An Introduction to Contract Theory for Computer Scientists

PAUL DÜTTING
Google Research, Switzerland
and
MICHAL FELDMAN
Tel Aviv University, Israel
and
INBAL TALGAM-COHEN
Tel Aviv University, Israel

This survey introduces contract design to computer scientists: using outcome-based payments to incentivize effort. We discuss applications that motivate a computational approach to contract design, ranging from online labor markets, to incentive-aware algorithmic classification, to emerging ecosystems of AI agents. We present the classic principal-agent model and introduce basic concepts. We show how optimal contracts are computable via linear programming, and why they can be opaque and brittle. Motivated by this we turn to simpler contract formats, specifically linear contracts. We discuss approximation guarantees, and robust (max–min) optimality of such contracts. We then move beyond the single-agent setting to combinatorial contract settings with multiple agents, mapping the algorithmic landscape of tractability and approximation. Finally, we discuss partial information settings through the lens of learning algorithms, and outline open directions. The goal is a concise, accessible entry point that connects economic foundations with algorithmic tools and highlights opportunities for new theory and applications.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms: Algorithms, Design, Economics, Theory

Additional Key Words and Phrases: Algorithmic Game Theory, Contract Design, Principal-Agent Model, Hidden Action

1. INTRODUCTION

The field of Algorithmic Game Theory (AGT) addresses the challenges and opportunities that arise when algorithms interact with real-life, strategic players. One of the founding papers of the field, by Nisan and Ronen (2001), was motivated by the rising prominence of the internet circa 2000 —"a distributed setting where the participants cannot be assumed to follow the algorithm but rather their own self-interest". Nisan and Ronen suggested a framework for studying algorithms in such settings, based on *mechanism design* from microeconomics. This led to a vast body of research on algorithmic mechanism design, with a focus on payment schemes that incentivize strategic players to *report truthfully* to algorithms.

However, in many instances, strategic reaction to algorithms is unrelated to truth-

Authors' addresses: duetting@google.com, mfeldman@tau.ac.il, inbaltalgam@gmail.com,

fulness and mechanism design. In this survey we introduce *algorithmic contract design*—a new frontier for AGT. Contract design is a renowned subfield of microeconomics. It is similar to mechanism design in its theme of incentivizing strategic agents, but the incentives are not to report truthfully, but rather to exert effort earnestly. A contract is essentially a performance-based payment scheme, which ties an agent's compensation to that agent's performance and created value.

A premier application of contracts is in online markets of services, which, similar to markets of goods in the first two decades of the internet, have recently seen a massive growth both in importance and scale—creating a unique opportunity for computational approaches.

Example 1 (Influencer marketing). *Consider a brand that wants to pay a social media influencer (say a famous TikToker) to post sponsored/branded content. How should the contract be structured? Should it be a flat payment? Or should the payment be tied to how successful the post is? Should it be tied to the number of views, or the number of conversions?*

A second, perhaps less obvious applications of contract theory, is as a framework for understanding how humans react to algorithms such as a classification algorithm (even absent payments). This connection plays a role, for example, in the design of massive open online courses (MOOCs).

Example 2 (Strategic reaction to a Classifier). *Consider a classification algorithm, classifying human subjects according to some features they exhibit. This could be a prediction algorithm that assigns credit scores, or an exam at a university course or MOOC. The subjects of classification may invest strategic effort into developing more favorable features. In other words, an algorithm cannot measure without changing what it measures. Contract design helps design classifiers that anticipate this strategic effect, and can help route the efforts to modify features into productive avenues.*

We are now on the cusp of a new revolution, where AI agents are becoming increasingly capable of autonomously carrying out more and more tasks. The classic, pre-set algorithms of Nisan and Ronen are being replaced with opaque GenAI models, operating autonomously on behalf of their users and interacting with other AI agents. This growing ecosystem of AI agents and LLM providers depends crucially on effective delegation and cooperation, as enabled by contract design.

Example 3 (Paying an AI agent). *Consider a marketing GenAI agent, creating content for a campaign on behalf of a brand. What is the ideal payment scheme? Current LLM monetization by providers like OpenAI or Anthropic is usually per-token or subscription-based. In comparison, a performance-based contract would enable the utilization of key performance indicators like profit or audience engagement, thus leading to aligned incentives and elevated performance.*

The examples demonstrate some of the fundamental computational questions that arise when considering contracts: Perhaps the simplest possible performance-based payment scheme is a fixed fraction of the profit, known as a *linear* contract; how far from optimal is such a scheme, relative to a more complex one that relies on a combination of multiple indicators? What if instead of a single agent, there is a team of agents; how should they split the profit? What if certain aspects that are relevant to the design of contracts are not fully known, and have to be learned? These and other challenges require a combination of algorithmic and learning techniques with economic theory, and form the new research area of algorithmic contract design.

This survey aims to provide an accessible entry point to this new area for computer scientists. A more comprehensive survey by Dütting et al. (2024) covers additional topics within this new frontier and gives additional details on the topics surveyed here. A separate survey by Feldman (2025) covers in greater depth recent advances in multi-agent and multi-action combinatorial settings.

Organization. We introduce the canonical contract model in Section 2. In Section 3 we explore optimal contracts, and discuss simple, linear contracts in Section 4. Section 5 explores combinatorial, multi-agent contracts. In Section 6, we take a learning perspective. A discussion of what's ahead appears in Section 7.

2. MODEL

In this section, we introduce the hidden-action principal agent model (Holmström, 1979; Grossman and Hart, 1983). Our coverage follows (Dütting et al., 2019).

Setting. There is a principal (she) and an agent (he). The agent can take one of n actions. Each action $i \in \mathcal{A} = [n]$ has a cost $c_i \geq 0$. Actions are sorted by non-increasing cost. The cost of an action is borne by the agent. We model the fact that the agent can opt out of the contract by assuming that one of the actions has zero cost. Each action stochastically leads to one of m outcomes. We write q_{ij} for the probability of outcome j under action i, and use $\mathbf{q}_i$ to denote the probability distribution over outcomes induced by action i. The agent's choice of action is *hidden* from the principal, who can only observe the stochastic outcome of the action. This reflects that the principal is unable or unwilling to keep track of the agent's action. Each outcome j is associated with a reward $r_j \geq 0$, enjoyed by the principal. Rewards are sorted in non-increasing order. We use $R_i := \mathbb{E}_{j \sim \mathbf{q}_i}[r_j] = \sum_{j \in [m]} q_{ij} r_j$ to denote the expected reward of action i.

Example 4 (Running Example). *Consider a principal-agent setting with three actions $i \in [3]$ and three outcomes $j \in [3]$:*

reward:	$r_1 = 0$	$r_2 = 1$	$r_3 = 7$	*cost*
action 1*:*	$q_{11} = 1$	$q_{12} = 0$	$q_{13} = 0$	$c_1 = 0$
action 2*:*	$q_{21} = 0$	$q_{22} = 1/2$	$q_{23} = 1/2$	$c_2 = 1$
action 3*:*	$q_{31} = 0$	$q_{32} = 1/6$	$q_{33} = 5/6$	$c_3 = 2$

If the agent takes action 2*, then he bears a cost of* 1*. The outcome is $j = 2$ or $j = 3$ with equal probability. Note that if the principal observes outcome* 2*, she cannot be certain this is because the agent chose action* 2*, but action* 2 *is more likely than action* 3*.*

It is possible to impose additional structure on the distributions, like (first-order) stochastic dominance, which reflects that more rewarding outcomes become more

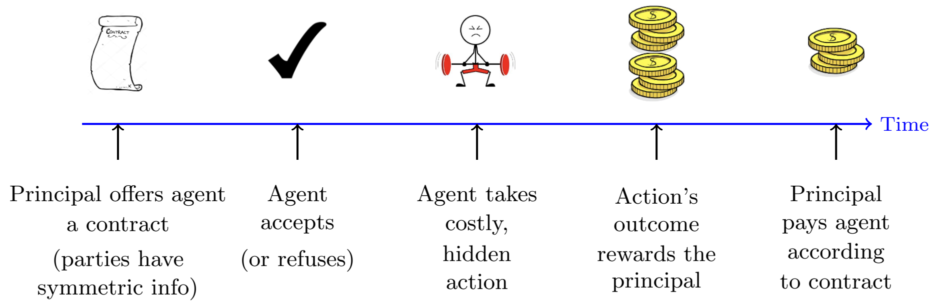


Fig. 1: Timeline.

likely as the agent exerts more costly effort. Other common assumptions include the stronger *Monotone Likelihood-Ratio Property* (MLRP), or the *Concavity of Distribution Function Property* (CDFP).

Contract. The principal enjoys the rewards while the agent bears the cost; these misaligned preferences create an incentive problem called *moral hazard*, which the principal seeks to address by defining a contract. A contract is a payment rule $\mathbf{t}$ that consists of m non-negative payments or *transfers* $(t_1, ..., t_m)$, one for each outcome.[1] The transfers are associated with outcomes rather than actions since the actions are hidden from the principal. For action $i \in [n]$ let

$$T_i := \mathbb{E}_{j \sim \mathbf{q}_i}[t_j] = \sum_{j \in [m]} q_{ij} t_j \tag{1}$$

denote the expected payment from principal to agent for taking action i.

Both the principal and the agent are assumed to be *risk neutral*. For a fixed contract $\mathbf{t}$ and choice of action i, the agent's expected utility and principal's expected utility (a.k.a. *revenue*) are given by

$$U_A(i \mid \mathbf{t}) := T_i - c_i \quad \text{and} \quad U_P(i \mid \mathbf{t}) := R_i - T_i.$$

Notice that the sum of the players' expected utilities is always equal to the expected welfare $W_i := R_i - c_i$ of the action i chosen by the agent. The contract thus influences the agent's choice of the welfare "pie" (through his choice of action), in addition to determining how this pie is divided between the principal and the agent.

Example 4, Continued. *Consider the contract $\mathbf{t} = (0, 1, 3)$. The expected payment for action 1 under this contract is $T_1 = 0$, for action 2 it is $T_2 = 2$, and for action 3 it is $T_3 = 8/3$. The agent's expected utility is thus maximized by action 2, which yields an expected utility of $U_A(2 \mid \mathbf{t}) = 1$, compared to an expected utility of $U_A(1 \mid \mathbf{t}) = 0$ for action 1 and an expected utility of $U_A(3 \mid \mathbf{t}) = 2/3$ for action 3. The principal's utility is then $U_p(2 \mid \mathbf{t}) = 2$.*

Incentives. The contract design problem defines a game between the principal and the agent, in which the principal moves first and defines a contract $\mathbf{t}$ and the agent

[1] The requirement that payments should be non-negative is referred to as *limited liability*. It reflects the asymmetric roles of the principal and the agent in contractual relations, and also rules out trivial but unrealistic solutions.

reacts with a utility-maximizing action $i^\star(\mathbf{t})$ (see Figure 1). Formally, for a fixed contract $\mathbf{t}$ let $\mathcal{A}^\star(\mathbf{t}) := \arg\max_{i\in[n]} U_A(i \mid \mathbf{t}) \subseteq [n]$ denote the set of actions that maximize the agent's expected utility. Using this notation, the agent chooses an action

$$i^\star \in \mathcal{A}^\star(\mathbf{t}) = \arg\max_{i\in[n]}\ U_A(i \mid \mathbf{t}). \tag{2}$$

Any such choice $i^\star$ is *incentive compatible* (IC) for the agent, because it is preferred over any other action. Note that it is also *individually rational* (IR) for the agent, meaning that the agent is guaranteed non-negative utility. This is because payments are non-negative, and we assumed that the agent has a zero-cost action.

Importantly, the principal's utility may be different for different actions $i^\star \in \mathcal{A}^\star(\mathbf{t})$. It is thus important to specify how the agent breaks ties. By default, and as is standard in the contracts literature, we assume that the agent breaks ties in favor of the principal.[2] We refer to this tie-breaking rule as the *canonical* tie-breaking rule.

3. OPTIMAL CONTRACTS

The principal's design problem is to select a contract $\mathbf{t}$ that maximizes her expected utility (i.e., revenue from the contract), given that the agent chooses an action $i^\star \in \mathcal{A}^\star(\mathbf{t})$ that maximizes his own expected utility. This contract is referred to as *optimal*. The next result is an algorithmic reformulation of a characterization of optimal contracts by (Grossman and Hart, 1983), showing that the optimal contract can be computed (in polyomial time) by solving *n linear programs* (LPs), one per action.

Proposition 1 (Grossman and Hart (1983)). *An optimal contract can be found by solving n linear programs, one per action.*

To outline the LP approach, we call an action $i \in [n]$ *implementable* (up to tie-breaking) if there exists a contract $\mathbf{t}$ such that $U_A(i \mid \mathbf{t}) \geq U_A(i' \mid \mathbf{t})$ for every $i' \neq i$. The approach is to construct an LP for each action $i \in [n]$ that determines whether the action is implementable and, if so, to find the minimum expected payment required to implement it. This LP and its dual are given in Figure 2. An optimal contract can then be found by applying this procedure to all actions, and choosing the action that maximizes the principal's utility. The corresponding pair of action and contract can be shown to be compatible with the canonical tie breaking rule.

An important corollary of the LP formulation is a characterization of the actions that the principal can implement.

Proposition 2 (Hermalin and Katz (1991)). *Action i is implementable if and only if there is no convex combination $\{\gamma_{i'}\}_{i'\neq i}$ of the actions other than i that results in the same distribution over outcomes, i.e., $\sum_{i'\neq i} \gamma_{i'} q_{i'j} = q_{i,j}$ for all outcomes j, with lower weighted cost, i.e., $\sum_{i'\neq i} \gamma_{i'} c_{i'} < c_i$.*

The characterization follows from LP duality: MINPAY-LP(i) is infeasible (equiv., action i cannot be implemented) precisely when its dual (which is always feasible)

[2] This tie-breaking rule is justified by the fact that a small perturbation would make the agent strictly prefer that action (see, e.g., Carroll (2015); Dütting et al. (2019) for additional discussion).

$$\begin{aligned} \min \quad & \sum_j q_{ij} t_j \\ \text{s.t.} \quad & \sum_j q_{ij} t_j - c_i \geq \sum_j q_{i'j} t_j - c_{i'} && \forall i' \neq i \\ & t_j \geq 0 && \forall j \end{aligned}$$

(a) MINPAY-LP(i)

$$\begin{aligned} \max \quad & \sum_{i' \neq i} \lambda_{i'} (c_i - c_{i'}) \\ \text{s.t.} \quad & \sum_{i' \neq i} \lambda_{i'} (q_{ij} - q_{i'j}) \leq q_{ij} && \forall j \\ & \lambda_{i'} \geq 0 && \forall i' \neq i \end{aligned}$$

(b) DUAL-MINPAY-LP(i)

Fig. 2: The MINPAY-LP(i) for action i **(left)** and its dual **(right)**. The primal variables are t_j for $j \in [m]$, and the dual variables are $\lambda_{i'}$ for $i' \in [n] \setminus \{i\}$.

is unbounded. If action i's outcome distribution can be replicated by a lower-cost convex combination of other actions, then a scaled version of the coefficients of this combination both satisfies the dual constraints, and achieves arbitrarily-high dual objective. The opposite direction is also not hard to show, establishing the characterization.

Optimal Contracts in Special Cases. In the special cases of binary actions and binary outcomes, optimal contracts have a particularly simple structure. For binary outcomes, the optimal contract is linear (see Section 4). For binary actions, the optimal contract only pays for a single outcome, the one that maximizes the so-called likelihood ratio. This also holds more generally for instances that, beyond the two actions, have a zero-cost action that leads to a zero-reward outcome with certainty, as in our running example.

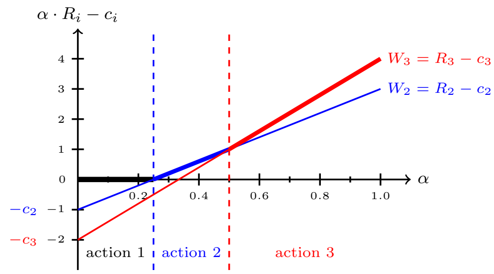
Example 4, Continued. *The optimal contract in our running example incentivizes the agent to take action* 3*, with contract* $t = (0, 0, 3)$ *which has a non-zero payment only for outcome* 3*. Note that outcome* 3 *is the outcome that maximizes the likelihood ratio* q_{3j}/q_{2j} *over all outcomes* j*.*

Shortcomings of Optimal Contracts. Beyond special cases, optimal contracts are often opaque and generally do not lend themselves to a clear, intuitive interpretation. Moreover, they are known to exhibit certain counterintuitive properties from an economic perspective. One particularly notable issue is that optimal contracts are not necessarily *monotone*, meaning that in the optimal contract $\mathbf{t}$, a higher principal reward r_j may entail a lower payment t_j. This is unnatural as the agent would be paid less, even though he achieved a better outcome.

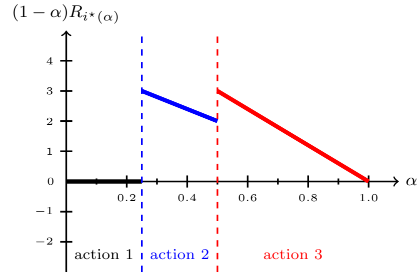
Another drawback of optimal contracts is their reliance on perfect knowledge of inputs, such as distributions $\mathbf{q}$ and costs $\mathbf{c}$. This issue is exacerbated by the fact that they are sensitive to small perturbations. These shortcomings motivate some of the work covered in the following sections.

4. LINEAR CONTRACTS

A contract is *linear* if it specifies a fixed fraction $\alpha \in [0, 1]$ of the rewards to be transferred to the agent by the principal. The parameter α determines all the payments: $t_j = \alpha r_j$ for every outcome $j \in [m]$. The agent's expected utility from the ith action is then $\alpha R_i - c_i$. So as α becomes closer to 1, the agent puts more weight on the expected reward R_i relative to the cost c_i when choosing an action. At the extreme $\alpha = 1$, the agent chooses the welfare-maximizing action. Linear contracts



(a) The agent's perspective



(b) The principal's perspective

Fig. 3: The agent's expected utility as a function of the linear contract's parameter α **(left)**, and the principal's expected utility as a function of α (**right**), for the principal-agent setting in Example 4.

have several nice properties like simplicity, interpretable payments (e.g., paying per conversion in Example 1), and monotonicity. Due to their widespread use, linear contracts were extensively studied in the economic literature. The computational perspective provides new insights, underscoring the potential of an algorithmic theory of contracts. We describe here two such insights: the approximation guarantees of linear contracts, and their robust optimality to uncertainty about distributions.

A Geometric Perspective. An important tool in the analysis of linear contracts, referred to as the geometric approach (Dütting et al., 2019), utilizes the fact that the agent's utility $\alpha R_i - c_i$ for each action i is a linear function in α. Specifically, the agent's best response to each contract α is given by the action i that results in the highest utility for the agent at that given α. In other words, by tracing the *upper envelope* of the set of functions $\{\alpha R_i - c_i\}_{i\in[n]}$ we can determine the agent's best response for each $\alpha \in [0, 1]$. Figure 3a illustrates this for Example 4. From this perspective, it becomes clear that the unit interval of possible linear contracts will be divided into contiguous intervals where the agent's choice of action is constant. Also, if we rename the actions as they appear on the upper envelope (from left to right), then actions are sorted by non-decreasing welfare, reward, and cost.

Switching the roles, when we consider the principal's utility as a function of the contract $\alpha \in [0, 1]$, we arrive at a graph like the one in Figure 3b (which is for the same example). We see that the principal's utility is a piece-wise linear function. Each segment corresponds to a different best-response of the agent, and within the segment corresponding to action i the principal's utility decreases at rate $-R_i$. Finally, under the standard tie-breaking rule, at the left-most point of a segment the agent will always choose the action with higher welfare (higher index, after re-indexing). Thus, the principal's utility is continuous when going to the right, while it might "jump" when going to the left. In particular, the best way to incentivize action i is by choosing the α_i that corresponds to the left-most endpoint of action i's interval. By checking the principal's utility at all *critical values* of α, i.e., all endpoints where the agent switches from one action to another, we can determine an optimal linear contract.

Example 4, Continued. *Example 4 demonstrates the suboptimality of linear* *contracts, which can obtain no more than* 3 *as expected utility for the principal (see Figure 3b), in comparison to* 3.5 *by the optimal nonlinear contract.*

Approximation Guarantees. A key insight of the algorithmic study of contracts is the following set of tight (multiplicative) approximation guarantees, showing that linear contracts provide a good approximation except in cases where there is simultaneously a large number of actions, a large spread in expected rewards, and a large spread in costs.

Theorem 1 (Dütting, Roughgarden, and Talgam-Cohen (2019)). *Let ρ denote the worst-case ratio between the principal's expected utility under the optimal contract, and the principal's expected utility under the optimal* linear *contract. Then among all principal-agent settings with n actions, $\rho = n$; among all settings with a ratio of H between the highest and lowest expected reward, $\rho = \Theta(\log H)$; and among all settings with a ratio of C between the highest and lowest cost, $\rho = \Theta(\log C)$.*

The upper bounds in Theorem 1 in fact apply against the stronger benchmark of optimal welfare, while the lower bounds apply even under the regularity assumption of MLRP.

The direction that $\rho \leq n$ follows from the characterization of optimal linear contracts above. The tightness direction ($\rho \geq n$) is established for a particular family of contract settings with n actions known as the *equal revenue* setting, parameterized by a small $\epsilon > 0$. This setting has exponentially increasing expected rewards among the actions: $R_i = 1/\epsilon^{i-1}$; as well as exponentially increasing costs: $c_i = R_i - i + \epsilon(i-1)$; but the welfare of the actions increases linearly: $W_i = i - \epsilon(i-1)$. The key observation is that any linear contract extracts utility of 1 for the principal, while the setting can be set up such that a general contract can extract $W_n \approx n$. The other bounds follow by similar constructions.

Robust Optimality. A different approach, spearheaded in the context of contract design by Carroll (2015), is the robust (or max-min) optimality approach. This approach assumes that certain aspects of the contracting problem are *known* to the principal, while other aspects are *unknown*. The principal then aims to design a contract $\mathbf{t}$ that achieves the best-possible worst-case utility over all instances $\mathcal{I}$ compatible with what is known. That is, $\mathbf{t}$ maximizes $U_P(\mathcal{I} \mid \mathbf{t}) = \inf_{I \in \mathcal{I}} U_P(I \mid \mathbf{t})$, where $I \in \mathcal{I}$ is the worst-case compatible instance. It turns out that under a broad range of assumptions, it is a *linear* contract that achieves optimal such worst-case guarantees.

In Carroll's work, the aspect known to the principal is a subset of actions A_0 available to the agent, while the actual set of actions A available to the agent can be any superset of A_0.

Theorem 2 (Carroll (2015)). *Linear contracts are max-min optimal when the principal knows a subset of actions A_0 available to the agent, and $\mathcal{I}$ includes instances in which the full action set is any superset A, i.e., $A_0 \subseteq A$.*

An alternative formulation of the robustness of linear contracts was given by Dütting et al. (2019), in a natural model of moment information that is easy to interpret.

Theorem 3 (Dütting, Roughgarden, and Talgam-Cohen (2019)). *Linear contracts are max-min optimal when the principal knows the costs $c_1, \ldots, c_n$, the rewards $r_1, \ldots, r_m$, and the expected rewards $R_1, \ldots, R_n$ available to the agent, and $\mathcal{I}$ includes all instances in which the distributions over rewards induced by the actions are compatible with the expected rewards.*

In both cases, the proof approach is based on showing that whatever non-linear contract $\mathbf{t}$ is considered as a candidate, an instance $I \in \mathcal{I}$ and a linear contract $\mathbf{t}'$ exist such that $U_P(I \mid \mathbf{t}) \leq U_P(\mathcal{I} \mid \mathbf{t}')$. Combined with $U_P(\mathcal{I} \mid \mathbf{t}) \leq U_P(I \mid \mathbf{t})$, we get that $U_P(\mathcal{I} \mid \mathbf{t}) \leq U_P(\mathcal{I} \mid \mathbf{t}')$, as desired.

5. COMBINATORIAL CONTRACTS

In the basic contracting model (Section 2), a single principal interacts with a single agent, who chooses one of n actions leading to one of m outcomes, and the revenue-optimal contract can be computed efficiently via LPs (see Section 3).

Real-world contract settings, however, are rarely this simple. A brand may engage multiple influencers (agents), each promoting multiple brands (principals) across various platforms (actions), evaluated by diverse metrics (outcomes). Such generalizations introduce new computational challenges, which the growing body of work on algorithmic contract theory aims to address. This body of work provides both efficient algorithms and impossibility results for settings with multiple agents (Babaioff et al., 2012; Dütting et al., 2026), combinatorial actions (Dütting et al., 2025a, 2024; Deo-Campo Vuong et al., 2024; Dütting et al., 2026), combinatorial outcomes (Dütting et al., 2021), multiple principals (Alon et al., 2024), and combinations thereof (Dütting et al., 2025b). As is typical at the interface of economics and computation, the computational perspective reveals key structural insights. Below we present a single paradigmatic example to illustrate the kinds of challenges that arise, and approaches to address them.

Paradigmatic Example: Multiple Agents. A natural extension of the contracting problem arises when a principal seeks to incentivize a team of agents. The classic reference here is Holmström (1982) on moral hazard in teams. The effectiveness of a team depends heavily on its composition, leading to the algorithmic challenge of designing (near-)optimal contracts for subsets of agents. This is already nontrivial in the basic setting where agents can either exert effort or not (the binary action setting), reducing to the question of which agents to contract with.

This line of work was initiated by Babaioff et al. (2012) under the name *combinatorial agency*. We present here the more general model of Dütting et al. (2026). There is a set N of n agents. Each agent $i \in N$ incurs a cost $c_i \in \mathbb{R}_{\geq 0}$ to exert effort, and the project either succeeds (reward 1) or fails (reward 0). A set function $f : 2^N \to [0, 1]$ specifies the success probability as a function of the subset of agents exerting effort. As in the single agent case, in this binary-outcome model one can focus on linear contracts: a vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$, where α_i is the share of the reward agent i receives if the project succeeds. Given a contract $\boldsymbol{\alpha}$ and a set $S \subseteq N$ of agents who exert effort, the principal's utility is $U_P(S \mid \boldsymbol{\alpha}) := (1 - \sum_{i \in N} \alpha_i) f(S)$, and agent i's utility is $U_i(S \mid \boldsymbol{\alpha}) := \alpha_i f(S) - \mathbb{1}[i \in S] \cdot c_i$, where $\mathbb{1}[i \in S] = 1$ if $i \in S$ and $\mathbb{1}[i \in S] = 0$ otherwise.

Each contract thus defines a game among the agents. We focus on pure Nash equilibria, namely action profiles where no agent has an incentive to deviate. A set S is an equilibrium under a contract $\boldsymbol{\alpha}$ if: (i) for every $i \in S$, $\alpha_i f(S) - c_i \geq \alpha_i f(S \setminus \{i\})$ (no incentive to deviate to shirking), and (ii) for every $i \notin S$, $\alpha_i f(S) \geq \alpha_i f(S \cup \{i\}) - c_i$ (no incentive to deviate to exerting effort). It is not too difficult to show that every contract $\boldsymbol{\alpha}$ admits an equilibrium (Deo-Campo Vuong et al., 2024; Dütting et al., 2025b). The problem is to find the contract $\boldsymbol{\alpha}$ that maximizes the principal's utility, in the best equilibrium of $\boldsymbol{\alpha}$.

The optimal choice of $\boldsymbol{\alpha}$ for a given set S is $\alpha_i = \frac{c_i}{f(i|S\setminus\{i\})}$ for $i \in S$, and $\alpha_i = 0$ for $i \notin S$. That is, the payment to an agent in S is proportional to his cost and inverse proportional to his marginal contribution to the success probability, reflecting the fact that an agent whose work matters less requires high compensation to avoid free riding on others' work. Thus, finding the optimal contract reduces to identifying the set $S^\star$ that maximizes the function

$$g : 2^N \to \mathbb{R} \cup \{-\infty\}, \qquad \text{where} \quad g(S) := \left(1 - \sum_{i \in S} \frac{c_i}{f(i \mid S \setminus \{i\})}\right) f(S).$$

Approximation Results. Dütting et al. (2026) study the problem of computing near-optimal contracts, assuming the success function f belongs to the hierarchy of complement-free set functions (e.g., additive, submodular, XOS, subadditive), capturing situations where an agent contribution diminishes as the team grows. They show that even for additive f, the problem is NP-hard, but admits an FPTAS. For more general set functions, they establish approximation results, given access to common primitives: A value query is given a set S and returns $f(S)$. A demand query gets a price vector $p_1, \ldots, p_m$ and returns a set maximizing $f(S) - \sum_{j \in S} p_j$.

Theorem 4 (Dütting, Ezra, Feldman, and Kesselheim (2026)). *For submodular (resp., XOS) success functions f, an $O(1)$-approximation to the optimal contract can be computed using poly-many value queries (resp., value and demand queries).*

The proof of Theorem 4 reveals a (perhaps surprising) connection to pricing. Namely, it can be shown that using a demand query with carefully chosen prices allows one to identify a "good" set of agents—one whose success value $f(S)$ is at least half that of the optimal set. However, to ensure a high $g(S)$ value, the payments must remain low; in particular, agents' marginal contributions to f must be sufficiently large. The crux is therefore a scaling lemma, which shows that it is always possible to remove agents from the set S, while also preserving a sufficiently large fraction of the original marginal contributions. For submodular functions, the latter part is straightforward, since the marginal contributions of the remaining agents can only go up. For XOS functions, however, the marginal contributions of the agents that remain may decrease, so a more careful argument is required to establish such a scaling property.

This proof pattern yields an efficient algorithm with value and demand queries. For submodular functions, the argument can be further refined, by replacing demand queries with an appropriately defined relaxation of demand queries that can be computed with poly-many value queries.

Impossibility Results. Follow-up work by Ezra et al. (2024) shows that for submodular functions, there exists a constant $c > 1$ such that no polynomial-time algorithm with value oracle access can approximate the optimal contract to within a factor better than c, assuming P≠NP. More recently, Dütting et al. (2025b) showed that even with both value and demand oracle access to the submodular function, there exists a constant $c > 1$, such that any algorithm that uses a sub-exponential number of queries returns a c-approximation with probability exponentially-small in n. In addition, for XOS functions, Ezra et al. (2024) show that no algorithm that makes poly-many value queries can approximate the optimal contract (with high probability) to within a factor $\Omega(n^{1/6})$.

6. LEARNING CONTRACTS

Another natural direction for an algorithmic study of contracts pioneered in Ho et al. (2016), is to study the contract design problem from a learning perspective. Here we focus on the online learning problem with bandit feedback studied in Ho et al. (2016), and discuss the state-of-the-art results of Zhu et al. (2023). Other variants of the problem have been studied, including under additional restrictions on the model (small action spaces, distributions satisfying additional conditions, etc.) (Bacchiocchi et al., 2024; Chen et al., 2024), and under different feedback models (Dütting et al., 2023; Chen et al., 2024; Dütting et al., 2025). Finally, Wang et al. (2023) propose a deep neural network architecture for designing near-optimal contracts from samples.

Online Learning with Bandit Feedback. There is a single principal interacting repeatedly with a single agent, whose type is drawn from an unknown type distribution $\mathcal{D}$. The interaction takes place over S rounds. In each round s, the agent's type θ^s is drawn from the underlying type distribution $\mathcal{D}$, independently. Over all rounds, the principal has fixed rewards $\{r_j\}_{j \in [m]}$. The agent's type θ^s determines the cost $c_i^\theta \geq 0$ of each action $i \in [n]$, as well as the probability distribution $\mathbf{q}_i^\theta$ over outcomes $j \in [m]$. It is assumed that both rewards and costs are bounded in $[0, 1]$.[3]

In each round s, the principal posts a contract $\mathbf{t}^s = (t_1^s, \ldots, t_m^s)$ (a non-negative payment for each outcome). The choice of contract may depend on what the principal has observed so far, and may be randomized. We consider two classes of contracts. In a (general) *bounded* contract we have $\mathbf{t}^s \in [0, 1]^m$, while a *linear* contract is defined as before. After the principal has posted contract $\mathbf{t}^s$, a type θ^s is drawn from $\mathcal{D}$, the agent takes a best response action $i^\star(\theta^s, \mathbf{t}^s)$, and an outcome j^s is sampled from $\mathbf{q}_{i^\star(\theta^s, \mathbf{t}^s)}$. The principal learns about the outcome j^s, receives the corresponding reward r_{j^s}, and pays the agent the amount specified by contract $\mathbf{t}^s$ for outcome j^s.

The principal's goal is to minimize *regret* with respect to the best single contract in hindsight. To formally define this, let $\mathcal{T}$ denote a class of contracts. Let $U_P(\theta \mid \mathbf{t})$ denote the expected principal utility for contract $\mathbf{t}$ when the agent's type is θ, and let π be a policy which maps each history $\mathcal{H}^{s-1}$ to a distribution over contracts.

[3] Since regret is an additive metric, we need to specify the range of the key quantities involved. Normalization to $[0, 1]$ can be achieved through appropriate scaling, but also scales the regret with respect to the original unscaled instance.

Then

$$\mathsf{regret}(\pi, \mathcal{T}) := \sup_{\bar{\mathbf{t}} \in \mathcal{T}} \sum_{s=1}^{S} \mathbb{E}_{\mathbf{t}^s \sim \pi(\mathcal{H}^{s-1})} \left(\mathbb{E}_{\theta^s}[U_P(\theta^s \mid \bar{\mathbf{t}})] - \mathbb{E}_{\theta^s}[U_P(\theta^s \mid \mathbf{t}^s)] \right).$$

Separation: General vs. Linear Contracts. The main results of Zhu et al. (2023) are near-tight bounds on the regret, showing that while the regret for (general) bounded contracts has to be essentially linear, linear contracts admit sublinear regret bounds.

Theorem 5 (Zhu, Bates, Yang, Wang, Jiao, and Jordan (2023)).
(1) For (general) bounded contracts there is an online learning algorithm that incurs a regret of at most $\tilde{O}(\sqrt{m} \cdot S^{1-1/(2m+1)})$, and no online learning algorithm can incur a regret better than $\Omega(S^{1-1/(m+2)})$.
(2) For linear contracts there is an online learning algorithm that incurs a regret of at most $\tilde{O}(S^{2/3})$, and no online learning algorithm can incur a regret better than $\Omega(S^{2/3})$.

The intuitive reason for the sharp separation between (general) bounded contracts and linear contracts is the very different geometry of the principal's utility in the two cases. For linear contracts (Figure 3b), the optimal principal utility is a well-behaved, right-continuous function. In particular, there is always a "safe" direction: if $\alpha^\star$ is the optimal linear contract and we over-shoot by ϵ, then we only lose ϵ utility. By contrast, for general (bounded) contracts, the principal's utility admits only a weak continuity property: for each contract $\mathbf{t}$, there exists a direction (a cone) along which the utility does not drop too much. While this enables a cover-by-cones argument, yielding the exponential upper bound, the lower-bound construction shows there is no additional structure that would enable more efficient learning.

Notably, the impossibility for (general) bounded contracts already holds for a fixed agent (i.e, when $\mathcal{D}$ is a point-mass distribution), and requires exponentially many (in m) actions. The case of a fixed agent with a small action space is handled in Bacchiocchi et al. (2024). An appealing open problem is whether the exponential lower bound can be circumvented under additional structure, e.g., MLRP (it can, for instances satisfying FOSD and CDFP (Chen et al., 2024)).

7. LOOKING AHEAD

Attesting to the rapid growth of the new research area of algorithmic contract design, in this brief survey we have not been able to touch upon all lines of research. For example, there has been much interest in *typed* contract design: Bayesian environments where agents have private type information, which call for a unified treatment of mechanism design and contract design (Guruganesh et al., 2021; Alon et al., 2021, 2023; Castiglioni et al., 2025, 2023).

While substantial progress has been made in developing the foundations of algorithmic contract design, several directions remain only partially explored. These additional directions include inspection and monitoring of actions and outcomes (Ezra et al., 2026), the interplay with information revelation (Babichenko et al., 2024; Castiglioni and Chen, 2025), ambiguous contract design (Dütting et al., 2024, 2025), sequential decision making (Ezra et al., 2026), and classification (Kleinberg and Raghavan, 2019; Alon et al., 2020) (as in Example 2). It would also be interesting to further explore the connection to closely related approaches such as scoring rules (Hartline et al., 2023) and delegation (Kleinberg and Kleinberg, 2018).

Finally, circling back to Example 3, the rise of an "agentic era," where autonomous AI agents act as both principals and agents, raises new challenges only beginning to be addressed (Saig et al., 2024). Vast and dynamic action spaces, shifting assumptions about rationality and commitment, and the computational capabilities of AI agents open fresh avenues for contract design. As in algorithmic mechanism design, we expect the computational lens to reveal new structures, tractability frontiers, and insights, shaping the next phase of contract theory.

Acknowledgments

This survey received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 101077862) and the Horizon Europe program for research and innovation (grant agreement No. 101170373), by the Israel Science Foundation (grant No. 3331/24), by the Israel Science Foundation Breakthrough Program (grant No. 2600/24), by the NSF-BSF (grant No. 2020788 and grant No. 2021680), by a Google Research Scholar Award, and by an Amazon Research Award.

REFERENCES

Alon, T., Dobson, M., Procaccia, A. D., Talgam-Cohen, I., and Tucker-Foltz, J. 2020. Multiagent evaluation mechanisms. In *Proc. of AAAI 2020*. AAAI Press, Washington, DC, USA, 1774–1781.

Alon, T., Dütting, P., Li, Y., and Talgam-Cohen, I. 2023. Bayesian analysis of linear contracts. In *Proc. of EC 2023*. ACM, New York, NY, USA. Full version available at: `https://arxiv.org/abs/2211.06850`.

Alon, T., Dütting, P., and Talgam-Cohen, I. 2021. Contracts with private cost per unit-of-effort. In *Proc. of EC 2021*. ACM, New York, NY, USA, 52–69.

Alon, T., Lavi, R., Shamash, E. S., and Talgam-Cohen, I. 2024. Incomplete information VCG contracts for common agency. *Oper. Res. 72,* 1, 288–299. An earlier version appeared in EC 2021.

Babaioff, M., Feldman, M., Nisan, N., and Winter, E. 2012. Combinatorial agency. *J. Econ. Theory 147,* 3, 999–1034. An earlier version appeared in EC 2006.

Babichenko, Y., Talgam-Cohen, I., Xu, H., and Zabarnyi, K. 2024. Information design in the principal-agent problem. In *Proc. EC 2024*. ACM, New York, NY, USA, 669–670.

Bacchiocchi, F., Castiglioni, M., Marchesi, A., and Gatti, N. 2024. Learning optimal contracts: How to exploit small action spaces. In *Proc. of ICLR 2024*. PMLR.

Carroll, G. 2015. Robustness and linear contracts. *Am. Econ. Rev. 105,* 2, 536–63.

Castiglioni, M. and Chen, J. 2025. Hiring for an uncertain task: Joint design of information and contracts. In *Proc. of SODA 2025*. SIAM, Philadelphia, PA, USA, 1758–1794.

Castiglioni, M., Chen, J., Li, M., Xu, H., and Zuo, S. 2025. A reduction from multi-parameter to single-parameter Bayesian contract design. In *Proc. of SODA 2025*. SIAM, Philadelphia, PA, USA, 1795–1836.

Castiglioni, M., Marchesi, A., and Gatti, N. 2023. Designing menus of contracts efficiently: The power of randomization. *Artif. Intell. 318*, 103881. An earlier version appeared in EC 2022.

Chen, Y., Chen, Z., Deng, X., and Huang, Z. 2024. Are bounded contracts learnable and approximately optimal? In *Proc. of EC 2024*. ACM, New York, NY, USA.

Deo-Campo Vuong, R., Dughmi, S., Patel, N., and Prasad, A. 2024. On supermodular contracts and dense subgraphs. In *Proc. of SODA 2024*. SIAM, Philadelphia, PA, USA, 109–132.

Dütting, P., Ezra, T., Feldman, M., and Kesselheim, T. 2025a. Combinatorial contracts. *SIAM J. Comput. 54,* 6, 1427–1455. An earlier version appeared in FOCS 2021.

Dütting, P., Ezra, T., Feldman, M., and Kesselheim, T. 2025b. Multi-agent combinatorial contracts. In *Proc. of SODA 2025*. SIAM, Philadelphia, PA, USA, 1857–1891.

Dütting, P., Ezra, T., Feldman, M., and Kesselheim, T. 2026. Multi-agent contracts. *J. ACM 73,* 2, 13:1–13:37. An earlier version appered in STOC 2023.

Dütting, P., Feldman, M., and Gal-Tzur, Y. 2024. Combinatorial contracts beyond gross substitutes. In *Proc. of SODA 2024*. SIAM, Philadelphia, PA, USA, 92–108.

Dütting, P., Feldman, M., Gal-Tzur, Y., and Rubinstein, A. 2026. When contracts get complex: Information-theoretic barriers. In *Proc. of SODA 2026*. SIAM, Philadelphia, PA, USA, 6464–6493.

Dütting, P., Feldman, M., Peretz, D., and Samuelson, L. 2024. Ambiguous contracts. *Econometrica 92,* 6, 1967–1992. An earlier version appeared in EC 2023.

Dütting, P., Feldman, M., Ponitka, T., and Soumalias, E. 2025. The pseudo-dimension of contracts. In *Proc. EC 2025*. ACM, New York, NY, USA, 514–539.

Dütting, P., Feldman, M., and Rashti, Y. 2025. Succinct ambiguous contracts. Available at `https://arxiv.org/abs/2503.02592`.

Dütting, P., Feldman, M., and Talgam-Cohen, I. 2024. Algorithmic contract theory: A survey. *Found. Trends Theor. Comput. Sci. 16,* 3-4, 211–412. `https://arxiv.org/abs/2412.16384`.

Dütting, P., Guruganesh, G., Schneider, J., and Wang, J. 2023. Optimal no-regret learning of one-sided Lipschitz functions. In *Proc. of ICML 2023*. PMLR.

Dütting, P., Roughgarden, T., and Talgam-Cohen, I. 2019. Simple versus optimal contracts. In *Proc. of EC 2019*. ACM, New York, NY, USA, 369–387. Full version available at `https://arxiv.org/pdf/1808.03713`.

Dütting, P., Roughgarden, T., and Talgam-Cohen, I. 2021. The complexity of contracts. *SIAM J. Comput. 50,* 1, 211–254. An earlier version appeared in SODA 2020.

Ezra, T., Feldman, M., and Schlesinger, M. 2024. On the (in)approximability of combinatorial contracts. In *Proc. of ITCS 2024*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Wadern, Germany, 44:1–44:22.

Ezra, T., Feldman, M., and Schlesinger, M. 2026. Contract design for sequential actions. In *Proc. of SODA 2026*. SIAM, Philadelphia, PA, USA, 6537–6570.

Ezra, T., Leonardi, S., and Russo, M. 2026. Contract design beyond hidden actions. In *Proc. of SODA 2026*. SIAM, Philadelphia, PA, USA, 2113–2133.

Feldman, M. 2025. Combinatorial contract design: Recent progress and emerging frontiers. *arXiv preprint arXiv:2510.15065*.

Grossman, S. J. and Hart, O. D. 1983. An analysis of the principal-agent problem. *Econometrica 51,* 1, 7–45.

Guruganesh, G., Schneider, J., and Wang, J. 2021. Contracts under moral hazard and adverse selection. In *Proc. of EC 2021*. ACM, New York, NY, USA, 563–582.

Hartline, J. D., Shan, L., Li, Y., and Wu, Y. 2023. Optimal scoring rules for multi-dimensional effort. In *Proc. of COLT 2023*. PMLR, 2624–2650.

Hermalin, B. E. and Katz, M. L. 1991. Moral hazard and verifiability: The effects of renegotiation in agency. *Econometrica 59*, 1735–1753.

Ho, C., Slivkins, A., and Vaughan, J. W. 2016. Adaptive contract design for crowdsourcing markets: Bandit algorithms for repeated principal-agent problems. *J. Artif. Intell. Res. 55*, 317–359. An earlier version appeared in EC 2014.

Holmström, B. 1979. Moral hazard and observability. *Bell J. Econ. 10*, 74–91.

Holmström, B. 1982. Moral hazard in teams. *Bell J. Econ. 13*, 324–340.

Kleinberg, J. and Raghavan, M. 2019. How do classifiers induce agents to invest effort strategically? In *Proc. of EC 2019*. ACM, New York, NY, USA, 825–844.

Kleinberg, J. M. and Kleinberg, R. 2018. Delegated search approximates efficient search. In *Proc. of EC 2018*. ACM, New York, NY, USA, 287–302.

Nisan, N. and Ronen, A. 2001. Algorithmic mechanism design. *Games Econ. Behav. 35,* 1, 166–196. An earlier version appeared in STOC 1999.

Saig, E., Einav, O., and Talgam-Cohen, I. 2024. Incentivizing quality text generation via statistical contracts. In *Proc. of NeurIPS 2024*. PMLR.

Wang, T., Dütting, P., Ivanov, D., Talgam-Cohen, I., and Parkes, D. C. 2023. Deep contract design via discontinuous networks. In *Proc. of NeuRIPS 2023*. PMLR.

Zhu, B., Bates, S., Yang, Z., Wang, Y., Jiao, J., and Jordan, M. I. 2023. The sample complexity of online contract design. In *Proc. of EC 2023*. ACM, New York, NY, USA, 1188.