Multi-agent Contracts*

Paul Dütting
Google Research
Zurich, Switzerland
duetting@google.com

Tomer Ezra
Sapienza University of Rome
Rome, Italy
tomer.ezra@gmail.com

Michal Feldman
Tel Aviv University
Tel Aviv, Israel
Microsoft Research
Herzliya, Israel
mfeldman@tauex.tau.ac.il

Thomas Kesselheim
University of Bonn
Bonn, Germany
thomas.kesselheim@uni-bonn.de

ABSTRACT

We study a natural combinatorial single-principal multi-agent contract design problem, in which a principal motivates a team of agents to exert effort toward a given task. At the heart of our model is a *reward function*, which maps the agent efforts to an expected reward of the principal. We seek to design computationally efficient algorithms for finding optimal (or near-optimal) linear contracts for reward functions that belong to the complement-free hierarchy.

Our first main result gives constant-factor approximation algorithms for submodular and XOS reward functions, with value and demand oracles, respectively. It relies on an unconventional use of "prices" and (approximate) demand queries for selecting the set of agents that the principal should contract with, and exploits a novel scaling property of XOS functions and their marginals, which may be of independent interest.

Our second main result is an $\Omega(\sqrt{n})$ impossibility for settings with n agents and subadditive reward functions, even with demand oracle access. A striking feature of this impossibility is that it applies to subadditive functions that are constant-factor close to submodular. This presents a surprising departure from previous literature, e.g., on combinatorial auctions.

CCS CONCEPTS

• **Theory of computation** → **Algorithmic game theory**; **Algorithmic mechanism design**.

KEYWORDS

contract theory; moral hazard; XOS; submodularity

ACM Reference Format:
Paul Dütting, Tomer Ezra, Michal Feldman, and Thomas Kesselheim. 2023. Multi-agent Contracts. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC '23), June 20–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3564246.3585193

1 INTRODUCTION

Contract theory is a core problem in economic theory (c.f. the 2006 "Nobel Prize" to Oliver Hart and Bengt Holmström), which seeks, using incentive mechanisms, to achieve desirable outcomes in the presence of unobservable actions. It plays a major role in the design of markets for effort (or services), similar to the role that mechanism design and auction theory play in the design of markets for goods. Applications of contract design range from multi-million dollar markets, such as influencer marketing on social media platforms [e.g., 39], to contracts for social goods, such as government-run programs to encourage afforestation/prevent deforestation [e.g., 2, 37].

At its heart is the hidden-action principal-agent problem [e.g., 28, 32], in which a principal seeks to incentivize an agent to take a costly action, whose stochastic outcome determines a reward for the principal. A contract defines monetary transfers from the principal to the agent based on the observable outcome. The principal's goal is to find the contract that maximizes her utility (expected reward minus transfer), when the agent chooses the action that maximizes his utility (expected transfer minus cost).

In its vanilla version, with a single principal and a single agent, the problem can be solved in polynomial time by solving one LP for each action [28]. While this approach can also be applied in more complex scenarios, its running time will usually be exponential in the (succinct) representation. Therefore, one has to understand the structure of optimal contracts in order to obtain computationally efficient algorithms for finding optimal contracts [e.g. 4, 17, 21].

A natural extension of the classic single-principal single-agent model, are settings where a principal seeks to incentivize a *team* of agents [33]. In such scenarios, the complexity arises from the complex combinatorial structure of dependencies between the agent actions, and is already compelling when each agent can either exert effort or not [4]. In this work, we provide computationally

*Part of the work of Thomas Kesselheim was done while the author was visiting the Simons Institute for the Theory of Computing. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 866132), by the Israel Science Foundation (grant number 317/17), by an Amazon Research Award, and by the NSF-BSF (grant number 2020788). Tomer Ezra was partially supported by the ERC Advanced Grant 788893 AMDROMA "Algorithmic and Mechanism Design Research in Online Markets" and MIUR PRIN project ALGADIMAR "Algorithms, Games, and Digital Markets".

STOC '23, June 20–23, 2023, Orlando, FL, USA

ACM ISBN 978-1-4503-9913-5/23/06.
https://doi.org/10.1145/3564246.3585193

efficient algorithms and impossibilities for (approximating) the optimal contract in this multi-agent setting.

As in many economic models, the problem of optimally incentivizing teams of agents boils down to a clean and compelling combinatorial optimization problem. The challenge is that, even in cases where the dependencies between the agent actions exhibit nice structure, this structure does *not* transfer to the objective function.

Model – Key ingredients. To state and discuss our results, it will be useful to describe the key ingredients of the multi-agent hidden-action principal-agent problem [c.f. 4] that we study in this paper.

In this model, a principal interacts with a set A of n agents. Agents have binary actions: They can either exert effort or not. Exerting effort comes with cost $c_i \in \mathbb{R}_{\geq 0}$ for agent i. Each subset of agents $S \subseteq A$ induces a probability distribution over possible outcomes $\omega \in \Omega$, and the principal associates a reward $r(\omega) \in \mathbb{R}_{\geq 0}$ with each possible outcome. The reward function $f : 2^A \to \mathbb{R}_{\geq 0}$ associates with each set $S \subseteq A$ of agents that exert effort an expected reward.

Our focus in this work is on a particularly important class of contracts, so-called linear contracts, and the optimal contract design problem in that class.

A linear contract consists of a vector $\alpha \in [0,1]^n$; and specify that agent i should be paid $\alpha_i \cdot r(\omega)$ for each outcome $\omega \in \Omega$. We say that a linear contract incentivizes a set of agents S if for each $i \in S$ it holds that $\alpha_i \cdot f(S) - c_i \geq \alpha_i \cdot f(S \setminus \{i\})$ and for each $i \notin S$ it holds that $\alpha_i \cdot f(S) \geq \alpha_i \cdot f(S \cup \{i\}) - c_i$. The goal is to find a linear contract α and a set S such that S is incentivized by α, and maximizes the principal's expected utility given by $g(S) = (1 - \sum_{i \in A} \alpha_i) f(S)$.

Linear contracts are arguably the most important class of contracts in practice, and they also exhibit good properties analytically. In particular, for the important special case of a binary outcome (e.g., a project that can succeed or fail), linear contracts are in fact optimal; and for more general outcome spaces, they are known to be max-min optimal when only the expected reward of each set of agents is known [17, 20].

1.1 Our Contribution

We study the computational complexity of computing optimal and near-optimal linear contracts for the multi-agent hidden-action principal-agent problem for different classes of "complement-free" reward functions [36].

As it turns out, moving from a single-agent to a multi-agent setting significantly complicates matters. Even for the simplest class of reward functions — additive reward — it is NP-hard to compute the optimal contract; but the problem admits an FPTAS (see Appendix A). As we progress in the hierarchy, the problem becomes significantly more challenging.

Our first main result concerns submodular, and in fact, the more general class of XOS reward functions (for definitions see Section 2). We show that for both of these classes there exist algorithms that provide constant-factor approximation to the optimal contract.

Our result relies on an uncoventional use of "prices" and (approximate) demand queries for selecting the set of agents that the principal should seek to incentivize, and exploits a novel scaling property for XOS functions and their marginals, which may be of independent interest.

Main Result 1 (Theorem 3.1): For any multi-agent setting with *submodular* reward function f, there exists a polynomial-time $O(1)$-approximation to the optimal contract, that uses a *value* oracle. If f is *XOS*, an $O(1)$-approximation can be achieved using *demand* and *value* oracles.

We complement this result by showing that, for the broader class of *subadditive* functions, the following impossibility applies:

Main Result 2 (Theorem 4.1): For any polynomial-time algorithm, with demand or value oracles, there exists a multi-agent setting with *subdditive* reward function f, such that the algorithm achieves no better than $\Omega\left(\sqrt{n}\right)$-approximation to the optimal contract, where n is the number of agents.

This impossibility result is particularly interesting in light of the fact that it applies to a subadditive function that is "constant-factor close" to submodular: It can be approximated by a submodular function to within a factor of 2. Yet, the gap between the optimal contract that can be found in polynomial time and the optimal contract jumps from constant to $\Omega(\sqrt{n})$.

So, on the one hand we show that—just like in pure combinatorial optimization and algorithmic mechanism design—submodularity, or more generally being XOS, enables efficient computation of near-optimal contracts. On the other hand we show that—in stark contrast to these closely related domains, where usually being within a factor of β of submodular or XOS implies that the bounds only deteriorate by a factor of $O(\beta)$ [e.g., 10, 26]—in our case the bound does not degrade gracefully with the closeness to submodularity.

Our positive result is "tight" in an additional way: In Theorem 4.9 we show that it is impossible to obtain a better than constant approximation for XOS functions (even with demand queries).

1.2 Our Techniques

In Section 2, we observe that the underlying optimization problem we need to solve is as follows. Given a cost $c_i \geq 0$ for each agent i and a reward function $f : 2^A \to \mathbb{R}_{\geq 0}$, our goal is to maximize the function $g : 2^A \to \mathbb{R}$ defined by

$$g(S) = \left(1 - \sum_{i \in S} \frac{c_i}{f(i \mid S \setminus \{i\})}\right) f(S).$$

The difficulty is that, even in cases where f is highly structured such as submodular, XOS, or subadditive (see definitions in Section 2), this structure does not carry over to g. For example, even in cases where f is non-negative monotone and submodular, the induced g will usually not be monotone and take negative values. If f is only XOS, g may even not be subadditive.

Constant-factor approximations for submodular and XOS. Let $S^\star$ be a set that maximizes $g(S^\star)$. Our goal is to find a set S such that $g(S) \geq \text{constant} \cdot g(S^\star)$. For the purpose of conveying the intuition behind our approach, assume in the following that $f(S^\star)$ is known to the algorithm (but not $S^\star$ itself) and the contribution of a single agent is negligible. In the technical sections, these assumptions will not be necessary.

A key ingredient in our proof is a pair of lemmas that draw connections between the value of the optimal solution and its marginals to the costs. Specifically, in Lemma 3.3, we show that

$\sum_{i\in S^\star}\sqrt{c_i} \le \sqrt{f(S^\star)}$. In Lemma 3.4, we show that if for a set S we have $f(i \mid S\setminus\{i\}) \ge \sqrt{2c_i f(S)}$ for every $i \in S$, then $g(S) \ge \frac{1}{2}f(S)$.

These observations give rise to guiding the choice of our set S by defining a "price" for each agent. Namely, let $p_i = \frac{1}{2}\sqrt{c_i f(S^\star)}$ for each agent i and consider the *demand set* T, which is defined to maximize $f(T) - \sum_{i\in T} p_i$. We now have $f(T) \ge f(T) - \sum_{i\in T} p_i \ge f(S^\star) - \sum_{i\in S^\star} p_i \ge \frac{1}{2}f(S^\star)$ using the definition of a demand set and Lemma 3.3. By definition, the marginal contribution of every agent in the demand set must exceed its price, namely $f(i \mid T\setminus\{i\}) \ge p_i = \frac{1}{2}\sqrt{c_i f(S^\star)}$. This condition looks almost like the one that is necessary to invoke Lemma 3.4. However, note that we only have a lower bound on $f(T)$, no upper bound. Therefore possibly $f(T)$ is much larger than $f(S^\star)$.

To deal with this, we establish a novel scaling property of XOS functions, showing that one can scale down the value of any set T to essentially any level, by removing some of its elements, while keeping the marginals of the remaining elements sufficiently high with respect to their original marginals. Namely, for every set T and every $\Psi < f(T)$, one can compute a subset $U \subseteq T$ such that $\frac{1}{2}\Psi \le f(U) \le \Psi$ and $f(i \mid U\setminus\{i\}) \ge \frac{1}{2}f(i \mid T\setminus\{i\})$. While this property is not too surprising for submodular functions, for XOS functions this is far from obvious, given the apparent lack of control over marginals, and may be of independent interest. Setting $\Psi = \frac{1}{32}f(S^\star)$, we can invoke Lemma 3.4 because with this choice $f(i \mid U\setminus\{i\}) \ge \frac{1}{2}f(i \mid T\setminus\{i\}) \ge \frac{1}{2}p_i = \frac{1}{4}\sqrt{c_i f(S^\star)} \ge \sqrt{2c_i f(U)}$ and conclude that $g(U) \ge \frac{1}{2}f(U) \ge \frac{1}{128}f(S^\star) \ge \frac{1}{128}g(S^\star)$.

Inapproximability results for subadditive and XOS. To show the impossibility result for subadditive reward functions f we follow the common approach of "hiding a good set" (see, e.g., [7]). We first construct a subadditive function f on subsets $S \subseteq A$ of n agents. It has the property that $g(S) = O(1)$ for $|S| < \sqrt{n}$ and $g(S) \le 0$ if $|S| \ge \sqrt{n}$. That is, the optimal principal utility is constant and any attempt to incentivize more than $\sqrt{n}$ agents would result in a negative (or zero) utility.

We then slightly modify f by choosing a random set $T^\star$ of size $\frac{n}{2}+1$, and increasing its value. Monotonicity and subadditivity of f are preserved by this change. This change significantly increases the principal's utility from the set $T^\star$ to $g(T^\star) = \Omega(\sqrt{n})$. At the same time, it does not increase the utility from any other set, so that, not only is $T^\star$ the unique optimal set, but it is also the only one that approximates it well. The principal's problem thus boils down to finding $T^\star$.

It then remains to show that $T^\star$ cannot be found by a polynomial number of (demand or value) queries. We then use the fact that the modified function is almost identical to the original (symmetric) one to show that every (demand or value) query reveals information only on a small set of candidates for $T^\star$, and thus every algorithm that uses only a polynomial number of queries, cannot guess $T^\star$ with high enough probability, resulting in an $\Omega(\sqrt{n})$ approximation to the principal's optimal utility.

Our proof providing a constant lower bound for XOS follows the same approach, but the additional structure of XOS functions limits how big the gap between $g(T)$ and $g(T^\star)$ can be.

1.3 Related Work

Optimizing the effort of others. Our work is part of an emerging frontier in Algorithmic Game Theory on optimizing the effort of others (see, e.g., the STOC 2022 TheoryFest workshop with the same title). This includes work on algorithmic contract design [e.g., 17, 20, 21], strategic classification [e.g., 8, 35], optimal scoring rule design [e.g., 14, 38], and delegation [e.g., 9, 34].

Combinatorial optimization and auctions. A number of fundamental papers has explored combinatorial optimization problems with "complement-free" set functions. In a landmark paper, Feige [23] gives constant-factor approximation algorithms for the welfare maximization problem in combinatorial auctions with submodular, XOS, and subadditive bidders. An exciting line of work seeks to understand whether it is possible to match these bounds with truthful mechanisms, with the current state of the art being polyloglog approximations [3, 16]. Complement-free valuations also play a crucial role in combinatorial auctions with item bidding, where subadditive valuations enable constant-factor Price of Anarchy bounds [10, 15, 25, 41]; and the prophet inequalities/posted-price literature, where constant-factor approximations are known for XOS and the state of the art for subadditive is a loglog approximation [18, 19, 26]. There are also polynomial-time constant-factor approximation results for truthful revenue maximization with unit-demand bidders [13], additive bidders [42], and XOS bidders [11].

Computational approaches to contracts. A computational approach to contracts was pioneered in [4, 24] and [20].

Most relevant for us is work on *combinatorial contracts*, which can be divided into three categories, depending on whether it concerns exponentially large outcome spaces [21], taking sets of actions [17], or settings with multiple agents [4–6, 22]. Out of these, the most closely related papers to our work are [17] and [4, 22].

Dütting et al. [17] study a single-principal single-agent setting, where the agent can take any subset of n actions. The main result is a polynomial-time (in n) algorithm for gross-substitutes reward functions. They also show NP-hardness for general submodular reward functions (with value oracle access). While both [17] and the present paper deal with a reward function that maps any subset of n actions to some expected reward, the induced optimization problems are fundamentally different: In [17] a single agent chooses a subset of n actions. A contract is defined by a single parameter α, and the agent chooses a set of actions that is better than any other set of actions. In particular, not every set of actions can be incentivized. Here, in contrast, each one of the n agents makes a binary choice over actions (exert effort or not), thus an action profile corresponds to a subset of the n agents. A contract is now defined by a vector $(\alpha_1, \ldots, \alpha_n)$. Typically, every set of agents can be incentivized, so there are exponentially many feasible solutions. The challenge is to find a feasible solution of high value.

Babaioff et al. [4] and Emek and Feldman [22] study the same model studied here—n agents with binary actions. They assume that each agent succeeds in his individual task with a certain probability, depending on whether he exerts effort or not; then a Boolean function maps individual successes and failures to a success or failure of the project. Babaioff et al. [4] show that for Boolean functions represented by read-once networks the optimal contract problem

is #P-complete, and give a polynomial-time algorithm for AND networks. Emek and Feldman [22] show that for OR networks (a special case of submodular) the problem is NP-hard, but admits an FPTAS. Our work significantly expands the landscape of this model, by considering general submodular, XOS, and subadditive reward functions.

Finally, recent work has considered settings exhibiting both hidden-action and hidden-type, through a computational lens in [1, 12, 30].

1.4 Organization

We formally set up the model in Section 2. Our main positive result—the constant-factor approximation guarantees for submodular and XOS reward functions—appear in Section 3. We prove the impossibility results for subadditve and XOS reward functions in Section 4. Our results for additive reward functions appear in Appendix A.

2 PRELIMINARIES

The multi-agent hidden-action principal-agent setting. In our model a single principal interacts with a set of n agents $A = [n] = \{1, \dots, n\}$. We focus on the following combinatorial binary-action setup: Agents can either take action (exert effort) or not. Taking action comes with a cost $c_i \in \mathbb{R}_{\geq 0}$ for agent i.

There is a set of outcomes Ω. To each subset of agents $S \subseteq A$ that exerts effort we associate a probability distribution $q_S : \Omega \to [0, 1]$ over outcomes. The principal derives reward $r(\omega) \in \mathbb{R}_{\geq 0}$ from outcome $\omega \in \Omega$. The interpretation is that if the set of agents S exerts effort, then an outcome ω is drawn from q_S, and the reward $r(\omega)$ goes to the principal.

We define the *reward function* $f : 2^A \to \mathbb{R}_{\geq 0}$ as the mapping from a given set of agents that exert effort to expected reward. We generally assume that f is monotone increasing so that for any two sets of agents S, S' with $S \subseteq S' \subseteq A$ it holds that $f(S) \leq f(S')$. We also assume that f is normalized in the sense that $f(\emptyset) = 0$. We write $f(i \mid S) := f(S \cup \{i\}) - f(S)$ for the marginal contribution of $i \in A$ to $S \subseteq A$.

One of the defining feature of the model is that the principal cannot directly observe the actions chosen by the agents, only their outcome, which is determined stochastically based on the actions.

Moral hazard and contracts. A main challenge in our problem is what economists refer to as *moral hazard*: In and by itself the agents have no interest in exerting effort, as exerting effort is costly and the benefits from that effort go to the principal.

The principal therefore designs a *contract* $t : \Omega \to \mathbb{R}^n_{\geq 0}$, which maps each outcome to a non-negative vector of transfers $t(\omega) = (t_1(\omega), \dots, t_n(\omega))$ to the agents. I.e., $t_i(\omega)$ is the transfer to agent i under outcome ω.

A particularly important class of contracts in practice, and the class of contracts we focus on are linear contracts. A *linear contract* is defined by a vector $\alpha = (\alpha_1, \dots, \alpha_n)$, and sets $t_i(\omega) = \alpha_i r(\omega)$ for all $i \in A$ and $\omega \in \Omega$. Thus, when $S \subseteq A$ is the set of agents that exert effort, the expected transfer to agent i is $\alpha_i f(S)$. Note that this only depends on the expected reward $f(S)$, and not the details of the distribution.

Focusing on linear contracts is without loss of generality in the important special case of a binary outcome space $\Omega = \{0, 1\}$, where $\omega = 0$ stands for "failure" and comes with no reward (i.e., $r(0) = 0$) and $\omega = 1$ stands for "success" and comes with reward $r(1) \in \mathbb{R}_{\geq 0}$. In this case an appropriately scaled f can be interpreted as success probability function.

For more general outcome spaces it is known that linear contracts are max-min optimal when only the expected rewards $f(S)$ for each $S \subseteq A$ (and not the full details of the underlying distributions) are known [17, 20].

Utility functions and equilibria. Consider a linear contract α, and let S be the set of agents that exert effort. Then the *principal's utility* is given by $(1 - \sum_{i \in A} \alpha_i) f(S)$; while *agent i's utility* is $\alpha_i f(S) - \mathbb{1}[i \in S] \cdot c_i$, where $\mathbb{1}[i \in S] = 1$ if $i \in S$ and $\mathbb{1}[i \in S] = 0$ otherwise. Note that this means that agent i is paid $\alpha_i f(S)$ irrespective of whether $i \in S$, while the cost c_i is only incurred when $i \in S$ (i.e., agent i exerts effort).

To analyze linear contracts, we therefore consider the (pure) Nash equilibria of the induced game among the agents. We capture Nash equilibria through the following definition. We say that a linear contract α incentivizes a set of agents $S \subseteq A$ to exert effort if

$$\begin{aligned} &\alpha_i f(S) - c_i \geq \alpha_i f(S \setminus \{i\}) && \text{for all } i \in S, \text{ and} \\ &\alpha_i f(S) \geq \alpha_i f(S \cup \{i\}) - c_i && \text{for all } i \notin S. \end{aligned}$$

Since equilibria will generally not be unique, we will think of contracts as consisting of a vector α and a set S that is icentivized by α which should exert effort.

The contract design problem. For a fixed set of agents S, the best way for the principal to incentivize the set (if it can be incentivized at all), is via

$$\begin{aligned} &\alpha_i = \frac{c_i}{f(S) - f(S \setminus \{i\})} = \frac{c_i}{f(i \mid S \setminus \{i\})} && \text{for all } i \in S \text{ and} \\ &\alpha_i = 0 && \text{for all } i \notin S, \end{aligned}$$

where we interpret $\frac{c_i}{f(i|S\setminus\{i\})}$ as zero if $c_i = 0$ and $f(i \mid S \setminus \{i\}) = 0$ and as infinity when $c_i > 0$ and $f(i \mid S \setminus \{i\}) = 0$. The principal thus tries to solve $\max_{S \in 2^A} g(S)$ where

$$g(S) := \left(1 - \sum_{i \in S} \frac{c_i}{f(i \mid S \setminus \{i\})}\right) f(S).$$

Let $S^\star$ be the optimal choice of agents, i.e., the set that maximizes g. We say that S is a *β-approximation* (where $\beta \geq 1$) if $\beta \cdot g(S) \geq g(S^\star)$.

Classes of reward functions f. We focus on reward functions $f : 2^A \to \mathbb{R}_{\geq 0}$ that belong to one of the following classes of complement-free set functions [36]:

- Set function f is *additive* if there exist values $v_1, \dots, v_n \in \mathbb{R}_{\geq 0}$ such that $f(S) = \sum_{i \in S} v_i$.
- Set function f is *submodular* if for any two sets $S, S' \subseteq A$ with $S \subseteq S'$ and any $i \in A$ it holds that $f(i \mid S) \geq f(i \mid S')$.
- Set function f is *XOS* if there exists a collection of additive functions $\{a_i : 2^A \to \mathbb{R}_{\geq 0}\}_{i=1,\dots,k}$ such that for each set $S \subseteq A$ it holds that $f(S) = \max_{i=1,\dots,k} a_i(S) = \max_{i=1,\dots,k} \sum_{j \in S} a_{ij}$. Given an XOS function f and a set $S \subseteq A$, there exists an additive function a_i such that $a_i(S) = f(S)$ and $a_i(T) \leq f(T)$ for all $T \subseteq A$; this function is called the additive supporting function of f on S.

- Set function f is *subadditive* if for any two sets $S, S' \subseteq A$ it holds that $f(S) + f(S') \geq f(S \cup S')$.

It is well known that submodular $\subset$ XOS $\subset$ subadditive and all containment relations are strict [36].

Primitives for accessing f. As is common in the combinatorial optimization literature involving set functions, we assume two primitives for accessing f:

- A *value oracle* for f is given $S \in 2^A$ and returns $f(S)$.
- A *demand oracle* for f is given a vector of prices $p = (p_1, \ldots, p_n) \in \mathbb{R}_{\geq 0}$ and returns a set $S \in 2^A$ that maximizes $f(S) - \sum_{j \in S} p_j$.

Both value and demand queries are considered standard in combinatorial optimization problems over set functions. In markets for goods (e.g., combinatorial auctions), a demand query corresponds to the best bundle to purchase given item prices. In our combinatorial contracting problem, a demand query corresponds to the best set of actions to take, had the principal to perform the actions herself. Demand oracles have proven useful in previous studies on combinatorial contracts, see [17].

Auxiliary lemma for XOS functions. For our main result on XOS reward functions we need the following lemma, which generalizes a well-known property of submodular functions to XOS functions.

Lemma 2.1. *[Cf. Lemma 1 in [27]] For any XOS function f and any sets $S \subseteq T$,*

$$\sum_{i \in S} f(i \mid T \setminus \{i\}) \leq f(S).$$

Proof. Let a be an additive supportive function for f on T so that $a(T) = f(T)$ and $a(T') \leq f(T')$ for all $T' \subseteq T$. Then, for any $i \in T$, it holds that

$$f(i \mid T \setminus \{i\}) = f(T) - f(T \setminus \{i\}) \leq a(T) - a(T \setminus \{i\}) = a(\{i\}).$$

Summing over all $i \in S$ we obtain

$$\sum_{i \in S} f(i \mid T \setminus \{i\}) \leq \sum_{i \in S} a(\{i\}) = a(S) \leq f(S),$$

as claimed. □

3 CONSTANT FACTOR FOR SUBMODULAR AND XOS

In this section, we present our main positive results: polynomial-time constant-factor approximation algorithms for submodular and XOS multi-agent combinatorial contracts.

Theorem 3.1. (1) *For submodular f and n agents, it is possible to compute an $O(1)$-approximation to the optimal contract in polynomial time using value queries.*
(2) *For XOS f and n agents, it is possible to compute an $O(1)$-approximation to the optimal contract in polynomial time using value and demand queries.*

Recall that we use $g : 2^A \to \mathbb{R}$ to denote the principal's utility as a function of the set of incentivized agents. I.e., $g(S) := f(S)\left(1 - \sum_{i \in S} \frac{c_i}{f(i \mid S \setminus \{i\})}\right)$. Let $S^\star$ be the optimal set of agents, i.e., the set that maximizes g.

Below we present the full argument for submodular/XOS reward functions assuming value and demand oracle access to the reward function. The result to submodular reward functions with only value oracle access requires only small modifications and relies on known algorithms for computing approximate demand sets [31, 40]. We defer the details of this extension to Appendix B.

3.1 Decomposing the Benchmark

Our first lemma provides a useful decomposition of the benchmark by showing that $g(S^\star)$ is upper bounded by the sum of $f(S^\star \cap A')$, where $A' = \{i \in A \mid \frac{c_i}{f(\{i\})} \leq \frac{1}{2}\}$, and $\max_{i \in A} g(\{i\})$ — the best contract for incentivizing a single agent.

This may look innocent, but is not, because—as we already observed earlier—generally none of the nice structural properties of f (such as non-negativity, monotonicity, submodularity or being XOS) carry over to g.

An important consequence of the lemma is that, since it's easy to find the best contract for incentivizing a single agent, we can focus on the non-trivial task of finding a contract that approximates $f(S^\star \cap A')$.

Lemma 3.2. *If f is XOS (or, more generally, subadditive), then*

$$g(S^\star) \leq f(S^\star \cap A') + \max\{0, \max_{i \in A} g(\{i\})\}.$$

Proof. If $g(S^\star) = 0$, then the claim is trivial. Otherwise, we first prove that $|S^\star \setminus A'| \leq 1$. This is since,

$$\begin{aligned} 0 \quad < \quad & g(S^\star) = f(S^\star)\left(1 - \sum_{i \in S^\star} \frac{c_i}{f(i \mid S^\star \setminus \{i\})}\right) \\ \leq \quad & f(S^\star)\left(1 - \sum_{i \in S^\star \setminus A'} \frac{c_i}{f(\{i\})}\right) \leq f(S^\star)\left(1 - \frac{|S^\star \setminus A'|}{2}\right), \end{aligned}$$

where the second inequality follows from subadditivity of f which implies that $f(i \mid S^\star \setminus \{i\}) = f(S^\star) - f(S^\star \setminus \{i\}) \leq f(\{i\})$. Since $f(S^\star) > 0$, this implies that $|S^\star \setminus A'| \leq 1$.

If $|S^\star \setminus A'| = 0$, the statement follows since $g(S^\star) = g(S^\star \cap A') \leq f(S^\star \cap A')$. Else, let $i^\star$ be the single item in $S^\star \setminus A'$. We have

$$\begin{aligned} g(S^\star) \quad & \leq \quad f(S^\star \cap A') + f(\{i^\star\})\left(1 - \frac{c_{i^\star}}{f(i^\star \mid S^\star \setminus \{i^\star\})}\right) \\ & \leq \quad f(S^\star \cap A') + \max_{i \in A} g(\{i\}). \end{aligned}$$

The first inequality follows from subadditivity of f, implying that $f(S^\star) \leq f(S^\star \cap A') + f(\{i^\star\})$, and by decreasing the payments. The second inequality follows again by subadditivity of f, implying that $f(i^\star \mid S^\star \setminus \{i^\star\}) \leq f(\{i^\star\})$. This concludes the proof. □

3.2 Relaxing the Problem

We next present two crucial lemmas that draw connections between rewards, marginal rewards, and costs. This pair of lemmas relaxes the problem and motivates our approach for finding a good set of agents to incentivize in a contract via prices and demand queries.

The next lemma shows that for the optimal set of agents $S^\star$, the costs are not too high. Clearly, for the optimal set of agents we must have $\sum_{i \in S^\star} c_i \leq f(S^\star)$ because otherwise the reward cannot compensate the incurred cost. In the multi-agent hidden-action

setting with XOS rewards, we can strengthen this observation as follows.

Lemma 3.3. *If f is XOS, then for all $S \subseteq S^\star$ we have*

$$\sum_{i \in S} \sqrt{c_i} \leq \sqrt{f(S)}.$$

Proof. Note that $f(i \mid S^\star \setminus \{i\}) > 0$ for all $i \in S^\star$ with $c_i > 0$ because otherwise $g(S^\star) = -\infty$, whereas $g(\emptyset) = 0$, contradicting the optimality of $S^\star$.

Let's first consider the case where $f(S^\star) = 0$. In this case, we have $f(i \mid S^\star \setminus \{i\}) \leq f(\{i\}) \leq f(S^\star) = 0$ for all $i \in S^\star$. This means that $c_i = 0$ for all $i \in S^\star$, implying the statement.

So consider the case where $f(S^\star) > 0$. By optimality $g(S^\star) \geq 0$, so

$$\begin{aligned} & g(S^\star) = f(S^\star)\left(1 - \textstyle\sum_{i \in S^\star} \frac{c_i}{f(i|S^\star \setminus \{i\})}\right) \geq 0 \\ \implies \quad & \textstyle\sum_{i \in S^\star} \frac{c_i}{f(i|S^\star \setminus \{i\})} \leq 1, \end{aligned}$$

where we used that $f(S^\star) > 0$ for the implication.

For $i \in S$ let $x_i = \sqrt{\frac{c_i}{f(i|S^\star \setminus \{i\})}}$ and $y_i = \sqrt{f(i \mid S^\star \setminus \{i\})}$. This is well-defined because $\frac{c_i}{f(i|S^\star \setminus \{i\})} \geq 0$ by our initial observation. The Cauchy-Schwarz inequality states

$$\left(\sum_{i \in S} x_i y_i\right)^2 \leq \left(\sum_{i \in S} x_i^2\right)\left(\sum_{i \in S} y_i^2\right).$$

Using this we obtain

$$\left(\sum_{i \in S} \sqrt{c_i}\right)^2 \leq \underbrace{\left(\sum_{i \in S} \frac{c_i}{f(i \mid S^\star \setminus \{i\})}\right)}_{\leq 1} \left(\sum_{i \in S} f(i \mid S^\star \setminus \{i\})\right) \leq f(S),$$

where the last inequality holds by Lemma 2.1. Taking the square root on both sides of the inequality establishes the claim. □

The next lemma shows that if for some set S, the marginal of every agent i is not too small, then the principal pays at most half of the reward as transfers. Therefore, our approach will be to find a set S for which $f(S)$ is high and also all marginals fulfill these constraints.

Lemma 3.4. *If f is XOS, then for any set S that fulfills $f(S) > 0$ and*

$$f(i \mid S \setminus \{i\}) \geq \sqrt{2 c_i f(S)} \quad \text{for all } i \in S,$$

we have $g(S) \geq \frac{1}{2} f(S)$.

Proof. Consider any $i \in S$. If $c_i > 0$, note that we have to have $f(i \mid S \setminus \{i\}) > 0$. So

$$f(i \mid S \setminus \{i\}) \geq \sqrt{2 c_i f(S)}$$

is equivalent to

$$\frac{c_i}{f(i \mid S \setminus \{i\})} \leq \frac{1}{2} \frac{f(i \mid S \setminus \{i\})}{f(S)}.$$

For any $i \in S$ with $c_i = 0$, we defined $\frac{c_i}{f(i|S \setminus \{i\})} = 0$, even when the denominator is zero; so also $\frac{c_i}{f(i|S \setminus \{i\})} \leq \frac{1}{2} \frac{f(i|S \setminus \{i\})}{f(S)}$ because $f(i \mid S \setminus \{i\}) \geq 0$. Summing over all $i \in S$ we obtain

$$\sum_{i \in S} \frac{c_i}{f(i \mid S \setminus \{i\})} \leq \frac{1}{2} \frac{\sum_{i \in S} f(i \mid S \setminus \{i\})}{f(S)} \leq \frac{1}{2},$$

where the second inequality holds by Lemma 2.1. Therefore

$$g(S) = \left(1 - \sum_{i \in S} \frac{c_i}{f(i \mid S \setminus \{i\})}\right) f(S) \geq \frac{1}{2} f(S). \qquad \square$$

3.3 A Scaling Property of XOS Functions and Their Marginals

The other crucial ingredient in our argument for finding a contract is a novel scaling property of XOS functions and their marginals, which roughly says that we can scale down the reward $f(T)$ of any set T to whatever level we wish, while also ensuring that the marginals of the elements that remain stay high (with respect to the original marginals).

This property is not too suprising for submodular f, and is indeed easy to obtain for this class of functions: Consider iteratively dropping elements from T one by one to get down to the desired level. Then, by submodularity, the marginals of the items that remain can only go up. For XOS f, however, this is far from obvious (given the apparent lack of control over marginals), and may be of independent interest.

Lemma 3.5. *Let $f : 2^A \to \mathbb{R}_{\geq 0}$ be an XOS function. Given a set $T \subseteq A$, and parameters $\delta \in (0, 1]$ and $0 \leq \Psi < f(T)$, Algorithm 1 runs in polynomial time with value oracle access to f and finds a set $U \subseteq T$ such that*

$$(1 - \delta)\Psi \leq f(U) \leq \Psi + \max_{i \in T} f(\{i\}), \tag{1}$$

and

$$f(i \mid U \setminus \{i\}) \geq \delta f(i \mid T \setminus \{i\}) \qquad \text{for all } i \in U. \tag{2}$$

ALGORITHM 1: Scaling sets for XOS

Data: An XOS function $f : 2^A \to \mathbb{R}_{\geq 0}$, a set T, parameters $0 \leq \Psi < f(T)$ and $\delta \in (0, 1]$
Result: A set $U \subseteq T$ satisfying Equations (1) and (2)
Let T_0 be an inclusion-wise minimal subset of T with $f(T_0) = f(T)$ ▷ I.e., $f(S) < f(T_0), \forall S \subsetneq T_0$
for $t = 1, \ldots, |T_0|$ **do**
  Let $i_t = \arg\min_{i \in T_{t-1}} \frac{f(i|T_{t-1} \setminus \{i\})}{f(i|T_0 \setminus \{i\})}$
  Let $T_t = T_{t-1} \setminus \{i_t\}$
  Let $\delta_t = \frac{f(i_t|T_t)}{f(i_t|T_0 \setminus \{i_t\})}$
end
Let $j^\star = \min\{j \mid f(T_j) \leq \Psi\}$
Let $k^\star = \min\{k \mid f(T_k) \leq (1 - \delta) f(T_{j^\star - 1})\}$
Let $t^\star = \arg\max_{t \in \{j^\star, \ldots, k^\star\}} \delta_t$
return $U = T_{t^\star - 1}$

Proof. It is easy to see that Algorithm 1 can be implemented in polynomial time with value oracle access to f; in particular, T_0 can be obtained by iteratively dropping any element from T whose removal does not decrease the value until reaching a point where no such element exists. So we focus on showing that it finds a set U that satisfies (1) and (2).

We begin by arguing that the algorithm is well defined. First note that $f(i \mid T_0 \setminus \{i\}) > 0$ for all $i \in T_0$ because otherwise T_0 would not be inclusion-wise minimal. Furthermore, we have to argue that the indices $j^\star$ and $k^\star$ as defined in the algorithm exist and satisfy $1 \le j^\star \le k^\star \le |T_0|$. To this end observe that $f(T) = f(T_0) \ge f(T_1) \ge \ldots \ge f(T_{|T_0|})$ and $T_{|T_0|} = \emptyset$, so $f(T_{|T_0|}) = 0$. Therefore there must be a smallest j with $0 \le j \le |T_0|$ such that $f(T_j) \le \Psi$. This is $j^\star$. Moreover, since $\Psi < f(T) = f(T_0)$, it must be that $j^\star \ge 1$. Hence there must also be a smallest k with $j^* - 1 \le k \le |T_0|$ such that $f(T_k) \le (1-\delta) f(T_{j^\star - 1})$. This is $k^\star$. Finally, since $\delta > 0$, and $f(T_{j^\star-1}) > \Psi \ge 0$, this $k^\star$ must satisfy $k^\star \ge j^\star$.

It remains to show that $U = T_{t^\star - 1}$ fulfills the respective properties.

By definition, it holds that

$$f(T_{j^\star-1}) = f(T_{k^\star}) + \sum_{t \in \{j^\star,\ldots,k^\star\}} f(i_t \mid T_t).$$

By rearranging and replacing $f(i_t \mid T_t)$ by $\delta_t \cdot f(i_t \mid T_0 \setminus \{i_t\})$ we get that

$$\begin{aligned} f(T_{j^\star-1}) - f(T_{k^\star}) &= \sum_{t \in \{j^\star,\ldots,k^\star\}} \delta_t \cdot f(i_t \mid T_0 \setminus \{i_t\}) \\ &\le \sum_{t \in \{j^\star,\ldots,k^\star\}} \delta_{t^\star} \cdot f(i_t \mid T_0 \setminus \{i_t\}) \\ &\le \delta_{t^\star} \cdot f(\{i_{j^\star}, \ldots, i_{k^\star}\}) \\ &\le \delta_{t^\star} \cdot f(\{i_{j^\star}, \ldots, i_{|T_0|}\}) = \delta_{t^\star} \cdot f(T_{j^\star-1}), \end{aligned}$$

where the first inequality is by the definition of $t^\star$, the second inequality is by Lemma 2.1, and the third inequality is by monotonicity. Thus, for all $i \in U = T_{t^\star-1}$,

$$\begin{aligned} \frac{f(i \mid T_{t^\star-1} \setminus \{i\})}{f(i \mid T_0 \setminus \{i\})} &\ge \delta_{t^\star} \ge \frac{f(T_{j^\star-1}) - f(T_{k^\star})}{f(T_{j^\star-1})} \\ &= 1 - \frac{f(T_{k^\star})}{f(T_{j^\star-1})} \ge 1 - \frac{(1-\delta) f(T_{j^\star-1})}{f(T_{j^\star-1})} = \delta, \end{aligned}$$

where the first inequality holds by definition of $i_{t^\star}$ and $\delta_{t^\star}$. Therefore, in order to establish Equation (2), it remains to show that $f(i \mid T_0 \setminus \{i\}) \ge f(i \mid T \setminus \{i\})$. Indeed, since $f(T_0) = f(T)$ and $T_0 \subseteq T$, monotonicity of f implies that

$$f(i \mid T_0 \setminus \{i\}) = f(T_0) - f(T_0 \setminus \{i\}) \ge f(T) - f(T \setminus \{i\}) = f(i \mid T \setminus \{i\}).$$

To prove Equation (1) observe that $f(T_{t^\star-1}) \ge f(T_{k^\star-1}) \ge (1-\delta) f(T_{j^\star-1}) \ge (1-\delta)\Psi$, and that $f(T_{t^\star-1}) \le f(T_{j^\star-1}) \le f(T_{j^\star}) + \max_{i \in T} f(\{i\}) \le \Psi + \max_{i \in T} f(\{i\})$. □

3.4 Computing a Good Set via a Demand Oracle

We next present a subroutine that formalizes the argument outlined in the introduction that if we knew the value of $f(S^\star \cap A')$, then by running a demand query at appropriately chosen prices and using the scaling property of XOS functions from the previous section we can find a contract that constant-factor approximates $f(S^\star \cap A')$.

The actual argument is a bit more complicated than what we sketched before as it has to deal with the fact that we don't know the value of $f(S^\star \cap A')$; and that we cannot simply ignore the contribution of individual agents.

ALGORITHM 2: Approximately optimal contract given an estimate of $f(S^\star \cap A')$

Data: An XOS function $f : 2^A \to \mathbb{R}_{\ge 0}$, costs $\{c_i\}_{i \in A}$, an estimate $\tilde{y} \ge 0$ of $f(S^\star \cap A')$
Result: A set $U \subseteq A'$
Let $\Psi = \frac{\tilde{y}}{32} - \max_{i \in A'} f(\{i\})$.
For every $i \in A'$, let $p_i = \frac{1}{2} \cdot \sqrt{c_i \cdot \tilde{y}}$ ($p_i = \infty$ for $i \in A \setminus A'$)
Let T be a demand set with prices p_i over the set A'
if $0 < \Psi < f(T)$ **then**
  $U \leftarrow$ the output of Algorithm 1 on $(f, T, \Psi, \delta = \frac{1}{2})$
else
  $U \leftarrow \emptyset$
end
return U

Lemma 3.6. *Algorithm 2 runs in polynomial time with access to a demand and value oracle to f. If $\tilde{y} \le f(S^\star \cap A')$, then the set $U \subseteq A'$ that it returns satisfies*

$$g(U) \ge \max\left\{\frac{\tilde{y}}{128} - \frac{\max_{i \in A'} f(\{i\})}{4}, 0\right\}.$$

Proof. Algorithm 2 runs in polynomial time with access to a demand and value oracle to f, because it issues a single demand query to f and makes at most one call to Algorithm 1 which runs in polynomial-time with value oracle access to f.

We next prove the lower bound on $g(U)$, under the assumption that $\tilde{y} \le f(S^\star \cap A')$. If $\Psi \le 0$ then the algorithm returns the empty set, which satisfies the claim. So consider the case $\Psi > 0$. In this case we have

$$\begin{aligned} f(T) &\ge f(T) - \sum_{i \in T} p_i \\ &\ge f(S^\star \cap A') - \sum_{i \in S^\star \cap A'} p_i \\ &= f(S^\star \cap A') - \frac{\sqrt{\tilde{y}}}{2} \cdot \sum_{i \in S^\star \cap A'} \sqrt{c_i} \\ &\ge f(S^\star \cap A') - \frac{1}{2}\sqrt{f(S^\star \cap A') \cdot \tilde{y}} \\ &\ge \frac{1}{2} \cdot f(S^\star \cap A'), \end{aligned} \tag{3}$$

where the first inequality holds since the prices are positive, the second inequality uses that T is a demand set over the set A', the third inequality is by Lemma 3.3, and the last inequality holds because $\tilde{y} \le f(S^\star \cap A')$.

Furthermore, note that, as T is a demand set, we have to have $f(i \mid T \setminus \{i\}) \ge p_i$ for all $i \in T$ (because otherwise it would be beneficial to remove i).

Now, since $\Psi = \tilde{y}/32 - \max_{i\in A'} f(\{i\}) > 0$, we must have $\tilde{y} > 0$ and hence $\tilde{y}/2 > \tilde{y}/32 \geq \Psi$. This shows

$$f(T) \overset{(3)}{\geq} \frac{1}{2} \cdot f(S^\star \cap A') \geq \frac{\tilde{y}}{2} > \Psi.$$

By Lemma 3.5, the set $U \subseteq T$ therefore fulfills

$$f(U) \geq (1-\delta)\,\Psi = \frac{1}{64}\tilde{y} - \frac{1}{2}\max_{i\in T} f(\{i\}) \geq \frac{1}{64}\tilde{y} - \frac{1}{2}\max_{i\in A'} f(\{i\}).$$

Furthermore, U fulfills

$$f(U) \leq \Psi + \max_{i\in T} f(\{i\}) \leq \frac{\tilde{y}}{32} \tag{4}$$

and for all $i \in U$

$$f(i \mid U \setminus \{i\}) \geq \delta f(i \mid T \setminus \{i\}) = \frac{1}{2} \cdot f(i \mid T \setminus \{i\}). \tag{5}$$

We conclude that, for all $i \in U$, we have

$$\begin{aligned} f(i \mid U \setminus \{i\}) &\overset{(5)}{\geq} \frac{1}{2} \cdot f(i \mid T \setminus \{i\}) \geq \frac{1}{2} \cdot p_i \\ &= \frac{1}{4} \cdot \sqrt{c_i \tilde{y}} = \sqrt{2c_i \frac{\tilde{y}}{32}} \overset{(4)}{\geq} \sqrt{2c_i f(U)}. \end{aligned}$$

So, Lemma 3.4 implies $g(U) \geq \frac{1}{2} f(U)$. In combination, we obtain

$$g(U) \geq \frac{1}{2} f(U) \geq \frac{1}{2}\left(\frac{1}{64}\tilde{y} - \frac{1}{2}\max_{i\in A'} f(\{i\})\right),$$

as claimed. □

3.5 Putting It All Together

We are finally in a position to wrap up our argument. Below we show that Algorithm 3 achieves a constant-factor approximation to $g(S^\star)$ for XOS reward function f, and runs in polynomial time with access to a demand and a value oracle to f.

The idea is simple: Try out all contracts that incentivize a single agent; and use the subroutine from the previous section with polynomially many guesses for the value of $f(S^\star \cap A')$ to obtain additional candidate sets. Argue that for the right guess of $f(S^\star \cap A')$ the better of the best contract that incentivizes a single agent, and the candidate set U from the subroutine for this guess yields the desired approximation.

ALGORITHM 3: Approximate optimal contract for XOS

Data: An XOS function $f : 2^A \to \mathbb{R}_{\geq 0}$, costs $\{c_i\}_{i\in A}$, parameter $\xi > 1$
Let $C = \{\{i\} \mid i \in A\} \cup \{\emptyset\}$
Let $A' = \{i \in A \mid \frac{c_i}{f(\{i\})} \leq \frac{1}{2}\}$
if $A' \neq \emptyset \wedge \max_{i\in A'} f(\{i\}) > 0$ **then**
  Let $x = \max_{i\in A'} f(\{i\})/2$
  for $j = 0, .., \lceil \log_\xi 2n \rceil$ **do**
    Let $x_j = x \cdot \xi^j$
    $U^{(j)} \leftarrow$ the output of Algorithm 2 on $(f, \{c_i\}_{i\in A}, A', \tilde{y} = x_j)$
    $C \leftarrow C \cup \{U^{(j)}\}$
  end
end
return $\arg\max_{S\in C} g(S)$

Proof of Theorem 3.1. We show that Algorithm 3 achieves the claims in the theorem for XOS reward functions f, assuming that we have access to a demand and a value oracle to f. We present details on how to obtain the result for submodular reward functions f when we only have value oracle access to f in Appendix B.

We begin by observing that, for any $\xi > 1$, Algorithm 3 runs in polynomial time with access to a demand and a value oracle to f, because we make only polynomially many calls to Algorithm 2, which itself runs in polynomial time with demand/value oracle access to f.

We next show that Algorithm 3 obtains an approximation guarantee of $1/(256\xi+2)$ which tends to $1/258$ as $\xi \to 1$. To this end, we distinguish between two cases. The first case is that $f(S^\star \cap A') \leq 128\xi \cdot \max\{0, \max_{i\in A} g(\{i\})\}$. Then by Lemma 3.2

$$\begin{aligned} g(S^\star) &\leq f(S^\star \cap A') + \max\{0, \max_{i\in A} g(\{i\})\} \\ &\leq (128\xi + 1) \cdot \max\{0, \max_{i\in A} g(\{i\})\}. \end{aligned}$$

So, the best single action (or the empty set) gives at least an approximation of $\frac{1}{128\xi+1}$, which completes the proof of the theorem for this case.

In the other case, we have

$$f(S^\star \cap A') > 128\xi \cdot \max\{0, \max_{i\in A} g(\{i\})\}.$$

So, in particular, we need to have $A' \neq \emptyset$ and $\max_{i\in A'} f(\{i\}) > 0$. Furthermore, $g(\{i\}) \geq \frac{1}{2} f(\{i\})$ for all $i \in A'$, resulting in $\max_{i\in A} g(\{i\}) \geq x$. So, in combination in this case $f(S^\star \cap A') \geq x$. On the other hand, by subadditivity of f and the definition of x, $f(S^\star \cap A') \leq \sum_{i\in S^\star \cap A'} f(\{i\}) \leq 2n \cdot x$. Therefore, there is a unique $j^\star \in \{0, 1, \ldots, \lceil \log_\xi 2n \rceil\}$ for which $x_j \leq f(S^\star \cap A') < \xi \cdot x_j$.

Now, since $\frac{1}{\xi} \cdot f(S^\star \cap A') \leq x_j \leq f(S^\star \cap A')$,

$$\begin{aligned} g(U^{(j^\star)}) &\geq \frac{x_{j^\star}}{128} - \frac{\max_{i\in A'} f(\{i\})}{4} \\ &\geq \frac{f(S^\star \cap A')}{128 \cdot \xi} - \frac{\max_{i\in A} g(\{i\})}{2} \\ &\geq f(S^\star \cap A') \cdot \left(\frac{1}{128 \cdot \xi} - \frac{1}{256 \cdot \xi}\right) \\ &\geq \frac{128\xi \cdot g(S^\star)}{128\xi + 1} \frac{1}{256 \cdot \xi} = \frac{g(S^\star)}{256\xi + 2}, \end{aligned}$$

where the first inequality is by Lemma 3.6, the second inequality is since for every $i \in A'$ it holds that $\frac{c_i}{f(\{i\})} \leq \frac{1}{2}$ and thus $g(\{i\}) = f(\{i\})(1 - \frac{c_i}{f(\{i\})}) \geq f(\{i\})/2$, the third and fourth inequality both use that $f(S^\star \cap A') > 128\xi \cdot \max\{0, \max_{i\in A} g(\{i\})\}$, and the the fourth inequality additionally uses Lemma 3.2. This gives an approximation of $\frac{1}{256\xi+2}$. □

4 INAPPROXIMABILITY RESULTS FOR XOS AND SUBADDITIVE

In this section, we present our main inapproximability results. We show a lower bound of $\Omega(\sqrt{n})$ for subadditive reward functions, and a constant lower bound for XOS reward functions (both with demand queries).

A striking feature of both of these results is that the reward functions used in the proofs are constant-factor close to submodular (see Observation 4.3).

4.1 Inapproximability Result for Subadditive

We begin by stating and proving our inapproximability result for subadditive reward functions.

Theorem 4.1. *For every randomized algorithm that performs only polynomially many value and demand queries, there is a subadditive function f such that the approximation ratio is $\Omega(\sqrt{n})$ with high probability.*

To prove the theorem, we define a probability distribution over instances and consider an arbitrary deterministic algorithm on a randomly drawn instance. By Yao's principle, the worst-case approximation ratio of any randomized algorithm is lower-bounded by the the approximation ratio of any deterministic algorithm on this randomized instance.

We describe the random instances in Section 4.1.1, and establish properties of the reward functions and the principal's utility function used in the construction in Section 4.1.2 and Section 4.1.3. We prove a lemma that limits the power of demand queries in these random instances in Section 4.1.4, and give the proof of Theorem 4.1 in Section 4.1.5.

4.1.1 Distribution Over Subadditive Reward Functions f_T. To define the probability distribution over instances, let $n \in \mathbb{N}$ be an even square number. In every instance, the agents' costs are the same, namely $c_i = \frac{2}{n}$ for every agent i.

The instances differ in terms of the reward functions. To define them, consider the following subadditive function f over n agents:

$$f(S) = \begin{cases} 3 + \frac{2\cdot|S|}{\sqrt{n}} & |S| \le \frac{n}{2} \\ 4 + \sqrt{n} & |S| = \frac{n}{2} + 1 \\ 5 + \sqrt{n} & |S| = \frac{n}{2} + 2 \\ 6 + \sqrt{n} & |S| \ge \frac{n}{2} + 3 \end{cases}$$

Given a set T of size $n/2+1$, let f_T be the function where $f_T(S) = f(S) + \mathbb{1}[S = T]$. Accordingly, let g_T be the associated principal's utility function. The random instance consists of $f_{T^\star}$, where $T^\star$ is a uniformly drawn set of size $n/2+1$.

4.1.2 Properties of the Reward Function f_T. We state and prove properties of the reward functions f_T. In particular, we show that they are subadditive, and close to submodular functions, and even to symmetric ones; i.e., submodular functions that depend only on the cardinality of the set.[1]

Claim 4.2. *For every T, the function f_T is subadditive.*

Proof. Consider any disjoint sets $S_1, S_2 \subseteq A$. We show that $f_T(S_1) + f_T(S_2) \ge f_T(S_1 \cup S_2)$. Distinguish two cases.

Case 1: $|S_1|, |S_2| \le \frac{n}{2}$. In this case $f_T(S_1) + f_T(S_2) = 6 + \frac{2\cdot(|S_1|+|S_2|)}{\sqrt{n}}$. If $|S_1| + |S_2| \le \frac{n}{2}$, then $f_T(S_1 \cup S_2) \le 3 + \frac{2\cdot(|S_1|+|S_2|)}{\sqrt{n}} \le 6 + \frac{2\cdot(|S_1|+|S_2|)}{\sqrt{n}} = f_T(S_1) + f_T(S_2)$. If $|S_1| + |S_2| > \frac{n}{2}$, then $f_T(S_1 \cup S_2) \le 6 + \sqrt{n} \le 6 + \frac{2\cdot(|S_1|+|S_2|)}{\sqrt{n}} = f_T(S_1) + f_T(S_2)$.

Case 2: $|S_1| > \frac{n}{2}, |S_2| \ge 1$. In this case $f_T(S_1) + f_T(S_2) \ge 4 + \sqrt{n} + 3 \ge f_T(S_1 \cup S_2)$. □

Observation 4.3. *There is a symmetric submodular function f' that fulfills*

$$f'(S) \le f_T(S) \le \left(1 + \frac{3}{3+\sqrt{n}}\right) f'(S)$$

for all S and T.

Proof. Let $f'(S) = \min\{3 + \frac{2\cdot|S|}{\sqrt{n}}, 3 + \sqrt{n}\}$. Note that f', as the minimum between two submodular functions, is submodular. Observe that $f'(S) \le f_T(S)$ for all $S \subseteq A$ with equality for $|S| \le \frac{n}{2}$. For $|S| > \frac{n}{2}$, we have $f'(S) = 3 + \sqrt{n}$ and $f_T(S) \le 6 + \sqrt{n}$. □

4.1.3 Properties of the Principal's Utility Function g_T. We next show that T is the optimal contract for reward function f_T, and no other set gives a better than $O(\sqrt{n})$-approximation to T. This implies that one needs to identify T in order to get a better than $O(\sqrt{n})$-approximation.

Lemma 4.4. *We have that $g_T(T) \ge \frac{\sqrt{n}}{4}$ and $g_T(S) \le 5$ for all $S \ne T$.*

Proof. First note that for every $i \in T$, $f_T(i \mid T \setminus \{i\}) = 2$. Thus, $g_T(T) = f_T(T)\left(1 - \sum_{i\in T} \frac{c_i}{f_T(i|T\setminus\{i\})}\right) \ge \frac{\sqrt{n}}{4}$. Next, consider any $S \ne T$. We can distinguish the following cases.

- If $|S| \le 1$, then $g_T(S) \le f_T(S) \le 3 + \frac{2}{\sqrt{n}}$.
- If $1 < |S| \le \frac{n}{2}$, then $f_T(i \mid S \setminus \{i\}) = \frac{2}{\sqrt{n}}$, thus

$$g_T(S) = f_T(S)\left(1 - \sum_{i\in S} \frac{c_i}{f_T(i \mid S \setminus \{i\})}\right) = f_T(S)\left(1 - \frac{1}{\sqrt{n}}|S|\right) \le \begin{cases} 5 \cdot \left(1 - \frac{|S|}{\sqrt{n}}\right) & |S| < \sqrt{n} \\ 0 & |S| \ge \sqrt{n} \end{cases}$$

- If $|S| \ge \frac{n}{2} + 1$ and $S \ne T$, then $f_T(i \mid S \setminus \{i\}) \le 1$ for all $i \in S$, thus

$$g_T(S) = f_T(S)\left(1 - \sum_{i\in S} \frac{c_i}{f_T(i \mid S \setminus \{i\})}\right) \le f_T(S)\left(1 - |S| \cdot \frac{2}{n}\right) \le 0.$$

Hence for any $S \ne T$ it holds that $g_T(S) \le 5$. □

4.1.4 Limiting the Power of Demand Queries. We now show that with each demand query one can distinguish at most n^{29} sets $T^\star$. This together with Lemma 4.4 will derive our impossibility result.

Lemma 4.5. *For $n > 4096$, for every vector of prices $p = (p_1, \ldots, p_n)$, the set of $\{T^\star \mid D(f,p) \ne D(f_{T^\star}, p)\}$ is at most of size n^{29}, where $D(f,p)$ (resp. $D(f_{T^\star}, p)$) is the demand set[2] of function f (resp. $f_{T^\star}$) with respect to prices p.*

Let $S_p = \{i \mid p_i \le 1/4\}$, and $B_p = \{i \mid p_i \ge 1/2\}$.

Claim 4.6. *If $|B_p| \le n/2 - 3$, then for every T it holds that $D(f_T, p) = D(f, p)$.*

[1] Since symmetric submodular functions are gross-substitutes [29], these subadditive functions are even close to gross-substitutes functions.

[2] For simplicity of the proof, we assume that when there are multiple sets in demand, the tie breaking is consistent across all f_T.

Proof. Since f_T and f disagree only on the value of T, it is sufficient to show that $D(f_T, p) \neq T$. Assume towards contradiction that $D(f_T, p) = T$, then, because $|A \setminus B_p| \geq n/2+3$ and $|T| = n/2+1$, the set $\Delta := A \setminus B_p \setminus T$ is of size at least 2. Let x, y be two arbitrary different elements in Δ, and note that for these $p(x) < 1/2$ and $p(y) < 1/2$. Now $f_T(T \cup \{x,y\}) - p(T \cup \{x,y\}) \geq 1 + f_T(T) - p(T) - p(x) - p(y) > f(T) - p(T)$ which contradicts that T is the demand set for f_T. □

Claim 4.7. *For $n > 4096$, if $|S_p| \leq n/2 - 8$, then for every T it holds that $D(f_T, p) = D(f, p)$.*

Proof. Since f_T and f disagree only on the values of T, it is sufficient to show that $D(f_T, p) \neq T$. Assume towards contradiction that $D(f_T, p) = T$, then the set $\Delta := T \setminus S_p$ is of size at least 9. Let X be an arbitrary subset of Δ of size 9. It holds that

$$\begin{aligned} f_T(T \setminus X) - p(T \setminus X) &= f_T(T) - 2 - \frac{16}{\sqrt{n}} - p(T) + p(X) \\ &\geq f_T(T) - p(T) + \frac{|X|}{4} - 2 - \frac{16}{\sqrt{n}} \\ &> f_T(T) - p(T), \end{aligned}$$

which contradicts that T is the demand set. □

Claim 4.8. *If $|S_p| > n/2-8$ and $|B_p| > n/2-3$, then for all $T \notin \{S \mid (|S| = n/2+1) \wedge (|S \setminus SP_p| \leq 14)\}$ it holds that $D(f_T, p) = D(f, p)$.*

Proof. Since f_T and f disagree only on the values of T, it is sufficient to show that $D(f_T, p) \neq T$. Assume towards contradiction that $D(f_T, p) = T$, it holds that the set $\Delta_1 := B_p \cap T$ is of size at least 5 since $|B_p \cap T| \geq |T \setminus S_p| - |A \setminus B_p \setminus S_p| \geq 14 - 9 = 5$, and the set $\Delta_2 := S_p \setminus T$ is of size at least 6 since $|S_p \setminus T| = |T \setminus S_p| - |T| + |S_p| \geq 14 - 8 = 6$. Let X be an arbitrary subset of Δ_1 of size 5 and let Y be an arbitrary subset of Δ_2 of size 5. It holds that

$$\begin{aligned} & f_T((T \setminus X) \cup Y) - p((T \setminus X) \cup Y) \\ = \; & f_T(T) - 1 - p(T) + p(X) - p(Y) \\ \geq \; & f_T(T) - p(T) + \frac{|X|}{4} - 1 > f_T(T) - p(T), \end{aligned}$$

which contradicts that T is the demand set. □

Proof of Lemma 4.5. By combining Claim 4.6 and Claim 4.7, a demand query can only reveal information about T if $|S_p| > n/2-8$ and $|B_p| > n/2-3$, and even then, by Claim 4.8, a demand query cannot distinguish between T's not in $\{S \mid (|S| = n/2+1) \wedge (|S \setminus S_p| \leq 14)\}$. Now, the lemma follows since for every choice of S_p of size greater than $n/2-8$, the set $\{S \mid (|S| = n/2+1) \wedge (|S \setminus S_p| \leq 14)\}$ is at most of size n^{29}. (One can bound it by counting the options to select a set $S \setminus S_p$ of size at most 14 and then select a set $S_p \setminus S$ which is of size at most 15 since $|S_p \setminus S| = |S \setminus S_p| + |S_p| - |S| \leq 14 + n - |B_p| - (n/2+1) \leq 14 + n - (n/2-2) - (n/2+1) = 15$.) □

4.1.5 Putting it All Together. We are now ready to prove the theorem.

Proof of Theorem 4.1. Consider any fixed deterministic algorithm *ALG*, and the random instances described in Section 4.1.1. Let S_{ALG} be the set that the algorithm computes when all demand and value queries are answered according to the function f. We claim that on the vast majority of functions f_T the algorithm will also compute S_{ALG}.

To this end, let $\mathcal{T}_i$ be the family of all sets T of size $\frac{n}{2}+1$ such that the first i demand or value queries performed by the algorithm would have led to the same answers on f_T as on f. Clearly $|\mathcal{T}_0| = \binom{n}{n/2+1}$. By Lemma 4.5 any demand query allows us to distinguish at most n^{29} many sets from f (and of course, any value query can distinguish at most one set). That is, $|\mathcal{T}_{i+1}| \geq |\mathcal{T}_i| - n^{29}$.

This implies that $|\mathcal{T}_i| \geq \binom{n}{n/2+1} - i \cdot n^{29}$. Furthermore, by Lemma 4.4, the approximation ratio of the algorithm that uses at most i queries is no better than $\frac{20}{\sqrt{n}}$ whenever $T^\star \in \mathcal{T}_i$ and $T^\star \neq S_{ALG}$. This happens with probability of at least $1 - \frac{|\mathcal{T}_i|-1}{\binom{n}{n/2+1}}$ since $T^\star$ is uniformly distributed in $\mathcal{T}_i$ and the algorithm can choose just one of them. Overall, we get that the approximation of an algorithm *ALG* that uses at most i demand and value queries is at most

$$\begin{aligned} E\left[\frac{g(S_{ALG})}{g(T^\star)}\right] &\leq \Pr[T^\star \notin \mathcal{T}_i \vee T^\star = S_{ALG}] \cdot 1 \\ &+ (1 - \Pr[T^\star \notin \mathcal{T}_i \vee T^\star = S_{ALG}]) \cdot \frac{20}{\sqrt{n}} \\ &\leq \frac{\binom{n}{n/2+1} - |\mathcal{T}_i| + 1}{\binom{n}{n/2+1}} + \frac{|\mathcal{T}_i| - 1}{\binom{n}{n/2+1}} \cdot \frac{20}{\sqrt{n}} \\ &\leq \frac{20}{\sqrt{n}} + \frac{i \cdot n^{29} + 1}{\binom{n}{n/2+1}} \leq O\left(\frac{1}{\sqrt{n}}\right), \end{aligned}$$

where the last inequality holds for every polynomial amount of demand and value queries. □

4.2 Inapproximability Result for XOS

We conclude by stating our inapproximability result for XOS reward functions. The proof follows the same structure as Theorem 4.1, so we focus on stating the family of XOS functions and defer the rest of the proof to Appendix C.

Theorem 4.9. *For every randomized algorithm that performs only polynomially many value and demand queries, there is a subadditive function f such that the approximation ratio is at least 1.136 with high probability.*

Proof. Let n be an even number. Consider the following symmetric XOS function f over n agents:

$$f(S) = \begin{cases} 1 + \frac{3 \cdot |S|}{n} & |S| \leq \frac{n}{2} \\ 1/2 + \frac{4 \cdot |S|}{n} & |S| > \frac{n}{2} \end{cases}$$

Given a set T of size $n/2+1$, let f_T be the function where $f_T(S) = f(S) + \mathbb{1}[S = T] \cdot \frac{1}{n}$. The costs of all actions are $\frac{5}{n(n+2)}$.

The rest of the proof follows exactly the same steps as in the proof of Theorem 4.1, just instead of using Claim 4.2, Lemma 4.4 and Lemma 4.5, it uses Claim C.1, Lemma C.2 and Lemma C.3 which are deferred to Appendix C. □

5 DISCUSSION

In this paper, we study a combinatorial variant of the principal-agent contract model with multiple agents, seeking to maximize the principal's expected utility. We explore the design of near-optimal

contracts for reward functions from the hierarchy of complement-free functions of [36]. We show that XOS reward functions admit constant-factor approximation, while for subadditive reward functions there is a polynomial lower bound. Our work suggests many natural open questions including:

- Is there a PTAS for submodular reward functions? Our lower bound only works for XOS reward functions.
- What is the best approximation that can be achieved for subadditive reward functions with access to demand oracles? Our work shows a lower bound of $\Omega(\sqrt{n})$, while an $O(n)$ upper bound can be achieved trivially by incentivizing at most one agent.

More generally, it is an interesting direction to combine our model and techniques with the ones in [17] to design approximation algorithms for a contract problem with multiple agents, each choosing from multiple actions.

A ADDITIVE REWARD FUNCTIONS

In this appendix we show that it is NP-hard to find the best contract for additive f, but it is possible to find an FPTAS for approximating the best contract.

A.1 NP-Hardness

Proposition A.1. *The optimal contract problem for additive reward functions over n agents is NP-hard.*

Proof. The proof is by reduction from PARTITION. We are given a multi-set $\{w_1, \ldots, w_n\}$ of positive integers $w_j > 0$ with $\sum_{j=1}^{n} w_j = W$, and have to decide whether $[n]$ can be partitioned into I_1 and $I_2 = [n] \setminus I_1$ such that $\sum_{j \in I_1} w_j = \sum_{j \in I_2} w_j = W/2$.

The corresponding contract problem is: f is additive over n agents, where agent i has a value of $v_i = w_i$ and a cost $c_i = w_i^2/W$. Since agent i's marginal is v_i, the indifference point for agent i is $\alpha_i = c_i/v_i = w_i/W$. The principal's utility for incentivizing a set of agents $S \subseteq [n]$ is

$$g(S) = \left(1 - \sum_{i \in S} \frac{w_i}{W}\right) \cdot \sum_{i \in S} w_i.$$

g is maximized when $\sum_{i \in S} w_i$ is closest to $W/2$. Thus, instance $\{w_1, \ldots, w_n\}$ is a yes-instance, if and only if $g(S^\star) = \frac{W}{4}$. □

A.2 FPTAS

Proposition A.2. *There is an FPTAS for the optimal contract problem for an additive reward functions over n agents.*

Suppose we know $b = \max_{i \in S^\star} f(\{i\})$. We can assume this because there are only n choices and can run the following algorithm with each choice.

Let $\delta = \frac{\epsilon}{n}$ and define an additive function $\tilde{f}$ by rounding down each $f(\{i\})$ to the next multiple of δb. That is, $\tilde{f}(\{i\}) = \left\lfloor \frac{f(\{i\})}{\delta b} \right\rfloor \delta b$ and $\tilde{f}(S) = \sum_{i \in S} \tilde{f}(\{i\})$. This way, each $\tilde{f}(S)$ is a multiple of δb.

Let T_x be the set S that minimizes $\sum_{i \in S} \frac{c_i}{f(\{i\})}$ subject to $\tilde{f}(T) \geq x$. Our algorithm returns the set T_x that maximizes $\left(1 - \sum_{i \in T_x} \frac{c_i}{f(\{i\})}\right) x$ among all $x = k\delta b$ for $k \in \{0, 1, \ldots, \lceil \frac{n}{\delta} \rceil\}$.

Claim A.3. *The algorithm can be implemented in polynomial time in n and $\frac{1}{\epsilon}$.*

Proof. Observe that we only need to compute polynomially many sets T_x. Furthermore, these sets can be computed by dynamic programming in polynomial time.

To this end, let

$$A(j, x) = \min \left\{ \sum_{i \in S} \frac{c_i}{f(\{i\})} \mid S \subseteq \{1, \ldots, j\}, \tilde{f}(S) \geq x \right\},$$

$A(0, x) = 0$ if $x \leq 0$ and $A(0, x) = \infty$ if $x > 0$. Observe that

$$A(j, x) = \min \left\{ A(j-1, x), A(j-1, x - \tilde{f}(\{i\})) + \frac{c_i}{f(\{i\})} \right\},$$

which completes the proof. □

Claim A.4. *The algorithm computes a $(1-\epsilon)$-approximation to the optimal contract.*

Proof. Note that for every set S, we have $\tilde{f}(S) \leq f(S)$. Therefore for every set T_x

$$g(T_x) = \left(1 - \sum_{i \in T_x} \frac{c_i}{f(\{i\})}\right) f(T_x) \geq \left(1 - \sum_{i \in T_x} \frac{c_i}{f(\{i\})}\right) x.$$

In particular, for the set T_x that we return, we have

$$\begin{aligned} \left(1 - \sum_{i \in T_x} \frac{c_i}{f(\{i\})}\right) x &\geq \left(1 - \sum_{i \in T_{\tilde{f}(S^\star)}} \frac{c_i}{f(\{i\})}\right) \tilde{f}(S^\star) \\ &\geq \left(1 - \sum_{i \in S^\star} \frac{c_i}{f(\{i\})}\right) \tilde{f}(S^\star) \end{aligned}$$

because $\tilde{f}(S^\star) = k\delta b$ for some k in the range.

Finally, observe that we have

$$\begin{aligned} \tilde{f}(S^\star) &= \sum_{i \in S^\star} \tilde{f}(\{i\}) \geq \sum_{i \in S^\star} (f(\{i\}) - \delta b) \\ &\geq f(S^\star) - n\delta b \geq (1-\epsilon) f(S^\star). \end{aligned}$$

Therefore

$$(1-\epsilon) g(S^\star) \leq \left(1 - \sum_{i \in S^\star} \frac{c_i}{f(\{i\})}\right) \tilde{f}(S^\star),$$

as claimed. □

B SUBMODULAR WITH VALUE QUERIES

In this appendix we show how to adjust the proof of Theorem 3.1 in Section 3 to submodular reward functions with value queries.

B.1 Approximate Demand Query via Value Queries

Our adjusted argument relies on the ability to compute an approximate demand set as formalized in the following lemma.

Lemma B.1 ([31, 40]). *For submodular f given access to a value oracle, there exists a poly-time algorithm that finds a set S such that*

$$f(S) - \sum_{i \in S} p_i \geq (1 - 1/e) f(T) - \sum_{i \in T} p_i \quad \text{for all } T.$$

We call the set S a β-approximate demand set, where $\beta = 1 - 1/e$. Note that this notion of an approximate demand query is not a fully multiplicative approximation, it's weaker in that we need to subtract the full price of a set.

B.2 Adjusting the Proof to Approximate Demand Queries

We next show how to adapt our proof to the case of submodular f, using only value oracle calls. To do so, we first show that if we modify Algorithm 2 so that (1) we initialize Ψ to be $\Psi = \frac{\beta^2 \cdot \tilde{y}}{32} - \max_{i \in A'} f(i)$, (2) we initialize the prices to be $p_i = \frac{\beta}{2} \cdot \sqrt{c_i \cdot \tilde{y}}$, (3) we replace the initialization of T to be a β-approximate demand set instead of an exact demand set (which can be done for $\beta = 1 - \frac{1}{e}$ using value oracle by Lemma B.1), and (4) we remove actions with negative utility from T, then the algorithm returns a set U with the following guarantee:

Lemma B.2. *Algorithm 4 runs in polynomial time with access to a demand and value oracle to f. If $\tilde{y} \le f(S^\star \cap A')$, then the set $U \subseteq A'$ that it returns satisfies*

$$g(U) \ge \max\left\{\frac{\tilde{y}}{128} - \frac{\max_{i \in A'} f(\{i\})}{4}, 0\right\}.$$

ALGORITHM 4: Approximately optimal contract given an estimate of $f(S^\star)$

Data: An XOS function $f : 2^A \to \mathbb{R}_{\ge 0}$, costs $\{c_i\}_{i \in A}$, an estimate $\tilde{y}$, a parameter β
Result: A set $U \subseteq A'$
Let $\Psi = \frac{\beta^2 \cdot \tilde{y}}{32} - \max_{i \in A'} f(i)$.
For every $i \in A'$, let $p_i = \frac{\beta}{2} \cdot \sqrt{c_i \cdot \tilde{y}}$ ($p_i = \infty$ for $i \in A \setminus A'$)
Let T be a β-demand set with prices p_i over the set A'
while *Exists $i \in T$ such that $f(i \mid T \setminus \{i\}) < p_i$* **do**
  $T = T \setminus \{i\}$
end
if $0 < \Psi < f(T)$ **then**
  $U \leftarrow$ the output of Algorithm 1 on $(f, T, \Psi, \delta = \frac{1}{2})$
else
  $U \leftarrow \emptyset$
end
return U

Proof. If $\Psi \le 0$ then the algorithm returns the empty set, which satisfies the proof. Otherwise, if we denote by T^1 the β-demand set, and by T^2 the β-demand set after removing actions with negative utility, then we have

$$\begin{aligned}
f(T^2) &\ge f(T^2) - \sum_{i \in T^2} p_i \\
&\ge f(T^1) - \sum_{i \in T^1} p_i \\
&\ge \beta f(S^\star \cap A') - \sum_{i \in S^\star \cap A'} p_i \\
&= \beta f(S^\star \cap A') - \frac{\beta\sqrt{\tilde{y}}}{2} \cdot \sum_{i \in S^\star \cap A'} \sqrt{c_i} \\
&\ge \beta f(S^\star \cap A') - \frac{\beta}{2}\sqrt{f(S^\star \cap A') \cdot \tilde{y}} \\
&\ge \frac{\beta}{2} \cdot f(S^\star \cap A'), \qquad (6)
\end{aligned}$$

where the first inequality is since the prices are non-negative, the second inequality is since we remove actions with negative utilities, the third inequality since T^1 is an approximate demand set over the set A', the fourth inequality is by Lemma 3.3, and the last inequality holds because $\tilde{y} \le f(S^\star \cap A')$. Furthermore, as in T^2 we deleted all actions with negative utility, we have to have $f(i \mid T^2 \setminus \{i\}) \ge p_i$ for all $i \in T^2$.

Thus, since $\tilde{y} \le f(S^\star \cap A')$ by assumption and $\Psi = \beta^2\tilde{y}/32 - \max(0, \max_{i \in A'} f(\{i\})) < \beta^2\tilde{y}/2$ it holds that

$$f(T^2) \overset{(6)}{\ge} \frac{\beta}{2} \cdot f(S^\star \cap A') \ge \frac{\beta\tilde{y}}{2} > \Psi.$$

By Lemma 3.5, the set $U \subseteq T^2$ fulfills

$$f(U) \ge (1 - \delta)\,\Psi = \frac{\beta^2}{64}\tilde{y} - \frac{1}{2}\max_{i \in T^2} f(i) \ge \frac{\beta^2}{64}\tilde{y} - \frac{1}{2}\max_{i \in A'} f(i).$$

Furthermore, U fulfills

$$f(U) \le \Psi + \max_{i \in T^2} f(i) \le \frac{\beta^2\tilde{y}}{32} \qquad (7)$$

and for all $i \in U$ it holds

$$f(i \mid U \setminus \{i\}) \ge \delta f(i \mid T^2 \setminus \{i\}) = \frac{1}{2} \cdot f(i \mid T^2 \setminus \{i\}). \qquad (8)$$

Therefore, for all $i \in U$, we have

$$\begin{aligned}
f(i \mid U \setminus \{i\}) &\overset{(8)}{\ge} \frac{1}{2} \cdot f(i \mid T^2 \setminus \{i\}) \ge \frac{1}{2} \cdot p_i \\
&= \frac{\beta}{4} \cdot \sqrt{c_i\tilde{y}} = \sqrt{2\beta^2 c_i \frac{\tilde{y}}{32}} \overset{(7)}{\ge} \sqrt{2c_i f(U)}.
\end{aligned}$$

So, Lemma 3.4 implies $g(U) \ge \frac{1}{2}f(U)$. So, in combination,

$$g(U) \ge \frac{1}{2}f(U) \ge \frac{1}{2}\left(\frac{\beta^2}{64}\tilde{y} - \frac{1}{2}\max_{i \in A'} f(i)\right),$$

as claimed. □

We next show that Algorithm 3 that uses as a subroutine the modified algorithm (Algorithm 4) fulfills Theorem 3.1.

Proof of Theorem 3.1 for the case of submodular. If $f(S^\star \cap A') \le \frac{128\xi}{\beta^2} \cdot \max\{0, \max_{i \in A} g(\{i\})\}$ then by Lemma 3.2 the best single action (or the empty set) gives at least an approximation of

$\frac{\beta^2}{128\xi+\beta^2}$. Else, $A' \neq \emptyset$ (thus Ψ is well defined), and it holds that

$$f(S^\star \cap A') \geq \frac{128\xi}{\beta^2} \cdot \max\{0, \max_{i\in A'} g(\{i\})\} \geq \max_{i\in A'} f(\{i\})/2 \geq \Psi,$$

and $f(S^\star \cap A') \leq 2n \cdot \Psi$.

Let $j^\star$ be the unique j in which $x_j \leq f(S^\star \cap A') < \xi \cdot x_j$. Note that because of the bounds we just established we must have $j^\star \geq 0$ and $j^\star \leq \lceil \log_{\frac{1}{\xi}} 2n \rceil$. Now, since $\frac{1}{\xi} \cdot f(S^\star \cap A') \leq x_j \leq f(S^\star \cap A')$,

$$\begin{aligned} g(U^{(j^\star)}) &\geq \frac{\beta^2 x_{j^\star}}{128} - \frac{\max_{i\in A'} f(\{i\})}{4} \\ &\geq \frac{\beta^2 f(S^\star \cap A')}{128 \cdot \xi} - \frac{\max_{i\in A} g(\{i\})}{2} \\ &\geq \beta^2 f(S^\star \cap A') \cdot \left(\frac{1}{128 \cdot \xi} - \frac{1}{256 \cdot \xi}\right) \\ &\geq \beta^2 \frac{\frac{128\xi}{\beta^2} \cdot g(S^\star)}{\frac{128\xi}{\beta^2}+1} \frac{1}{256 \cdot \xi} = \frac{\beta^2 \cdot g(S^\star)}{256\xi + 2 \cdot \beta^2}, \end{aligned}$$

where the first inequality is by Lemma B.2, the second inequality is since for every $i \in A'$ it holds that $\frac{c_i}{f(\{i\})} \leq \frac{1}{2}$ and thus $g(\{i\}) = f(\{i\})(1 - \frac{c_i}{f(\{i\})}) \geq f(\{i\})/2$, the third and fourth inequalities are since we are considering the case that $f(S^\star \cap A') > \frac{128\xi}{\beta^2} \cdot \max\{0, \max_{i\in A} g(\{i\})\}$. This gives an approximation of $\frac{\beta^2}{256\xi+2\cdot\beta^2}$. □

C INAPPROXIMABILITY FOR XOS REWARD FUNCTIONS

In this appendix, we provide details for the proof of Theorem 4.9. The claims, observations, and lemmas below refer to the function f and f_T defined in the proof of the theorem.

Claim C.1. *For every T, the function f_T is XOS.*

Proof. Consider the set of additive functions where there is for every agent $i \in A$ an additive function a_i such that $a_i(i) = 1 + \frac{3}{n}$, and $a_i(j) = \frac{3}{n}$ for every $j \neq i$, and there is another additive function a'_i, where $a'_i(i) = 0.5 + \frac{4}{n}$, and $a'_i(j) = \frac{4}{n}$ for every $j \neq i$. It is easy to verify that the maximum over $\{a_i, a'_i\}_{i\in A}$ is the function f. To create f_T, we add another additive function a_T, where $a_T(i) = \frac{5}{n} \cdot \mathbb{1}[i \in T]$. It holds that $a_T(T) = \frac{n+2}{2} \cdot \frac{5}{n} = 2.5 + \frac{5}{n} = f_T(T)$, and it holds that $a_T(S) \leq f_T(S)$ for every $S \neq T$, since (1) If $|S| > n/2 + 1$, then $a_T(S) \leq a_T(T) = 2.5 + \frac{5}{n} \leq f_T(S)$. (2) If $|S| = n/2 + 1$ and $S \neq T$, then $a_T(S) \leq \frac{n}{2} \cdot \frac{5}{n} \leq f_T(S)$. (3) If $|S| < n/2 + 1$, then $a_T(S) \leq |S| \cdot \frac{5}{n} \leq f_T(S)$. Thus f_T is the maximum of $\{a_i, a'_i\}_{i\in A} \cup \{a_T\}$. □

Lemma C.2. *We have that $g_T(T) \geq \frac{5}{4}$ and $g_T(S) \leq \frac{11}{10}$ for all $S \neq T$.*

Proof. First note that for every $i \in T$, $f_T(i \mid T \setminus \{i\}) = \frac{5}{n}$. Thus, $g_T(T) = f_T(T)\left(1 - \sum_{i\in T} \frac{c_i}{f_T(i|T\setminus\{i\})}\right) = (2.5 + \frac{5}{n}) \cdot (1 - \frac{1}{2}) \geq \frac{5}{4}$. Next, consider any $S \neq T$. We can distinguish the following cases.

- If $|S| \leq 1$, then $g_T(S) \leq f_T(S) \leq 1 + \frac{3}{n}$.
- If $1 < |S| \leq \frac{n}{2}$, then $f_T(i \mid S \setminus \{i\}) = \frac{3}{n}$, thus
$$\begin{aligned} g_T(S) &= f_T(S)\left(1 - \sum_{i\in S} \frac{c_i}{f_T(i \mid S \setminus \{i\})}\right) \\ &= \left(1 + \frac{3 \cdot |S|}{n}\right) \cdot \left(1 - \frac{5 \cdot |S|}{3 \cdot (n+2)}\right) \leq \frac{11}{10}, \end{aligned}$$
where the last inequality holds for large enough n and any $|S| \leq \frac{n}{2}$.
- If $|S| \geq \frac{n}{2} + 1$ and $S \neq T$, then $f_T(i \mid S \setminus \{i\}) \leq \frac{4}{n}$ for all $i \in S$, thus
$$\begin{aligned} g_T(S) &\leq \left(1/2 + \frac{4 \cdot |S|}{n}\right)\left(1 - \sum_{i\in S} \frac{c_i}{f_T(i \mid S \setminus \{i\})}\right) \\ &\leq \left(1/2 + \frac{4 \cdot |S|}{n}\right)\left(1 - \frac{5 \cdot |S|}{4 \cdot (n+2)}\right) \leq \frac{11}{10}, \end{aligned}$$
where the last inequality holds for large enough n and any $|S| \geq \frac{n}{2} + 1$.

Hence for any $S \neq T$ it holds that $g_T(S) \leq \frac{11}{10}$. □

Lemma C.3. *For every vector of prices $p = (p_1, \ldots, p_n)$, the set of $\{T^\star \mid D(f,p) \neq D(f_{T^\star}, p)\}$ is at most of size n^{29}, where $D(f,p)$ (resp. $D(f_{T^\star}, p)$) is the demand set of function f (resp. $f_{T^\star}$) with respect to prices p.*

For simplicity of the proof, we assume that when there are multiple sets in demand, the tie breaking is consistent across all f_T.

Proof. Let $S_p = \{i \mid p_i \leq \frac{3.25}{n}\}$, and $B_p = \{i \mid p_i \geq \frac{3.5}{n}\}$.

Claim C.4. *If $|B_p| \leq n/2-3$, then for every T it holds that $D(f_T, p) = D(f, p)$.*

Proof. Since f_T and f disagree only on the value of T, it is sufficient to show that $D(f_T, p) \neq T$. Assume towards contradiction that $D(f_T, p) = T$, then, because $|A \setminus B_p| \geq n/2+3$ and $|T| = n/2+1$, the set $\Delta := A \setminus B_p \setminus T$ is of size at least 2. Let x, y be four arbitrary different elements in Δ, and note that for these $p(x) < \frac{3.5}{n}$ and $p(y) < \frac{3.5}{n}$. Now $f_T(T \cup \{x,y\}) - p(T \cup \{x,y\}) \geq \frac{7}{n} + f_T(T) - p(T) - p(x) - p(y) > f_T(T) - p(T)$, which contradicts that T is the demand set. □

Claim C.5. *If $|S_p| \leq n/2-8$, then for every T it holds that $D(f_T, p) = D(f, p)$.*

Proof. Since f_T and f disagree only on the values of T, it is sufficient to show that $D(f_T, p) \neq T$. Assume towards contradiction that $D(f_T, p) = T$, then the set $\Delta := T \setminus S_p$ is of size at least 9. Let X be an arbitrary subset of Δ of size 9. It holds that

$$\begin{aligned} f_T(T \setminus X) - p(T \setminus X) &= f_T(T) - \frac{29}{n} - p(T) + p(X) \\ &\geq f_T(T) - p(T) + \frac{3.25 \cdot |X|}{n} - \frac{29}{n} \\ &> f_T(T) - p(T), \end{aligned}$$

which contradicts that T is the demand set. □

Claim C.6. *If $|S_p| > n/2-8$ and $|B_p| > n/2-3$, then for all $T \notin \{S \mid (|S| = n/2+1) \wedge (|S \setminus SP_p| \leq 14)\}$ it holds that $D(f_T, p) = D(f, p)$.*

Proof. Since f_T and f disagree only on the values of T, it is sufficient to show that $D(f_T, p) \neq T$. Assume towards contradiction that $D(f_T, p) = T$, it holds that the set $\Delta_1 := B_p \cap T$ is of size at least of size 5 since $|B_p \cap T| \geq |T \setminus S_p| - |A \setminus B_p \setminus S_p| \geq 14 - 9 = 5$, and the set $\Delta_2 := S_p \setminus T$ is of size at least 6 since $|S_p \setminus T| = |T \setminus S_p| - |T| + |S_p| \geq 14 - 8 = 6$. Let X be an arbitrary subset of Δ_1 of size 5 and let Y be an arbitrary subset of Δ_2 of size 5. It holds that

$$\begin{aligned} & f_T((T \setminus X) \cup Y) - p((T \setminus X) \cup Y) \\ = \; & f_T(T) - \frac{1}{n} - p(T) + p(X) - p(Y) \\ \geq \; & f_T(T) - p(T) + \frac{|X|}{4n} - \frac{1}{n} \\ > \; & f_T(T) - p(T), \end{aligned}$$

which contradicts that T is the demand set. □

By combining Claim C.4 and Claim C.5, a demand query can only reveal information about T if $|S_p| > n/2 - 8$ and $|B_p| > n/2 - 3$, and even then, by Claim C.6, a demand query cannot distinguish between T's not in $\{S \mid (|S| = n/2 + 1) \wedge (|S \setminus S_p| \leq 14)\}$. Now, the lemma follows since for every choice of S_p of size greater than $n/2 - 8$, the set $\{S \mid (|S| = n/2+1) \wedge (|S \setminus S_p| \leq 14)\}$ is at most of size n^{29}. (One can bound it by counting the options to select a set $S \setminus S_p$ of size at most 14 and then select a set $S_p \setminus S$ which is of size at most 15 since $|S_p \setminus S| = |S \setminus S_p| + |S_p| - |S| \leq 14 + n - |B_p| - (n/2 + 1) \leq 14 + n - (n/2 - 2) - (n/2 + 1) = 15$.) □

REFERENCES

[1] Tal Alon, Paul Dütting, and Inbal Talgam-Cohen. 2021. Contracts with Private Cost per Unit-of-Effort. In *Proc. ACM EC 2021*. 52–69.

[2] Itai Ashlagi, Wanyi Li, and Irene Lo. 2022. Simple and Approximately Optimal Contracts for Payment for Ecosystem Services. *Management Science* (2022). Articles in advance, published online.

[3] Sepehr Assadi, Thomas Kesselheim, and Sahil Singla. 2021. Improved Truthful Mechanisms for Subadditive Combinatorial Auctions: Breaking the Logarithmic Barrier. In *Proc. ACM-SIAM SODA 2021*. 653–661.

[4] Moshe Babaioff, Michal Feldman, and Noam Nisan. 2006. Combinatorial agency. In *Proc. ACM EC 2006*. 18–28.

[5] Moshe Babaioff, Michal Feldman, and Noam Nisan. 2009. Free-Riding and Free-Labor in Combinatorial Agency. In *Proc. SAGT 2009*. 109–121.

[6] Moshe Babaioff, Michal Feldman, and Noam Nisan. 2010. Mixed Strategies in Combinatorial Agency. *Journal of Artificial Intelligence Research* 38 (2010), 339–369.

[7] Ashwinkumar Badanidiyuru, Shahar Dobzinski, and Sigal Oren. 2012. Optimization with demand oracles. In *Proceedings of the 13th ACM conference on electronic commerce*. 110–127.

[8] Yahav Bechavod, Chara Podimata, Zhiwei Steven Wu, and Juba Ziani. 2022. Information Discrepancy in Strategic Learning. In *Proc. ICML 2022*. 1691–1715.

[9] Curtis Bechtel, Shaddin Dughmi, and Neel Patel. 2022. Delegated Pandora's Box. In *Proc. ACM EC 2022*. 666–693.

[10] Kshipra Bhawalkar and Tim Roughgarden. 2011. Welfare Guarantees for Combinatorial Auctions with Item Bidding. In *Proc. ACM-SIAM SODA 2011*. 700–709.

[11] Yang Cai, Argyris Oikonomou, and Mingfei Zhao. 2022. Computing simple mechanisms: Lift-and-round over marginal reduced forms. In *Proc. ACM STOC 2022*. 704–717.

[12] Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. 2022. Designing Menus of Contracts Efficiently: The Power of Randomization. In *Proc. ACM EC 2022*. 705–735.

[13] Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. 2010. Multi-parameter mechanism design and sequential posted pricing. In *Proc. ACM STOC 2010*. 311–320.

[14] Yiling Chen and Fang-Yi Yu. 2021. Optimal Scoring Rule Design. *CoRR* abs/2107.07420 (2021). https://arxiv.org/abs/2107.07420

[15] George Christodoulou, Annamária Kovács, and Michael Schapira. 2016. Bayesian Combinatorial Auctions. *Journal of the ACM* 63, 2 (2016), 11:1–11:19.

[16] Shahar Dobzinski. 2021. Breaking the Logarithmic Barrier for Truthful Combinatorial Auctions with Submodular Bidders. *SIAM J. Comput.* 50, 3 (2021).

[17] Paul Dütting, Tomer Ezra, Michal Feldman, and Thomas Kesselheim. 2021. Combinatorial Contracts. In *Proc. IEEE FOCS 2021*. 815–826.

[18] Paul Dütting, Michal Feldman, Thomas Kesselheim, and Brendan Lucier. 2020. Prophet Inequalities Made Easy: Stochastic Optimization by Pricing Nonstochastic Inputs. *SIAM J. Comput.* 49, 3 (2020), 540–582.

[19] Paul Dütting, Thomas Kesselheim, and Brendan Lucier. 2020. An O(log log m) Prophet Inequality for Subadditive Combinatorial Auctions. In *Proc. IEEE FOCS 2020*. 306–317.

[20] Paul Dütting, Tim Roughgarden, and Inbal Talgam-Cohen. 2019. Simple versus Optimal Contracts. In *Proc. ACM EC 2019*. 369–387.

[21] Paul Dütting, Tim Roughgarden, and Inbal Talgam-Cohen. 2021. The Complexity of Contracts. *SIAM J. Comput.* 50, 1 (2021), 211–254.

[22] Yuval Emek and Michal Feldman. 2012. Computing optimal contracts in combinatorial agencies. *Theoretical Computer Science* 452 (2012), 56–74.

[23] Uriel Feige. 2009. On maximizing welfare when utility functions are subadditive. *SIAM J. Comput.* 39 (2009), 122–142. Issue 1.

[24] Michal Feldman, John Chuang, Ion Stoica, and Scott Shenker. 2007. Hidden-Action in Network Routing. *IEEE Journal on Selected Areas in Communications* 25, 6 (2007), 1161–1172.

[25] Michal Feldman, Hu Fu, Nick Gravin, and Brendan Lucier. 2013. Simultaneous auctions are (almost) efficient. In *Proc. ACM STOC 2013*. 201–210.

[26] Michal Feldman, Nick Gravin, and Brendan Lucier. 2015. Combinatorial Auctions via Posted Prices. In *Proc. ACM-SIAM SODA 2015*. 123–135.

[27] Hu Fu, Robert Kleinberg, and Ron Lavi. 2012. Conditional equilibrium outcomes via ascending price processes with applications to combinatorial auctions with item bidding.. In *EC*. Citeseer, 586.

[28] Sanford J. Grossman and Oliver D. Hart. 1983. An Analysis of the Principal-Agent Problem. *Econometrica* 51, 1 (1983), 7–45.

[29] Faruk Gul and Ennio Stacchetti. 1999. Walrasian equilibrium with gross substitutes. *Journal of Economic theory* 87, 1 (1999), 95–124.

[30] Guru Guruganesh, Jon Schneider, and Joshua R. Wang. 2021. Contracts under Moral Hazard and Adverse Selection. In *Proc. ACM EC 2021*. 563–582.

[31] Chris Harshaw, Moran Feldman, Justin Ward, and Amin Karbasi. 2019. Submodular Maximization beyond Non-negativity: Guarantees, Fast Algorithms, and Applications. In *Proc. ICML 2019*. 2634–2643.

[32] Bengt Holmström. 1979. Moral Hazard and Observability. *The Bell Journal of Economics* 10, 1 (1979), 74–91.

[33] Bengt Holmström. 1982. Moral Hazard in Teams. *The Bell Journal of Economics* 13 (1982), 324–340. Issue 2.

[34] Jon M. Kleinberg and Robert Kleinberg. 2018. Delegated Search Approximates Efficient Search. In *Proc. ACM EC 2018*. 287–302.

[35] Jon M. Kleinberg and Manish Raghavan. 2019. How Do Classifiers Induce Agents to Invest Effort Strategically?. In *Proc. ACM EC 2019*. 825–844.

[36] Benny Lehmann, Daniel Lehmann, and Noam Nisan. 2006. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior* 55, 2 (2006), 270–296.

[37] Wanyi Dai Li, Nicole Immorlica, and Brendan Lucier. 2021. Contract Design for Afforestation Programs. In *Proc. WINE 2021*. 113–130.

[38] Yingkai Li, Jason D. Hartline, Liren Shan, and Yifan Wu. 2022. Optimization of Scoring Rules. In *Proc. ACM EC 2022*. 988–989.

[39] Statista Research Department. 2021. Instagram influencer marketing spending worldwide from 2013 to 2020. https://www.statista.com/statistics/950920/global-instagram-influencer-marketing-spending/.

[40] Maxim Sviridenko, Jan Vondrák, and Justin Ward. 2017. Optimal Approximation for Submodular and Supermodular Optimization with Bounded Curvature. *Mathematics of Operations Research* 42, 4 (2017), 1197–1218.

[41] Vasilis Syrgkanis and Éva Tardos. 2013. Composable and efficient mechanisms. In *Proc. ACM STOC 2013*. 211–220.

[42] Andrew Chi-Chih Yao. 2015. An n-to-1 bidder reduction for multi-item auctions and its applications. In *Proc. ACM-SIAM SODA 2015*. 92–109.

Received 2022-11-07; accepted 2023-02-06