Combinatorial Contracts

Paul Dütting*, Tomer Ezra†, Michal Feldman†‡ and Thomas Kesselheim§
*Google Research, Zürich, Switzerland, duetting@google.com
†School of Computer Science, Tel Aviv University, Tel Aviv, Israel, tomer.ezra@mail.tau.ac.il, mfeldman@tau.ac.il
‡Microsoft Research, Herzliya, Israel
§Institute of Computer Science, University of Bonn, Bonn, Germany, thomas.kesselheim@uni-bonn.de

***Abstract*—We introduce a new model of combinatorial contracts in which a principal delegates the execution of a costly task to an agent. To complete the task, the agent can take any subset of a given set of unobservable actions, each of which has an associated cost. The cost of a set of actions is the sum of the costs of the individual actions, and the principal's reward as a function of the chosen actions satisfies some form of diminishing returns. The principal incentivizes the agents through a contract, based on the observed outcome.**

Our main results are for the case where the task delegated to the agent is a project, which can be successful or not. We show that if the success probability as a function of the set of actions is gross substitutes, then an optimal contract can be computed with polynomially many value queries, whereas if it is submodular, the optimal contract is NP-hard. All our results extend to linear contracts for higher-dimensional outcome spaces, which we show to be robustly optimal given first moment constraints.

Our analysis uncovers a new property of gross substitutes functions, and reveals many interesting connections between combinatorial contracts and combinatorial auctions, where gross substitutes is known to be the frontier for efficient computation.

***Keywords*-contract theory; moral hazard; gross substitutes**

I. Introduction

Contract theory is one of the pillars of microeconomic theory (cf., the 2016 Nobel Prize in Economics for Hart and Holmström [1]). Indeed, contract theory and its central principal-agent (hidden-action) model play a similar role for markets of *services*, as the theory of mechanism design and its central (combinatorial) auctions model play for markets of *goods*. The past few years have seen classic applications of contract theory moving online, and with it computational, algorithmic, and optimization approaches are becoming more relevant. Applications range from crowdsourcing platforms, to online labor markets, to online marketing.

In the classic hidden-action / principal-agent model of Holmström [2] and Grossman and Hart [3], a principal delegates a task to an agent. The agent can take one of n costly actions (e.g., effort levels), and these trigger some distribution over m possible rewards that go to the principal. The principal cannot observe the action taken by the agent; she can only observe the obtained outcome. This model captures an incentive problem, which is quite different from that in mechanism design and auctions, commonly referred to as "moral hazard": in and by itself the agent has no interest in taking a costly action. A contract defines a transfer—from the principal to the agent—for each possible stochastic outcome, and serves to incentivize the agent to exert effort.

In this setting, an *optimal* contract, that is, one that maximizes the principal's utility assuming that the agent best responds to the contract, can be computed in time polynomial in n and m (as was already shown in [3]). The idea is to determine for each action whether the principal can incentivize the agent to take it via linear programming. Out of these, the one maximizing the principal's utility will be selected.

One thing that the classic model does not explicitly capture—a well recognized fact in the Econ literature (cf., the influential multi-tasking paper of Holmström and Milgrom [4])—is that typically performing a complex task entails taking a *set* of actions and that the principal's reward as a function of the chosen actions satisfies some form of diminishing returns.

Of course, this could be modeled by the classic model by writing down one meta-action for each of the 2^n subsets of actions. However, this approach would ignore all structure of the underlying problem and, in particular, following the above blueprint to compute an optimal contract would require considering exponentially many subsets of actions.

In this work we propose a new model of combinatorial contracts that captures the additional structure in problems where a task entails taking a set of actions. Within this model we study the design of optimal contracts through a computational lens. We establish a non-trivial positive computational result, as well as hardness results, and en-route we reveal several interesting connections to combinatorial auctions.

A model of combinatorial contracts: In our base model a principal seeks to delegate a project to an agent, and the project can either succeed or fail. The principal has a value for the project to succeed, which we normalize to 1. There is a ground set A of n actions. The agent can choose any

Full version available from https://arxiv.org/abs/2109.14260. This research is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 866132), the ISF (grant number 317/17), and the NSF-BSF (grant number 2020788).

2575-8454/21/$31.00 ©2021 IEEE
DOI 10.1109/FOCS52979.2021.00084

subset $S \subseteq A$ of these actions. If the agent chooses to take actions S, the project succeeds with probability $f(S)$. Each of the actions $a \in A$ has a cost $c(a)$, and the cost of a set of actions S is the sum of the costs $c(a)$ for $a \in S$.

To incentivize the agent to take a certain set of actions, the principal defines a contract. As in the classic model we assume that the actions are hidden, so the contract can only depend on the outcome of the project. Therefore, the principal's only choice is to define payments $t(1)$ and $t(0)$ to the agent for the case that the project succeeds or fails, respectively. The agent will then choose actions so that the expected difference of payment and costs is maximized. The principal's goal in turn is to maximize the expected value minus the payment, where the expected value is equal to the success probability because of normalization.

We also consider a generalization of the base model, in which the outcome space is not binary (a project that succeeds or fails), but rather a vector of m rewards to the principal. In this more general model, each set of actions entails a different probability distribution over rewards, with $f_j(S)$ for $j \in \{1, \ldots, m\}$ being the probability of outcome j under actions S.

The fact that the principal's reward satisfies some form of diminishing returns (e.g., some form of submodularity) then naturally translates to corresponding assumptions on the probability distributions.

Computing optimal contracts: Our main results are for the base model where the principal delegates a project to an agent, and the project can be successful or not. We observe that in this case, a linear contract is optimal, that is setting $t(1) = \alpha$ and $t(0) = 0$ in the notation from above (recall that the principal's reward upon success is normalized to 1). Under such a contract, the agent chooses the set S that maximizes $\alpha \cdot f(S) - \sum_{a \in S} c(a)$, or equivalently, $f(S) - \frac{1}{\alpha} \sum_{a \in S} c(a)$. That is, the agent's problem is equivalent to resolving a *demand query* at prices $\frac{1}{\alpha} c(a)$ in the framework of combinatorial auctions.

Our main positive result is for the case where the success-probability function f is gross substitutes — a strict subclass of submodular functions that includes natural functions as special cases (e.g., additive, unit demand, and matroid rank functions), and plays a central role in both economics and computer science.

Main Theorem 1: For gross substitutes success probability functions, the optimal contract can be computed in time polynomial in the number of actions n given access to a value oracle (namely an oracle that for any set S computes the value $f(S)$ in polynomial time).

A key object in our analysis is the set of *critical* values of α — values that are potential candidates for an optimal contract. Our key technical observations are, firstly, that for every function f there are only finitely many critical values of α. Secondly, in case of a gross-substitutes function f, the number of such critical points is bounded polynomially in n. Thirdly, we can efficiently iterate over these critical points. Together this gives us a polynomial-time algorithm.

In order to bound the number of critical points, our main insight is that for a gross-substitutes function f there can be only $O(n^2)$ maximizers S of $f(S) - \frac{1}{\alpha} \sum_{a \in S} c(a)$ for different values of α. In the language of combinatorial auctions, this means that the number of changes in the demand correspondence when prices (of all items simultaneously) are being scaled linearly is bounded by $O(n^2)$. We prove this by uncovering a new property of gross substitutes functions, according to which, generically, as we increase α (decrease prices), whenever the demand set changes, either a single item enters the demand set or an expensive item replaces a cheaper one in a one-to-one fashion. This then implies the claimed bound through a potential function argument.

For cases beyond gross substitutes success probability functions f, we show a hardness result, which applies even for budget additive functions.

Main Theorem 2: For budget additive (and, hence, submodular) success probability functions, computing the optimal contract is NP-hard.

We prove this result by a reduction from the problem of subset sum. In addition, we show that our approach used for the gross substitutes case, of going over all the critical points, does not work for the case of submodular success probability functions (or even coverage functions). We show this by recursively constructing a coverage function with exponentially many (in the number of actions) critical points.

Finally, we show that for every function f (whether submodular or not) there is an FPTAS, which computes a $(1 - \epsilon)$-approximation with only $\mathrm{poly}(k, \frac{1}{\epsilon})$ evaluations of a contract, where k is the number of bits used to represent values of f and c. To obtain this result, we observe that we can restrict our attention to $\mathrm{poly}(k, \frac{1}{\epsilon})$ candidate contracts, which we have to compare, one of which will be a $(1 - \epsilon)$-approximation. Moreover, we show that one can iterate over all critical points in weakly polynomial time. This implies that if the number of critical points is polynomially bounded, then we can compute the optimal contract in weakly polynomial time.

We then turn to more general outcome spaces, where there is a vector of m rewards, and each set of actions $S \subseteq A$ induces a distribution over these rewards. A linear contract for this setting specifies which fraction α of the reward goes to the agent. Unlike in the case of a binary outcome, linear contracts are no longer optimal for this more general setting. However, as we show, they are robustly optimal in a max-min sense, when for each action only the expected reward rather than the exact distribution is known, and the principal wishes to maximize her utility in the worst-case over all compatible distributions.

Moreover, all our results for optimal (linear) contracts for the case of a binary outcome translate to linear contracts for this more general setting: We can compute an optimal linear contract (which is max-min optimal among all contracts) for the case where the principal's expected reward as a function of the set of actions is a gross substitute function in time polynomial in n and m; computing an optimal contract for submodular functions is hard, while computing an approximately optimal contract for general functions is weakly polynomial.

Open problems: Our work leaves a number of exciting research questions. One direction is to explore whether it is possible to compute the optimal contract given access to a demand oracle. Our algorithm for gross substitutes functions provides the blueprint for an algorithm that runs in polynomially many demand oracle queries if there are only polynomially many critical values of α. While we know that this set can be exponential in size for coverage functions, we don't know whether this is the case for other subclasses of submodular functions, such as budget-additive. It would also be interesting to understand whether for submodular functions there can be a polytime algorithm with demand queries despite the fact that there can be exponentially many critical values of α. A second direction is to explore the existence of polytime approximation algorithms without access to a demand oracle for classes of principal reward functions where computing the exact optimal contract is hard. A third direction is to explore the computation of optimal (possibly non-linear) contracts when the outcome is multi-dimensional. Finally, it would be interesting to study settings with non-additive costs.

A. Related Work

Contract theory: The two foundational papers of contract theory are the aforementioned papers of Holmström [2] and Grossman and Hart [3]. In addition to the basic model, these papers contain the linear programming approach to computing optimal contracts. In another classic paper, Holmström and Milgrom [5] study a multi-round interaction between a principal and an agent, and show that under the assumptions of that model a linear contract is optimal. Holmström and Milgrom [4] consider a model similar to ours, but consider a fractional allocation of efforts to actions which makes their model less amenable to a computational analysis.

Carroll [6] assumes that there is a fixed set of known actions but the actual set of actions is a superset of these. He shows that then a linear contract is max-min optimal. In a similar spirit, Dütting et al. [7] consider the case that only the *expected* rewards of actions are known but not their actual distributions. They show that linear contracts are also max-min optimal in this setting (we show a similar result in Section VI for the case where the agent chooses any set of actions, not necessarily a single action). Besides this, they also discuss how well a linear contract can approximate an optimal one.

Babaioff et al. [8] turn to a setting in which there are multiple agents. They introduce the combinatorial agency model, where a principal incentivizes a *team* of agents to exert costly effort on his behalf in equilibrium, and the outcome depends on the complex combinations of the efforts by the agents. It generalizes an earlier work by Feldman et al. [9] for a simple multi-hop routing. Follow-up work by Babaioff et al. [10], [11] study the effect of mixed strategies and free riding in combinatorial agency. The combinatorial explosion in these papers comes from a similar source as in our paper. For example, if there are k agents and each agent has 2 actions, then there are 2^k action profiles. An important difference to our work is in the incentive compatibility constraint, which in these papers has to hold for each agent individually, and leads these papers to study equilibria, while in our paper there is only a single agent who will choose the best set of actions. Another difference is that because of the many agents perspective these papers did not make the connection to demand queries in combinatorial auctions. They also did not study gross substitutes or submodular principal reward functions.

Dütting et al. [12] consider a problem with one agent choosing one of n actions. There are m different success events, each action causes each event to happen with a certain probability. As these random draws are independent, there are 2^m different subsets of success events that can take place, so the number of outcomes is exponential in the input size. This is a similar but orthogonal question to what we study in this paper, where the exponential growth comes from the number of different combinations of actions the agent can choose.

Ho et al. [13] consider a contract design question in a repeated setting from a bandit-learning angle. Kleinberg and Kleinberg [14] and Kleinberg and Raghavan [15] consider problems that can be thought of as contract design without money. Guruganesh et al. [16] and Alon et al. [17] consider settings in which screening (that is, hidden types) is combined with moral hazard (that is, hidden action).

Gross substitutes functions: Gross substitute functions play a central role in economics (e.g., [18]). They have been independently discovered in mathematics, under a different name in the context of discrete convex analysis, see [19], [20]. The class of gross substitutes functions is a strict subclass of submodular functions [21], which includes natural functions such as additive, unit demand, and matroid rank functions as special cases. This class plays a central role in the analysis of combinatorial markets; for example, it is the frontier for both market equilibrium existence [18], [22], and for the efficient computation of a welfare-maximizing allocation [23]. Its original definition uses the notions of prices, utility and demand [18], but due to its centrality in combinatorial markets, it has attracted a lot of work that furthers our understanding of its characteristics (e.g., [24], [25], [26]); see [27] for an influential algorithmic survey.

B. Paper Structure

We present our problem in Section II. In Section III we present some useful insights on the structure of the problem. Our polynomial time algorithm for gross substitutes success probability functions is presented in Section IV. In Section V we study success probability functions beyond gross substitutes: in Section V-A we provide negative results for submodular functions; in Section V-B we present a FPTAS for general success probability functions; and in Section V-C we present a weakly poly-time algorithm for instances with poly-size critical sets. In Section VI, we discuss the case of non-binary outcomes. Due to space limitations, some proofs are deferred to the full version.

II. The Combinatorial Contracts Problem

Hidden-action principal-agent setting: There is a single principal and a single agent. There is a set $A = \{1, \ldots, n\}$ of n possible actions. The strategy of the agent consists of a set of actions $S \in 2^A$. Every action $a \in A$ is associated with a positive cost $c(a) > 0$. The cost of a set of actions $S \in 2^A$ is additive; i.e., $c(S) = \sum_{a \in S} c(a)$. The cost of not taking any actions $c(\emptyset)$ is zero.

We focus on the case of a binary outcome space $\Omega = \{0, 1\}$. (We consider more general, higher dimensional outcome spaces in Section VI.) Outcome 0 corresponds to failure, and outcome 1 corresponds to success. The principal derives a reward $r(1) \in \mathbb{R}_{\geqslant 0}$ from outcome 1 (i.e., success), and a reward of $r(0) = 0$ from outcome 0 (i.e., failure).

Every strategy $S \in 2^A$ by the agent has an associated success probability $f : 2^A \to [0, 1]$. We assume that $f(\emptyset) = 0$, and that f is monotonically non-decreasing, i.e., $S \subseteq S'$ implies that $f(S) \leqslant f(S')$.

The principal cannot directly observe the set of actions chosen by the agent, but she can observe the stochastic outcome of the chosen set of actions.

The contract design problem: A contract $t : \Omega \to \mathbb{R}_{\geqslant 0}$ is a mapping from outcomes to non-negative payments for each outcome. In the binary case, a contract thus corresponds to two numbers, $t(0)$ and $t(1)$, the payment upon failure and success.

The principal's expected reward for a set of actions $S \in 2^A$ is $R(S) = f(S) \cdot r(1)$. The expected payment from the principal to the agent for a set of actions $S \in 2^A$ is defined as $T(S) = (1 - f(S)) \cdot t(0) + f(S) \cdot t(1)$. The principal's expected utility from a set of actions $S \in 2^A$ is

$$u_p(S, t) = R(S) - T(S).$$

The agent's expected utility from a set of actions $S \in 2^A$ is

$$u_a(S, t) = T(S) - c(S).$$

A set of actions $S \in 2^A$ is *a best response* to a contract t (we also say it is *incentivized* by contract t) if it yields the highest possible utility to the agent, where we assume that the agent breaks ties in favor of the principal.

Formally, let

$$\mathcal{D}(t) = \arg\max_{S' \in 2^A} u_a(S', t)$$

be the collection of sets of actions that maximize the agent's utility. Then the collection of sets of actions that are incentivized by the contract is

$$\mathcal{D}^\star(t) = \arg\max_{S' \in \mathcal{D}(t)} u_p(S', t) \subseteq \mathcal{D}(t).$$

The assumption that the agent breaks ties in favor of the principal is a standard one in the contracts literature (see, e.g., [6]). It is motivated by the fact that one could perturb payments slightly to achieve the same effect.

Note that we have set things up so that the best response condition (or the "IC constraint") implies individual rationality, i.e., that the agent's utility is non-negative. Also note that for any $S \in \mathcal{D}^\star(t)$, the principal's utility $u_p(S, t)$ is the same. We can thus define $u_p(t)$ to be the principal's utility $u_p(S, t)$ from any set $S \in \mathcal{D}^\star(t)$.

The computational problem that we are interested in is that of computing a contract that maximizes the principal's utility.

OPT-CONTRACT:
Input: Action set A, outcome set Ω, rewards $r(j)$ for $j \in \Omega$, costs $c(i)$ for $i \in A$, oracle access to f
Output: Contract t that maximizes $u_p(t)$

The following simple observation, will allow us to narrow down the search space:

Observation II.1. *For any contract t there is a contract t' such that $t'(0) = 0$ that yields a weakly higher utility to the principal.*

We can thus focus on contracts t such that $t(0) = 0$. Any such contract can be expressed by a single parameter α such that $t(1) = \alpha \cdot r(1)$. This motivates identifying contracts t with their α, and replacing t with α and $t(1)$ with $\alpha \cdot r(1)$ in the definitions above. For example, we replace $\mathcal{D}(t)$ and $\mathcal{D}^\star(t)$ by $\mathcal{D}(\alpha)$ and $\mathcal{D}^\star(\alpha)$, respectively.

Hereafter, we normalize $r(1)$ to be 1. Hence, $R(S) = f(S)$, $t(1) = \alpha$, and $f(S) \cdot r(1)$ can be replaced by $f(S)$.

Example II.1. *Consider the success probability function $f : 2^{\{1,2,3\}} \to [0, 1]$ where: $f(\emptyset) = 0$, $f(\{1\}) = f(\{2\}) = 0.3, f(\{1, 2\}) = 0.5$, $f(\{3\}) = f(\{1, 3\}) = f(\{2, 3\}) = f(\{1, 2, 3\}) = 0.6$, and $c(1) = c(2) = 0.1$, $c(3) = 0.3$. (One can verify that f is submodular but not gross substitutes, see definitions below.) Consider two contracts, $\alpha_1 = 1$ and $\alpha_2 = 0.5$. For $\alpha_1 = 1$ it holds that $\mathcal{D}(\alpha_1) = \{\{1, 2\}, \{3\}\}$ since both sets give the agent an expected utility of 0.3. On* *the other hand $\mathcal{D}^\star(\alpha_1) = \{\{3\}\}$ since this is the only set in the demand that maximizes the utility of the principal. For $\alpha_2 = 0.5$ it holds that $\mathcal{D}(\alpha_2) = \mathcal{D}^\star(\alpha_2) = \{\{1\},\{2\}\}$ since both sets give the maximized expected utility to the agent and principal.*

The agent's problem: For a fixed contract t with $t(0) = 0$ or the corresponding α, the agent's problem is to find a set $S \in \mathcal{D}^\star(\alpha)$.

BEST-RESPONSE:
Input: Contract $\alpha \in (0,1]$
Output: Some set $S \in \mathcal{D}^\star(\alpha)$

To determine whether $S \in \mathcal{D}(\alpha)$ for some $\alpha \in (0,1]$ we need to compare the agent's utility for pairs of sets of actions $S, S' \in 2^A$. Rather than comparing $u(S,\alpha)$ to $u(S',\alpha)$ we can equivalently compare $u(S,\alpha)/\alpha$ and $u(S',\alpha)/\alpha$, i.e.,

$$f(S) - \sum_{i \in S} c(i)/\alpha \quad \text{and} \quad f(S') - \sum_{i \in S'} c(i)/\alpha.$$

Success probability functions: It is natural to impose some form of "decreasing marginal returns" on the success probabilities $f : 2^A \to [0,1]$ as a function of the set of actions taken. We consider the following classes of functions:

- The function f is *submodular* if for every $S, S' \in 2^A$ with $S \subseteq S'$ and every $i \in A \backslash S'$ we have $f(S \cup \{i\}) - f(S) \geqslant f(S' \cup \{i\}) - f(S')$.
- The function f is *budget additive* if there exists a budget $B \in [0,1]$ such that for every $S \in 2^A$ we have $f(S) = \min\{B, \sum_{i \in S} f(\{i\})\}$.
- The function f is *coverage* if there exists a finite set U, where every element $j \in U$ is associated with a weight $w_j \in \mathbb{R}_{\geqslant 0}$, and a function $g : A \to 2^U$ such that for every set $S \in 2^A$, $f(S) = \sum_{j \in \bigcup_{i \in S} g(i)} w_j$.
- The function f is *gross substitutes* if for any two vectors $p, q \in \mathbb{R}^n_{\geqslant 0}$ such that $q \geqslant p$ and any $S \in 2^A$ such that $f(S) - \sum_{i \in S} p_i \in \arg\max_{S' \in 2^A} f(S') - \sum_{i \in S'} p_i$ there is a $T \in 2^A$ such that $f(T) - \sum_{i \in T} q_i \in \arg\max_{T' \in 2^A} f(T') - \sum_{i \in T'} q_i$ and $T \supseteq \{i \in S \mid q_i = p_i\}$.
- The function f is *unit demand* if for every $S \in 2^A$ we have $f(S) = \max_{i \in S} f(\{i\})$.

Unit demand functions are gross substitutes. Gross substitutes functions, coverage functions, and budget additive functions are submodular. Gross substitutes functions, coverage functions, and budget additive functions are incomparable to each other.

Value vs. demand queries: The success probabilities $f : 2^A \to [0,1]$ are combinatorial objects, whose explicit description size is exponential in n. For computational questions we therefore consider the following two types of oracle access to these functions:

- A *value oracle* receives a set $S \in 2^A$ as input and returns $f(S)$.
- A *demand oracle* receives a vector $p \in \mathbb{R}^n_{\geqslant 0}$ of "prices" as input and returns some set S that maximizes the "utility" $f(S) - \sum_{i \in S} p_i$.

Demand oracles are a natural assumption in the context of combinatorial auctions, where they correspond to asking the agent for a set of items that maximizes his utility given item prices. As we shall see, they also play a natural role in combinatorial contracts. For example, the agent's problem above is essentially solving a demand query.

III. Structural Insights

In this section we present some useful insights regarding the structure of $\mathcal{D}_{f,c}(\alpha)$ and $\mathcal{D}^\star_{f,c}(\alpha)$ (recall that $\mathcal{D}_{f,c}(\alpha)$ is the collection of sets of actions maximizing the agent's utility under contract α, and $\mathcal{D}^\star_{f,c}(\alpha)$ is the subset among these that maximize the principal's utility). We also define *critical* values of α as ones for which the demand changes (formal details below). As we shall see, the set of critical values of α is useful in calculating the optimal contract.

Given some α, let $V_{f,c}(\alpha)$ denote the value of $f(S)$ for $S \in \mathcal{D}^\star_{f,c}(\alpha)$. Note that this value is well defined since all sets $S \in \mathcal{D}^\star_{f,c}(\alpha)$ have the same value.

For example, in Example II.1, it holds that $V_{f,c}(1) = 0.6$ since $\mathcal{D}^\star_{f,c}(1) = \{\{3\}\}$ and $f(\{3\}) = 0.6$. Similarly, $V_{f,c}(0.5) = 0.3$ since $\mathcal{D}^\star_{f,c}(0.5) = \{\{1\},\{2\}\}$ and $f(\{1\}) = f(\{2\}) = 0.3$.

The following proposition establishes a monotonicity property of the demand as a function of α.

Proposition III.1. *Let $0 \leqslant \alpha_1 < \alpha_2$. For every $S_1 \in \mathcal{D}_{f,c}(\alpha_1)$ and $S_2 \in \mathcal{D}_{f,c}(\alpha_2)$, it holds that $f(S_1) \leqslant f(S_2)$.*

Proof: Let $S_1 \in \mathcal{D}_{f,c}(\alpha_1)$ and let $S_2 \in \mathcal{D}_{f,c}(\alpha_2)$. By these definitions, we have $\alpha_1 f(S_1) - c(S_1) \geqslant \alpha_1 f(S_2) - c(S_2)$ and $\alpha_2 f(S_2) - c(S_2) \geqslant \alpha_2 f(S_1) - c(S_1)$. Adding these inequalities implies $\alpha_1 f(S_1) + \alpha_2 f(S_2) \geqslant \alpha_2 f(S_1) + \alpha_1 f(S_2)$, or equivalently, $(\alpha_2 - \alpha_1) f(S_1) \leqslant (\alpha_2 - \alpha_1) f(S_2)$. Dividing by $\alpha_2 - \alpha_1$ implies the claim. ■

Note that $V_{f,c}(\alpha) = \max_{S \in \mathcal{D}_{f,c}(\alpha)} f(S)$. Thus, Proposition III.1 implies monotonicity of $V_{f,c}$.

Corollary III.2. *For every $0 \leqslant \alpha_1 < \alpha_2$, $V_{f,c}(\alpha_1) \leqslant V_{f,c}(\alpha_2)$.*

We next observe that $V_{f,c}$ is right continuous. The proof exploits the fact that the agent breaks ties in favor of the principal.

Observation III.3. *For every $\alpha > 0$, $\lim_{\epsilon \to 0^+} V_{f,c}(\alpha + \epsilon)$ and $\lim_{\epsilon \to 0^+} V_{f,c}(\alpha - \epsilon)$ are well defined. Moreover,*

$$V_{f,c}(\alpha) = \lim_{\epsilon \to 0^+} V_{f,c}(\alpha + \epsilon).$$

To summarize, $V_{f,c}$ is a monotone, right-continuous function whose image is contained in the image of f. Thus, it must be a "step function" with at most 2^n steps. This is cast in the following corollary.

Corollary III.4. *There exists some $k < 2^n$ and a series of α values $0 = \alpha_0 < \alpha_1 < \ldots < \alpha_k \leqslant 1$ such that for every $x \in [0,1]$, $V_{f,c}(x) = V_{f,c}(\alpha)$, where $\alpha = \max\{\alpha_i \mid \alpha_i \leqslant x\}$. Moreover, for every $0 \leqslant i < j \leqslant k$, $V_{f,c}(\alpha_i) < V_{f,c}(\alpha_j)$.*

Every α_i from the last corollary (except for α_0) is said to be a *critical* value of α. The set $\{\alpha_1, \ldots, \alpha_k\}$ is termed the *critical set* with respect to f, c, and is denoted by $\mathsf{C}_{f,c}$; i.e.,

$$\mathsf{C}_{f,c} = \{\alpha \in (0,1] \mid V_{f,c}(\alpha) \neq V_{f,c}(\alpha - \epsilon) \quad \forall \epsilon > 0\}.$$

The principal's utility is $(1-\alpha) \cdot V_{f,c}(\alpha)$. By Corollary III.4, $V_{f,c}(\alpha)$ is constant for every α that lies between two consecutive critical α's. This implies that the principal can restrict the search space to α values in the critical set $\mathsf{C}_{f,c}$ (or zero); that is:

Observation III.5. *Let $\alpha^\star$ be the optimal contract with respect to f, c. Then, $\alpha^\star \in \mathsf{C}_{f,c} \cup \{0\}$.*

For example, suppose f is an additive success probability function. Then, for every contract α, the best response of the agent is to select all actions $a \in A$ such that $\alpha \cdot f(a) \geqslant c(a)$. Consequently, the critical set $\mathsf{C}_{f,c}$ is $\{\frac{c(a)}{f(a)} \mid a \in A\}$. Thus, the optimal contract $\alpha^\star$ is the best point among these critical α values. As we shall see, things become more complex for richer classes of success probability functions.

IV. Gross Substitutes Functions

In this section, we devise a polynomial time algorithm for the case that the success probability function f is gross-substitutes, which includes additive, unit demand, and matroid rank functions as special cases.

Theorem IV.1. *Given a gross substitutes success probability function f, and a cost function c, one can compute the optimal contract α in polynomial time.*

To prove Theorem IV.1, we present Algorithm 1, which computes the optimal contract for any success probability function f, and show that it can be implemented in polynomial time for the case where f is gross substitutes.

Algorithm 1 uses the *successor* function, which, for any value in $[0,1]$, returns the next critical α. Formally, the successor is a function $\mathsf{succ}_{f,c} : [0,1] \to [0,1] \cup \{\text{NULL}\}$ which, for any value of $\alpha \in [0,1]$, returns the smallest $\alpha' > \alpha$ such that $\alpha' \in \mathsf{C}_{f,c}$ or NULL if such an α' does not exist.

Algorithm 1 is a generic algorithm for finding the optimal contract. The algorithm goes over all critical α's and returns the best one among them (which by Observation III.5 is the optimal contract). It assumes the existence of $\mathsf{succ}_{f,c}$ and $V_{f,c}$ oracles. Its complexity is bounded by the size of the critical set $\mathsf{C}_{f,c}$ and the complexity required for computing succ_{f_c} and $V_{f,c}$.

ALGORITHM 1: Optimal contract

Result: Optimal contract α
Input: Success probability function f, and costs c;
Initialization: $\alpha^\star = \alpha^{(0)} = 0$;
$t = 1$
$\alpha^{(1)} = \mathsf{succ}_{f,c}(\alpha^{(0)})$
while $\alpha^{(t)} \neq$ NULL **do**
    if $(1-\alpha^{(t)}) \cdot V_{f,c}(\alpha^{(t)}) > (1-\alpha^\star) \cdot V_{f,c}(\alpha^\star)$ **then**
        $\alpha^\star = \alpha^{(t)}$;
    end
    $t = t + 1$
    $\alpha^{(t)} = \mathsf{succ}_{f,c}(\alpha^{(t-1)})$
end
Return $\alpha^\star$

Theorem IV.2. *Algorithm 1 returns the optimal contract. Its time complexity is bounded by $|\mathsf{C}_{f,c}|$ multiplied by the complexity of computing $\mathsf{succ}_{f,c}$ and $V_{f,c}$.*

In the remainder of this section we show that for every gross substitutes success probability function f, Algorithm 1 runs in polynomial time. In Section IV-A we show that $\mathsf{succ}_{f,c}$ and $V_{f,c}$ can be implemented in polynomial time, and in Section IV-B we show that $|\mathsf{C}_{f,c}| \leqslant \frac{n(n+1)}{2}$.

A. Implementation of $\mathsf{succ}_{f,c}$ and $V_{f,c}$

It is well known that a demand query for gross substitutes functions can be computed in polynomial time in n by the following greedy algorithm: As long as there is an action with non-negative marginal utility (added value minus cost of the action), pick an action with maximal marginal utility (see, e.g., [27]). Generally, ties can be broken arbitrarily in this greedy procedure. For our purposes, it will be helpful to consider a particular tie-breaking rule, as follows.

Definition IV.3. *The ordered demanded set with respect to a contract α, denoted $S_\alpha \in \mathcal{D}_{f,c}(\alpha)$, is the ordered set obtained by the greedy algorithm with the following tie-breaking rule:*

- *Among multiple actions with the same (highest) marginal utility, pick an action with maximal cost. Among these, pick the action with the smallest index.*
- *If the highest marginal utility is 0, pick such an action.*

We refer to the greedy algorithm with the tie-breaking rule in Definition IV.3 as GREEDY. Let $S_\alpha[t]$ denote the action selected by GREEDY in the t-th iteration, and let S_α^t denote the set of actions selected within the first t iterations, that is, $S_\alpha^t = \{S_\alpha[1], \ldots, S_\alpha[t]\}$.

We first claim that the set S_α maximizes the principal's utility among all sets in the agent's demand.

Proposition IV.4. *For every α, it holds that $S_\alpha \in \mathcal{D}^\star_{f,c}(\alpha)$.*

Proof: Consider a contract $\alpha' = \alpha + \epsilon$ for a sufficiently small ϵ. By the structure of GREEDY, it holds that $S_{\alpha'} = S_\alpha$, thus S_α maximizes the agent's utility with respect to α'. By

Proposition III.1, $f(S_\alpha) \geqslant f(S)$ for every $S \in \mathcal{D}_{f,c}(\alpha)$, implying that $S_\alpha \in \mathcal{D}^\star_{f,c}(\alpha)$. ∎

We next show that $\mathsf{succ}_{f,c}(\alpha)$ can be computed in polynomial time.

Lemma IV.5. *Given a gross substitutes function f, a cost function c, and $\alpha \geqslant 0$, one can compute $\mathsf{succ}_{f,c}(\alpha)$ in polynomial time.*

Proof: Consider the case where $\mathsf{succ}_{f,c}(\alpha) \neq$ NULL. Let $\alpha' = \mathsf{succ}_{f,c}(\alpha)$, let S_α and $S_{\alpha'}$ be the respective sets returned by GREEDY, and let $d = |S_\alpha|$. Observe that there has to be an $i \in [d]$ such that $S_\alpha[i] \neq S_{\alpha'}[i]$ or that $|S_{\alpha'}| > d$. In the former case, consider the smallest such i. We have $S^{i-1}_\alpha = S^{i-1}_{\alpha'}$ and $(\alpha' - \epsilon) f(S_\alpha[i] \mid S^{i-1}_\alpha) - c(S_\alpha[i]) \geqslant (\alpha' - \epsilon) f(S_{\alpha'}[i] \mid S^{i-1}_\alpha) - c(S_{\alpha'}[i])$ for all sufficiently small $\epsilon > 0$ as well as $\alpha' f(S_{\alpha'}[i] \mid S^{i-1}_{\alpha'}) - c(S_{\alpha'}[i]) \geqslant \alpha' f(S_\alpha[i] \mid S^{i-1}_{\alpha'}) - c(S_\alpha[i])$. This implies that $\alpha' = \frac{c(S_{\alpha'}[i]) - c(S_\alpha[i])}{f(S_{\alpha'}[i] \mid S^{i-1}_\alpha) - f(S_\alpha[i] \mid S^{i-1}_\alpha)}$. Analogously, if $|S_{\alpha'}| > d$, then $\alpha' = \frac{c(S_{\alpha'}[d+1])}{f(S_{\alpha'}[d+1] \mid S_\alpha)}$.

So, in order to compute $\mathsf{succ}_{f,c}(\alpha)$, it suffices to consider all (finite) ratios $\frac{c(a) - c(S_\alpha[i])}{f(a \mid S^{i-1}_\alpha) - f(S_\alpha[i] \mid S^{i-1}_\alpha)}$ and $\frac{c(a)}{f(a \mid S_\alpha)}$ for all $a \in A$ and $i \in [d]$. The smallest one that is bigger than α and has a larger value of $V_{f,c}$ is $\mathsf{succ}_{f,c}(\alpha)$. If there is none, then $\mathsf{succ}_{f,c}(\alpha) =$ NULL. Since there are at most n^2 such ratios, this can be implemented in polynomial time. ∎

B. Bounding the Size of the Critical Set

In this section we establish the following upper bound on the size of the critical set, for every gross substitutes function and every cost function.

Theorem IV.6. *Let f be a gross substitutes function, and let c be a cost function. It holds that $|\mathsf{C}_{f,c}| \leqslant \frac{n(n+1)}{2}$.*

In Section IV-B1, we introduce the notion of *generic* cost functions and prove the theorem for this subclass. In Section IV-B2 we show that the proof extends to arbitrary cost functions.

1) Generic Cost Functions: We wish to define the notion of a *generic* cost function with respect to a success probability function so that for every α, GREEDY has at most one round where tie breaking occurs. In order to formally define this notion, we first define sets of candidate critical values.

Definition IV.7. *Given a function f, a cost function c, and an (unordered) pair of actions $a_1, a_2 \in A \cup \{\text{NULL}\}$ such that $a_1 \neq a_2$, let*

$$\Gamma_{f,c}(a_1, a_2) = \{\alpha \mid \exists_{S_1, S_2 \subseteq A}\ \alpha f(a_1 \mid S_1) - c(a_1) = \alpha f(a_2 \mid S_2) - c(a_2) \geqslant 0\},$$

where $c(\text{NULL}) = 0$ and $f(\text{NULL} \mid S) = 0$ for every set S.

That is, $\Gamma_{f,c}(a_1, a_2)$ is the set of values of α such that the marginal utility of a_1 with respect to some set S_1 equals the marginal utility of a_2 with respect to some set S_2. These are candidate values of α for which GREEDY may be indifferent between adding a_1 and adding a_2, where a_1 (or a_2) may be NULL.

We next observe that only α values that belong to some $\Gamma_{f,c}(a_1, a_2)$ may be critical.

Observation IV.8. *If for every $a_1, a_2 \in A \cup \{\text{NULL}\}$ such that $a_1 \neq a_2$ it holds that $\alpha \notin \Gamma_{f,c}(a_1, a_2)$, then $\alpha \notin \mathsf{C}_{f,c}$.*

Proof: Since $\alpha \notin \Gamma_{f,c}(a_1, a_2)$ for any $a_1, a_2 \in A \cup \{\text{NULL}\}$, $a_1 \neq a_2$, the execution of GREEDY does not involve any tie breaking because always $\alpha f(S_\alpha[t] \mid S^{t-1}) - c(S_\alpha[t]) > \alpha f(a \mid S^{t-1}) - c(a)$ for all $a \neq S_\alpha[t]$. As all of these finitely many inequalities are strict, there has to be an $\epsilon > 0$ such that still $(\alpha - \epsilon) f(S_\alpha[t] \mid S^{t-1}) - c(S_\alpha[t]) > (\alpha - \epsilon) f(a \mid S^{t-1}) - c(a)$ for all t and a. That is, on $\alpha - \epsilon$, the execution of GREEDY is identical to the one on α. So, α cannot be critical. ∎

We are now ready to define generic cost functions.

Definition IV.9. *A cost function c is said to be generic w.r.t. a success probability function f if for every $\alpha > 0$, there exists at most one (unordered) pair of actions $a_1, a_2 \in A \cup \{\text{NULL}\}$, $a_1 \neq a_2$, such that $\alpha \in \Gamma_{f,c}(a_1, a_2)$.*

By definition, for every generic cost function c, and every $\alpha > 0$, there could be at most one iteration in GREEDY (i.e., when generating S_α) in which tie breaking occurs.

The following lemma establishes an upper bound on the size of the critical set for every generic cost function.

Proposition IV.10. *Let f be a gross substitutes function, and let c be a generic cost function w.r.t. f. It holds that $|\mathsf{C}_{f,c}| \leqslant \frac{n(n+1)}{2}$.*

We will prove the claim using a potential function $\Phi \colon 2^A \to \mathbb{Z}_{\geqslant 0}$, which associates every set of actions S with a non-negative integer. In order to define Φ, for every action $a \in A$, we let r_a be the rank of a according to c (i.e., the rank of the highest cost action is n, the rank of the 2nd highest cost action is $n - 1$, and so on).

Since the cost function is assumed to be generic, the rank is unique. To see this, observe that for every generic cost function it holds that for any two actions $a_1 \neq a_2$ it must be that $c(a_1) \neq c(a_2)$, or else $\Gamma_{f,c}(a_1, a_2)$ would be the set of all real numbers (by setting $S_1 = S_2 = A$).

Now, we define $\Phi(S) = \sum_{a \in S} r_a$. The core insight is the following lemma, showing that the potential increases in α.

Lemma IV.11. *For every $\alpha', \alpha \in \mathsf{C}_{f,c}$, $\alpha' < \alpha$, we have $\Phi(S_{\alpha'}) \leqslant \Phi(S_\alpha) - 1$.*

Before proving Lemma IV.11, we show how it implies Proposition IV.10. Letting $\mathsf{C}_{f,c} = \{\alpha_1, \ldots, \alpha_k\}$ with $\alpha_1 < \ldots < \alpha_k$, we have $\Phi(S_{\alpha_1}) \geqslant 1$ because $S_{\alpha_1} \neq \varnothing$. We get that $\Phi(S_{\alpha_{j+1}}) \geqslant \Phi(S_{\alpha_j}) + 1$ for all j, implying that $\Phi(S_{\alpha_k}) \geqslant k$. However, we also have, $\Phi(S_{\alpha_k}) \leqslant \sum_{a \in A} r_a = \sum_{i=1}^n i = \frac{n(n+1)}{2}$. This implies that $k \leqslant \frac{n(n+1)}{2}$.

We now prove Lemma IV.11.

Proof of Lemma IV.11: In order to prove Lemma IV.11, it suffices to prove that $\Phi(S_{\alpha'}) < \Phi(S_\alpha)$ for every *neighboring* $\alpha, \alpha' \in \mathsf{C}_{f,c}$. That is, α, α' such that $\alpha' < \alpha$ and $(\alpha', \alpha) \cap \mathsf{C}_{f,c} = \emptyset$.

Specifically, it suffices to prove that for any neighboring α' and α, the set $S_{\alpha'}$ takes one of the following two forms: either (i) $S_{\alpha'} = S_\alpha \backslash \{a\}$ for some $a \in S_\alpha$, or (ii) $S_{\alpha'} = (S_\alpha \backslash \{a_1\}) \cup \{a_2\}$ for some $a_1 \in S_\alpha$, and $a_2 \notin S_\alpha$, where $c(a_2) < c(a_1)$. Indeed, in each one of these cases, the potential of $S_{\alpha'}$ is smaller than the potential of S_α by at least 1.

We note that since α, α' are neighboring, it holds that $S_{\alpha'} = S_{\alpha-\epsilon}$ for every $\epsilon \in (0, \alpha - \alpha')$. Also note that, for small enough ϵ, if GREEDY has a unique action that maximizes the marginal utility with respect to α (also with respect to not choosing any action), then it also maximizes the marginal utility with respect to $\alpha - \epsilon$ (as long as the set of actions chosen by GREEDY so far has not changed.).

Since by Observation IV.8, every critical α must be a candidate critical α for some set in $\Gamma_{f,c}(a_1, a_2)$, $a_1, a_2 \in A \cup \{\text{NULL}\}$, $a_1 \neq a_2$, we can distinguish between the following two cases:

Case 1: $\alpha \in \Gamma_{f,c}(a, \text{NULL})$ for some action $a \in A$ (and thus not in any of $\Gamma_{f,c}(a_1, a_2)$ for $a_1, a_2 \in A$ nor in $\Gamma_{f,c}(a', \text{NULL})$ for $a' \neq a$ by genericity of c).

We first observe that $a \in S_\alpha$. To see this, assume that $a \notin S_\alpha$. Then by genericity, $\alpha f(S_\alpha[t] \mid S_\alpha^{t-1}) - c(S_\alpha[t]) > \alpha f(a' \mid S_\alpha^{t-1}) - c(a')$ for all t and all $a' \neq S_\alpha[t]$. As these are finitely many strict inequalities, there is an $\epsilon > 0$ such that we also have $(\alpha - \epsilon) f(S_\alpha[t] \mid S_\alpha^{t-1}) - c(S_\alpha[t]) > (\alpha - \epsilon) f(a' \mid S_\alpha^{t-1}) - c(a')$ for all t and all $a' \neq S_\alpha[t]$. That is, the execution of GREEDY on $\alpha - \epsilon$ is the same as on α: This is a contradiction to α being critical.

Since $a \in S_\alpha$, let ℓ be the step in GREEDY such that $S_\alpha[\ell] = a$.

One can further assume that the marginal utility $\alpha \cdot f(a \mid S_\alpha^{\ell-1}) - c(a) = 0$. Indeed, it must be non-negative, and if it is strictly positive then in every step GREEDY has only one action that maximizes the marginal utility. This implies that for small enough ϵ GREEDY on $\alpha - \epsilon$ will choose exactly the same set S_α, which contradicts α being critical.

Furthermore, $\alpha \cdot f(a' \mid S_\alpha^{\ell-1}) - c(a') < 0$ for all $a' \neq a$ because otherwise $\alpha \in \Gamma_{f,c}(a, a')$, contradicting genericity. Similarly, for α, GREEDY has a positive marginal utility for every step $t < \ell$, and in every such step, the action that maximizes the marginal utility is unique. Thus, for small enough ϵ, GREEDY on $\alpha - \epsilon$ selects the same action $S_\alpha[t]$ for all $t < \ell$, and selects no action at step ℓ since the marginal utility of all remaining actions at $\alpha - \epsilon$ is negative. It follows that, for α', GREEDY would have chosen the exact same actions except for a, and $S_{\alpha'} = S_\alpha \backslash \{a\}$.

Case 2: $\alpha \in \Gamma_{f,c}(a_1, a_2)$ for some actions $a_1, a_2 \in A$. Let $c(a_1) > c(a_2)$.

We first show that at least one of a_1 and a_2 is contained in S_α. With the goal of a contradiction, let us assume that $a_1, a_2 \notin S_\alpha$. Let $d = |S_\alpha|$. By genericity, for every $t \leqslant d$, it holds that the selected action $S_\alpha[t]$ in iteration t is the unique optimal action (also with respect to not choosing any action). Therefore, for small enough ϵ, we have $S_{\alpha-\epsilon} = S_\alpha$, which contradicts α being critical.

Next we show that indeed $a_1 \in S_\alpha$. To this end, let ℓ be the first iteration where one of a_1 and a_2 is added to S_α. That is, $S_\alpha[\ell] \in \{a_1, a_2\}$ and $a_1, a_2 \notin S_\alpha^{\ell-1}$. We observe that $\alpha \cdot f(a_1 \mid S_\alpha^{\ell-1}) - c(a_1) = \alpha \cdot f(a_2 \mid S_\alpha^{\ell-1}) - c(a_2)$ because otherwise due to genericity there is no tie-breaking occurring in GREEDY, implying that for small enough ϵ, we have $S_\alpha = S_{\alpha-\epsilon}$, which contradicts α being critical. Since GREEDY breaks ties in favor of actions with higher cost, in iteration ℓ, GREEDY chooses to add a_1, meaning that $S_\alpha[\ell] = a_1$, thus $a_1 \in S_\alpha$.

Note also that, by genericity, the marginal utility $\alpha \cdot f(a_1 \mid S_\alpha^{\ell-1}) - c(a_1) = \alpha \cdot f(a_2 \mid S_\alpha^{\ell-1}) - c(a_2)$ must be strictly positive.

We next consider running GREEDY on instances in which we perturb the costs of a_1 and a_2, and relate the obtained outcome to S_α and $S_{\alpha-\epsilon}$. To this end, let $Z = \{\alpha \cdot (f(a \mid S_1) - f(a' \mid S_2)) - c(a) + c(a') \mid a, a' \in A, S_1, S_2 \subseteq A\} \cup \{\alpha \cdot f(a \mid S) - c(a) \mid a \in A, S \subseteq A\}$. Furthermore, let

$$\delta = \min\left\{\frac{z}{2} \mid z \in Z, z > 0\right\}.$$

Define c' by $c'(a_1) = c(a_1) + \delta$, and $c'(a) = c(a)$ for $a \neq a_1$. Let X be the outcome of GREEDY on (f, c', α). Because f is gross substitutes, there is a set in the demand $\mathcal{D}_{f,c'}(\alpha)$ at (f, c', α) that contains $S_\alpha \backslash \{a_1\}$. Since all steps of GREEDY on (f, c', α) are unique up to step ℓ and in step ℓ it adds a_2, a_2 must be in every demand set. Combining these two observations implies that $(S_\alpha \backslash \{a_1\}) \cup \{a_2\}$ is contained in a demand set. By genericity and the choice of δ, all steps of GREEDY on (f, c', α) after step ℓ must also be unique, showing that $X \supseteq (S_\alpha \backslash \{a_1\}) \cup \{a_2\}$.

Note that since all steps of GREEDY on (f, c', α) are unique, GREEDY on $(f, c, \alpha - \epsilon)$ gives the same outcome for a sufficiently small ϵ. Thus, $S_{\alpha-\epsilon} = X$.

Now define c'' by $c''(a_2) = c'(a_2) + \delta$, and $c''(a) = c'(a)$ for $a \neq a_2$. Let Y be the outcome of GREEDY on (f, c'', α). We claim that GREEDY on (f, c'', α) makes the exact same choices as GREEDY on (f, c, α). Up to step $\ell - 1$, this holds by genericity. At step ℓ, both a_1 and a_2 used to be the strict best choices (with a strictly positive marginal) and a_1 was chosen by tie-breaking (due to the higher cost). By our choice of δ and since we raised the cost of both a_1 and a_2 by the same amount, this is still the case. After step ℓ, all choices that were unique are still unique. Note that this includes decisions involving a_2 because of our choice of δ, and because the only element it could tie with is a_1 (by genericity), and a_1 was already chosen.

We conclude that $Y = S_\alpha$.

By gross substitutes, Y must contain $X \backslash \{a_2\}$.

Combining everything we have shown so far it must hold that S_α contains action a_1, S_α may or may not contain action a_2, and S_α may contain some additional actions $x_1, \ldots, x_k$ different from a_1 and a_2; while $S_{\alpha-\epsilon}$ must contain action a_2, $S_{\alpha-\epsilon}$ may or may not contain action a_1, $S_{\alpha-\epsilon}$ must contain all other actions $x_1, \ldots, x_k$ contained in S_α (if any), and it may not contain any other action.

First suppose that $a_2 \in S_\alpha$. Note that then we cannot have $a_1 \in S_{\alpha-\epsilon}$ because this would mean that $S_{\alpha-\epsilon} = S_\alpha$ in contradiction to α being critical. So we must have $S_{\alpha'} = S_{\alpha-\epsilon} = S_\alpha \backslash \{a_1\}$.

Now consider the case where $a_2 \notin S_\alpha$. In this case we can't have $a_1 \in S_{\alpha-\epsilon}$ because this would mean that $S_{\alpha-\epsilon} \supset S_\alpha$ in contradiction to Proposition III.1. Hence $a_1 \notin S_{\alpha-\epsilon}$ and thus $S_{\alpha'} = S_{\alpha-\epsilon} = (S_\alpha \backslash \{a_1\}) \cup \{a_2\}$. ∎

2) Arbitrary Cost Functions: It remains to extend the bounded critical set result from generic cost functions to arbitrary ones. To do so, we define a perturbation over cost functions that leads to a generic cost function with probability 1 and where the size of the critical set can only increase.

Given a cost function c, a cost function $\hat{c}$ is said to be an *ϵ-perturbation* of c if for every action $a \in A$, $\hat{c}(a) \in [c(a), c(a) + \epsilon]$.

We first observe that a small enough perturbation in the cost function cannot insert new sets into the demand:

Observation IV.12. *For every success probability function f, cost function c and $\alpha > 0$, there exists $\epsilon > 0$ such that for every ϵ-perturbation cost function $\hat{c}$ of c, $\mathcal{D}_{f,\hat{c}}(\alpha) \subseteq \mathcal{D}_{f,c}(\alpha)$.*

Proof: Let $\delta = \left(\max_{S \in \mathcal{D}_{f,c}(\alpha)} \alpha \cdot f(S) - c(S)\right) - \left(\max_{S' \notin \mathcal{D}_{f,c}(\alpha)} \alpha \cdot f(S') - c(S')\right)$ be the smallest utility gap between a set inside $\mathcal{D}_{f,c}$ and outside $\mathcal{D}_{f,c}$. Note that $\delta > 0$ since there are only finitely many sets. We now claim that for $\epsilon < \frac{\delta}{|A|}$, we have $\mathcal{D}_{f,\hat{c}}(\alpha) \subseteq \mathcal{D}_{f,c}(\alpha)$. To this end, consider any $S' \notin \mathcal{D}_{f,c}(\alpha)$. We claim that $S' \notin \mathcal{D}_{f,\hat{c}}(\alpha)$. Let $S \in \mathcal{D}_{f,c}(\alpha)$. We have $\alpha \cdot f(S') - \hat{c}(S') \overset{(\star)}{\leqslant} \alpha \cdot f(S') - c(S') \overset{(\star\star)}{\leqslant} \alpha \cdot f(S) - c(S) - \delta \overset{(\star)}{\leqslant} \alpha \cdot f(S) - \hat{c}(S) + \epsilon \cdot |A| - \delta < \alpha \cdot f(S) - \hat{c}(S)$, where $(\star)$ follows from the definition of $\hat{c}$ and $(\star\star)$ from the fact that $S \in \mathcal{D}_{f,c}(\alpha)$ but $S' \notin \mathcal{D}_{f,c}(\alpha)$. ∎

With Observation IV.12 at hand, the following lemma shows that a perturbed cost function can only increase the size of the critical set.

Lemma IV.13. *For every success probability function f and a cost function c, there exists an $\epsilon > 0$ such that for every ϵ-perturbation $\hat{c}$ of c, $|\mathsf{C}_{f,c}| \leqslant |\mathsf{C}_{f,\hat{c}}|$.*

Proof: Let $\alpha_1 < \ldots < \alpha_k$ be the critical values in $\mathsf{C}_{f,c}$. Let $\beta_0 = \alpha_1/2$, $\beta_i = \frac{\alpha_i + \alpha_{i+1}}{2}$ for $1 \leqslant i < k$, and let $\beta_k = 2\alpha_k$. By Proposition III.1 and Corollary III.4 we get that for every i, and $S \in \mathcal{D}_{f,c}(\beta_i)$, $V(\alpha_i) \leqslant f(S) < V(\alpha_{i+1})$, and therefore $\{\mathcal{D}_{f,c}(\beta_i)\}_i$ are disjoint (i.e., for every $i \neq j$, $\mathcal{D}_{f,c}(\beta_i) \cap \mathcal{D}_{f,c}(\beta_j) = \varnothing$).

By Observation IV.12, there exist $\epsilon_0, \ldots, \epsilon_k$ such that for every i, and $\hat{c}_i$ which is an ϵ_i-perturbation of c, it holds that $\mathcal{D}_{f,\hat{c}_i}(\beta_i) \subseteq \mathcal{D}_{f,c}(\beta_i)$. Thus, for $\epsilon = \min_i \epsilon_i$ it holds that for every ϵ-perturbation $\hat{c}$ of c, $\mathcal{D}_{f,\hat{c}}(\beta_i) \subseteq \mathcal{D}_{f,c}(\beta_i)$, and thus $\{\mathcal{D}_{f,\hat{c}}(\beta_i)\}_i$ are disjoint. Therefore, every interval (β_{i-1}, β_i) must have a critical $\hat{\alpha}_i$ w.r.t. $\hat{c}$. This concludes the proof. ∎

We are now ready to prove Theorem IV.6, namely to establish the upper bound of $\frac{n(n+1)}{2}$ on the size of the critical set for an arbitrary cost function.

Proof of Theorem IV.6: By Lemma IV.13 there exists $\epsilon > 0$ such that for every ϵ-perturbation $\hat{c}$ of c, it holds that $|\mathsf{C}_{f,c}| \leqslant |\mathsf{C}_{f,\hat{c}}|$. Suppose one draws $\hat{c}(a)$ uniformly at random from $[c(a), c(a) + \epsilon]$ for every action a. We show that $\hat{c}$ would be generic with probability 1.

To see this, consider the event that $\Gamma_{f,c}(a_1, a_2) \cap \Gamma_{f,c}(a_3, a_4)$ contains some $\alpha > 0$ for two different (unordered) pairs $(a_1, a_2), (a_3, a_4)$. By the union bound, it holds that

$$\begin{aligned} &\Pr\Big[\Gamma_{f,c}(a_1,a_2) \cap \Gamma_{f,c}(a_3,a_4) \neq \varnothing\Big] \\ \leqslant &\sum_{\substack{S_1,S_2,\\ S_3,S_4}} \Pr\Big[\exists \alpha > 0 : \alpha \cdot f(a_1 \mid S_1) - \hat{c}(a_1) \qquad (1) \\ &\qquad = \alpha \cdot f(a_2 \mid S_2) - \hat{c}(a_2) \geqslant 0 \\ &\qquad \text{AND} \quad \alpha \cdot f(a_3 \mid S_3) - \hat{c}(a_3) \\ &\qquad = \alpha \cdot f(a_4 \mid S_4) - \hat{c}(a_4) \geqslant 0\Big]. \end{aligned}$$

If $f(a_1 \mid S_1) = f(a_2 \mid S_2)$, then

$$\Pr\Big[\Gamma_{f,c}(a_1,a_2) \cap \Gamma_{f,c}(a_3,a_4) \neq \varnothing\Big] \overset{(1)}{\leqslant} \Pr[\hat{c}(a_1) = \hat{c}(a_2)] = 0,$$

where the last equality follows since this is a measure 0 event. Else,

$$\begin{aligned} &\Pr\Big[\Gamma_{f,c}(a_1,a_2) \cap \Gamma_{f,c}(a_3,a_4) \neq \varnothing\Big] \\ \overset{(1)}{\leqslant} &\sum_{\substack{S_1,S_2,\\ S_3,S_4}} \Pr\Big[[f(a_1 \mid S_1) - f(a_2 \mid S_2)]\,[\hat{c}(a_3) - \hat{c}(a_4)] = \\ &\qquad [f(a_3 \mid S_3) - f(a_4 \mid S_4)]\,[\hat{c}(a_1) - \hat{c}(a_2)]\Big] \\ = \;& 0, \end{aligned}$$

where the last equality follows again since this is a measure 0 event. Applying the union bound once more on all such (finitely many) events shows that $\hat{c}$ is generic with probability 1. Thus, there exists a generic ϵ-perturbation $\hat{c}$ of c. We get that $|\mathsf{C}_{f,c}| \leqslant |\mathsf{C}_{f,\hat{c}}| \leqslant \frac{n(n+1)}{2}$, where the last inequality follows by Lemma IV.10. This concludes the proof. ∎

V. Beyond Gross Substitutes Functions

In this section we study success probability functions beyond gross substitutes. In Section V-A we show that submodular functions are more complex than gross substitutes: unlike gross substitutes, they may exhibit an exponential critical set, and the optimal contract is NP-hard to compute. In Section V-B we devise an FPTAS for arbitrary success probability functions. In Section V-C we present a weakly poly-time algorithm for instances with poly-size critical sets.

A. Submodular Functions

Our first result shows that our approach for gross substitutes functions of iterating over all critical points cannot yield a polytime algorithm for submodular functions or even for coverage functions.

Theorem V.1. *There exists a coverage success probability function f and a cost function c such that $|\mathsf{C}_{f,c}| = 2^n - 1$.*

Proof: For simplicity of presentation, we present a proof for f values that are not necessarily in $[0, 1]$; this can be easily scaled.

We prove the theorem by induction on the size of the action set. For $n = 1$, it is trivial. (E.g., $f(1) = 2$, and $c(1) = 1$, gives $|\mathsf{C}_{f,c}| = 1$.)

Assume there exist a coverage success probability function f and a cost function c over a set of actions A of size k such that $|\mathsf{C}_{f,c}| = 2^k - 1$, and let $\alpha_1 < \ldots < \alpha_{2^k-1}$ be the critical values of α in $\mathsf{C}_{f,c}$.

Let g be a success probability function over the set of actions $A \cup \{k+1\}$, given by

$$g(S) = \begin{cases} \beta_1 \cdot f(S) & \text{if } S \subseteq A \\ \beta_2 \cdot f(A) + f(S \backslash \{k+1\}) & \text{if } k+1 \in S \end{cases}$$

where $\beta_1 = \frac{10 \cdot \alpha_{2^k-1}}{\alpha_1}$ and $\beta_2 = 10 \cdot \beta_1$. The function g is a coverage function (see full version). Let $\hat{c}$ be the cost function over actions in $A \cup \{k+1\}$ defined as $\hat{c}(a) = c(a)$ for every $a \in A$, and $\hat{c}(k+1) = 20 \cdot \alpha_{2^k-1} \cdot f(A)$.

We show that there are $2^k - 1$ critical values of α in the range $\alpha \leqslant \frac{\alpha_{2^k-1}}{\beta_1}$, a single critical value of α in the range $(\frac{\alpha_{2^k-1}}{\beta_1}, \frac{\alpha_1}{2})$, and $2^k - 1$ additional critical values of α in the range $\alpha > \frac{\alpha_1}{2}$, amounting to $2^{k+1} - 1$ critical α's.

For every $\alpha \leqslant \frac{\alpha_{2^k-1}}{\beta_1}$ the marginal utility of action $k+1$ with respect to any set $S \subseteq A$ is at most

$$\begin{aligned} & \alpha\,(\beta_2 f(A) + f(S) - \beta_1 f(S)) - \hat{c}(k+1) \\ \leqslant\; & \alpha \cdot \beta_2 \cdot f(A) - \hat{c}(k+1) \\ \leqslant\; & 10 \cdot \alpha_{2^k-1} \cdot f(A) - 20 \cdot \alpha_{2^k-1} \cdot f(A) < 0, \end{aligned}$$

where the first inequality follows by $\beta_1 \geqslant 1$, and the second inequality follows by the range of α and by substituting $\beta_2/\beta_1 = 10$. It follows that for every $\alpha \leqslant \frac{\alpha_{2^k-1}}{\beta_1}$, action $k+1$ is never included in any demanded set, thus $g(S) = \beta_1 f(S)$. Therefore, for every $i = 1, \ldots, 2^k - 1$, $\frac{\alpha_i}{\beta_1}$ is a critical α .

We next show that in the range $(\frac{\alpha_{2^k-1}}{\beta_1}, \frac{\alpha_1}{2})$, there must exists an additional critical α. By Corollary III.4, every critical α leads to a demanded set of a strictly higher value. Thus, the demanded set at $\alpha = \frac{\alpha_{2^k-1}}{\beta_1}$ must be A. We now show that for $\alpha = \alpha_1/2$, action $k+1$ must be in every set in the demand. Indeed, the utility from action $k+1$ alone is

$$\begin{aligned} & \alpha \cdot \beta_2 \cdot f(A) - \hat{c}(k+1) \\ = \; & \frac{\alpha_1}{2} \cdot \frac{100 \cdot \alpha_{2^k-1}}{\alpha_1} \cdot f(A) - 20 \cdot \alpha_{2^k-1} \cdot f(A) \\ = \; & 30 \cdot \alpha_{2^k-1} \cdot f(A), \end{aligned}$$

while the utility from any set that does not contain action $k+1$ is at most

$$\alpha \cdot \beta_1 \cdot f(A) = \frac{\alpha_1}{2} \cdot \frac{10 \cdot \alpha_{2^k-1}}{\alpha_1} \cdot f(A) = 5 \cdot \alpha_{2^k-1} \cdot f(A).$$

The same argument shows that $k+1$ must be in the demand of every set for $\alpha > \frac{\alpha_1}{2}$. We conclude that there must exists a critical alpha in the range $(\frac{\alpha_{2^k-1}}{\beta_1}, \frac{\alpha_1}{2})$ and that action $k+1$ must be in the demand of this critical. Moreover, for $\alpha = \frac{\alpha_1}{2}$, the demand is exactly $\{k+1\}$. This is since for every action $a \neq k+1$ it holds that $\alpha f(a \mid \{k+1\}) - c(a) = \alpha f(a) - c(a) < 0$, since $\alpha < \alpha_1$ Thus, all other actions has a negative marginal utility since, and are not in the demand set.

For $\alpha > \frac{\alpha_1}{2}$, for every set S, the marginal utility of S with respect to the action $k+1$ is the same for f and g. Thus, every α_i in this range is also critical. ■

Our next theorem establishes NP-hardness for computing an optimal contract under submodular functions, using a reduction from subset sum.

Theorem V.2. *The optimal contract problem for submodular success probability (or even budget additive) functions is NP-hard.*

Proof: We prove the theorem by a reduction from SUBSET-SUM. Subset-sum receives as input a (multi-)set of positive integer values $X = \{x_1, \ldots, x_n\}$ and an integer value Z. The question is whether there exists a subset $S \subseteq X$ such that $\sum_{j \in S} x_j = Z$. W.l.o.g., assume that $x_i < Z$ for all i (all numbers greater than Z can be ignored), and that $\sum_{i \in X} x_i > Z$ (otherwise this is an easy instance).

Given an instance $(x_1, \ldots, x_n, Z)$ to subset-sum, construct an instance to the optimal contract problem for budget additive functions over n actions as follows[1]. For every action $i = 1, \ldots, n$, set $f(\{i\}) = x_i$, and set $B = Z$. I.e., for every set S, $f(S) = \min(Z, \sum_{i \in S} x_i)$. Let the cost function be $c(i) = \epsilon \cdot x_i$, where $\epsilon = \frac{1}{Z^2}$.

If there exists a set S such that $\sum_{i \in S} x_i = Z$, then for a contract of $\alpha \geqslant \epsilon$ the agent's best-response is the set S, and for $\alpha < \epsilon$ the agent's best response is the empty-set. Thus, the optimal contract is to set $\alpha = \epsilon$ where the principal utility is $(1 - \epsilon) \cdot Z$.

[1] Recall that f is budget additive if there exists a budget B such that for every $S \subseteq A$ we have $f(S) = \min\{B, \sum_{i \in S} f(\{i\})\}$.

Consider next the case where there does not exist a set S such that $\sum_{i \in S} x_i = Z$. Let $Z_1 = \arg\min\{z > Z \mid \exists S \subseteq [n]. \sum_{i \in S} x_i = z\}$, and let S_1 be the set that sums to Z_1. Similarly, let $Z_2 = \arg\max\{z < Z \mid \exists S \subseteq [n]. \sum_{i \in S} x_i = z\}$, and let S_2 be the set that sums to Z_2.

Every set S such that $\sum_{i \in S} x_i > Z$ gives an agent's utility of $\alpha Z - \epsilon \sum_{i \in S} x_i$. Thus, S_1 is optimal among all these sets. Similarly, every set S such that $\sum_{i \in S} x_i < Z$ gives an agent's utility of $(\alpha - \epsilon) \sum_{i \in S} x_i$. Thus, for $\alpha \geqslant \epsilon$, S_2 is optimal among all these sets. It follows that there are exactly two critical α's, namely $\alpha_1 = \epsilon$, where the agent selects S_2 and the principal's utility is $(1-\epsilon)Z_2$, and $\alpha_2 = \frac{Z_1 - Z_2}{Z - Z_2} \cdot \epsilon$, where the agent selects S_1 and the principal's utility is $(1 - \frac{Z_1 - Z_2}{Z - Z_2} \cdot \epsilon)Z$.

We claim that the latter contract is better for the principal. To see this, observe first that $Z_1 - Z_2 < Z$. Indeed, let i be an arbitrary action in $S_1 \backslash S_2$ (such an action must exist). It holds that $Z_2 + x_i > Z$ (else, contradicting the choice of Z_2). It further follows that $Z_2 + x_i \geqslant Z_1$ (else, contradicting the choice of Z_1). On the other hand, $x_i < Z$. We get $Z_1 \leqslant Z_2 + x_i < Z_2 + Z$, as claimed. Furthermore, $Z - Z_2 \geqslant 1$. It follows that $\frac{Z_1 - Z_2}{Z - Z_2} < Z$. We get $(1 - \frac{Z_1 - Z_2}{Z - Z_2}\epsilon)Z > (1 - Z\epsilon)Z > (1-\epsilon)Z_2$, where the last inequality follows by $Z_2 \leqslant Z - 1$ and $\epsilon = 1/Z^2$. Thus, the principal's utility in this case is $(1 - \frac{Z_1 - Z_2}{Z - Z_2} \cdot \epsilon)Z$, which is strictly smaller than $(1-\epsilon)Z$.

It follows that the optimal contract is $\alpha = \epsilon$ if and only if the subset-set instance is a YES instance. ∎

B. FPTAS for Arbitrary Functions

Next we devise an FPTAS for arbitrary success probability functions. We first show that when all values of the success probability function f and the cost function c are represented using k bits, all critical values of α are ratios of two k-bit integers.[2]

Lemma V.3. *If all values of f and c are multiples of 2^{-k}, then $\mathsf{C}_{f,c} \subseteq Q$, where $Q = \left\{ \frac{a}{b} \in \mathbb{Q} \mid a, b \in [2^k] \right\}$.*

We now establish an FPTAS for the optimal contract problem for cases where all f and c values are k-bit representable.

Theorem V.4. *Consider the case where all values of f and c are multiples of 2^{-k}. For every $\epsilon \in (0,1)$, a $(1-\epsilon)$-approximation to the optimal contract can be computed using $\left\lceil \frac{k}{-\log_2(1-\epsilon)} \right\rceil$ queries to a $V_{f,c}$ oracle.*[3]

C. Weakly Poly-Time Algorithm for Instances with Poly-Size Critical Sets

Finally we show that in cases where the critical set is of polynomial size, one can compute the optimal contract in weakly polynomial time. Recall that the number of $V_{f,c}$ queries required by Algorithm 1 is the size of $|\mathsf{C}_{f,c}|$ multiplied by the number of $V_{f,c}$ queries required to implement $\mathsf{succ}_{f,c}$.

We show that if all f and c values are multiples of 2^{-k}, $\mathsf{succ}_{f,c}$ can be implemented using $2k+1$ $V_{f,c}$ queries. Thus, Algorithm 1 can be computed using $(2k+1)|\mathsf{C}_{f,c}|$ queries.

Theorem V.5. *Consider the case where all values of f and c are multiples of 2^{-k}. Then, $\mathsf{succ}_{f,c}(\alpha)$ can be computed using $2k+1$ queries to a $V_{f,c}$ oracle.*

VI. Beyond Binary Rewards

In this section we study a generalization of the binary outcome model, where the outcome space is a vector of m rewards to the principal $r(1), \ldots, r(m)$, and every set of actions entails some probability distribution over rewards, with $f_j(S)$ for $j \in \{1, \ldots, m\}$ being the probability of reward $r(j)$ under actions S. We use $R(S) = \sum_j f_j(S) \cdot r(j)$ to denote the expected reward to the principal given action set S. We assume that R is monotone and normalized (i.e., $R(S) \geqslant R(S')$ for every $S' \subseteq S$, and $R(\varnothing) = 0$). A contract in this model is a function $t : \mathbb{R}_{\geqslant 0} \to \mathbb{R}_{\geqslant 0}$, specifying the payment from the principal to the agent for every observed reward. A contract is said to be *linear* if there exists some α such that $t(r) = \alpha \cdot r$ for every r.

As in the binary outcome case, we can make some structural assumptions on the expected principal's reward $R(S)$. For example, that $R(S)$ is submodular or gross substitutes.

We show that when restricting attention to linear contracts, all our (positive and negative) results from the binary outcome model continue to hold in the general model. Moreover, we show that linear contracts are *max-min optimal* among all possible contracts, meaning that if the principal knows the expected reward $R(S)$ for every set of actions $S \subseteq A$, but not the probability distribution over rewards obtained from S, then a linear contract maximizes the principal's utility in the worst case over all distributions compatible with the known expected rewards.

A. Robust Optimality of Linear Contracts

We start by showing that linear contracts are robustly optimal in the general model. To do so we introduce some notation. Let $\mathcal{D}$ denote the collection of all sets of probability distributions over a finite set of rewards (not fixed) that are compatible with the known expected rewards. I.e., $\mathcal{D}$ is the collection of distribution sets $\{D_S\}_{S \subseteq A}$ over rewards such that $\mathbb{E}_{X_S \sim D_S}[X_S] = R(S)$ for every $S \subseteq A$.

Let $\underline{u}_p(t)$ denote the worst-case (over all distributions in $\mathcal{D}$, fixing expected rewards $\{R(S)\}_{S \subseteq A}$) principal's utility under contract t. I.e., $\underline{u}_p(t) = \min_{D \in \mathcal{D}} \mathbb{E}_{X \sim D_{S_t}}[X - t(X)]$, where S_t is the agent's best response action set for contract t. The following theorem shows that there exists a linear contract that maximizes $\underline{u}_p(t)$ among all contracts t.

[2] A similar analysis holds for the case that f and c take values $\frac{a}{b}$ for $a, b \in \{1, \ldots, M\}$ for some M.

[3] One can verify that the $V_{f,c}$ oracle can be easily replaced by a demand oracle to f.

Theorem VI.1. *For every function R of expected rewards, cost function c, and contract t, there exists a linear contract α such that $\underline{u}_p(t) \leqslant \underline{u}_p(\alpha)$.*

B. Optimal Linear Contracts in the General Model

Finally, we show that our computational results for the binary case translate to linear contracts in the general case. For this, observe that when restricting attention to linear contracts, the agent's best response depends only on the expected rewards of action sets, not on their distributions. Therefore, finding the optimal contract among all contracts in the binary outcome model is equivalent to finding the optimal one among all linear contracts in the general model.

Corollary VI.2. *The following hold in the general model:*

- *Given a gross substitutes function R of expected rewards, and a cost function c, one can compute the optimal linear contract in polynomial time.*
- *For a budget additive function R of expected rewards, and a cost function c, it is NP-hard compute the optimal linear contract.*
- *Given an arbitrary function R of expected rewards, and a cost function c, one can compute a $(1 - \epsilon)$-approximation to the optimal linear contract in weakly polynomial time.*

REFERENCES

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

[2] B. Holmström, "Moral hazard and observability," *The Bell Journal of Economics*, vol. 10, no. 1, pp. 74–91, 1979.

[3] S. J. Grossman and O. D. Hart, "An analysis of the principal-agent problem," *Econometrica*, vol. 51, no. 1, pp. 7–45, 1983.

[4] B. Holmström and P. Milgrom, "Multitask principal-agent analyses: Incentive contracts, asset ownership, and job design," *Journal of Law, Economics, and Organization*, vol. 7, pp. 24–52, 1991.

[5] B. Holmström and P. Milgrom, "Aggregation and linearity in the provision of intertemporal incentives," *Econometrica*, vol. 55, no. 2, pp. 303–328, 1987.

[6] G. Carroll, "Robustness and linear contracts," *The American Economic Review*, vol. 105, no. 2, pp. 536–563, 2015.

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

[8] M. Babaioff, M. Feldman, N. Nisan, and E. Winter, "Combinatorial agency," *Journal of Economic Theory*, vol. 147, no. 3, pp. 999–1034, 2012.

[9] M. Feldman, J. Chuang, I. Stoica, and S. Shenker, "Hidden-action in network routing," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 6, pp. 1161–1172, 2007.

[10] M. Babaioff, M. Feldman, and N. Nisan, "Free-riding and free-labor in combinatorial agency," in *Proceedings of SAGT'09*, 2009, pp. 109–121.

[11] ——, "Mixed strategies in combinatorial agency," *Journal of Artificial Intelligence Research*, vol. 38, pp. 339–369, 2010.

[12] P. Dütting, T. Roughgarden, and I. Talgam-Cohen, "The complexity of contracts," *SIAM Journal on Computing*, vol. 50, no. 1, pp. 211–254, 2021.

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

[14] J. M. Kleinberg and R. Kleinberg, "Delegated search approximates efficient search," in *Proceedings of ACM EC'18*, 2018, pp. 287–302.

[15] J. M. Kleinberg and M. Raghavan, "How do classifiers induce agents to invest effort strategically?" in *Proceedings of ACM EC'19*, 2019, pp. 825–844.

[16] G. Guruganesh, J. Schneider, and J. Wang, "Contracts under moral hazard and adverse selection," in *Proceedings of ACM EC'21*, 2021, pp. 563–582.

[17] T. Alon, P. Dütting, and I. Talgam-Cohen, "Contracts with private cost per unit-of-effort," in *Proceedings of ACM EC'21*, 2021, pp. 52–69.

[18] A. S. Kelso Jr and V. P. Crawford, "Job matching, coalition formation, and gross substitutes," *Econometrica*, pp. 1483–1504, 1982.

[19] K. Murota and A. Shioura, "M-convex function on generalized polymatroid," *Mathematics of Operations Research*, vol. 24, no. 1, pp. 95–105, 1999.

[20] K. Murota, "Convexity and steinitz's exchange property," in *Proceedings of IPCO'96*, 1996, pp. 260–274.

[21] B. Lehmann, D. Lehmann, and N. Nisan, "Combinatorial auctions with decreasing marginal utilities," *Games and Economic Behavior*, vol. 55, no. 2, pp. 270–296, 2006.

[22] F. Gul and E. Stacchetti, "Walrasian equilibrium with gross substitutes," *Journal of Economic Theory*, vol. 87, no. 1, pp. 95–124, 1999.

[23] N. Nisan and I. Segal, "The communication requirements of efficient allocations and supporting prices," *Journal of Economic Theory*, vol. 129, no. 1, pp. 192–224, 2006.

[24] E. Balkanski and R. P. Leme, "On the construction of substitutes," in *Proceedings of ACM EC'18*, 2018, p. 643.

[25] K. Murota, "Discrete convex analysis: A tool for economics and game theory," *Journal of Mechanism and Institution Design*, vol. 1, no. 1, pp. 151–273, 2016.

[26] S. Dobzinski, U. Feige, and M. Feldman, "Are gross substitutes a substitute for submodular valuations?" in *Proceedings of ACM EC'21*, 2021, pp. 390–408.

[27] R. P. Leme, "Gross substitutability: An algorithmic survey," *Games and Economic Behavior*, vol. 106, pp. 294–316, 2017.