{x(z_i)} \cdot z_i &>= && \\
 & && \sum_{j \in [m]} F_{k, j} \cdot t_j^{z_{i'}} - \gamma_k \cdot z_i &&& \forall i, i', k \in [n], \\
 & && \sum_{j \in [m]} F_{x(z_{i-1}), j} \cdot t_j^{z_{i-1}} - \gamma_{x(z_{i-1})} \cdot z_i &>= && \\
 & && \sum_{j \in [m]} F_{k, j} \cdot t_j^{z_{i'}} - \gamma_k \cdot z_i &&& \forall 0 < i, i', k \in [n], \\
 & && t_j^{z_i} \geq 0 &&& \forall i \in [n], j \in [m].
 \end{aligned} \tag{LP2}$$

The dual of (LP2) has two sets of  $O(n^2 \cdot m)$  nonnegative variables; Right variables: indexed by  $\lambda_{(R, i, i', k)}$  for  $i, i', k \in [n]$  and Left variables:  $\lambda_{(L, i, i', k)}$  for  $i \neq 0, i', k \in [n]$ . In our intuitive interpretation, the indexing  $(\{R, L\}, i, i', k)$  corresponds to side, type, report, and action (respectively). The dual is:

$$\begin{aligned}
 & \text{maximize} && \sum_{i=1}^n \sum_{i'=0}^n \sum_{k \in [n]} \lambda_{(L, i, i', k)} \cdot (\gamma_{x(z_{i-1})} \cdot z_i - \gamma_k \cdot z_i) && + \\
 & && \sum_{i=0}^n \sum_{i'=0}^n \sum_{k \in [n]} \lambda_{(R, i, i', k)} \cdot (\gamma_{x(z_i)} \cdot z_i - \gamma_k \cdot z_i) \\
 & \text{subject to} && \sum_{i'=0}^n \sum_{k \in [n]} \lambda_{(R, i', i, k)} \cdot F_{k, j} + \sum_{i'=1}^n \sum_{k \in [n]} \lambda_{(L, i', i, k)} \cdot F_{k, j} &>= && \\
 & && F_{x(z_i), j} \cdot \sum_{i'=0}^n \sum_{k \in [n]} (\lambda_{(R, i, i', k)} + \lambda_{(L, i+1, i', k)}) &&& \forall j \in [m], i \in [n-1], \\
 & && \sum_{i'=0}^n \sum_{k \in [n]} \lambda_{(R, i', n, k)} \cdot F_{k, j} + \sum_{i'=1}^n \sum_{k \in [n]} \lambda_{(L, i', n, k)} \cdot F_{k, j} &>= && \\
 & && F_{x(z_n), j} \cdot \sum_{i'=0}^n \sum_{k \in [n]} \lambda_{(n, R, i', k)} &&& \forall j \in [m]
 \end{aligned}$$

Rearranging the objective and the constraints, using the fact that  $\lambda_{(L, 0, i', k)} = 0 \forall i', k \in [n]$  (at the end of the theorem statement), and that  $x(z_i) = n - i$  for a monotone piecewise constant allocation rule with breakpoints  $z_1 \leq \dots \leq z_n$ . The objective is:

$$\sum_{i=0}^n \sum_{i'=0}^n \sum_{k \in [n]} (\lambda_{(L, i, i', k)} \cdot \gamma_{n+1-i} + \lambda_{(R, i, i', k)} \cdot \gamma_{n-i}) \cdot z_i - \tag{3}$$

$$\sum_{i=0}^n \sum_{i'=0}^n \sum_{k \in [n]} (\lambda_{(R, i, i', k)} + \lambda_{(L, i, i', k)}) \cdot \gamma_k \cdot z_i \tag{4}$$

The constraints are:

$$\begin{aligned}
 & \sum_{i'=0}^n \sum_{k \in [n]} \lambda_{(R, i', i, k)} \cdot F_{k, j} + \sum_{i'=1}^n \sum_{k \in [n]} \lambda_{(L, i', i, k)} \cdot F_{k, j} &>= && \\
 & F_{x(z_i), j} \cdot \sum_{i'=0}^n \sum_{k \in [n]} (\lambda_{(R, i, i', k)} + \lambda_{(L, i+1, i', k)}) &&& \forall j \in [m], i \in [n]
 \end{aligned}$$

By the strong duality theorem, (LP2) is feasible if and only if there are no variables such that the objective (3) is strictly positive and all the above constraints satisfy. This completes the proof.  $\square$

The following corollary is implied by the polynomial-time reduction specified in Lemma 3.10, the observation that the contract for the discrete instance (in Lemma 3.10) can be translated to a contract for the continuous instance, and the LP-based proof of Theorem 3.8.

**COROLLARY 3.11.** *The problem of determining whether or not a piecewise constant allocation rule is implementable is solvable in time  $O(\text{poly}(n, m, |C|^2))$ . Moreover, if an allocation rule is implementable, then we can find optimal payments for this allocation rule in time  $O(\text{poly}(n, m, |C|^2))$ .*

## 4 USING THE CHARACTERIZATION

We present three applications and/or extensions of our theory. First we use the observation that allocation rules have to be monotone to obtain a straightforward but still important corollary, namely that there is a poly-time algorithm for computing the optimal contract for discrete types and a constant number of actions. Then we give a non-trivial example of a “divide-and-conquer”

separation of the incentive constraints for private type and hidden action for the case of uniformly distributed costs. The idea is to ignore the IC constraints introduced by hidden action, this brings us into Myerson-land and we can just optimize over monotone allocation rules. In a second step we then verify that this optimal monotone rule happens to satisfy the incentive constraints due to hidden action. Finally, we discuss additional computational results and structural insights for other natural classes of distributions.

#### 4.1 A Poly-Time Algorithm for Fixed Number of Actions

We start by describing the poly-time algorithm for discrete types and constant number of actions. The idea is that in this case Proposition 3.7 allows us to search over all monotone rules, and we can find the optimal payments for a given rule via Corollary 3.5.

This positive result is in sharp contrast to the multi-dimensional model of Guruganesh et al. [15], where even computing the best single contract (rather than menu of contracts) with constant number of actions is APX-hard.

**THEOREM 4.1.** *The problem of finding the optimal contract for discrete types is solvable in polynomial time for a constant number of actions  $n$ .*

Given Proposition 3.7 and Corollary 3.5, Theorem 4.1 is straightforward to prove (the proof appears in the full version for completeness).

#### 4.2 Optimal Contract for Uniform Costs

Returning to the continuous-types setting, in this section we explore a Myersonian approach of optimizing over all monotone allocation rules, hoping that it turns out to be implementable. So in a sense, while not every monotone allocation rule is implementable as per our example, it could still be true that the optimal one is implementable. We show this to be the case assuming uniform distributions over cost (see Theorem 4.5). The proof turns out to be rather tricky, with a verification part that requires somewhat non-standard machinery and arguments.

We start by stating (for completeness) the definition of virtual cost, which is the “reverse” version of virtual value as introduced in [21]. Next, we verify that the classic result that expected revenue equals expected virtual welfare applies for any allocation rule in our model.

**DEFINITION 4.2.** *The virtual cost is given by*

$$\varphi(c) = c + \frac{G(c)}{g(c)} \quad \forall c \in C.$$

**PROPOSITION 4.3.** *For every allocation rule  $x : C \rightarrow \mathbb{R}^n$ , and payment rule  $t$  that implements  $x$ , the expected revenue is equal to the expected virtual welfare:*

$$\mathbb{E}_{c \sim G}[R_{x(c)}(c) - T_{x(c)}^c(c)] = \mathbb{E}_{c \sim G}[R_{x(c)}(c) - \varphi(c) \cdot Y_{x(c)}(c)].$$

The proof is deferred to the full version. It follows that the revenue maximizing allocation rule  $x^*$  is obtained by maximizing virtual welfare pointwise, as follows.

**COROLLARY 4.4.** *The revenue maximizing allocation rule  $x^*$  is given by*

$$x^*(c) = \arg \max_{i \in [n]} \{R_i - \varphi(c) \cdot \gamma_i\} \quad \forall c \in C.$$

The following theorem shows that when the distribution over types is uniform, the optimal monotone rule is implementable — a non-trivial result in light of Proposition 3.7, which showed that monotonicity alone is not sufficient.

**THEOREM 4.5.** *If the distribution over types  $G$  is the uniform distribution  $U[0, a]$  for  $a > 0$ , the virtual welfare maximizing allocation rule  $x^*$  is implementable.*

At a high level the proof proceeds as follows (the formal proof is given in the full version).

**PROOF OVERVIEW.** Suppose towards a contradiction that  $x^*$  is unimplementable. According to the characterization in Theorem 3.8, there exists a deviation schedule  $\lambda$  of  $z_0, \dots, z_n$  that gives weakly dominant distributions with a strictly lower joint cost. We modify the allocation  $x^*$  according to  $\lambda$  to attain weakly higher expected reward with strictly lower virtual cost, that is, strictly higher virtual welfare. This is a contradiction to the above (that  $x^*$  is a virtual welfare maximizer).

In more detail, the main parts of the proof are as follows: (i) Weakly dominant distributions result in a (weakly) higher expected reward. To see this, multiply each inequality in Theorem 3.8.2 by  $r_j$ , and sum the inequalities over all outcomes and types. (ii) For uniform distributions, strictly lower joint cost implies strictly lower joint *virtual* cost. This is not the case for other distributions, e.g., exponential.<sup>10</sup> (iii) A modification of the allocation rule  $x^*$ : We define  $n$  small intervals near each breakpoint  $z_i$ . Each interval is proportional to  $\sum_{i' \in [n]} \sum_{k \in [n]} \lambda_{(R, i, i', k)} + \lambda_{(L, i, i', k)}$ . (Changing only the breakpoints will not affect the expectation and we want to maintain the proportions of the agents' interplay.) Then, relying on the fact that our results hold for randomized allocation rules (see full version), for each breakpoint  $i$ , we allocate according to the weights in  $\lambda$ .  $\square$

### 4.3 Beyond Uniform Costs

We present additional observations about other natural classes of distributions, which lead to additional computational results and structural insights, and may prove useful in future work.

**Regular Distributions.** The first class of distributions is the “reverse” version of Myerson’s regular distributions definition.

**DEFINITION 4.6.** *A distribution  $G$  is regular if the corresponding virtual cost  $\varphi(\cdot)$  function is nondecreasing.*

As one would expect, regular distributions give a sufficient condition for the monotonicity of the optimal allocation rule. This implies that when the distribution is regular, the optimal allocation rule is monotone piecewise constant. Combining this with Corollary 3.11 yields the following.

**COROLLARY 4.7.** *Assuming regular distributions. The problem of determining whether or not the optimal allocation  $x^*$  is implementable is solvable in time  $O(\text{poly}(n, m, |C|^2))$ . Moreover, if  $x^*$  is implementable, then we can find optimal payments for this allocation rule in time  $O(\text{poly}(n, m, |C|^2))$ .*

**Diminishing Marginal Returns.** We give another regularity assumption—which we could not find in the contracts literature. Namely, “decreasing marginal returns”.

**DEFINITION 4.8.** *Distributions  $F$ , rewards  $r$  and required efforts  $\gamma$  satisfy diminishing marginal returns if  $\frac{R_i - R_{i-1}}{\gamma_i - \gamma_{i-1}}$  is (weakly) decreasing with  $i \in [n]$ .*

In words, the ratio between the gained revenue when shifting from action  $i - 1$  to action  $i$  and the added amount of effort is decreasing with  $i \in [n]$ . This natural assumption, along with regular distributions, suggests the following nice interpretation of the optimal allocation function.

<sup>10</sup>To see this, let  $\gamma_1 = 1, \gamma_2 = 2, \gamma_3 = 3, \gamma_4 = 7$  and types  $c_1 = 1, c_2 = 2$ . The joint cost of  $x(1) = 2, x(2) = 3$  is:  $\gamma_3 \cdot 2 + \gamma_2 \cdot 1 = 8$ . The joint cost of  $x'(1) = 4, x'(2) = 1$  is:  $\gamma_1 \cdot 2 + \gamma_4 \cdot 1 = 9$ . Thus the joint cost of  $x$  is lower than the joint cost of  $x'$ . The virtual cost function of exponential distribution with parameter 1 is  $\varphi(c) = c + e^c - 1$ . Thus, The joint virtual cost of  $x(1) = 2, x(2) = 3$  is:  $\gamma_3 \cdot (2 + e^2 - 1) + \gamma_2 \cdot (1 + e - 1) \approx 30.6$ . The joint cost of  $x'(1) = 4, x'(2) = 1$  is:  $\gamma_1 \cdot (2 + e^2 - 1) + \gamma_4 \cdot (1 + e - 1) \approx 27.41$ . Thus, the joint virtual cost of  $x$  is higher than the joint virtual cost of  $x'$ .

**THEOREM 4.9.** *Assuming regular distributions and diminishing marginal return, the revenue maximizing rule  $x^*$  is monotone piecewise constant, and for every breakpoint  $z_i$  for  $1 < i \in [n]$*

$$\varphi(z_i) = \frac{R_{n+1-i} - R_{n-i}}{\gamma_{n+1-i} - \gamma_{n-i}}.$$

**PROOF.** According to Corollary 4.4, it suffices to show that for every  $c \in [z_i, z_{i+1})$ , the virtual welfare of action  $n - i$ ,  $R_{n-i} - \varphi(c) \cdot \gamma_{n-i}$ , is (weakly) higher than that of any other action  $k \in [n]$ . There are two analogue cases depending on whether  $k < n - i$  or not. Starting with  $k < n - i$ , we prove that  $R_k - \varphi(c) \cdot \gamma_k \leq R_i - \varphi(c) \cdot \gamma_i$  by downward induction on  $k$ . Initially,  $k = n - i - 1$ . Since  $c \leq z_{i+1}$  and the virtual costs are non-decreasing;  $\varphi(c) \leq \varphi(z_{i+1})$ . By the choice of  $z_i$ , this implies that  $\varphi(c) \leq \frac{R_{n-i} - R_{n-i-1}}{\gamma_{n-i} - \gamma_{n-i-1}}$ . That is,  $R_{n-i-1} - \varphi(c) \cdot \gamma_{n-i-1} \leq R_{n-i} - \varphi(c) \cdot \gamma_{n-i}$ . This completes the base case. For the inductive step, suppose  $R_{k+1} - \varphi(c) \cdot \gamma_{k+1} \leq R_{n-i} - \varphi(c) \cdot \gamma_{n-i}$  for  $k + 1 < n - i$ . By diminishing marginal returns,  $\frac{R_{n-i} - R_{n-i-1}}{\gamma_{n-i} - \gamma_{n-i-1}} \leq \frac{R_{k+1} - R_k}{\gamma_{k+1} - \gamma_k}$ . As shown above,  $\varphi(c) \leq \frac{R_{n-i} - R_{n-i-1}}{\gamma_{n-i} - \gamma_{n-i-1}}$ . Combining the last two inequalities, we have  $R_k - \varphi(c) \cdot \gamma_k \leq R_{k+1} - \varphi(c) \cdot \gamma_{k+1}$ . Which, by the induction hypothesis, implies that  $R_k - \varphi(c) \cdot \gamma_k \leq R_{n-i} - \varphi(c) \cdot \gamma_{n-i}$ . The proof for  $k > n - i$  is analogue.  $\square$

*Remark.* We can show that decreasing marginal returns implies that every action is implementable in the untyped principal-agent problem. Notice that this implication goes away when (single-dimensional) types are added (see Proposition 3.7).

## 5 CONCLUSION

In this work we proposed a natural principal-agent problem with hidden action and single-dimensional private types, which seems to strike a balance between applicability and tractability. We provided analogs to Myerson’s theory — two characterization results, one for discrete and one for continuous types — on which the agenda of  $CT \cap MD$  can build.

We provided two “proofs of concept” which show that this direction is promising: (1) a much more positive computational result than the corresponding result in the multi-parameter model of Guruganesh et al. [15], and (2) a non-trivial “divide and conquer” result, which suggests that the optimal monotone rule could actually be implementable (more) generally, despite the fact that monotonicity is not a sufficient condition in our model.

An interesting question going forward is to verify or disprove that this approach works for all regular distribution functions, or all regular distributions under some “regularity” assumption on the contract setting. A natural candidate for the latter is to assume “diminishing marginal returns,” which makes sense economically but (as we showed) also results in a nice allocation rule (Theorem 4.9), and ensures that all actions can be incentivized absent hidden types.

## ACKNOWLEDGMENTS

This research was supported by the Israel Science Foundation (grant no. 336/18) and by the Taub Family Foundation.

## REFERENCES

- [1] Moshe Babaioff, Michal Feldman, and Noam Nisan. 2006. Combinatorial agency. In *Proceedings of the 7th ACM Conference on Electronic Commerce (EC)*. 18–28.
- [2] Dirk Bergemann and Martin Pesendorfer. 2007. Information structures in optimal auctions. *Journal of Economic Theory* 137, 1 (2007), 580–609.
- [3] B. Douglas Bernheim and Michael Whinston. 1986. Common agency. *Econometrica* 54, 4 (1986), 923–42.
- [4] Patrick Bolton and Mathias Dewatripont. 2005. *Contract Theory*. MIT press.
- [5] Yang Cai, Constantinos Daskalakis, and S. Matthew Weinberg. 2013. Understanding Incentives: Mechanism Design Becomes Algorithm Design. In *54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 618–627.

- [6] Gabriel Carroll. 2015. Robustness and linear contracts. *American Economic Review* 105, 2 (2015), 536–63.
- [7] Hector Chade and Jeroen Swinkels. 2019. Disentangling moral hazard and adverse selection. Working paper.
- [8] Paul Dütting, Tim Roughgarden, and Inbal Talgam-Cohen. 2019. Simple versus optimal contracts. In *Proceedings of the 20th ACM Conference on Economics and Computation (EC)*. 369–387.
- [9] Paul Dütting, Tim Roughgarden, and Inbal Talgam-Cohen. 2020. The complexity of contracts. In *Proceedings of the 31st ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2688–2707.
- [10] Edith Elkind. 2007. Designing and learning optimal finite support auctions. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 736–745.
- [11] Christian Gollier, Pierre-François Koehl, and Jean-Charles Rochet. 1997. Risk-taking behavior with limited liability and risk aversion. *Journal of Risk and Insurance* 64, 2 (1997), 347–370.
- [12] Daniel Gottlieb and Humberto Moreira. 2013. Simultaneous adverse selection and moral hazard. Working paper.
- [13] Daniel Gottlieb and Humberto Moreira. 2015. Simple contracts with adverse selection and moral hazard. Working paper.
- [14] Sanford J. Grossman and Oliver D. Hart. 1983. An analysis of the principal-agent problem. *Econometrica* 51, 1 (1983), 7–45.
- [15] Guru Guruganesh, Jon Schneider, and Joshua Wang. 2021. Contracts under moral hazard and adverse selection. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC)*. To appear.
- [16] Chien-Ju Ho, Aleksandrs Slivkins, and Jennifer Wortman Vaughan. 2016. Adaptive contract design for crowdsourcing markets: Bandit algorithms for repeated principal-agent problems. *Journal of Artificial Intelligence Research* 55 (2016), 317–359.
- [17] B. Holmström. 1984. On the theory of delegation. In *Bayesian Models in Economic Theory*, M. Boyer and R. Kihlstrom (Eds.). Elsevier.
- [18] Robert D Innes. 1990. Limited liability and incentive contracting with ex-ante action choices. *Journal of Economic Theory* 52, 1 (1990), 45–67.
- [19] Jon Kleinberg and Manish Raghavan. 2019. How do classifiers induce agents to invest effort strategically?. In *Proceedings of the 20th ACM Conference on Economics and Computation (EC)*. 825–844.
- [20] Jean-Jacques Laffont and David Martimort. 2009. *The Theory of Incentives: The Principal-Agent Model*. Princeton University Press.
- [21] Roger B. Myerson. 1981. Optimal auction design. *Mathematics of Operations Research* 6, 1 (1981), 58–73.
- [22] Roger B. Myerson. 1982. Optimal coordination mechanisms in generalized principal-agent problems. *Journal of Mathematical Economics* 10 (1982), 67–81.
- [23] Royal Swedish Academy of Sciences. 2016. Scientific Background on the 2016 Nobel Prize in Economic Sciences.
- [24] Bernard Salanie. 2017. *The Economics of Contracts: A Primer*. MIT press.