Multidimensional Budget-Feasible Mechanism Design

Rian Neogi* Kanstantsin Pashkovich* Chaitanya Swamy*

Abstract

In *budget-feasible mechanism design*, a buyer wishes to procure a set of items of maximum value from self-interested rational players. We have a nonnegative valuation function $v : 2^U \mapsto \mathbb{R}_+$, where U is the set of all items, where $v(S)$ specifies the value obtained from set S of items. The entirety of current work on budget-feasible mechanisms has focused on the *single-dimensional* setting, wherein each player holds a *single* item e and incurs a *private cost* c_e for supplying item e (and each item is held by some player).

We introduce *multidimensional budget feasible mechanism design*: the universe U is now partitioned into item-sets $\{G_i\}$ held by the different players, and each player i incurs a private cost $c_i(S_i)$ for supplying the set $S_i \subseteq G_i$ of items. A *budget-feasible mechanism* is a mechanism (i.e., an algorithm and a payment scheme) that is *truthful*, i.e., where players are incentivized to report their true costs, and where the total payment made to the players is at most some given budget B. The goal (as in the single-dimensional setting) is to devise a budget-feasible mechanism that procures a set of items of large value.

We obtain the first approximation guarantees for multidimensional budget feasible mechanism design.

Our contributions are threefold. First, we prove an impossibility result showing that the standard benchmark used in single-dimensional budget-feasible mechanism design, namely the algorithmic optimum OPT_{Alg} is inadequate in that no budget-feasible mechanism can achieve good approximation relative to this. We identify that the chief underlying issue here is that there could be a monopolist, i.e., a single player who contributes a large fraction of OPT_{Alg}, which prevents a budget-feasible mechanism from obtaining good guarantees. Second, we devise an alternate benchmark, OPT_{Bench}, that allows for meaningful approximation guarantees, thereby yielding a metric for comparing mechanisms. Third, we devise budget-feasible mechanisms that achieve *constant-factor approximation guarantees* with respect to this benchmark for XOS valuations. Our most general results pertain to XOS valuations and arbitrary cost functions, where we obtain a budget-feasible-in-expectation mechanism that runs in polytime given a demand oracle, and a universally budget-feasible mechanism. We also obtain polytime universally budget-feasible mechanisms for: (a) additive valuations and additive costs; and (b) XOS valuations and superadditive cost functions, assuming access to a variant of a demand oracle.

Our guarantees for XOS valuations also yield an $O(\log k)$-approximation for subadditive valuations (with respect to our benchmark), where k is the number of players.

*{rneogi,kpashkovich,cswamy}@uwaterloo.ca. Dept. of Combinatorics and Optimization, Univ. Waterloo, Waterloo, ON N2L 3G1. Supported in part by the NSERC Discovery grants of K. Pashkovich and C. Swamy.

1 Introduction

In *budget-feasible mechanism design*, a buyer wishes to procure a set of items of maximum value from self-interested rational players. We have a ground set U of items, a *valuation function* $v : 2^U \mapsto \mathbb{R}_+$ satisfying $v(\emptyset) = 0$, where $v(S)$ specifies the value obtained from set S of items, and a budget B. The *vast majority* of work on budget-feasible mechanisms has focused on the *single-dimensional setting*, wherein each player holds a *single, distinct* element e and incurs a *private cost* c_e for supplying item e. In order to incentivize players to reveal their true costs, the buyer needs to make suitable payments to the players. The utility of a player is then equal to the (payment received by it) $-$ (cost incurred by it). A mechanism, i.e., an algorithm and a payment scheme, is: (a) *truthful*, if each player maximizes her utility by revealing her true cost and thereby has no incentive to misreport her cost; and (b) *individually-rational*, if the utility of every truthful player is nonnegative. A *budget-feasible mechanism* is a truthful, individually rational (IR) mechanism where the total payment made to the players is at most the given budget B. The goal is to devise a budget-feasible mechanism that procures a set of items of large value, where we measure the quality of the mechanism by comparing its value returned against the *algorithmic optimum* $OPT_{\mathsf{Alg}} := OPT_{\mathsf{Alg}}(v, B, c) := \max\{v(S) : \sum_{e \in S} c_e \le B,\ S \subseteq U\}$, which denotes the maximum obtainable value if the costs were public information.

Budget feasible mechanisms were introduced by Singer [55] and have been extensively studied (see, e.g., [14, 56, 49, 12] and the references therein), but (as noted earlier) *almost exclusively* in the (above) single-dimensional setting. A few works consider a somewhat richer, albeit *still single-dimensional* setting called the level-of-service (LOS) setting, wherein a player can provide multiple units of an item (or levels of service), incurring the same cost for each unit supplied [21, 5, 45, 3]. This focus on single-dimensional settings is in stark contrast with other prominent mechanism-design problem domains, such as social-welfare maximization and profit-maximization in combinatorial auctions (see [46, 40]), or cost-sharing mechanism design [35, 27], which considers richer multidimensional settings involving more-expressive players (notwithstanding the significant challenges that arise in multidimensional mechanism design).

1.1 Our contributions and results

We *initiate the study of multidimensional budget feasible mechanism design*: each player i now owns a publicly-known set G_i of items, where these player-sets partition U and incurs cost $c_i(S_i)$ for supplying the set $S_i \subseteq G_i$ of items, where $c_i : 2^{G_i} \mapsto \mathbb{R}_+$ is i's *private cost function*. (Throughout, we use i to index the players, and e to index items in U.) We assume only that the c_is are monotone ($c_i(S) \le c_i(T)$ for all $S, T \subseteq G_i$), and normalized ($c_i(\emptyset) = 0$). The goal (as before) is to design a budget-feasible mechanism that maximizes the value of the procured set of items.[1] Let $\mathcal{C}_i$ denote the set of all possible player-i private cost functions, and $\mathcal{C} := \Pi_i \mathcal{C}_i$. Let $n = |U|$. Note that v, B, and $\{(G_i, \mathcal{C}_i)\}_i$ are public knowledge.

We obtain the first approximation guarantees for multidimensional budget feasible mechanism design.

Multidimensional mechanism design is in general substantially more challenging than single-dimensional mechanism design, mostly due to the fact that there is no simple and conveniently-leverageable characterization of truthfulness that is analogous to Myerson's monotonicity-based characterization of truthfulness in single-dimensional settings. With budget-feasible mechanism design, where payments also feature in the constraints, this challenge also manifests itself in another distinct (but related) way, namely, that there is a complicated relationship between truthfulness-inducing payments and the underlying algorithm[2] compared to the single-dimensional setting, where payments correspond to threshold values.

[1] Note that this model is rich enough to to capture even a *multidimensional* LOS setting, wherein $c_i(S_i)$ is a monotone (but not necessarily linear) function of $|S_i|$.

[2] In the multidimensional setting, there are characterizations of truthfulness based on cycle monotonicity and weak-monotonicity, and payments can be obtained by computing shortest paths in a certain graph, but these have been difficult to leverage.

Furthermore, multidimensional budget-feasible mechanism design poses an (orthogonal) modeling challenge, namely, the benchmark used in the single-dimensional setting, the algorithmic optimum, which now translates to $OPT_{\mathsf{Alg}}(v, B, c) := \max\{v(S) : \sum_i c(S \cap G_i) \le B,\ S \subseteq U\}$, turns out to be quite inadequate: *no budget-feasible mechanism can achieve any non-trivial approximation with respect to* OPT_{Alg}, even for additive valuations and additive cost functions. This impossibility result (see Theorem 1.1) extends to *budget-feasible-in-expectation* mechanisms, which are *truthful-in-expectation* mechanisms—i.e., truthful reporting maximizes the expected utility of each player—where the expected total payment is at most the budget B (and IR holds with probability 1). It also extends to the *Bayesian* setting, wherein there is a prior distribution from which players' types are drawn, and we compare the expected value of the mechanism and $\mathbb{E}\big[OPT_{\mathsf{Alg}}\big]$.

We say that a budget-feasible mechanism achieves approximation ratio α with respect to a benchmark Bmark, where $\alpha \ge 1$, if it always obtains value, or expected value in case of a randomized mechanism, at least $\mathsf{Bmark}(v, B, c)/\alpha$. We sometimes consider a natural "no-overbidding" assumption, which states that every item, by itself, constitutes a feasible solution: formally, for every i, every $c \in \mathcal{C}_i$, and every $e \in G_i$, we have $c_i(\{e\}) \le B$. This is without loss of generality in the single-dimensional (single-item and LOS) settings (as dropping a player i with $c_i > B$ does not impact truthfulness or approximation), but does impose a restriction on player cost functions in the multidimensional setting.[3]

Theorem 1.1 (**Impossibility results:** Informal versions of Theorems 3.1–3.3)**.** *(Recall that $n = |U|$.)*

(a) No deterministic budget-feasible mechanism can achieve any bounded approximation ratio relative to $OPT_{\mathsf{Alg}}(v, B, c)$.

(b) No budget-feasible-in-expectation mechanism can achieve approximation ratio better than n relative to $OPT_{\mathsf{Alg}}(v, B, c)$.

(c) ([21]) Under no-overbidding, even for the single-dimensional LOS *setting, no deterministic budget-feasible mechanism can achieve approximation ratio better than n, and no budget-feasible-in-expectation mechanism can achieve approximation ratio better than $O(\log n)$, relative to $OPT_{\mathsf{Alg}}(v, B, c)$.*

These impossibility results hold even when the valuation v and the c_is are additive functions. The lower bounds for budget-feasible-in-expectation mechanisms extend to Bayesian budget-feasible mechanisms.

(We include part (c) above mainly for the sake of comparison (when we do not assume no-overbidding) and completeness.[4]) We remark that the lower bounds mentioned above are *tight* for XOS valuations (see Section 8.2); this follows as a by-product of some of our results, and we elaborate upon this later.

Suitable benchmark. In light of the above impossibility results, a pertinent question that arises is: *can one come up with alternative suitable benchmarks that enable one to circumvent the above impossibility result and obtain meaningful approximation guarantees*? One of our contributions is to define a novel benchmark that answers this question *affirmatively*.

Before delving into our benchmark, we first note that the natural idea of comparing against the maximum value obtainable by a budget-feasible mechanism, fails. This is because for every input (v, B, c), one can always tailor a budget-feasible mechanism that obtains value $OPT_{\mathsf{Alg}}(v, B, c)$ on this particular input;[5] so this maximum-value benchmark coincides with OPT_{Alg}. Thus some care and insight is needed to define a suitable benchmark.

[3] In particular, with overbidding, one can capture a richer space of private inputs, namely the "unknown" setting where there is a *private* subset $Z_i \subseteq G_i$ of items that a player i can provide; with overbidding, observe that this can be easily encoded by player i reporting (a monotone) c_i with $c_i(\{e\}) > B$ for all $e \in G_i - Z_i$.

[4] In [3], this lower bound is bypassed in the LOS setting by making a strong "all-in" assumption. Under such a strong assumption, one can obtain $O(1)$-approximation relative to OPT_{Alg} even in our multidimensional settings; see Remark 8.3.

[5] Let $S^* \subseteq U$ be such that $v(S^*) = OPT_{\mathsf{Alg}}(v, B, c)$ and $S_i^* = S^* \cap G_i$. Consider the mechanism that on input d, returns $\bigcup_i T_i$, where $T_i = S_i^*$ if $d(S_i^*) \le c(S_i^*)$ and $\emptyset$ otherwise, and pays player i, $c(S_i^*)$, if $d(S_i^*) \le c(S_i^*)$ and 0 otherwise.

Roughly speaking, the impossibility results stem from the fact that there could be a single player i responsible for a large fraction of the total value, who can then act as a monopolist: a budget-feasible mechanism can be forced to spend the entire budget on player i, even if only one item from G_i is chosen, which leads to a poor approximation factor. There are two approaches for bypassing this issue: (1) restrict attention to inputs where no player is a monopolist (and still compare against OPT_{Alg}); or (2) come up with a new benchmark whose definition captures that there is no monopolist. We adopt the latter approach, as it has the benefit that it yields guarantees for *all* inputs. Given a vector $c = (c_1, \ldots, c_k)$ of player cost functions, define $c(S) := \sum_i c_i(S \cap G_i)$. Define

$$OPT_{\mathsf{Bench}}(v, B, c) \;:=\; \max_{S \subseteq U} \Big\{ \min_{i \in [k]} v(S - G_i) :\; c(S) \le B \Big\}.$$

This benchmark safeguards against a monopolist in that its objective function $\min_i v(S - G_i)$ degrades in the presence of a a monopolist.[6] An appealing aspect of OPT_{Bench} is that it is parameter-free. Observe also that for subadditive v (i.e., $v(S \cup T) \le v(S) + v(T)$ for all $S, T \subseteq U$), on inputs where there is no monopolist, OPT_{Bench} essentially coincides with OPT_{Alg}: if we have some $S^* \subseteq U$ with $v(S^*) = OPT_{\mathsf{Alg}}(v, B, c)$ satisfying $v(S^* \cap G_i) \le \varepsilon \cdot v(S^*)$ for all $i \in [k]$, then $OPT_{\mathsf{Bench}} \ge (1 - \varepsilon) OPT_{\mathsf{Alg}}$.

We discuss this benchmark further in Section 1.3, relating it also to other mechanism-design domains where the need for considering suitable banchmarks arises.

Approximation results. We devise budget-feasible mechanisms that achieve *constant-factor approximation guarantees with respect to the above OPT_{Bench} benchmark.* Unless otherwise stated, in the sequel, when we refer to the approximation factor of a mechanism, it is always with respect to the OPT_{Bench} benchmark, and without assuming no-overbidding. (As noted earlier, overbidding provides increased modeling power, allowing us to also model settings where player i's set of items is private.)

Note that since budget-feasibility imposes a condition on the payments of the mechanism, even the *existence* of budget-feasible mechanisms, *bereft of computational concerns*, that achieve a "good" approximation with respect to the above benchmark is not guaranteed.[7] This situation applies also to the single-dimensional setting (where we seek approximation with respect to OPT_{Alg}), and therefore even the development of good budget-feasible mechanisms *setting aside* computational considerations has been the focus of a significant body of work. For instance, Dobzinski et al. [28] explicitly raised the question of whether there exists an $O(1)$-approximation budget-feasible mechanism for subadditive valuations, which was answered affirmatively by Bei et al. [14], albeit in a very non-constructive fashion, and only very recently an explicit, but non-polytime, mechanism was obtained [53]; a polytime mechanism for subadditive valuations remains elusive. Similarly, for multidimensional cost-sharing mechanism design, the work of [27] focuses on proving the existence of good cost-sharing mechanisms, regardless of computational considerations.

Our main results are for *XOS valuations* (and subclasses), and are summarized in Table 1. A function v is XOS if it is the maximum of a collection of additive functions.

Our mechanisms access the valuation function v via a suitably generalized form of demand oracle, and leverage the VCG mechanism (see Theorem 2.2). (Recall that given $c \in \mathcal{C} := \Pi_i \mathcal{C}_i$, we define $c(S) := \sum_i c_i(S \cap G_i)$ for $S \subseteq U$.) In the multidimensional setting, since player-cost functions are not necessarily additive, it is natural to extend the notion of a demand oracle to consider general price functions: a demand

[6] For subadditive v, this is always larger than the alternate objective $v(S) - \max_i v(S \cap G_i)$. Also, it is important to remove the *maximum* contribution of any player towards $v(S)$. For instance, if in the alternate objective, we subtract the *average* contribution of the players and consider the objective $v(S) - \big(\sum_i v(S \cap G_i)\big)/k'$, where $k' = |\{i : S \cap G_i \ne \emptyset\}|$, then the benchmark again becomes too strong; this is because one can always pad an instance with a monopolist with dummy players that contribute little value and incur 0 cost, thereby increasing k' and driving the subtracted term to 0.

[7] In fact, in the multidimensional setting, the existence of even just a truthful (but not necessarily budget-feasible) mechanism that achieves a good approximation relative to OPT_{Alg} is not guaranteed; see Theorem 3.5. This in contrast to the single-dimensional setting, where the algorithm that returns an optimal solution is monotone and hence truthfully implementable.

Valuations	Costs	$O(1)$-approximation wrt.	Mechanism type	Oracle
XOS	Arbitrary	OPT_{Bench}	BF-in-expectation (Theorem 4.1)	Demand
		OPT_{Bench}	Universally BF (Theorem 5.6)	*
	Superadditive	OPT_{Bench}	Universally BF (Theorem 5.11)	Constrained demand
Additive	Additive	OPT_{Bench}	Universally BF (Theorem 5.8)	–
Submodular	Arbitrary	$OPT_{\mathsf{Bench}} - O\big(\max_i v(G_i)\big)$	Universally BF (Theorem 6.1)	Demand

Table 1: Summary of our results for XOS valuations, and subclasses. BF stands for budget-feasible. BF-in-expectation denotes truthful-in-expectation with expected total payment at most B, where IR holds with probability 1; universally BF means that truthfulness, IR, and "total payment $\leq B$", all hold with probability 1. All our results hold without assuming no-overbidding. The value obtained is at least $\frac{1}{O(1)}$ times the quantity listed under "$O(1)$ approximation wrt." Our mechanisms run in polytime given access to the specified oracle; the entry marked * requires a more involved oracle. For additive valuations and costs, the valuation and costs are explicitly given.

oracle for class $\mathcal{C} = \Pi_i \mathcal{C}_i$, takes $q \in \mathcal{C}$, $\kappa \in \mathbb{R}_+$ as input, and returns $\operatorname{argmax}_{S\subseteq U} \big(v(S) - \kappa \cdot q(S)\big)$. Some of our mechanisms utilize a constrained demand oracle, where we are also given a cap Val, and the oracle returns $\operatorname{argmax}\{v(S) - \kappa \cdot c(S) : S \subseteq U,\ v(S) \leq Val\}$. (We remark that we can also work with an oracle that returns $\operatorname{argmax}_{S\subseteq U}\big(\min(v(S), Val) - \kappa \cdot c(S)\big)$.)

Our most general results apply to XOS valuations and *arbitrary* (monotone, normalized) cost functions. We devise two types of mechanisms that both achieve $O(1)$-approximation (with respect to OPT_{Bench}). The first mechanism is *budget-feasible-in-expectation* and runs in polytime given a demand oracle (Theorem 4.1). The second mechanism (Theorem 5.6) satisfies a much stronger mechanism-design guarantee, *universal budget-feasibility*—wherein truthfulness, IR, and the budget constraint on total payment, hold with probability 1—but requires stronger oracle access. Note that this also demonstrates the existence of a good universally budget-feasible mechanism, even in the most general setting of arbitrary cost functions.

We next design *polytime* universally budget-feasible mechanisms for various special cases of interest (that achieve $O(1)$ approximation). For XOS valuations and *superadditive* cost functions, our mechanisms run in polytime given a constrained demand oracle (Theorems 4.8 and 5.11).

For *additive valuations and additive cost functions*, a constrained demand oracle corresponds to solving a knapsack problem, and we show that, we can (roughly speaking) solve related (scaled and rounded) knapsack problems optimally and thereby obtain polytime mechanisms (Theorems 4.10 and 5.8). (We do not require any oracle access here since the additive valuation and costs are now explicitly specified in the input.)

In presenting our universally budget-feasible mechanisms, we present them first in the simpler setting where we assume no-overbidding (Section 4.2), as this helps to illustrate some of the main ideas, and the arguments become much simpler. We then discuss how to drop this assumption in Section 5. (Recall that all the results in Table 1 hold without assuming no-overbidding.)

For the subclass of monotone *submodular valuations* and arbitrary cost functions, we design a polytime universally budget-feasible mechanism using demand oracles, with a weaker approximation guarantee (Theorem 6.1). Interestingly, this mechanism is not VCG-based: we do not perform a global VCG computation, but only perform VCG computations locally, for individual players (see Algorithm Submod-UniBF).

Our mechanisms for XOS mechanisms also yield $O(\log k)$*-approximation guarantees for subadditive valuations* (Section 7). Recall that k is the number of players. This essentially capitalizes on the idea that an XOS function can pointwise approximate a subadditive function. However: (a) we cannot use this as a black-box since a pointwise approximation of v by $\tilde{v}$ does not imply that the benchmarks $OPT_{\mathsf{Bench}}(v, B, c)$ and $OPT_{\mathsf{Bench}}(\tilde{v}, B, c)$ are close to each other; (b) the best pointwise-approximation of v by an XOS function can incur an $O(\log n)$-factor loss [14, 16], as v is defined over a universe of n items. Instead, we obtain our results by working with a pointwise $O(\log k)$-approximation of v that satisfies a property weaker than XOS, and utilizing this internally within our mechanisms in a suitable fashion.

We obtain two other types of results as a byproduct of our techniques. First, we also obtain tight approximation guarantees using demand oracles for the algorithmic problem of approximating $OPT_{\mathsf{Alg}}(v, B, c)$ in polynomial time (Section 8.1). Theorem 8.1 shows that when the c_is are superadditive, we can obtain a polytime $(2 + \varepsilon)$-approximation, for any constant $\varepsilon > 0$, even for subadditive valuations (i.e., we have $v(S \cup T) \leq v(S) + v(T)$ for all $S, T \subseteq U$), which is *tight*, even for additive cost functions [11].

Second, we can show that the lower bounds mentioned in our impossibility result (Theorem 1.1) are *tight* for XOS valuations (see Section 8.2). For instance, we obtain randomized budget-feasible mechanisms that achieve $O(\log n)$-approximation relative to $OPT_{\mathsf{Alg}}(v, B, c)$ assuming no-overbidding (Theorem 8.2). This is a consequence of the fact (see Section 1.2) that our mechanisms work with an estimate V_1 of our benchmark, and one can isolate OPT_{Alg} within a multiplicative factor of n (in fact, within an $O(\max_i |G_i|)$-factor). Notably, this also implies substantially more-general results even for the single-dimensional LOS setting mentioned at the start of the Introduction, which is the very special case where $c_i(S) = c_i|S|$ for $S \subseteq G_i$. For this setup (where no-overbidding is without loss of generality), prior work yields approximation factors of $O(\log n)$ [21] and $O(\max_i |G_i|)$ [3], assuming that v is additive across players and concave for a single player. Our results yield similar guarantees, but *we do not need additivity* across players; they apply to *any XOS valuation*, and to the much-richer *multidimensional* LOS *setting*, where $c_i(S)$ is a monotone (as opposed to linear) function of $|S|$.

Our work opens up the area of multidimensional budget-feasible mechanism design as an exciting well-motivated research direction. Our results show that, notwithstanding the substantial challenges faced in multidimensional mechanism design, one can obtain interesting guarantees in this domain. While we make substantial progress in this area, a variety of interesting questions remain, and we hope that our work will stimulate further work in this area.

1.2 Technical overview

We give an overview of our techniques, highlighting the salient ideas underlying our mechanisms and their analysis. We use $v(e)$, $c_i(e)$, and $c(e)$ to denote the respective function value for the singleton set $\{e\}$. Broadly speaking, our mechanisms for XOS valuations capitalize on two main ideas. (A) We can obtain a good estimate of the target value to aim for via random partitioning; and (B) we can leverage such an estimate to select a suitable subset via a suitable demand-oracle computation, which amounts to a VCG computation. These ideas have been shown to be useful for single-dimensional budget-feasible mechanism design [14, 53], but the multidimensional setting inevitably leads to various challenges, including a rather tricky technical difficulty that arises when we aim for guarantees without assuming no-overbidding.

(A) Random partitioning. The idea here is to estimate the target value from a suitable random subset of U, which is then discarded. In the single-dimensional setting, this has been utilized in prior work (see, e.g., [14, 2, 53]), where one randomly samples each element in U with probability 0.5, and one can argue that this well-estimates OPT_{Alg}. In the single-dimensional setting, sampling players or elements amounts to the same thing, but when players hold multiple items, the two versions differ, and one needs to be careful in selecting the right sampling procedure. To ensure truthfulness, we do not want any player i's item-set G_i to be straddled by the sample as then a player i whose items are still in "play" after the sample is discarded can influence the computation on the sample. Therefore, we pick a random subset $\mathcal{N}_1$ of players by selecting each player with probability 0.5, and use the items owned by these players $U_1 := \bigcup_{i \in \mathcal{N}_1} G_i$ to estimate the target value to aim for and discard U_1. However, we hit an immediate snag, namely that if a single player contributes a large fraction of OPT_{Alg} (i.e., is a monopolist), then such a sample will be quite noisy and not yield a good estimate.

As shown by our impossibility result, this is a *real* issue that precludes good approximation with respect to OPT_{Alg} (and not simply an artifact of random sampling). Crucially (and conveniently) however, when we move to the OPT_{Bench} benchmark, this no longer presents a difficulty, as this benchmark specifically safeguards against a monopolist.

Specifically, we can argue (see Lemma 2.4 and Corollary 2.5) that this random sampling: (i) yields an $O(1)$-factor estimate V_1 of OPT_{Bench}, i.e., $V_1 = \Omega(OPT_{\mathsf{Bench}})$ with constant probability; and (ii) the set $U_2 = U - U_1$ also contains a good approximation to OPT_{Bench}. Moreover, we show that for XOS valuations, it also holds that, for any $j = 1, 2$, the optimal value of the LP-relaxation of the algorithmic problem on U_j yields a good estimate for OPT_{Bench} (see Lemma 2.6).

It is worth noting that the definition of our benchmark OPT_{Bench} to suitably account for a monopolist, is key here to showing that random sampling remains an effective tool that can be leveraged in the multidimensional setting.

(B) Leveraging the estimate V_1 via suitable demand oracles. Given the estimate V_1, we show how to exploit the VCG mechanism—the prototypical truthful mechanism for social-welfare maximization—in a careful fashion via suitable demand oracles. One insight to emerge from the work of [53] on the single-dimensional setting is that computing $S^* = \operatorname{argmax}_{S \subseteq U_2} \left\{ v(S) - \frac{\lambda V_1}{B} \cdot c(S) \right\}$ via a demand oracle, yields suitable payments. The argument therein is based on the monotonicity condition that characterizes truthfulness in single-dimensional settings, where payments correspond to threshold values, and they show that with an XOS valuation v, if we prune S^* to a set T with $v(T) \leq \lambda V_1$, then the payments satisfy the budget constraint.

The above demand-oracle computation is a VCG computation: it amounts to minimizing an affine function of players' costs. This is promising, since it implies that there are payments that can be combined with the algorithm to obtain a truthful mechanism (see Theorem 2.2 and (VCG)). However, these payments may not satisfy the budget constraint, even in the single-dimensional setting, which is why [53] need the postprocessing pruning step.

The pruning operation is however quite problematic and no longer works in the multidimensional setting, even if we are careful and select or discard the entire set $S^* \cap G_i$ of a player i. The crux of the issue is that, since players own multiple items, there is a much-richer space of possibilities in terms of how a player's cost function can influence S^* (and hence, the final outcome): a player who gets discarded by pruning can lie so that a different subset from G_i is included in S^* and she is no longer discarded, and the VCG payment then provides her with positive utility.[8] This makes pruning S^* and discarding items and/or players quite problematic. Indeed, in general, composing procedures (e.g., VCG + postprocessing) while maintaining truthfulness is quite challenging in multidimensional settings, due to the fact that one has much less control on how a player's reported cost can affect the output of a procedure, and hence the overall output.

We avoid this issue altogether by eliminating pruning. Instead, we use a *constrained demand oracle* that only considers sets with $v(S) \leq \lambda V_1$; that is, we compute $S^* = \operatorname{argmax}_{S \subseteq U_2} \left\{ v(S) - \frac{\lambda V_1}{B} \cdot c(S) : \ v(S) \leq \lambda V_1 \right\}$. This still corresponds to a VCG computation, and we can again use VCG to infer payments that ensure truthfulness. Also, when v is XOS, due to the upper bound on $v(S^*)$, budget-feasibility easily follows.

The proof of approximation guarantee requires one additional idea to show that $v(S^*) = \Omega(OPT_{\mathsf{Bench}})$. Due to our random sampling, we know that, with constant probability, U_2 contains some set T_2^* with $c(T_2^*) \leq B$, and $v(T_2^*) = \Omega(OPT_{\mathsf{Bench}})$; for instance, the maximum-value set in U_2 with cost at most B satisfies

[8] In contrast, in the one-item-per-player setting, the demand-set computation is immune to a player $i \in S^*$ in the sense that i cannot lie and cause a different demand-set to be computed that still contains i.

this. However, we may have $v(T_2^*) > \lambda V_1$, and so it is unclear how to utilize this to lower bound $v(S^*)$. The key is to argue that we can extract a suitable subset of T_2^* that can act as a witness for lower bounding $v(S^*) - \frac{\lambda V_1}{B} \cdot c(S^*)$: we show that we can always find $T \subseteq T_2^*$ such that $v(T)$ is roughly λV_1, and for which $c(T)$ is bounded away from B, say is at most $B/2$ (see Lemmas 2.7 and 2.8). This step incurs an additive loss bounded by $O\big(\max_{e \in T_2^*} v(e)\big)$ for superadditive cost functions, and $O\big(\max_i v(T_2^* \cap G_i)\big)$ for general cost functions. With superadditive costs, this loss is not a problem *when we assume no-overbidding*, since we can "recover" this by returning the maximum-value element in U with some constant probability. Without no-overbidding however, this poses a rather tricky issue, and one needs to come up with novel ideas to help offset this loss (as discussed below).

Putting everything together, with no-overbidding, our mechanisms for XOS valuations consist essentially of two steps: (1) random partitioning to estimate OPT_{Bench}; and (2) using a constrained demand-oracle computation to obtain S^*, which is tailored to ensure that the corresponding VCG payments yield a budget-feasible mechanism. The chief component in the analysis, and the differences in the various mechanisms, lie in the proof of approximation (since, as noted above, VCG payments easily ensure truthfulness and satisfy the budget constraint).

The above template yields universally budget-feasible mechanisms using a constrained demand oracle. We also show that using a (standard) demand oracle, one can obtain a budget-feasible-in-expectation mechanism (Algorithm XOS-BFInExp in Section 4.1). Here, we consider an *LP-relaxation for a constrained-demand oracle* (CDLP(A)), which we argue can be solved efficiently using a demand oracle. Viewing the LP-solution as a distribution, and again using VCG payments, we obtain a budget-feasible-in-expectation mechanism.

An important takeaway from our constructions and techniques is that the *VCG mechanism can be suitably adapted and controlled* to obtain payments that satisfy the budget condition.

Dropping no-overbidding (Section 5). As discussed above, one incurs an additive loss—$O\big(\max_{e \in T_2^*} v(e)\big)$ with superadditive costs, and a worse loss of $O\big(\max_{i \in \mathcal{N}_2} v(T_2^* \cap G_i)\big)$ with general costs—in arguing the existence of a good set $T \subseteq T_2^*$ with $v(T) \leq \lambda V_1$, $c(T) \leq B/2$. The question that arises is: *how do we offset this loss*?

While this may seem like a benign issue, it actually presents a serious obstacle. Let $e^* = \operatorname{argmax}\{v(e) : e \in U,\ c(e) \leq B\}$ and $v_{\max} = v(e^*)$. Also, let $\mathsf{opt}_i = \max_{S \subseteq G_i}\{v(S) : c_i(S) \leq B\}$ for $i \in \mathcal{N}$, and $\mathsf{opt}^* = \max_{i \in \mathcal{N}} \mathsf{opt}_i$; clearly $\mathsf{opt}^* \geq v_{\max}$. The additive loss incurred is bounded by opt^*, so a natural thought would be to return the set corresponding to opt^* with some probability; or, with superadditive costs, to return e^* with some probability. However, observe that, even for superadditive costs, and unlike the situation with no-overbidding, returning e^* does not yield a truthful mechanism: the player i owning e^* can benefit by *overbidding* on e^* and cause the mechanism to choose a lower-cost element from G_i. The same issue arises with returning the set corresponding to opt^*. More tellingly, and this points to the trickiness of this question, the impossibility result in Theorem 1.1 (a) implies that *no deterministic truthful mechanism can obtain value* $\Omega(v_{\max})$, as this would yield an $O(n)$-approximation relative to OPT_{Alg}.

To circumvent this issue, we develop novel (non-VCG-based) budget-feasible mechanisms (Mechanisms 2ndOpt, 2ndOpt-Poly, and 2ndOpt-CDemd) to serve as a kind of "proxy" for the "return set-corresponding-to-opt^* mechanism" (which is not truthfully implementable), which satisfy the following guarantee: they obtain expected value proportional to the *second-largest* opt_i *value*, which we denote by $\mathsf{opt}^{(2)}$ (Theorems 5.2, 5.7, and 5.9). Mechanism 2ndOpt has a particularly clean description: it carefully selects a player $\widehat{i}$ with $\mathsf{opt}_{\widehat{i}} = \mathsf{opt}^*$ and returns a minimum-cost set from $G_{\widehat{i}}$ whose value is at least $\mathsf{opt}^{(2)}$; Mechanisms 2ndOpt-Poly and 2ndOpt-CDemd build upon this idea to ensure *polytime computation* for additive valuations and additive costs, and superadditive costs with a constrained demand oracle, respectively.

The $\Omega\big(\mathsf{opt}^{(2)}\big)$-value guarantee that we obtain lies in a nice sweet spot: it is weak enough to not be precluded by our impossibility result, and yet is strong enough that we *can* still exploit this to recoup the additive loss incurred, provided that we can ensure that $\max_{i \in \mathcal{N}_2} v(T_2^* \cap G_i) \leq \mathsf{opt}^{(2)}$. We show that we can rework the *analysis* to achieve this, by choosing $T_2^* \subseteq U_2$ appropriately. The upshot is that, if we now run the random-sampling based mechanism with suitable probability p, and Mechanism 2ndOpt with probability $1-p$, then we obtain a good approximation relative to our benchmark (without assuming no-overbidding).

Submodular and subadditive valuations. Our mechanism for monotone, submodular valuations (Section 6) departs from the above template in that after obtaining an estimate V_1, it does not use a VCG mechanism. Instead, we use a greedy algorithm and process the players in a sequence, and for each player i select a suitable subset from $U_2 \cap G_i$ via a demand oracle. Thus, we perform "local" VCG computations but the overall mechanism is not VCG-based.

For subadditive valuations (Section 7), we utilize our mechanisms for XOS valuations, by working with a suitable pointwise-approximation of the subadditive valuation v in the demand-set computation. It is well known that any subadditive valuation can be pointwise-approximated by an XOS function within an $O(\log n)$-factor; but utilizing this this would only yield an $O(\log n)$-approximation. Instead, we observe that our mechanisms for XOS valuations work when the valuation function satisfies a limited fractional-cover property (see Section 2), namely, that for any $S \subseteq U$, every fractional cover $\{x_T\}_{T \subseteq S}$ of S by *player-respecting sets of S* has value at least $v(S)$, where $T \subseteq S$ is player-respecting if $T \cap G_i \in \{\emptyset, S \cap G_i\}$ for all i. One can show that a subadditive valuation can be pointwise-approximated within an $O(\log k)$-factor by a function satisfying this player-respecting fractional-cover property, and this yields $O(\log k)$-approximation factors for subadditive valuations.

1.3 Related work

As mentioned earlier, to our knowledge, all of the work on budget-feasible mechanism design has solely considered single-dimensional settings, and the vast majority of it has focussed on the setting where each player owns a single item. Following the work of Singer [55], which introduced budget-feasible mechanism design in the single-item setting, there has been much work on developing budget-feasible mechanisms for different types of valuation functions, with approximation factors comparing the value returned by the mechanism to the algorithmic optimum OPT_{Alg}. The prominent classes of valuation classes considered are submodular valuations [23, 42, 1, 2, 41, 12], XOS valuations [14, 15, 53], and subadditive valuations [28, 14, 15, 53]. For submodular and XOS valuations, the mechanism-design guarantees qualitatively match the guarantees known for the algorithmic problem. For subadditive valuations, there is a gap: Bei et al. [14] showed that an $O(1)$-approximation mechanism exists, and an explicit exponential-time mechanism was devised by [53], but obtaining a polytime $O(1)$-approximation mechanism using demand oracles remains an intriguing open question.

Some work [21, 5, 45, 3] has considered a richer, but still single-dimensional, level-of-service (LOS) setting, wherein a player can provide multiple levels of service (or multiple units of an item) and incurs the same incremental cost for each additional unit provided. Typically, one assumes that the valuation function v is additive across players, and is a linear or concave function of each player's provided level of service. In this setup, approximation factors of $O(\log n)$ and $O(\max_i |G_i|)$ were obtained by [21, 3] respectively; [21] also considered subadditive valuations and obtained an $O\big(\frac{\log^2 n}{\log \log n}\big)$-approximation. Another line of work considers the divisible-item setting, wherein one can buy a fraction of an item from a seller, and a related large-market assumption [4, 13, 42], which assumes that a single player has a negligible "effect" on the overall value. The latter assumption is similar in spirit to what our benchmark aims to capture, but the goal in these works is quite different. They still examine the single-dimensional setting, and the goal is to derive improved approximation guarantees (with respect to OPT_{Alg}) under this assumption. A natural extension of the large-market assumption to the multidimensional setting would be to assume that $v(G_i) \leq \varepsilon \cdot OPT_{\mathsf{Alg}}(v, B, c)$ for all $i \in \mathcal{N}$, for some parameter $\varepsilon < 1$. This precludes a monopolist by design, and so an alternate approach would have been to restrict attention to such inputs, but compare against OPT_{Alg}. As discussed earlier, the benefit of coming up with an alternate benchmark that applies to all inputs (as we do) is that one can then obtain guarantees for *all* inputs. Moreover, a guarantee relative to OPT_{Bench} is stronger, in the sense that any such guarantee translates to a guarantee relative to OPT_{Alg} for inputs satisfying the large-market assumption, as we have $OPT_{\mathsf{Bench}} \geq (1-\varepsilon)OPT_{\mathsf{Alg}}$ for such inputs.

The need for considering suitable alternate benchmarks invites comparison to the areas of prior-free profit maximization [36, 37] and frugal mechanism design [6, 57], where similar considerations arise. The issues we encounter in multidimensional budget-feasible mechanism design are similar in flavor (but the details differ) to those arising in prior-free profit-maximization. There as well, the need for alternate benchmarks arises because the natural comparison point proves to be too strong due to the presence of a single "dominant" player: one cannot obtain any non-trivial guarantees with respect to the optimal fixed-price revenue $\mathcal{F}$, as this could extract all its revenue from a single winner [36]. One therefore considers the $\mathcal{F}^{(2)}$-benchmark [36, 37] that hard-codes that there are at least two winners, which is perhaps closest in spirit as our benchmark. Frugal mechanism design considers a similar setup as budget-feasible mechanism design, where a buyer seeks to procure a suitable set from self-interested players. But any feasible set suffices, and the buyer's goal is to minimize the total payment made to the players. There is no clear benchmark here to compare the payment of a truthful mechanism, and a variety of benchmarks have been proposed in the literature, such as the cost of the second-cheapest set [6, 57], and quantities that are, loosely speaking, motivated by considering equilibria of first-price auctions [43, 33, 39], and various mechanisms have been devised that obtain good guarantees relative to these benchmarks [43, 33, 44, 22, 39]. With the sole exception of [52] who consider multidimensional vertex cover, all of this work focusses on single-dimensional problems. Finally, we remark that removing the maximum contribution from a player in the definition of OPT_{Bench} bears cosmetic similarity to the approximate envy-free fairness notions EF1 [50] and EFX [20].

The setting of cost-sharing mechanisms provides an interesting comparison point with budget-feasible mechanism design, being another example of a prominent domain where: (1) prices feature both in truthfulness and the constraints; (2) even the existence of mechanisms satisfying the desirable criteria (truthfulness, cost-recovery, and good approximation of the social-cost objective [54]) is not a given; and (3) most work has investigated only the single-dimensional setting. Unlike budget-feasible mechanism design, there has been some prior work that has explored multidimensional cost-sharing mechanism design, both in a combinatorial setting [35, 27, 17] that is similar in spirit to the setting we consider, as also in a level-of-service setting [51, 35]. While the technical aspects are quite different, it is worth pointing out that Dobzinski and Ovadia [27] also leverage VCG in novel ways, developing mechanisms that significantly advance the realm of problems for which one can obtain good cost-sharing mechanisms, and they focus on demonstrating the existence of good mechanisms bereft of computational considerations.

Multidimensional mechanism design has been most prominently considered in the setting of combinatorial auctions, both in the context of social-welfare maximization (see, e.g., [46, 48, 29, 8, 7]), and profit or revenue maximization, (see, e.g. [40, 30, 18, 19, 9, 31, 32]), where one often considers the goal of obtaining a good approximation via a simple mechanism. As noted earlier, one key source of difficulty in multidimensional mechanism design, compared to the single-dimensional setting, is the lack of a convenient characterization of truthfulness. Although characterizations based on cycle monotonicity and weak-monotonicity are known for truthfulness, and payments can be obtained by computing shortest paths in a certain graph, these have found quite scant application in the design of multidimensional truthful mechanisms; see [47, 10] for some exceptions. Instead, VCG-based mechanisms (leading to the MIDR approach) [48, 26, 25, 29] and posted-price mechanisms [34, 30], where one offers take-it-or-leave-it prices, have been the chief means for obtaining truthful mechanisms.

2 Preliminaries

We use $\mathbb{R}_+$ and $\mathbb{Z}_+$ to denote the set of nonnegative reals, and nonnegative integers respectively. For an integer $n \geq 1$, let $[n]$ denote $\{1, \ldots, n\}$. Let k be the number of players, and $\mathcal{N} = [k]$ denote the set of all players. Let $n = |U|$. Throughout, we use i to index players, and e to index items. Recall that: (a) $G_i \subseteq U$ is the publicly-known set of items owned by player i; (b) the G_is partition U; (c) $\mathcal{C}_i$ is the publicly-known collection of possible player-i cost functions; and (d) $\mathcal{C} = \Pi_{i=1}^{k} \mathcal{C}_i$. Given a cost-function vector $c \in \mathcal{C}$, we define $c(S) := \sum_{i \in [k]} c_i(S \cap G_i)$ for any $S \subseteq U$. Throughout, $OPT_{\mathsf{Alg}}(v, B, c) := \max\{v(S) : S \subseteq U,\ c(S) \leq B\}$ denotes the algorithmic optimum. It will be convenient to introduce the notation $v_{-1}(S) := \min_{i \in [k]} v(S - G_i)$ for $S \subseteq U$. Recall that we utilize the following benchmark

$$OPT_{\mathsf{Bench}}(v, B, c) \ := \ \max_{S \subseteq U} \Big\{ \min_{i \in [k]} v(S - G_i) : \ c(S) \leq B \Big\} \ = \ \max_{S \subseteq U} \Big\{ v_{-1}(S) : \ c(S) \leq B \Big\}.$$

To avoid cumbersome notation, we will frequently drop v, B from the arguments, as these will usually be fixed and clear from the context; we also drop c when this is clear from the context. For a singleton set $\{e\}$, we use $v(e)$, $c_i(e)$ and $c(e)$ to denote the respective function value on that set.

Valuation functions. Let $v : 2^U \mapsto \mathbb{R}_+$ be a valuation function. We will always assume that v is normalized, i.e., $v(\emptyset) = 0$. We say that v is *monotone*, if $v(S) \leq v(T)$ whenever $S \subseteq T \subseteq U$. We consider various classes of valuation functions. We say that v is:

- *additive* (or *modular*), if there exists some $a \in \mathbb{R}^U$ such that $v(S) = a(S)$ for all $S \subseteq U$. Note that an additive valuation is monotone iff it is nonnegative.
- *submodular*, if $v(S) + v(T) \geq v(S \cap T) + v(S \cup T)$ for all $S, T \subseteq U$.
- *XOS*, if v is the maximum of a finite collection of additive functions, i.e., there exist $a^1, \ldots, a^k \in \mathbb{R}^U$ such that $v(S) = \max_{i \in [k]} a^i(S)$ for all $S \subseteq U$. Note that we allow the additive functions to be negative, and this allows us to capture non-monotone XOS functions. The above definition is equivalent to saying that for every $S \subseteq U$, there exists some $w \in \mathbb{R}^U$ such that $v(S) = w(S)$ and $v(T) \geq w(T)$ for all $T \subseteq S$; we say that w *(or the corresponding additive valuation) supports* S. XOS valuations are also called *fractionally subadditive* valuations, and can be equivalently defined in terms of fractional covers. A *fractional cover* of a set $S \subseteq U$ is a collection $\{\mu_T\}_{T \subseteq S}$ such that $\sum_{T \subseteq S: e \in T} \mu_T = 1$ for all $e \in S$, and its value is defined as $\sum_{T \subseteq S} v(T)\mu_T$. We say that v is fractionally subadditive, if for every $S \subseteq U$, every fractional cover of S has value at least $v(S)$. Using LP duality, it is not hard to infer that this is equivalent to stating that every $S \subseteq U$ has a supporting additive valuation.
  When v is monotone and XOS, we can assume that v is the maximum of a collection of nonnegative additive valuations, and we can relax the fractional-cover condition to the inequality $\sum_{T \subseteq U: e \in T} \mu_T \geq 1$ for all $e \in S$. This is because we can always ensure that equality holds here by dropping elements from sets if needed, and with a monotone valuation, this does not increase the value of the fractional cover.
- *subadditive*, if $v(S \cup T) \leq v(S) + v(T)$ for all $S, T \subseteq U$.

It is well known that additive valuations are a strict subclass of submodular valuations, which in turn form a strict subclass of both XOS and subadditive valuations. Also, *monotone* XOS valuations form a strict subclass of monotone subadditive valuations.

Oracles used for accessing v. Our mechanisms chiefly utilize two types of oracles for accessing the valuation v, in addition to a value oracle: a (suitably generalized) demand oracle and a constrained demand oracle. As noted earlier, since in the multidimensional setting, we work with player-cost functions that are more general than additive functions (e.g., monotone, normalized functions, or superadditive functions), we need to extend the notion of a demand to consider such general price functions.

Recall that $\mathcal{C}_i \subseteq \mathbb{R}_+^{2^{G_i}}$ denotes the class of possible player-i cost functions, $\mathcal{C} = \Pi_i \mathcal{C}_i$, and for $c \in \mathcal{C}$ and $S \in U$, we define $c(S) := \sum_{i \in [k]} c_i(S \cap G_i)$. A *demand oracle for the class* $\mathcal{C}$ takes $q \in \mathcal{C}$, $\kappa \geq 0$, as input, and returns $\operatorname{argmax}_{S \subseteq U} \big(v(S) - \kappa \cdot q(S)\big)$. Note that a standard demand oracle amounts to a demand oracle for the class of additive functions: it takes item prices $q \in \mathbb{R}_+^U$ and performs the above computation for the additive function specified by q. The underlying motivation and rationale behind this definition is the same as that for a standard demand oracle. A demand oracle answers the economic question of what is the most profitable set for the buyer under given prices for the items, which turns out to be beneficial because it serves as a means of aggregating players' costs. Since players' costs are now given by general cost functions, it is only apt that we use the same expressive power when considering item prices, and therefore we now consider a price function $q \in \mathcal{C}$ when answering this question.[9]

A *constrained demand oracle for the class* $\mathcal{C}$ additionally takes a cap $Val \in \mathbb{R}_+$ as input, and returns $\operatorname{argmax}\{v(S) - \kappa \cdot q(S) : S \subseteq U,\ v(S) \leq Val\}$. It can be seen as answering the more-nuanced economic question of what is the most profitable value-capped set for the buyer.

We will often need to optimize the objective of a demand- or constrained-demand oracle only over subsets of some given set $A \subseteq U$. We assume that a demand-oracle or constrained-demand oracle has this flexibility.[10]

Theorem 2.1. *Let* $\big(v : 2^U \mapsto \mathbb{R}_+, B, \{G_i, \mathcal{C}_i \subseteq \mathbb{R}_+^{2^{G_i}}, c_i \in \mathcal{C}_i\}\big)$ *be a budget-feasible mechanism-design instance. Suppose we are given a demand oracle for* $\mathcal{C} = \Pi_i \mathcal{C}_i$*. For any* $A \subseteq U$*,* $\kappa \geq 0$*,* $Val \in \mathbb{R}_+$*, the following LPs can be solved in polytime:*

(a) $$\max \sum_{S \subseteq A} v(S) x_S \quad \text{s.t.} \quad \sum_{S \subseteq A} c(S) x_S \leq B, \quad \sum_{S \subseteq A} x_S \leq 1, \quad x \geq 0. \qquad \text{(BFLP}(A)\text{)}$$

(b) $$\max \sum_{S \subseteq A} \big(v(S) - \kappa \cdot c(S)\big) x_S \quad \text{s.t.} \quad \sum_{S \subseteq U_2} v(S) x_S \leq Val, \quad \sum_{S \subseteq A} x_S \leq 1, \quad x \geq 0. \quad \text{(CDLP}(A)\text{)}$$

Proof. Both parts follow by considering the dual of the respective LPs and observing that a demand oracle yields a separation oracle for the dual. Therefore, the ellipsoid method can be used to solve the dual LPs, and hence the primal LPs.

For part (a), the dual of (BFLP(A)) has the constraint $\alpha \cdot c(S) + \beta \geq v(S)$ for all $S \subseteq A$, where $\alpha, \beta \geq 0$ are the dual variables corresponding to the primal constraints. This amounts to determining if $\max_{S \subseteq A}\big(v(S) - \alpha \cdot c(S)\big) \leq \beta$, which we can determine by using a demand-oracle query over the set A to find S^*. For part (b), the dual of (CDLP(A)) has the constraint $\alpha \cdot v(S) + \beta \geq v(S) - \kappa \cdot c(S)$ for all $S \subseteq A$, where $\alpha, \beta \geq 0$. If $\alpha \geq 1$, then these constraints are trivially satisfied, so assume otherwise. Then, feasibility amounts to determining if $\max_{S \subseteq A}\big(v(S) - \frac{\kappa}{1-\alpha} \cdot c(S)\big) \leq \frac{\beta}{1-\alpha}$, which can again be answered via a demand-oracle query over A. ☐

Mechanism design. In the basic mechanism design setup, we have a set of k players, and a set A of possible outcomes. Each player i has a *private type* $c_i : A \mapsto \mathbb{R}_+$, where $c_i(a)$ gives the cost of alternative $a \in A$ to player i.[11] Let $\mathcal{C}_i$ be the publicly-known set of all valid types of player i (so $c_i \in \mathcal{C}_i$). Let $\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_k$ denote the space of all players' valid types. (We have deliberately overloaded notation

[9] From a computational perspective, it is known that value oracles are insufficient even for the algorithmic problem of computing a good approximation to OPT_{Alg} in polynomial time, even with additive costs.

[10] Most often, the set A will be of the form $\bigcup_{i \in I} G_i$ for some $I \subseteq [k]$. A demand oracle on A can be encoded as a query over the entire set U by taking c_ℓ, for $\ell \notin I$ to be the constant function: $c_\ell(S) = M_\ell$ for all $S \subseteq 2^{G_\ell} - \{\emptyset\}$, where M_ℓ is sufficiently large, say $2v(G_\ell)/\kappa$ (assuming v is monotone, subadditive). Thus, this added flexibility is a very benign requirement.

[11] We describe the setup in terms of cost incurred and payments, instead of the more-common choice of value obtained and prices, as this is what we encounter in budget-feasible mechanism design.

here, as the type c_i and set $\mathcal{C}_i$ above are essentially the same as the cost-function $c_i : 2^{G_i} \mapsto \mathbb{R}_+$ and set $\mathcal{C}_i$ of such cost functions in the budget-feasible mechanism-design (MD) setup.)

For example, multidimensional budget-feasible mechanism design can be cast in the above setup as follows. We have $A = 2^U$. Each player i's cost function $c_i : 2^{G_i} \mapsto \mathbb{R}_+$ yields a corresponding type, that we also denote by c_i, where $c_i(S) := c_i(S \cap G_i)$; the set $\mathcal{C}_i$ of player-i cost functions, correspondingly maps to the set of valid player types for player i. We use c to denote the tuple $(c_1, \ldots, c_k)$, and c_{-i} to denote the tuple that excludes i's type (or cost function). Similarly $\mathcal{C}_{-i} = \prod_{j \in [k]-\{i\}} \mathcal{C}_j$.

A (direct revelation) *mechanism* consists of an *algorithm* or allocation rule $f : \mathcal{C} \mapsto A$, and a *payment function* $p_i : \mathcal{C} \mapsto \mathbb{R}$ for each player i. Each player i reports some type $c_i \in \mathcal{C}_i$ (possibly deviating from her true type), and the mechanism computes the outcome $f(c)$ and pays $p_i(c)$ to each player i. Note that in budget-feasible mechanism design, f and the p_is can depend on the publicly-known information $(v, B, \{G_i, \mathcal{C}_i\})$, which we will treat implicitly as being fixed. The *utility* that i obtains when her true type is $\bar{c}_i$, she reports c_i, and other players report c_{-i} is $u_i(\bar{c}_i; c_i, c_{-i}) := p_i(c_i, c_{-i}) - \bar{c}_i(f(c_i, c_{-i}))$, and each player aims to maximize her own utility. In multidimensional budget-feasible mechanism design, we seek a mechanism $\mathcal{M} = \big(f, \{p_i\}_{i \in [k]}\big)$ satisfying the following properties.

- $\mathcal{M}$ is *truthful*: each player i maximizes her utility by reporting her true private type: for every $\bar{c}_i, c_i \in \mathcal{C}_i$ and $c_{-i} \in \mathcal{C}_{-i}$, we have $u_e(\bar{c}_e; \bar{c}_e, c_{-e}) \geq u_e(\bar{c}_e; c_e, c_{-e})$.
- $\mathcal{M}$ is *individually rational* (IR): $u_i(\bar{c}_i; \bar{c}_i, c_{-i}) \geq 0$ for every i, every $\bar{c}_i \in \mathcal{C}_i$ and $c_{-i} \in \mathcal{C}_{-i}$; note that this implies that $p_i(c) \geq 0$ for all c. We say that $\mathcal{M}$ makes *no positive transfers* (NPT) if $p_i(c) = 0$ whenever $S = f(c) \subseteq U$ is such that $S \cap G_i = \emptyset$. (More abstractly, we can say that $p_i(c) = 0$ whenever $a = f(c)$ is such that $c'_i(a) = 0$ for all $c'_i \in \mathcal{C}_i$.) In the sequel, we will always implicitly require NPT.[12]
- $\mathcal{M}$ is *budget feasible*: we have $\sum_i p_i(c) \leq B$ for every type $c \in \mathcal{C}$. Note that if $\mathcal{M}$ is individually rational, this implies that $\sum_i c_i\big(f(c)\big) \leq B$.

A *randomized* mechanism can use random bits to determine $f(c)$ and $\{p_i(c)\}$; so the cost-incurred by, payment made to, and utility of, a player are all random variables. We say that a randomized mechanism is:

(a) *universally budget feasible*, if truthfulness, IR, and budget feasibility hold with probability 1, i.e., the mechanism can be viewed as a distribution over deterministic budget-feasible mechanisms.
(b) *budget-feasible in expectation*, if $\mathcal{M}$ is *truthful in expectation*, the expected total payment is at most B, and IR (and NPT) holds with probability 1.[13] Truthful in expectation means that the expected utility of a player is maximized by truthful reporting.

We say that $\mathcal{M}$ achieves approximation ratio α with respect to a benchmark Bmark, if $v\big(f(c)\big) \geq \mathsf{Bmark}(c)/\alpha$ for all $c \in \mathcal{C}$. If $\mathcal{M}$ is randomized, then we have $\mathbb{E}\big[v\big(f(c)\big)\big] \geq \mathsf{Bmark}(c)/\alpha$ for all $c \in \mathcal{C}$.

A central tool that we utilize is the *VCG mechanism* [58, 24, 38], which is in fact a family of mechanisms, showing that if the algorithm f is such that it minimizes an affine function of the players' costs, then one can always combine it with suitable payments to obtain a truthful mechanism. This classical result is one of the predominant tools often leveraged in multidimensional mechanism design [48, 25, 29], forming the basis of the maximal-in-distributional-range (MIDR) approach.

Theorem 2.2 (VCG mechanism [58, 24, 38]). *Let $f : \mathcal{C} \mapsto \mathbb{R}_+$ be given by $f(c) = \operatorname{argmin}_{a \in A}\big(\sum_{i \in [k]} \alpha_i c_i(a) + \beta_a\big)$ for all $c \in \mathcal{C}$, where $\alpha_i > 0$ for every player i. Consider the following payments:*

$$p_i(c) = \frac{-1}{\alpha_i}\Big(\sum_{\ell \in [k]-\{i\}} \alpha_\ell c_\ell(a^*) + \beta_{a^*}\Big) + h_i(c_{-i}), \quad \text{where } a^* = f(c) \qquad \text{for all } i, c \in \mathcal{C} \tag{VCG}$$

where h_i is some function that depends only c_{-i}. Then the mechanism (f, p) is truthful.

[12] We remark that the lower bounds we prove in Section 3 hold also for mechanisms that may not satisfy NPT.

[13] If IR holds in expectation, i.e., the expected utility of a truthful player is nonnegative, then one can define random payments that ensure that IR holds with probability 1, without changing the expected payment made by the mechanism; see Lemma 4.3.

For budget-feasible mechanism design (where $A = 2^U$), one choice of h_is that ensures IR and NPT is to set $h_i(c_{-i}) := \frac{1}{\alpha_i} \cdot \min_{S \subseteq U - G_i} \big(\sum_\ell \alpha_{\ell \in [k]-\{i\}} c_\ell(S) + \beta_S\big)$ for all i, $c_{-i} \in \mathcal{C}_{-i}$.

Proof sketch. Fix player $i, \bar{c}_i, c_i \in \mathcal{C}_i$, $c_{-i} \in \mathcal{C}_{-i}$. Let $a^* = f(\bar{c}_i, c_{-i})$ and $b = f(c_i, c_{-i})$. Then

$$u(\bar{c}_i; \bar{c}_i, c_{-i}) = h_i(c_{-i}) - \frac{1}{\alpha_i}\Big(\sum_{\ell \in [k].\ell \neq i} \alpha_\ell c_\ell(a^*) + \alpha_i \bar{c}_i(a^*) + \beta_{a^*}\Big), \qquad \text{and}$$

$$u(\bar{c}_i; c_i, c_{-i}) = h_i(c_{-i}) - \frac{1}{\alpha_i}\Big(\sum_{\ell \in [k], \ell \neq i} \alpha_\ell c_\ell(b) + \alpha_i \bar{c}_i(b) + \beta_b\Big).$$

But by definition, $\sum_{\ell \in [k], \ell \neq i} \alpha_\ell c_\ell(a^*) + \alpha_i \bar{c}_i(a^*) + \beta_{a^*} = \min_{a \in A}\big(\sum_{\ell \in [k] \ell \neq i} \alpha_\ell c_\ell(a) + \alpha_i \bar{c}_i(a) + \beta_a\big)$. This also shows that with the given choice of h_is for budget-feasible mechanism design, we obtain IR, since h_i can be viewed as optimizing the objective function underlying f over a subset of A. We obtain IR because if $a^* = S^* \subseteq U$ is such that $S^* \cap G_i = \emptyset$, then $u(\bar{c}_i; \bar{c}_i, c_{-i}) = 0$ and $\bar{c}_i(a^*) = 0$, so $p_i(\bar{c}_i, c_{-i}) = 0$. ☐

Our mechanisms utilize the VCG mechanism in a bit more generality. We will usually have a set $\mathcal{N}'$ of players, with $\alpha_i > 0$ for all $i \in \mathcal{N}'$, and $\alpha_i = 0$ for all $i \notin \mathcal{N}'$. With this, one can still utilize Theorem 2.2, taking the alternative set A to be $2^{U'}$, where $U' = \bigcup_{i \in \mathcal{N}'} G_i$, and setting the payments of players not in $\mathcal{N}'$ to always be 0.

Properties of XOS and subadditive valuations.

Claim 2.3. *Let $v : 2^U \mapsto \mathbb{R}_+$ be an XOS valuation. Then, for any $S \subseteq U$, and any partition $A_1, \dots, A_r$ of U, we have we have $\sum_{i \in [r]}\big(v(S) - v(S - A_i)\big) \leq v(S)$.*

Proof. Let $I \subseteq [r]$ be the indices i for which $A_i \cap S \neq \emptyset$. The stated inequality is equivalent to showing that $\sum_{i \in I} \frac{v(S - A_i)}{|I|-1} \geq v(S)$. This inequality follows because taking $\mu_T = \frac{1}{|I|-1}$ for all $T \in \{S - A_i : i \in I\}$ yields a fractional cover of S. ☐

Our randomized mechanisms all use a random-sampling step to compute a good estimate of OPT_{Bench}. Let $\mathcal{N}_1, \mathcal{N}_2$ be a random partition of $\mathcal{N}$ obtained by placing each player independently with probability $\frac{1}{2}$ in $\mathcal{N}_1$ or $\mathcal{N}_2$. Let $U_j := \bigcup_{i \in \mathcal{N}_j} G_i$ for $j = 1, 2$ be the corresponding partition of U induced by $\mathcal{N}_1, \mathcal{N}_2$. The idea is to use U_1 to obtain a good estimate, and work with this estimate for U_2. Random partitioning has also been used in prior work [14, 2, 53] for single-dimensional budget-feasible mechanism design, where U is directly partitioned by assigning each element to a part with probability $\frac{1}{2}$. While in the single-dimensional setting, players and items are synonymous, as noted earlier, in the multidimensional setting, to ensure truthfulness, it is important to partition players and consider the partition of U induced by this partition of players, as defined above. We prove the following result.

Lemma 2.4 (Random-partitioning lemma). *Let $g : 2^U \mapsto \mathbb{R}_+$ be subadditive. Define $g_{-1}(S) := \min_{i \in [k]} g(S - G_i)$ for $S \subseteq U$. Consider any $S \subseteq U$, and let $S_1 = S \cap U_1$, $S_2 = S \cap U_2$. Let Ω be the event $\big\{g(S_2), g(S_1) \geq \frac{g_{-1}(S)}{4}\big\}$. Then, (a) $\Pr[\Omega] \geq \frac{1}{2}$, and (b) $\Pr\big[\{g(S_2) \geq \frac{g(S)}{2},\ g(S_2) \geq g(S_1)\} \cap \Omega\big] \geq \frac{1}{4}$.*

Proof. Let I be a minimal prefix of $[k]$ such that, letting $A_1 = \bigcup_{i \in I}(S \cap G_i)$, we have $g(A_1) \geq \frac{g_{-1}(S)}{2}$. Let $A_2 = S - A_1$. Letting ℓ be the last index in I, we have $g(A_1 - G_\ell) < \frac{g_{-1}(S)}{2}$. Also, $g(S - G_\ell) \geq g_{-1}(S)$. So since g is subadditive, we have $g(A_2) \geq g(S - G_\ell) - g(A_1 - G_\ell) > \frac{g_{-1}(S)}{2}$.

Now fix a partition A_1^H, A_1^L of A_1, and a partition A_2^H, A_2^L of A_2, where $g(A_1^H) \geq g(A_1)/2$ and $g(A_2^H) \geq g(A_2)/2$. We call A_1^H and A_2^H, the "big sets".

The random partition U_1, U_2 induces random partitions of A_1 and A_2. Consider the event Γ that for both $\ell = 1, 2$, the random partition of A_ℓ induced by U_1, U_2 is the same as the partition A_ℓ^H, A_ℓ^L, up to permutations of the parts. That is, Γ is the event that $U_j \cap A_\ell \in \{A_\ell^H, A_\ell^L\}$ for $j = 1, 2$ and $\ell = 1, 2$. For any $j = 1, 2$, we have $\Pr\big[U_j \cap A_1 = A_1^L,\ U_j \cap A_2 = A_2^L \,|\, \Gamma\big] = \frac{1}{4}$. So conditioned on Γ, with probability at least $\frac{1}{2}$, we have that both U_1, U_2 contain some big set. Removing the conditioning yields part (a).

For part (b), we observe that $g(S_1)$, $g(S_2)$ are identically distributed, and this remains true even when we condition on the event Ω. It follows that $\Pr[g(S_2) \geq g(S_1) | \Omega] \geq \frac{1}{2}$. Also $g(S_1) + g(S_2) \geq g(S)$, so $g(S_2) \geq g(S_1)$ also implies that $g(S_2) \geq \frac{g(S)}{2}$. □

Throughout, for $j = 1, 2$, we use $V_j^* := \max\{v(S) :\ S \subseteq U_j,\ c(S) \leq B\}$ to denote the optimal value that can be achieved using only players in $\mathcal{N}_j$ and elements in U_j.

Corollary 2.5. *Let $v : 2^U \mapsto \mathbb{R}_+$ be subadditive. We have $\Pr\big[V_2^* \geq \frac{OPT_{\mathsf{Alg}}(v,B,c)}{2},\ V_2^* \geq V_1^* \geq \frac{OPT_{\mathsf{Bench}}(v,B,c)}{4}\big] \geq \frac{1}{4}$.*

Proof. Let $O^* \subseteq U$ be such that $v_{-1}(O^*) = OPT_{\mathsf{Bench}}(v, B, c)$. Since $V_j^* \geq v(O^* \cap U_j)$ for $j = 1, 2$, applying Lemma 2.4 (a) to the set $S = O^*$, yields $\Pr\big[V_2^*, V_1^* \geq \frac{OPT_{\mathsf{Bench}}(v,B,c)}{4}\big] \geq \frac{1}{2}$. V_1^*, V_2^* are identically distributed, even conditioned on the above event, so $V_2^* \geq V_1^*$ with probability $1/2$, conditioned on this event. Also $V_1^* + V_2^* \geq OPT_{\mathsf{Alg}}(v, B, c)$, so $V_2^* \geq V_1^*$ implies that $V_2^* \geq \frac{OPT_{\mathsf{Alg}}(v,B,c)}{2}$. □

For XOS valuations, we prove an analogous result for the optimal value of the LP-relaxation for the algorithmic problem of computing OPT_{Alg}. For $A \subseteq U$, let $\mathsf{LP}^*(A)$ be the optimal value of (BFLP(A)), which recall is the following LP that can be solved in polytime using a demand oracle (Theorem 2.1).

$$\max \sum_{S \subseteq A} v(S) x_S \qquad \text{s.t.} \qquad \sum_{S \subseteq A} c(S) x_S \leq B, \qquad \sum_{S \subseteq A} x_S \leq 1, \qquad x \geq 0. \tag{BFLP(A)}$$

Throughout, for $j = 1, 2$, let LP_j^* denote the optimal value of (BFLP(U_j)), and let LP^* be the optimal value of (BFLP(U)).

Lemma 2.6. *Let $v : 2^U \mapsto \mathbb{R}_+$ be XOS. With probability at least $1/4$, we have $\mathsf{LP}_2^* \geq \mathsf{LP}_1^* \geq \frac{OPT_{\mathsf{Bench}}(v,B,c)}{4}$ and $\mathsf{LP}_2^* \geq \frac{\mathsf{LP}^*}{2}$.*

Proof. Again, let $O^* \subseteq U$ be such that $v_{-1}(O^*) = OPT_{\mathsf{Bench}}(v, B, c)$. Clearly, we have $\mathsf{LP}_j^* \geq V_j^* \geq v(O^* \cap U_j)$ for $j = 1, 2$. So by Lemma 2.4 (a), we have $\Pr\big[\mathsf{LP}_2^*, \mathsf{LP}_1^* \geq \frac{OPT_{\mathsf{Bench}}(v,B,c)}{4}\big] \geq \frac{1}{2}$. Conditioned on this event, with probability $\frac{1}{2}$, we have that $\mathsf{LP}_2^* \geq \mathsf{LP}_1^*$, which also implies that $\mathsf{LP}_2^* \geq \mathsf{LP}^*/2$ since $\mathsf{LP}_1^* + \mathsf{LP}_2^* \geq \mathsf{LP}^*$. □

In the analysis of our mechanisms, we will often need to demonstrate the existence of a good-value set whose cost is bounded away from the budget, say, is at most $B/2$. We obtain this by arguing that a good-value set satisfying the budget constraint can be suitably pruned, as shown by Lemmas 2.7 and 2.8 for general cost functions and superadditive cost functions respectively. Recall that $c(S) := \sum_{i \in [k]} c_i(S \cap G_i)$ for $S \subseteq U$. Note that the cost-function c inherits the properties of the c_is: it is monotone and normalized, and if all c_is are superadditive, then so is c.

Lemma 2.7. *Let $g : 2^U \mapsto \mathbb{R}_+$ be subadditive. Let $Val \in \mathbb{R}_+$, and $S \subseteq U$ be such that $c(S) \leq B$. We can find $T \subseteq S$ such that $c(T) \leq B/2$ and $\min\{g_{-1}(S) - Val,\ Val - \max_{e \in S} g(e)\} < g(T) \leq Val$.*

Proof. Let $\theta = \max_{e \in S} v(e)$. Let I be a minimal prefix of $[k]$ such that, letting $S_1 = \bigcup_{i \in I}(S \cap G_i)$, we have $c(S_1) > B/2$ or $g(S_1) \geq Val$. Let $S_2 = S - S_1$, and let $S_1' = S_1 - G_\ell$, where ℓ is the last index in I.

If $c(S_1) \leq B/2$, then we must have $g(S_1) > Val$. Considering elements of S_1 in some fixed order, we take T to be a maximal prefix of S_1 such that $g(T) \leq Val$. Then we have $c(T) \leq B/2$ (since $T \subseteq S_1$), and the maximality of T shows that $g(T) > Val - \theta$.

If $c(S_1) > B/2$, then $c(S_2) < B/2$. By the minimality of I, we have $g(S'_1) < Val$. We have $g(S-G_\ell) \geq g_{-1}(S)$, from the definition of $g_{-1}(S)$. So since g is subadditive, $g(S_2) \geq g(S-G_\ell) - g(S'_1) > g_{-1}(S) - Val$. So again taking T to be a maximal prefix of S_2 with $g(T) \leq Val$, we obtain that $T = S_2$ and $g(T) > g_{-1}(S) - Val$, or $g(T) > Val - \theta$. □

Lemma 2.8. *Let $g : 2^U \mapsto \mathbb{R}_+$ be subadditive, and $c : 2^U \mapsto \mathbb{R}_+$ be superadditive. Let $Val \in \mathbb{R}_+$, and $S \subseteq U$ be such that $c(S) \leq B$. We can find $T \subseteq S$ such that $c(T) \leq B/2$ and $\min\{g(S) - Val, Val\} - \max_{e \in S} g(e) < g(T) \leq Val$.*

Proof. Let $\theta = \max_{e \in S} g(e)$. We proceed as in the proof of Lemma 2.7, except that we can exploit superadditivity and do not need to consider whole player-sets when forming S_1 and S_2. Considering elements in some fixed order, we now take S_1 to be a minimal prefix of S such that $c(S_1) > B/2$ or $g(S_1) \geq Val$. If $c(S_1) \leq B/2$, then we again take T to be a maximal prefix of value at most Val. Otherwise, we have $g(S_1) < Val + \theta$. Letting $S_2 = S - S_1$, we than have $g(S_2) > g(S) - Val - \theta$. We again let T be a maximal prefix of S_2 of value at most Val, so that $T = S_2$ or $g(T) > Val - \theta$. Also, $c(T) \leq c(S_2)$, and since c is superadditive, we obtain that $c(S_2) \leq c(S) - c(S_1) < B/2$. □

3 Impossibility results and lower bounds

We prove here the impossibility results that were stated informally in Theorem 1.1, which rule out any good approximation with respect to $OPT_{\mathsf{Alg}}(v, B, c)$.

Theorem 3.1. *Consider any $\alpha \geq 1$. There is an additive valuation v and budget B, such that if $\mathcal{M}$ is a deterministic budget-feasible mechanism, then $\mathcal{M}$ obtains value at most $OPT_{\mathsf{Alg}}(v, B, c)/\alpha$ for some additive cost function c.*

Proof. Let $U = \{e, f\}$ with $v(e) = \alpha$, $v(f) = 1$. Let the budget B be 1. There is only one player. Consider the additive cost function $c^{(1)}$ given by $c^{(1)}_e = B$, $c^{(1)}_f = 0$, and $c^{(2)}$ given by $c^{(2)}_e = B+1$, $c^{(2)}_f = B$. (Note that $c^{(2)}$ is a valid input since we are not assuming no-overbidding.)

On input $c = c^{(2)}$, the mechanism cannot return e due to budget-feasibility, and must return f, as otherwise the statement holds for $c = c^{(2)}$. By IR, $\mathcal{M}$ must pay B to the player under $c^{(2)}$. Now, $\mathcal{M}$ cannot output e on input $c^{(1)}$, and so the statement holds for $c = c^{(1)}$. This is because, otherwise, on input $c^{(1)}$, the player obtains 0 utility by reporting $c^{(1)}$ (as she is paid at most B due to budget feasibility), but obtains utility B by reporting $c^{(2)}$, contradicting truthfulness. □

Theorem 3.2. *Consider any $\epsilon > 0$. There is an additive valuation v and budget B, such that if $\mathcal{M}$ is a budget-feasible-in-expectation mechanism, then $\mathcal{M}$ obtains value at most $OPT_{\mathsf{Alg}}(v, B, c) \cdot \frac{1+\epsilon}{n}$ for some additive cost function c.*

Proof. We may assume that $\epsilon \leq 1$. Let $\gamma = 1 + \frac{n}{\epsilon}$. Again, there is only one player. We identify U with $[n]$. Let v be the additive valuation defined by $v(e) = \gamma^{n-e}$ for all $e \in [n]$, and the budget B be 1. For each $\ell \in [n]$, let $c^{(\ell)}$ be the additive cost function defined by $c^{(\ell)}_e = M \geq (1 + n\gamma^n)B$ for all $e \in [\ell - 1]$, $c^{(\ell)}_\ell = 1$, and $c^{(\ell)}_e = 0$ for all $e \in \{\ell+1, \ldots, n\}$. We argue that on some input $c^{(\ell)}$, $\mathcal{M}$ obtains value at most $OPT_{\mathsf{Alg}}(v, B, c^{(\ell)}) \cdot \frac{1+\epsilon}{n}$. (Again, $c^{(\ell)}$ is allowed as input, since we are not assuming no-overbidding.)

For $\ell, r \in [n]$, let p_ℓ be the expected payment made by $\mathcal{M}$ when the player reports $c^{(\ell)}$, and let $b_{\ell,r}$ be the expected cost incurred by the player when her true cost is $c^{(\ell)}$ and she reports $c^{(r)}$. Note that on input $c^{(\ell)}$, the expected number of items returned by $\mathcal{M}$ from $[\ell - 1]$ is at most $\frac{1}{1+n\gamma^n}$. This is because otherwise, we would have $b_{\ell,\ell} > B$, and so $p_\ell > B$ by IR, which contradicts budget-feasibility in expectation. By truthfulness, we have $p_\ell - b_{\ell,\ell} \geq p_r - b_{\ell,r}$ for all $\ell, r \in [n]$. In particular, we have

$$p_\ell - p_{\ell+1} \geq b_{\ell,\ell} - b_{\ell,\ell+1} \qquad \text{for all } \ell \in [n].$$

Adding the above for all $\ell \in [n-1]$, along with $p_n \geq b_{n,n}$, which follows due to IR, we obtain that $p_1 \geq b_{1,1} + \sum_{\ell=2}^{n}(b_{\ell,\ell} - b_{\ell-1,\ell})$. By budget-feasibility in expectation, we have $p_1 \leq 1$. Therefore, we have $b_{1,1} \leq 1/n$ or there is some $\ell \in \{2, \dots, n\}$ with $b_{\ell,\ell} - b_{\ell-1,\ell} \leq \frac{1}{n}$. Since $b_{\ell,\ell} - b_{\ell-1,\ell}$ is at least $\Pr\big[\mathcal{M}$ returns a set containing item ℓ on input $c^{(\ell)}\big]$, this in turn implies that there is some $\ell \in [n]$ such that $\Pr\big[\mathcal{M}$ returns a set containing item ℓ on input $c^{(\ell)}\big] \leq 1/n$.

Finally, we argue that this last inequality implies that on input $c' = c^{(\ell)}$, $\mathcal{M}$ obtains value at most $OPT_{\mathsf{Alg}}(v, B, c') \cdot \frac{1+\epsilon}{n}$. We have $OPT_{\mathsf{Alg}}(v, B, c') = v(\{\ell, \dots, n\})$. The expected value Val obtained by $\mathcal{M}$ on input (v, B, c') is at most

$$\frac{v(\ell)}{n} + v(\{\ell+1, \dots, n\}) + v(1) \cdot \mathbb{E}\big[\text{no. of items returned by } \mathcal{M} \text{ from } [\ell-1]\big] \leq v(\ell) \cdot \Big(\frac{1}{n} + \frac{\gamma^{n-1}}{1+n\gamma^n}\Big) + v(\{\ell+1, \dots, n\}).$$

So we have $OPT_{\mathsf{Alg}} - v(\{\ell+1, \dots, n\}) \geq n\big(\mathit{Val} - v(\{\ell+1, \dots, n\}) - \frac{\gamma^{n-1}}{1+n\gamma^n} \cdot v(\ell)\big)$. Therefore

$$OPT_{\mathsf{Alg}} \geq n \cdot \mathit{Val} - (n-1)v(\{\ell+1, \dots, n\}) - \frac{v(\ell)}{\gamma} \geq n \cdot \mathit{Val} - \frac{n}{\gamma-1} \cdot v(\ell) \geq n \cdot \mathit{Val} - \epsilon \cdot OPT_{\mathsf{Alg}}. \qquad \square$$

The following impossibility result under no-overbidding was shown by [21], even for the single-dimensional LOS setting. We include a proof in Appendix A for completeness.

Theorem 3.3 ([21]). *Let v be the additive valuation with $v(e) = 1$ for all $e \in U$, and let the budget be $B = n = |U|$. Let $\mathcal{M}$ be a budget-feasible mechanism.*

(a) If $\mathcal{M}$ is deterministic, then for the above (v, B), there is some additive cost function c for which $\mathcal{M}$ obtains value at most $OPT_{\mathsf{Alg}}(v, B, c)/n$.

(b) If $\mathcal{M}$ is budget-feasible in expectation, there is some additive cost function c for which $\mathcal{M}$ obtains value at most $OPT_{\mathsf{Alg}}(v, B, c)/O(\log n)$.

The above lower bounds are obtained on instances involving a single player. One can therefore argue using Yao's minimax principle that the lower bounds for randomized mechanisms in Theorems 3.2 and 3.3 (b) extend to Bayesian budget-feasible mechanisms. We define the Bayesian setting, and prove the following corollary, in Appendix A.

Corollary 3.4. *No Bayesian budget-feasible mechanism can achieve approximation ratio better than n, and better than $O(\log n)$ assuming no-overbidding, relative to $OPT_{\mathsf{Alg}}(v, B, c)$.*

We also prove a lower bound on the approximation guarantee achievable with respect to OPT_{Alg} by any truthful (even non budget-feasible) mechanism. This is in stark contrast to the single-dimensional setting, where lower bounds only exist for budget-feasible mechanisms; the algorithm that returns an optimal solution $\operatorname{argmax}_{S \subseteq U} \{v(S) : c(S) \leq B\}$ (with consistent tie breaking) satisfies Myerson's monotonicity condition, and hence is truthfully implementable.

Theorem 3.5. *No deterministic truthful mechanism can achieve approximation ratio strictly larger than $\phi = \frac{1+\sqrt{5}}{2}$ with respect to OPT_{Alg}, even with additive valuations and additive cost functions.*

Proof. Consider again the setting with a single player. There are 2 items, with values $v(e_1) = 1$, $v(e_2) = \phi$, and v is additive, and the budget is $B = 2$. Consider the additive cost functions c_1, c_1' that assign costs $c_1(e_1) = 1, c_1(e_2) = B$ and $c_1'(e_1) = 1.5, c_1'(e_2) = 0.5$. We have $OPT_{\mathsf{Alg}}(c_1) = \phi$, so if the mechanism attains approximation ratio larger than ϕ, it must return e_2 on input c_1. But then on input c_1', due to truthfulness, the mechanism must still return only e_2. Suppose the mechanism returns $S \subseteq \{e_1, e_2\}$. Then, by weak-monotonicity, we must have $c_1(e_2)+c_1'(S) \leq c_1(S)+c_1'(e_2)$, i.e., $c_1'(S) \leq c_1(S)-(B-0.5)$, and only $S = \{e_2\}$ satisfies this inequality. But then the mechanism's approximation ratio on input c_1' is $\frac{\phi+1}{\phi} = \phi$. □

We remark that while the constructions above utilize only a single player, we can always pad the instance by introducing "dummy" players that incur 0 cost, and contribute very little value to OPT_{Alg}.

Lower bounds on the approximation factor relative to OPT_{Bench}. Lower bounds under a large-market assumption introduced by [4] in the single-dimensional setting, carry over to lower bounds on the approximation factor achievable relative to OPT_{Bench}. This is simply because under the large-market assumption $v_{\max} \ll OPT_{\mathsf{Alg}}$ in the single-dimensional setting, OPT_{Bench}, which is at least $OPT_{\mathsf{Alg}} - v_{\max}$, essentially coincides with OPT_{Alg}. Therefore, an $\frac{e}{e-1}$ approximation-factor lower bound (relative to OPT_{Alg}) shown by [4] in the large market setting, translates to the same lower bound against OPT_{Bench} in our setting. The lower bound is proved in [4] under the assumption $\max_e c(e) \ll B$, but they mention that it carries over to the setting $v_{\max} \ll OPT_{\mathsf{Alg}}$. We include a proof of the following theorem in Appendix A for completeness.

Theorem 3.6 (Follows from [4]). *No budget-feasible mechanism can achieve value better than $\left(1 - \frac{1}{e}\right) \cdot OPT_{\mathsf{Bench}}(v, B, c)$ on every instance, even with additive v and additive cost functions.*

4 XOS Valuations

We design and analyze mechanisms for XOS valuations that achieve $O(1)$-approximation ratio with respect to our benchmark OPT_{Bench}. Section 4.1 describes a budget-feasible-in-expectation mechanism for general cost functions, and Section 4.2 focuses on universally budget-feasible mechanisms under the no-overbidding assumption. We show how to drop the no-overbidding assumption in Section 5. As noted earlier, our mechanisms perform a VCG computation, which then defines the payments made to the players as in the VCG mechanism (see Theorem 2.2 and (VCG)), so we describe the underlying algorithm, and discuss payments given by (VCG) in the analysis.

4.1 Budget-feasible-in-expectation mechanism for general cost functions

Our budget-feasible-in-expectation mechanism uses random sampling to compute an estimate of OPT_{Bench} from one part $(\mathcal{N}_1, U_1)$, and utilizes this to solve an LP-relaxation of a constrained demand-oracle query on $(\mathcal{N}_2, U_2)$. By viewing the LP solution as a distribution, and using VCG payments, we obtain the desired mechanism. Recall that LP_j^* is the optimal value of (BFLP(U_j)), for $j = 1, 2$.

Theorem 4.1. *Taking $\lambda = 0.5$ in Algorithm XOS-BFInExp, along with suitable payments, we obtain a budget-feasible-in-expectation mechanism that obtains expected value at least $\frac{OPT_{\mathsf{Bench}}(v,B,c)}{64}$. The mechanism runs in polytime given a demand oracle.*

Proof. The polynomial running time follows from Theorem 2.1, which shows that we can solve LPs of the form (BFLP(A)) and (CDLP(A)) in polytime given a demand oracle. Lemma 4.3 specifies the payments, shows that they can be computed efficiently, and the resulting mechanism is budget-feasible in expectation. Lemma 2.6 shows that $\mathsf{LP}_2^* \geq \mathsf{LP}_1^* \geq \frac{OPT_{\mathsf{Bench}}(v,B,c)}{4}$ holds with probability at least $\frac{1}{4}$. Assuming that this event happens, Lemma 4.2 shows that we obtain value at least $\frac{V_1}{4} \geq \frac{OPT_{\mathsf{Bench}}(v,B,c)}{16}$, therefore the expected value returned at least $\frac{OPT_{\mathsf{Bench}}(v,B,c)}{64}$. □

Algorithm XOS-BFINEXP // budget-feasible-in-expectation mechanism for general costs

Input: Budget-feasible MD instance $\big(v : 2^U \mapsto \mathbb{R}_+, B, \{G_i, \mathcal{C}_i \subseteq \mathbb{R}_+^{2^{G_i}}, c_i \in \mathcal{C}_i\}\big)$; $\lambda \in [0,1]$
Output: subset of U; payments are VCG payments, as specified in Lemma 4.3

1 Partition $\mathcal{N}$ into two sets $\mathcal{N}_1, \mathcal{N}_2$ by placing each player independently with probability $\frac{1}{2}$ in $\mathcal{N}_1$ or $\mathcal{N}_2$. Let $U_j := \bigcup_{i\in\mathcal{N}_j} G_i$ be the induced partition of U. Compute $V_1 = \mathsf{LP}_1^*$ using a demand oracle.

2 Obtain x^* by solving the following constrained-demand-oracle LP, which is (CDLP(A)) with $A = U_2$, $\kappa = \frac{\lambda V_1}{B}$, $\mathit{Val} = \lambda V_1$.

$$\max \sum_{S\subseteq U_2} \big(v(S) - \tfrac{\lambda V_1}{B}\cdot c(S)\big)x_S \quad \text{s.t.} \quad \sum_{S\subseteq U_2} v(S)x_S \le \lambda V_1, \quad \sum_{S\subseteq U_2} x_S \le 1, \quad x \ge 0. \quad (1)$$

3 Sample a set T from the distribution $(x^*_S)_{S\subseteq U_2}$, and **return** T.

Lemma 4.2. *Suppose that* $\mathsf{LP}_2^* \ge V_1$. *Then the optimal value of* (1) *is at least* $\lambda V_1\big(1 - \frac{\lambda V_1}{\mathsf{LP}_2^*}\big) \ge \lambda(1-\lambda)V_1$.

Proof. Let $\overline{x}$ be an optimal solution to (BFLP(U_2)). We use S below to index over subsets of U_2. So $\sum_S v(S)\overline{x}_S = \mathsf{LP}_2^*$ and $\sum_S c(S)\overline{x}_S \le B$. Consider $x' = \frac{\overline{x}}{\mathsf{LP}_2^*/\lambda V_1}$. We have $\sum_S v(S)x'_S = \lambda V_1$, so x' is feasible to (1). We also have $\sum_S c(S)x'_S \le \frac{\lambda V_1}{\mathsf{LP}_2^*}\cdot B$. So the optimal value of (1) is at least the objective value of x', which is

$$\sum_S v(S)x'_S - \frac{\lambda V_1}{B}\cdot\sum_S c(S)x'_S \ge \lambda V_1 - \frac{\lambda V_1}{B}\cdot\frac{\lambda V_1}{\mathsf{LP}_2^*}\cdot B = \lambda V_1\Big(1 - \tfrac{\lambda V_1}{\mathsf{LP}_2^*}\Big). \qquad \square$$

Lemma 4.3. *Given a demand oracle, one can efficiently compute payments that when combined with Algorithm XOS-BFInExp yield a budget-feasible-in-expectation mechanism (where IR holds with probability* 1.

Proof. We first obtain expected payments that yield truthfulness in expectation and budget-feasibility in expectation. Given these, there is a standard way of obtaining actual payments for each random outcome of the mechanism that satisfy IR and NPT with probability 1.

Observe that the computation in step 2 amounts to a VCG computation over the domain of feasible *fractional* solutions to (1). Thus, we can still use expression (VCG) to obtain expected payments for the players. Let $\kappa = \frac{\lambda V_1}{B}$. The expected payment to player $i \in \mathcal{N}_2$ when step 2 returns the fractional solution x is given by $\sum_S x_S\big(\frac{1}{\kappa}\cdot v(S) - c(S - G_i)\big) - h_i(c_{-i})$, where $h_{-i}(c_i)$ is $\frac{1}{\kappa}$ times the optimal value of (1) when player i is excluded, that is, we are only allowed to use sets $S \subseteq U_2 - G_i$. Note that $h_{-i}(c_{-i})$ can be calculated efficiently given a demand oracle, so these expected payments can be computed efficiently. Since the expected cost to player i is $\sum_S c_i(S\cap G_i)x_S$, the expected utility is $\frac{1}{\kappa}\times$(objective value of x)$-h_{-i}(c_{-i})$, which is maximized by $x = x^*$, the outcome when player i reports truthfully. This shows truthfulness in expectation, and also shows that the expected utility under truthful reporting is nonnegative. One feasible solution to (1) when player i is excluded is given by setting $x'_T = \sum_{S\subseteq U: S-G_i=T} x^*_S$ for all $T \subseteq U_2 - G_i$, whose objective value is $\sum_{S\subseteq U} v(S-G_i)x^*_S - \kappa\cdot\sum_{\ell\in\mathcal{N}_2-\{i\}} c_\ell(S\cap G_\ell)x^*_S$. This yields a lower bound on $h_{-i}(c_{-i})$ and an upper bound of $\frac{1}{\kappa}\cdot\sum_S(v(S) - v(S-G_i))x^*_S$ on the payment made to player i. Therefore, the total expected payment is at most

$$\frac{B}{\lambda V_1}\cdot\sum_S x^*_S \sum_{\ell\in\mathcal{N}_1}(v(S) - v(S-G_\ell)) \le \frac{B}{\lambda V_1}\cdot\sum_S x^*_S v(S) \le B \qquad (2)$$

where the first inequality follows from Claim 2.3.

Fix a random partition $(\mathcal{N}_1, U_1)$, $(\mathcal{N}_2, U_2)$. We obtain payments for each random outcome T of the mechanism as follows. Consider a player $i \in \mathcal{N}_2$. Let μ_i be the expected payment to i, as computed above. Let $\mathbb{E}\left[cost_i\right] = \sum_S c_i(S \cap G_i) x^*_S$ be the expected cost incurred by i. We set the payment of i under outcome T to be $\frac{\mu_i}{\mathbb{E}\left[cost_i\right]} \cdot c_i(T \cap G_i)$. (Here c_i is i's reported cost, which we may assume is her true cost, due to truthfulness in expectation.) Since $\mu_i \geq \mathbb{E}\left[cost_i\right]$, this ensures that the payment is always the cost incurred by i, and is 0 if this cost is 0. ☐

Remark 4.4. With additive valuations and additive cost functions, observe that Algorithm XOS-BFInExp runs in polynomial time, since the LPs (BFLP(U_1)) and (CDLP(A)) can be solved in polytime.

4.2 Universally budget-feasible mechanisms assuming no-overbidding

We next describe how to obtain the stronger mechanism-design guarantee of universal budget-feasibility. We consider here the simpler setting where we assume no-overbidding; we show how to drop this assumption in Section 5. While the results here are subsumed by those obtained in Section 5 (modulo $O(1)$ approximation factors), we discuss things first in the simpler setting of no-overbidding as this will introduce many of the key underlying ideas, and the arguments are simpler.

Recall that no-overbidding imposes that for every player i, cost-function $c_i \in \mathcal{C}_i$, element $e \in G_i$, we have $c_i(e) \leq B$. Let $e^* = \operatorname{argmax}_{e \in U} v(e)$, and $v_{\max}$ denotes $v(e^*)$. Our mechanisms will exploit the fact that under no-overbidding, the mechanism that returns e^* and pays B to the player who owns e^* is a budget-feasible mechanism. (As noted earlier, without no-overbidding, the adaptation where we return $\operatorname{argmax}_{e \in U} \{v(e) : c(e) \leq B\}$ is not truthfully implementable.)

Algorithm XOS-UniBF describes the underlying algorithm for general cost functions. It yields an $O(1)$-approximation with respect to a weaker benchmark than OPT_{Bench}, which we describe below. This serves chiefly as a warm-up for the more sophisticated constructions in Section 5, where we drop the no-overbidding assumption by suitably mdifying portions of the algorithm (and obtain $O(1)$-approximation relative to OPT_{Bench}). We then consider superadditive costs (Section 4.2.1), and show that with some minor changes, we can achieve a stronger guarantee, namely, $O(1)$-approximation with respect to OPT_{Bench}, and we can do so in polytime given a constrained demand oracle.

Algorithm XOS-UniBF // universally budget-feasible mechanism for general costs

---

Input: Budget-feasible MD instance $\left(v : 2^U \mapsto \mathbb{R}_+, B, \{G_i, \mathcal{C}_i \subseteq \mathbb{R}_+^{2^{G_i}}\}, \{c_i \in \mathcal{C}_i\}\right)$; parameters $\lambda \in [0, 0.5]$, $p \in [0, 1]$

Output: subset of U; payments are VCG payments

1 Partition $\mathcal{N}$ into two sets $\mathcal{N}_1, \mathcal{N}_2$ by placing each player independently with probability $\frac{1}{2}$ in $\mathcal{N}_1$ or $\mathcal{N}_2$. For $j = 1, 2$, let $U_j := \bigcup_{i \in \mathcal{N}_j} G_i$ give the induced partition of U.

2 Compute $V_1 = \max_{S \subseteq U_1} \{v_{-1}(S) : \ c(S) \leq B\}$, the benchmark OPT_{Bench} associated with $\mathcal{N}_1$ and item-set U_1.

3 Compute $S^* \leftarrow \operatorname{argmax}_{S \subseteq U_2} \left\{v(S) - \frac{\lambda V_1}{B} \cdot c(S) : \ v(S) \leq \lambda V_1\right\}$ using a constrained demand oracle.

4 **return** S^* with probability p and e^* with probability $1 - p$.

Algorithm XOS-UniBF is based on the template used in [53] for budget-feasible mechanism design in the single-dimensional setting, and this works out because of the no-overbidding assumption. Without this, as discussed earlier, returning e^* in step 4 is no longer viable and one needs a much-more sophisticated approach to find a workaround; so the mechanism and its analysis become more involved, especially when we seek polytime guarantees using a constrained demand oracle (see Section 5.2.2).

To state the performance guarantee obtained by Algorithm XOS-UniBF, we introduce the following notation. For an integer $\ell \geq 1$, and function $g : 2^U \mapsto \mathbb{R}_+$ define $g_{-\ell}(S) := \min_{I \subseteq [k]:|I| \leq \ell} g\big(S - \bigcup_{i \in I} G_i\big)$ for $S \subseteq U$. Define $OPT_{\mathsf{Bench}}(\ell) = OPT_{\mathsf{Bench}}(\ell; v, B, c) := \max_{S \subseteq U} \{v_{-\ell}(S) : c(S) \leq B\}$. Also, define the parametrized benchmark

$$OPT_{\mathsf{Param}}(\varepsilon) \ = \ OPT_{\mathsf{Param}}(\varepsilon; v, B, c) \ := \ \max_{S \subseteq U} \Big\{ v(S) : \ c(S) \leq B, \quad v(S \cap G_i) \leq \varepsilon \cdot v(S) \ \ \forall i \in [k] \Big\}.$$

Note that $OPT_{\mathsf{Bench}}(\ell) \geq \max_{\varepsilon \in [0,1]} (1 - \ell\varepsilon) \cdot OPT_{\mathsf{Param}}(\varepsilon)$.

In Appendix B, we prove the following generalization of Lemma 2.4 (a).

Lemma 4.5. *Let $g : 2^U \mapsto \mathbb{R}_+$ be subadditive, and $\ell \geq 0$ be an integer. Consider any $S \subseteq U$, and let $S_1 = S \cap U_1$, $S_2 = S \cap U_2$. Then $\Pr\big[g_{-\ell}(S_1), g_{-\ell}(S_2) \geq \frac{g_{-(4\ell+1)}(S)}{4}\big] \geq \frac{1}{2}$.*

Theorem 4.6. *Taking $\lambda = 0.5$ and $p = 0.8$ in Algorithm XOS-UniBF, together with suitable payments, we obtain a universally budget-feasible mechanism that obtains expected value at least $\frac{1}{80} \cdot OPT_{\mathsf{Bench}}(5; v, B, c) \geq \frac{1}{80} \cdot \max_{\varepsilon \in [0,1]} (1 - 5\varepsilon) OPT_{\mathsf{Param}}(\varepsilon; v, B, c)$.*

Proof. Lemma 4.7 shows that we obtain a universally budget-feasible mechanism with suitable payments. We focus on proving the approximation guarantee.

Let $O^* = \operatorname{argmax}_{S \subseteq U} \{v_{-5}(S) : c(S) \leq B\}$. Let $O_j^* = O^* \cap U_j$ for $j = 1, 2$. Let $T_2^* := \operatorname{argmax}_{S \subseteq U_2} \{v_{-1}(S) : c(S) \leq B\}$, and $V_2 = v_{-1}(T_2^*)$ be the OPT_{Bench}-benchmark associated with player-set $\mathcal{N}_2$ and item-set U_2. Let Γ be the event that $v_{-1}(O_1^*), v_{-1}(O_2^*) \geq v_{-5}(O^*)/4$. Applying Lemma 4.5 to O^*, we obtain that $\Pr[\Gamma] \geq 0.5$. Note that V_1 and V_2 are identically distributed, and this remains true even when we condition on Γ. Therefore, we have that $\Pr[\{V_2 \geq V_1\} \cap \Gamma] \geq \frac{1}{4}$. Assume that this event happens.

Applying Lemma 2.7 on T_2^*, we can obtain $T \subseteq T_2^*$ such that $c(T) \leq B/2$ and $\lambda V_1 - v_{\max} < v(T) \leq \lambda V_1$. It follows that

$$v(S^*) \geq v(S^*) - \frac{\lambda V_1}{B} \cdot c(S^*) \geq v(T) - \frac{\lambda V_1}{B} \cdot c(T) \geq \lambda V_1 - v_{\max} - \lambda V_1/2 \geq \frac{\lambda V_1}{2} - v_{\max}.$$

We also have $V_1 \geq v_{-1}(O_1^*) \geq \frac{v_{-5}(O^*)}{4}$.

Putting everything together, and taking $\lambda = \frac{1}{2}$, we obtain that the expected value returned is at least

$$\frac{p}{4} \cdot \big(\tfrac{1}{16} \cdot OPT_{\mathsf{Bench}}(5) - v_{\max}\big) + (1-p) v_{\max} \geq \frac{1}{80} \cdot OPT_{\mathsf{Bench}}(5) \geq \frac{1}{80} \cdot \max_{\varepsilon \in [0,1]} (1 - 5\varepsilon) OPT_{\mathsf{Param}}(\varepsilon). \ \square$$

Lemma 4.7. *There exist payments that when combined with Algorithm XOS-UniBF yield a universally budget-feasible mechanism.*

Proof. We consider each possible outcome of the random choices in Algorithm XOS-UniBF and supply payments for which the resulting mechanism is budget feasible. If the outcome is to return e^* and i is such that $e^* \in G_i$, then we pay B to player i and 0 to the other players. The utility of player i is then $B - c_i(e^*) \geq 0$, regardless of her reported cost function. So we trivially obtain truthfulness, IR, and budget feasibility.

So suppose otherwise, and let $(\mathcal{N}_1, U_1)$, $(\mathcal{N}_2, U_2)$ be the partition obtained in step 1. Let $\kappa = \frac{\lambda V_1}{B}$. The computation in step 3 is a VCG computation. So using (VCG), paying each player $i \in \mathcal{N}_2$ the amount

$$\frac{v(S^*)}{\kappa} - \sum_{\ell \in \mathcal{N}_2 - \{i\}} c_\ell(S^* \cap G_\ell) - h_i(c_{-i}), \qquad \text{where } h_i(c_{-i}) = \frac{1}{\kappa} \cdot \max_{S \subseteq U_2 - G_i} \Big\{ v(S) - \kappa \cdot c(S) : \ v(S) \leq \lambda V_1 \Big\}$$

and the other players 0, yields truthfulness. Under this payment, the utility of every player $i \in \mathcal{N}_2$ is $\frac{1}{\kappa} \cdot \big(v(S^*) - \kappa \cdot c(S^*)\big) - h_{-i}(c_{-i})$, which is nonnegative since $v(S^*) - \kappa \cdot c(S^*) \geq v(S) - \kappa \cdot c(S)$ for all $S \subseteq U_2$ with $v(S) \leq \lambda V_1$. Moreover, if $S^* \cap G_i = 0$, then this utility, and hence payment to i are 0. This shows IR and NPT.

Finally, since $h_i(c_{-i}) \geq \frac{1}{\kappa} \cdot \big(v(S^* - G_i) - \kappa \cdot c(S^* - G_i)\big)$, the payment to $i \in \mathcal{N}_2$ is at most $\frac{1}{\kappa} \cdot \big(v(S^*) - v(S^* - G_i)\big)$. We have $\sum_{i \in \mathcal{N}_2} \big(v(S^*) - v(S^* - G_i)\big) \leq v(S^*)$ by Claim 2.3, and so the total payment is at most B. □

4.2.1 Superadditive cost functions: polytime $O(1)$-approximation with respect to OPT_{Bench}

With superadditive costs, we show that with minor tweaks to Algorithm XOS-UniBF, we can obtain a mechanism that runs in polytime given a constrained demand oracle, and achieves an $O(1)$-approximation with respect to OPT_{Bench} (Theorem 4.8). In step 2 of Algorithm XOS-UniBF, we now compute $V_1 = \mathsf{LP}_1^*$, and in step 3, we compute $S^* \leftarrow \operatorname{argmax}_{S \subseteq U_2} \big\{v(S) - \frac{\lambda V_1 + v_{\max}}{B} \cdot c(S) : \ v(S) \leq \lambda V_1 + v_{\max}\big\}$.

As in Lemma 4.7, we can obtain suitable payments that when combined with this modified algorithm yield a universally budget-feasible mechanism. The computations of V_1, S^* and the payments, can all be done in polytime, since we are given a constrained demand oracle. The only portion of the analysis that changes more significantly is the proof of the approximation guarantee.

Theorem 4.8. *Taking $\lambda = \frac{2}{5}$ and $p = \frac{14}{15}$ in the above modified version of Algorithm XOS-UniBF, together with suitable payments, we obtain a universally budget-feasible mechanism for superadditive costs with approximation ratio* 100 *that runs in polytime given a constrained demand oracle.*

Proof. As discussed above, we focus on proving the approximation guarantee. Fix an input (v, B, c), where the c_is are superadditive. We abbreviate $OPT_{\mathsf{Bench}}(v, B, c)$ to OPT_{Bench}. Lemma 2.6 shows that $\mathsf{LP}_2^* \geq \mathsf{LP}_1^* \geq \frac{OPT_{\mathsf{Bench}}}{4}$, holds with probability at least $\frac{1}{4}$. Assume that this event happens.

Let $\kappa = \lambda V_1 + v_{\max}$. Let $\overline{x}$ be an optimal solution to (BFLP(U_2)). We use S below to index over subsets of U_2. We transform $\overline{x}$ into a fractional solution x' such that $x'_S > 0$ only if $v(S) \leq \kappa$, and then use x' to obtain a lower bound on $v(S^*)$. Initialize $x' \leftarrow 0$. Consider a set S with $\overline{x}_S > 0$. By repeatedly finding a maximal subset of S having value at most κ, and deleting this, we obtain that as long as $v(S) > \kappa$, the deleted subset has value strictly larger than λV_1. So we can partition S into at most $\left\lceil \frac{v(S)}{\lambda V_1} \right\rceil$ sets in this fashion, each having value at most κ; we increase the x'-value of each of these sets by $\overline{x}_S$. Observe that $\sum_S x'_S \leq \sum_S \overline{x}_S \left\lceil \frac{v(S)}{\lambda V_1} \right\rceil \leq \frac{\mathsf{LP}_2^*}{\lambda V_1} + 1$. Also, $\sum_S v(S) x'_S \geq \sum_S v(S) \overline{x}_S = \mathsf{LP}_2^*$ since v is subadditive, and $\sum_S c(S) x'_S \leq \sum_S c(S) \overline{x}_S \leq B$ since the costs are superadditive. Since $x'_S > 0$ only if $v(S) \leq \kappa$, any such set is a candidate set for the constrained demand oracle. So we have

$$v(S^*) - \frac{\kappa}{B} \cdot c(S^*) \geq \frac{1}{\sum_S x'_S} \cdot \sum_S x'_S \Big(v(S) - \tfrac{\kappa}{B} \cdot c(S)\Big) \geq \frac{\mathsf{LP}_2^* - \kappa}{\mathsf{LP}_2^*/\lambda V_1 + 1}$$
$$= \frac{\lambda V_1}{1 + \lambda V_1/\mathsf{LP}_2^*} - \frac{\kappa}{1 + \mathsf{LP}_2^*/\lambda V_1} \geq \frac{\lambda V_1}{1 + \lambda} - \frac{\kappa}{1 + 1/\lambda} = V_1 \cdot \frac{\lambda(1-\lambda)}{1+\lambda} - \frac{v_{\max}}{1 + 1/\lambda}$$

where the final inequality above follows since $\mathsf{LP}_2^* \geq \mathsf{LP}_1^* = V_1$. We have $\frac{p}{4(1+1/\lambda)} = 1 - p$, and the expected value returned is at least $\frac{p}{4} \cdot v(S^*) + (1-p) v_{\max} \geq OPT_{\mathsf{Bench}}/100$. □

Remark 4.9. We can also tweak Algorithm XOS-UniBF differently, by taking $V_1 = V_1^*$ in step 2 (and no other changes). Recall that $V_j^* := \max\{v(S) : \ S \subseteq U_j, \ c(S) \leq B\}$ for $j = 1, 2$. This yields a slightly better 80-approximation (the same factor as in Theorem 4.6) relative to OPT_{Bench} for superadditive costs, but not in polytime, since computing V_1^* is *NP*-hard even for additive valuations and costs. To see this, let $\lambda = 0.5$, $p = 0.8$ (as in Theorem 4.6). As before, we can obtain suitable payments that when combined with this modified algorithm yield a universally-budget-feasible mechanism. For the approximation guarantee, suppose that the event $V_2^* \geq V_1^* \geq \frac{OPT_{\mathsf{Bench}}(v,B,c)}{4}$ occurs, which happens with probability $\frac{1}{4}$ (Corollary 2.5). Let T_2^* be an optimal solution to $\max\{v(S): S \subseteq U_2, c(S) \leq B\}$. So $c(T_2^*) \leq B$ and $v(T_2^*) = V_2^* \geq 2\lambda V_1$. Applying Lemma 2.8, we can obtain $T \subseteq T_2^*$ such that $c(T) \leq B/2$ and $\lambda V_1 - v_{\max} < v(T) \leq \lambda V_1$. We can now proceed as in the proof of Theorem 4.6 to obtain expected value at least $OPT_{\mathsf{Bench}}(v,B,c)/80$.

4.2.2 Polytime mechanism for additive valuations and additive costs

The above polytime mechanism for XOS valuations and superadditive costs can be adapted to run in polynomial time (i.e., without any oracle) for additive valuations and additive costs. With additive valuations and additive cost functions, a constrained demand oracle involves solving a knapsack problem. We argue that we can instead work with a related knapsack problem that can be solved efficiently using dynamic programming (DP), and thereby obtain a polytime $O(1)$-approximation mechanism.

This result is subsumed (modulo $O(1)$ approximation factors) by the result in Section 5.2.1, where we do not assume no-overbidding, but the mechanism and its analysis become much simpler assuming no-overbidding, so we include this below. The reader interested in the setting without assuming no-overbidding (for additive valuations and additive costs) can directly skip to Section 5.2.1.

Theorem 4.10. *There is a polytime mechanism for additive valuations and additive costs that achieves approximation ratio* 272 *with respect to* OPT_{Bench}.

Proof. In the oracle polytime algorithm for superadditive costs described at the start of Section 4.2.1, recall that we modify Algorithm XOS-UniBF by computing $V_1 = \mathsf{LP}_1^*$ in step 2, and $S^* = \mathrm{argmax}_{S \subseteq U_2}\{v(S) - \kappa \cdot c(S): v(S) \leq \kappa B\}$ in step 3, where $\kappa = \frac{\lambda V_1 + v_{\max}}{B}$. With additive v and additive costs, (BFLP(U_1)) becomes a polynomial-size LP, so it can be solved in polytime, and setting $u_e = v(e) - \kappa \cdot c(e)$ for all $e \in U_2$, computing S^* amounts to solving the knapsack problem, maximize $u(S)$ subject to $v(S) \leq \kappa B$, $S \subseteq U_2$.

We scale the $v(e)$'s to obtain polynomially-bounded weights, modifying the u_es and the knapsack budget correspondingly, to obtain a polytime-solvable knapsack problem. Set $B^{\mathsf{new}} = \frac{n\lambda V_1}{v_{\max}} + n$, and $w_e = \left\lceil \frac{nv(e)}{v_{\max}} \right\rceil$ and $u'_e = w_e - \frac{B^{\mathsf{new}}+n}{B} \cdot c(e)$ for all $e \in U_2$. We now obtain S^* by solving the knapsack problem over U_2, with item-values $\{u'_e\}$, item weights $\{w_e\}$ and knapsack budget $B^{\mathsf{new}} + n$; since the weights lie in $\{0\} \cup [n]$ this takes polynomial time. We return S^* with probability p and e^* with probability $1-p$.

The modified knapsack problem minimizes an affine function of the player costs. Since we solve this knapsack problem optimally, VCG again applies. So as in Lemma 4.7, one obtains universal truthfulness, and the payment to each player $i \in \mathcal{N}_2$ is at most $\frac{B}{B^{\mathsf{new}}+n} \cdot \big(w(S^*) - w(S^* \cap G_i)\big)$. Budget-feasibility follows since $w(S^*) \leq B^{\mathsf{new}} + n$; so we only need to analyze the approximation ratio.

We argue as in the proof of Lemma 4.8. We may assume that $\mathsf{LP}_2^* \geq \mathsf{LP}_1^* \geq \frac{OPT_{\mathsf{Bench}}(v,B,c)}{4}$, which happens with probability at least $\frac{1}{4}$. Let $\overline{x}$ be an optimal solution to (BFLP(U_2)). We transform $\overline{x}$ into a fractional solution x' such that $x'_S > 0$ only if $v(S) \leq \kappa$, and then use x' to obtain a lower bound on $v(S^*)$. Initialize $x' \leftarrow 0$. Consider a set S with $\overline{x}_S > 0$. We partition S into at most $\left\lceil \frac{w(S)}{B^{\mathsf{new}}} \right\rceil$ sets, each having w-weight at most $B^{\mathsf{new}} + n$, and increase x'-value of each of these sets by $\overline{x}_S$. We then have

$$\sum_S x'_S \leq \frac{\sum_S w(S)\overline{x}_S}{B^{\mathsf{new}}} + 1 \leq \frac{\frac{n}{v_{\max}} \cdot \mathsf{LP}_2^* + n}{B^{\mathsf{new}}} + 1 = \frac{\frac{\mathsf{LP}_2^*}{v_{\max}} + 1}{\frac{\lambda V_1}{v_{\max}} + 1} + 1 \leq \frac{\mathsf{LP}_2^*}{\lambda V_1} + 1$$

where the last inequality follows since $\mathsf{LP}_2^* \geq \lambda V_1$. Therefore,

$$
\begin{aligned}
w(S^*) \geq u'(S^*) &\geq \frac{\sum_S u'(S)x'_S}{\sum_S x'_S} \geq \frac{\sum_S w(S)x'_S - (B^{\mathsf{new}}+n)}{\mathsf{LP}_2^*/\lambda V_1 + 1} \\
&\geq \frac{\frac{n}{v_{\max}} \cdot \mathsf{LP}_2^* - (B^{\mathsf{new}}+n)}{\frac{\mathsf{LP}_2^*}{\lambda V_1}+1} = \frac{n}{v_{\max}} \cdot \frac{\mathsf{LP}_2^* - \lambda V_1}{\frac{\mathsf{LP}_2^*}{\lambda V_1}+1} - \frac{2n}{\frac{\mathsf{LP}_2^*}{\lambda V_1}+1} \\
\implies \quad v(S^*) &\geq \frac{\mathsf{LP}_2^* - \lambda V_1}{\frac{\mathsf{LP}_2^*}{\lambda V_1}+1} - v_{\max}\Big(\frac{2}{\frac{\mathsf{LP}_2^*}{\lambda V_1}+1}+1\Big) \\
&\geq V_1 \cdot \frac{\lambda(1-\lambda)}{1+\lambda} - v_{\max} \cdot \frac{3\lambda+1}{\lambda+1}.
\end{aligned}
$$

We take $\lambda = 0.5$ and $p = \frac{12}{17}$ so that $\frac{p}{4} \cdot \frac{3\lambda+1}{\lambda+1} = 1-p$. The expected value returned is at least $\frac{p}{4} \cdot v(S^*) + (1-p)v_{\max} \geq OPT_{\mathsf{Bench}}(v, B, c)/272$. □

5 Dropping the no-overbidding assumption

We now describe how to obtain universally-budget-feasible mechanisms for XOS valuations without assuming no-overbidding. Recall that $\mathcal{N} = [k]$ is the set of all players. Now define e^* to be $\operatorname{argmax}\{v(e) : e \in U,\ c(e) \leq B\}$, and let $v_{\max} = v(e^*)$. Let $\mathsf{opt}^* = \max_{i \in \mathcal{N}} \max_{S \subseteq G_i} \big\{v(S) : c_i(S) \leq B\big\}$.

As discussed in Section 1.2, dropping this assumption entails figuring out a way of offsetting the additive loss incurred in arguing the existence of a large-value set $T \subseteq U_2$ with $v(T) \leq \lambda V_1$, $c(T) \leq B/2$. While this additive loss is bounded by opt^*, as noted earlier, the mechanism that returns $S \subseteq G_i$, for some player i, with $v(S) = \mathsf{opt}^*$, $c(S) \leq B$, is not truthfully implementable.[14] So we need to devise an alternative to this "return (set corresponding to) opt^* mechanism." We devise such a mechanism in Section 5.1, and show in Section 5.2 how the analyses of algorithms from Section 4.2 can be modified so as to leverage the guarantee of this mechanism and obtain approximation guarantees relative to OPT_{Bench}.

5.1 Truthful mechanism to offset additive loss

Recall that given cost functions $\{c_i \in \mathcal{C}_i\}$, we define $c(S) := \sum_i c_i(S \cap G_i)$ for all $S \subseteq U$.

Mechanism 2NDOPT // budget-feasible mechanism: substitute for returning e^*

Input: Budget-feasible MD instance $\big(v : 2^U \mapsto \mathbb{R}_+, B, \{G_i, \mathcal{C}_i \subseteq \mathbb{R}_+^{2^{G_i}}, c_i \in \mathcal{C}_i\}\big)$
Output: subset of G_i for some player i; payment = B

For every player i, define $\mathsf{opt}_i := \max\big\{v(S) : S \subseteq G_i,\ c(S) \leq B\big\}$. Also, for each player i, define $\mathcal{S}_i = \mathcal{S}_i(c_{-i}) = \{S \subseteq G_i : v(S) \geq \max_{j<i} \mathsf{opt}_j,\ v(S) > \max_{j>i} \mathsf{opt}_j\}$.
Choose $\widehat{S} = \operatorname{argmin}\{c(S) : S \in \bigcup_{i \in \mathcal{N}} \mathcal{S}_i\}$, and let $\widehat{S} \in \mathcal{S}_{\widehat{i}}$. (We show in the analysis that $\bigcup_i \mathcal{S}_i \neq \emptyset$.)
return $\widehat{S}$, and pay B to player $\widehat{i}$.

Lemma 5.1. *There exists some player i' and some $S' \in \mathcal{S}_{i'}$ such that $c_{i'}(S') \leq B$.*

[14]Concretely, suppose $B = 4$, there is one player, $U = \{e, f\}$, v is additive with $v(e) > v(f)$, and we have two additive cost functions $c^{(1)}$, $c^{(2)}$, given by $c_e^{(1)} = B$, $c_f^{(1)} = 1$ and $c_e^{(2)} = B+1$, $c_f^{(2)} = B$. When the true cost function is $c^{(1)}$, truthful reporting would yield utility 0, whereas reporting $c^{(2)}$ would yield utility $B - 1 > 0$.

Proof. Let i' be the player with largest index such that $\mathsf{opt}_{i'} = \max_{i \in \mathcal{N}} \mathsf{opt}_i$, and let $S' \subseteq G_{i'}$ be such that $v(S') = \mathsf{opt}_{i'}$ and $c_{i'}(S') \leq B$. By definition then, we have $v(S') \geq \mathsf{opt}_j$ for all $j < i$, and $v(S') > \mathsf{opt}_j$ for all $j > i'$. So $S' \in \mathcal{S}_{i'}$. □

Theorem 5.2. *Mechanism 2ndOpt is budget-feasible, and its output $\widehat{S} \subseteq G_{\widehat{i}}$ satisfies $v(\widehat{S}) \geq \max_{i \neq \widehat{i}} \mathsf{opt}_i = \max\{v(S) : S \subseteq G_i$ for some $i \neq \widehat{i},\ c(S) \leq B\}$.*

Proof. Lemma 5.1 shows that the mechanism is well defined. The performance-guarantee statement follows from construction, since we have $\widehat{S} \in \mathcal{S}_{\widehat{i}}$. The payment made is B by construction. Individual rationality follows from Lemma 5.1, since this implies that there is some $S \in \bigcup_i \mathcal{S}_i$ with $c(S) \leq B$, We focus on proving truthfulness.

The key observation is that a player i cannot affect her collection of sets $\mathcal{S}_i$. We first claim that player $\widehat{i}$ cannot benefit by lying. We have that $\widehat{S}$ is a minimum-cost set from $\mathcal{S}_{\widehat{i}}$. So since $c_{\widehat{i}}$ does not affect $\mathcal{S}_{\widehat{i}}$, player $\widehat{i}$ cannot lie and cause a lower $c_{\widehat{i}}$-cost subset of $G_{\widehat{i}}$ to be chosen.

Next, consider a player $i \neq \widehat{i}$. We show that $c_i(S_i) > B$ for every $S \in \mathcal{S}_i$, which implies that i cannot obtain positive utility by reporting some $c' \in \mathcal{C}_i$, $c' \neq c_i$. Again, $\mathcal{S}_i$ does not depend on player i's reported cost (so we can unambiguously say $\mathcal{S}_i$ given that c_{-i} is fixed). Suppose $i < \widehat{i}$. We have $v(\widehat{S}) \geq \mathsf{opt}_i$, and so $v(\widehat{S}) \geq v(S)$ for every $S \subseteq G_i$ with $c_i(S) \leq B$. Therefore, if $S \subseteq G_i$ is such that $v(S) > \mathsf{opt}_{\widehat{i}} \geq v(\widehat{S})$, then we must have $c_i(S) > B$. Similarly, suppose $i > \widehat{i}$. Then, we have $v(\widehat{S}) > \mathsf{opt}_i$, and so $v(\widehat{S}) > v(S)$ for every $S \subseteq G_i$ with $c_i(S) \leq B$. So if $S \subseteq G_i$ satisfies $v(S) \geq \mathsf{opt}_{\widehat{i}} \geq v(\widehat{S})$, then we must again have $c_i(S) > B$. □

We obtain the following immediate corollary. Let $\mathsf{opt}^{(2)}$ be the second-largest opt_i value.

Corollary 5.3. *The set $\widehat{S}$ returned by Mechanism 2ndOpt satisfies $v(\widehat{S}) \geq \mathsf{opt}^{(2)}$.*

Remark 5.4. We remark that the asymmetry in the definition of $\mathcal{S}_i$ in Mechanism 2ndOpt is crucial. The proof evidently exploits this, and we can show that if we change the definition of $\mathcal{S}_i$ to $\{S \subseteq G_i : v(S) \geq \max_{j \neq i} \mathsf{opt}_j\}$, then we do not obtain truthfulness.

Remark 5.5 (Computation using oracles). Computing the set $\widehat{S}$ requires two types of oracles: an oracle for computing the opt_i quantities, and a *knapsack-cover oracle* [53] to find a minimum-cost set in $\mathcal{S}_i$ for every player i, in step 1. In the multidimensional setting, a *knapsack-cover oracle for the class $\mathcal{C} = \Pi_i \mathcal{C}_i$* takes $q \in \mathcal{C}$ and a target value Val as input, and returns $\operatorname{argmin}_{S \subseteq U} \{q(S) : v(S) \geq Val\}$, or "infeasible", if no feasible set exists. As with a demand- and constrained- demand oracle, we assume, somewhat more generally, that we can specify a subset $A \subseteq U$ of the form $A = \bigcup_{i \in I} G_i$, for some $I \subseteq [k]$, and the oracle returns the optimum over A (as opposed to U).[15]

Given a knapsack-cover oracle, we can compute $\widehat{S}$ as follows. By scaling, we may assume that $v(S)$ is an integer for all $S \subseteq U$. For every player i, we compute $S_i^* = \operatorname{argmin}_{S \in \mathcal{S}_i} c_i(S)$ by calling the knapsack-cover oracle with the set $A = G_i$, $q = c$, and target value $Val_i = \max\{\max_{j<i} \mathsf{opt}_j, 1 + \max_{j>i} \mathsf{opt}_j\}$. We then return $\widehat{S} = \operatorname{argmin}\{c_i(S_i^*) : i \in \mathcal{N}\}$.

5.2 Universally-budget-feasible mechanisms without assuming no-overbidding

We now utilize Mechanism 2ndOpt (and variants of it) in conjunction with mechanisms from Section 4.2 (with some changes) to obtain *universally-budget feasible mechanisms that achieve $O(1)$-approximation with respect to OPT_{Bench}, without assuming no-overbidding*.

[15] This can be achieved by taking q_ℓ for $\ell \notin I$ to be the constant function $q_\ell(S) = M$ for all $\emptyset \neq S \subseteq G_\ell$, where M is sufficiently large, say $2|A| \cdot \max_{i \in A} q_i(G_i)$. We call the knapsack-cover oracle with this modified q, and return the oracle's output if the output is "infeasible", or a set $S \subseteq A$; otherwise, we return "infeasible".

Recall that $\mathsf{opt}_i = \max\{v(S) : S \subseteq G_i,\ c(S) \le B\}$, and $\mathsf{opt}^{(2)}$ is the second-largest opt_i value. Examining the analysis of Algorithm XOS-UniBF in Theorem 4.6, we see that one of the places where we incur a loss in value is when we argue the existence of a large-value set of cost at most $B/2$, by applying Lemma 2.7 to a suitable set $S \subseteq U_2$ (the set T_2^* in the proof of Theorem 4.6): this incurs a loss bounded by roughly $O\big(\max_{i\in\mathcal{N}_2} v(S \cap G_i)\big)$ (due to the $g_{-1}(S)$ term in the statement of Lemma 2.7). The key insight is that we can now recover this loss using Mechanism 2ndOpt, provided that we can engineer things *in the analysis* so that $\max_{i\in\mathcal{N}_2} v(S \cap U_i)$ is at most $\mathsf{opt}^{(2)}$, and $v(S)$ is large. This requires a careful application of the random-partitioning lemma (Lemma 2.4) coupled with some additional observations. The procedure is described in Algorithm XOS-Gen below, and follows the same template as in prior algorithms, but we use Mechanism 2ndOpt in place of the "return-e^* mechanism."

In Section 5.2.1, we consider additive valuations and additive costs, and obtain a *polytime* $O(1)$-factor approximation, universally budget-feasible mechanism for such instances. Here, we exploit the fact that with additive valuations and costs, various computations in Mechanism 2ndOpt and the mechanism from Section 4.2 amount to solving knapsack or knapsack-cover problems, and we can make suitable changes to these mechanisms to move to related problems that can be solved efficiently using dynamic programming. In Section 5.2.2, we consider general XOS valuations and superadditive cost functions, and devise polytime mechanisms given access to a constrained demand oracle. The crucial (and only) change here compared to Algorithm XOS-Gen lies in coming up with a different "version" of Mechanism 2ndOpt that can be implemented in polytime using a constrained demand oracle.

Algorithm XOS-GEN // universally budget-feasible mechanism: general costs without no-overbidding

---

Input: Budget-feasible MD instance $\big(v : 2^U \mapsto \mathbb{R}_+, B, \{G_i, \mathcal{C}_i \subseteq \mathbb{R}_+^{2^{G_i}}\}, \{c_i \in \mathcal{C}_i\}\big)$; parameters $\lambda \in [0, 0.5], p \in [0,1]$

Output: subset of U; payments are VCG payments

Independently, for each player $i \in \mathcal{N}$, place i in $\mathcal{N}_1$ or $\mathcal{N}_2$, each with probability $\frac{1}{2}$. For $j = 1, 2$, let $U_j := \bigcup_{i\in\mathcal{N}_j} G_i$.

Compute $V_1 = V_1^* := \max_{S\subseteq U_1} \{v(S) :\ c(S) \le B\}$.

Use a constrained demand oracle to obtain $S^* \leftarrow \operatorname{argmax}_{S\subseteq U_2} \big\{v(S) - \frac{\lambda V_1}{B} \cdot c(S) :\ v(S) \le \lambda V_1\big\}$.

return S^* with probability p and the output of Mechanism 2ndOpt with probability $1 - p$.

Theorem 5.6. *Taking $\lambda = 0.5$ and $p = \frac{128}{145}$ in Algorithm XOS-Gen, together with suitable payments, we obtain a universally budget-feasible mechanism that obtains expected value at least $\frac{1}{145} \cdot OPT_{\mathsf{Bench}}(v, B, c)$.*

Proof. Universal budget-feasibility follows from the same arguments as in the proof of Lemma 4.7, so we focus on proving the approximation guarantee. We may assume that $n \ge 2$, otherwise $OPT_{\mathsf{Bench}} = 0$, and there is nothing to be shown.

We follow a similar approach as in the proof of Theorem 4.6. As alluded to earlier, we carefully identify a set $S \subseteq U_2$ with $c(S) \le B$ such that when we apply Lemma 2.7 to S to extract a subset of cost at most $B/2$ and value roughly λV_1, the additive loss incurred (relative to λV_1) can be charged to $\mathsf{opt}^{(2)}$. More precisely, we identify such a set S with $v(S) = \frac{1}{O(1)} \cdot \big(OPT_{\mathsf{Bench}} - \mathsf{opt}^{(2)}\big)$ assuming a certain good event happens, and argue that this good event happens with constant probability. Coupled with the $opt^{(2)}$ value returned by Mechanism 2ndOpt, this will yield the stated approximation bound.

Let $i^* \in \mathcal{N}$ be such that $\mathsf{opt}_{i^*} = \max_{i\in\mathcal{N}} \mathsf{opt}_i$, and let $G^* = G_{i^*}$. Essentially, the idea is that since opt_{i^*} is the only opt_i quantity that we cannot recover using Mechanism 2ndOpt, we simply consider the effect of random partitioning after excluding i^*, noting that excluding G^* only incurs a bounded loss in value. Let $O^* = \text{argmax}_{S\subseteq U}\{v_{-1}(S) : c(S) \le B\}$. Define $\overline{O} = O^* - G^*$, and $\overline{O}_j = \overline{O} \cap U_j$ for $j = 1, 2$. Note that $v(\overline{O}) \ge v_{-1}(O^*) = OPT_{\mathsf{Bench}}$. Also, for any $S \subseteq U - G^*$ with $c(S) \le B$, we have $v_{-1}(S) \ge v(S) - \mathsf{opt}^{(2)}$, since by definition, for every $i \in \mathcal{N}$ with $S \cap G_i \ne \emptyset$, we have $i \ne i^*$ and so $v(S \cap G_i) \le \mathsf{opt}_i \le \mathsf{opt}^{(2)}$. In particular, we have $v_{-1}(\overline{O}) \ge OPT_{\mathsf{Bench}} - \mathsf{opt}^{(2)}$.

Let Γ be the event that $v(\overline{O}_1), v(\overline{O}_2) \ge v_{-1}(\overline{O})/4$. Applying Lemma 2.4 (a) to $\overline{O}$, we obtain that $\Pr[\Gamma] \ge 0.5$. Let $V'_j = \max\{v(S) : S \subseteq U_j - G^*,\ c(S) \le B\}$ for $j = 1, 2$. Let $T'_2 \subseteq U_2 - G^*$ be such that $v(T'_2) = V'_2$ and $c(T'_2) \le B$. Note that V'_1 and V'_2 are identically distributed, and this remains true even when we condition on Γ. So we have $\Pr[\{V'_2 \ge V'_1\} \cap \Gamma] \ge \frac{1}{4}$. Finally, let Ω be the event $G^* \subseteq U_2$. Clearly, $\Pr[\Omega] = 0.5$. Event Ω depends only on the random choice made for player i^*, whereas event Γ and the random variables V'_1, V'_2 depend only on the random choices made for the other players. So Ω and the event $\{V'_2 \ge V_1\} \cap \Gamma$ are independent, and we have $\Pr[\{V'_2 \ge V'_1\} \cap \Gamma \cap \Omega] \ge \frac{1}{8}$.

Let us condition on the good event $\{V'_2 \ge V'_1\} \cap \Gamma \cap \Omega$. Note then that since $G^* \cap U_1 = \emptyset$, we have $V'_1 = V_1 = V^*_1$. Applying Lemma 2.7 on T'_2 with $\mathit{Val} = \lambda V_1$, we can obtain $T \subseteq T'_2$ such that $c(T) \le B/2$ and $\min\{v_{-1}(T'_2) - \lambda V_1, \lambda V_1 - \max_{e\in T'_2} v(e)\} < v(T) \le \lambda V_1$. We also have $v_{-1}(T'_2) \ge V'_2 - \mathsf{opt}^{(2)}$, and $\max_{e\in T'_2} v(e) \le \max_{i\in\mathcal{N}_2} v(T'_2 \cap G_i) \le \mathsf{opt}^{(2)}$. Since $V'_2 \ge 2\lambda V_1$, it follows that $v(T) > \lambda V_1 - \mathsf{opt}^{(2)}$. So we have

$$v(S^*) \ge v(S^*) - \frac{\lambda V_1}{B} \cdot c(S^*) \ge v(T) - \frac{\lambda V_1}{B} \cdot c(T) \ge \lambda V_1 - \mathsf{opt}^{(2)} - \frac{\lambda V_1}{2} \ge \frac{\lambda V_1}{2} - \mathsf{opt}^{(2)}.$$

We also have

$$V_1 \ge v(\overline{O}_1) \ge \frac{v_{-1}(\overline{O})}{4} \ge \frac{OPT_{\mathsf{Bench}} - \mathsf{opt}^{(2)}}{4}.$$

By Corollary 5.3, the value obtained by Mechanism 2ndOpt is at least $\mathsf{opt}^{(2)}$. Putting everything together, we obtain that the expected value returned is at least

$$p \cdot \frac{1}{8} \cdot \left(\frac{1}{16} \cdot \left(OPT_{\mathsf{Bench}} - \mathsf{opt}^{(2)}\right) - \mathsf{opt}^{(2)}\right) + (1-p)\mathsf{opt}^{(2)} \tag{3}$$

$$= \frac{p}{128} \cdot OPT_{\mathsf{Bench}} = \frac{1}{145} \cdot OPT_{\mathsf{Bench}}. \quad \square$$

5.2.1 Polytime mechanism for additive valuations and additive costs

With additive valuations and additive cost functions, a constrained demand oracle involves solving a knapsack problem. The computation of opt_i's in Mechanism 2ndOpt also amounts to solving knapsack problems. We argue that, using scaling and rounding, we can instead work with related knapsack problems that can be solved efficiently using dynamic programming (DP), and thereby obtain a polytime $O(1)$-approximation mechanism. But we need to exercise some care, because we cannot use $v_{\max}$ in the scaling.

We discuss below the changes to Algorithm XOS-Gen and Mechanism 2ndOpt; Algorithm Additive contains the entire description.

1. In step 2 of Algorithm XOS-Gen, we now use any β-approximation algorithm for the knapsack problem, where $\beta \ge 1$, to obtain $V_1 \ge V^*_1/\beta$.
2. In step 3 of Algorithm XOS-Gen, we compute S^* by solving the following roughly-equivalent knapsack problem. Set $B^{\mathsf{new}} = 4n$, and $w_e = \left\lceil \frac{nv(e)}{\lambda V_1} \right\rceil$ and $u'_e = w_e - \frac{B^{\mathsf{new}}}{B} \cdot c(e)$ for all $e \in U_2$. Solve the knapsack problem over U_2, with item-values $\{u'_e\}$, item weights $\{w_e\}$ and knapsack budget B^{new}, to obtain S^*. This takes polytime since $B^{\mathsf{new}} = 4n$. (Without the $\lceil . \rceil$ in the w_es, this would be the same as the problem $\max_{S\subseteq U_2}\left\{v(S) - \frac{4\lambda V_1}{B} \cdot c(S) : v(S) \le 4\lambda V_1\right\}$.)

3. Dealing with the (efficient computation of the) opt_i quantities in Mechanism 2ndOpt poses more difficulties, and the workaround is more involved. The issue is that the truthfulness guarantee of Mechanism 2ndOpt relies crucially on the fact that we are working with the exact opt_i quantities. To make this more approximation-friendly, we use random partitioning (again!). We obtain an estimate of $\max_i \mathsf{opt}_i$ from one part, for which we can use any approximation algorithm for computing the opt_is. For each player i in the second part, we solve the *knapsack-cover problem* of finding a min-cost set $T_i^* \subseteq G_i$ whose value is at least this estimate. Finding the smallest-index player i (from the second part) for which $c_i(T_i^*) \leq B$, if one exists, returning T_i^* and paying B to player i, yields a truthful mechanism with a guarantee similar to that stated in Theorem 5.2 for Mechanism 2ndOpt. Finally, since the valuation and costs are additive, we use scaling and rounding to solve a related knapsack-cover problem in polynomial time, which suffices.

   This modified version of Mechanism 2ndOpt is described below, and Theorem 5.7 states its performance guarantee.

Mechanism 2NDOPT-POLY

Input: Budget-feasible MD instance $\big(v \in \mathbb{R}_+^U, B, \{G_i, c_i \in \mathbb{R}_+^{G_i}\}\big)$ with additive valuation and additive costs

Output: subset of G_i for some player i; payment = B

1 For every player i, define $\mathsf{opt}_i := \max_{S \subseteq G_i} \big\{v(S) : c_i(S) \leq B\big\}$. Partition $\mathcal{N}$ into two sets $\mathcal{N}_1, \mathcal{N}_2$ by placing each player independently with probability $\frac{1}{2}$ in $\mathcal{N}_1$ or $\mathcal{N}_2$.

2 For every $i \in \mathcal{N}_1$, compute opt'_i, a γ-approximation to opt_i. Set $Val := \max_{i \in \mathcal{N}_1} \mathsf{opt}'_i$.

3 For every $i \in \mathcal{N}_2$, do the following. Set $\mathsf{wt}_e = \big\lfloor \frac{2nv_e}{Val} \big\rfloor$ for all $e \in G_i$. Let $T_i^* \subseteq G_i$ be an optimal solution to the following knapsack-cover problem: $\min_{S \subseteq G_i} \{c_i(S) : \mathsf{wt}(S) \geq n\}$.

4 If $c_i(T_i^*) > B$ for all $i \in \mathcal{N}_2$, **return** $\emptyset$ and make no payment to any player. Otherwise, let $\widehat{i} \in \mathcal{N}_2$ be the smallest index $i \in \mathcal{N}_2$ such that $c_i(T_i^*) \leq B$; **return** $\widehat{S} = T_{\widehat{i}}^*$ and pay B to player $\widehat{i}$.

We run steps 1–3 of Algorithm XOS-Gen with the above changes, and return S^* or the output of Mechanism 2ndOpt-Poly, each with some probability. The entire algorithm is descibed below.

Algorithm ADDITIVE // polytime universally budget-feasible mechanism: additive valuation and costs, without no-overbidding

---

Input: Budget-feasible MD instance $\big(v \in \mathbb{R}_+^U, B, \{G_i, c_i \in \mathbb{R}_+^{G_i}\}\big)$ with additive valuation and additive costs; parameters $\lambda \in [0, 0.5]$, $p \in [0, 1]$, $r \in \mathbb{Z}_+, r \geq 2$.

Output: subset of U; payments are VCG payments

1 Independently, for each player $i \in \mathcal{N}$, place i in $\mathcal{N}_1$ or $\mathcal{N}_2$, each with probability $\frac{1}{2}$. For $j = 1, 2$, let $U_j := \bigcup_{i \in \mathcal{N}_j} G_i$.

2 Compute V_1, a β-approximation to $V_1^* := \max_{S \subseteq U_1} \{v(S) : \ c(S) \leq B\}$.

3 Set $w_e = \left\lceil \frac{nv(e)}{\lambda V_1} \right\rceil$ and $u'_e = w_e - \frac{B^{\mathsf{new}}}{B} \cdot c(e)$ for all $e \in U_2$. Compute an optimal solution S^* to the knapsack problem over U_2, with item-values $\{u'_e\}$, item weights $\{w_e\}$ and knapsack budget $B^{\mathsf{new}} = 4n$.

4 **return** S^* with probability p and the output of Mechanism 2ndOpt-Poly with probability $1 - p$.

Theorem 5.7 states the performance guarantee of Mechanism 2ndOpt-Poly. We first prove that Mechanism Additive is an $O(1)$-approximation, universally budget-feasible mechanism assuming Theorem 5.7, and then prove Theorem 5.7.

Theorem 5.7. *Mechanism 2ndOpt-Poly is a polytime, universally budget-feasible mechanism that with probability at least* $\frac{1}{4}$*, returns a set* $\widehat{S}$ *such that* $v(\widehat{S}) \geq \frac{\mathsf{opt}^{(2)}}{2\gamma}$*, where* $\mathsf{opt}^{(2)}$ *is the second-largest* opt_i *value.*

Theorem 5.8. *Taking* $\lambda = 0.125$*, and for a suitable choice of* p*, Algorithm Additive, together with suitable payments, yields a polytime, universally budget-feasible mechanism for additive valuations and additive costs that achieves* $O(1)$ *approximation with respect to* OPT_{Bench}*.*

Proof. We proceed similarly to the proof of Theorem 5.6. Let p be such that $\frac{p}{8} \cdot \left(\frac{1}{64\beta} + 1\right) = \frac{1-p}{8\gamma}$. Note that such a $p \in [0, 1]$ always exists. Assume that $n \geq 2$.

Let $i^* \in \mathcal{N}$ be such that $\mathsf{opt}_{i^*} = \max_{i \in \mathcal{N}} \mathsf{opt}_i$, and let $G^* = G_{i^*}$. Let $O^* = \operatorname{argmax}_{S \subseteq U} \{v_{-1}(S) : c(S) \leq B\}$. Define $\overline{O} = O^* - G^*$, and $\overline{O}_j = \overline{O} \cap U_j$ for $j = 1, 2$. We have $v_{-1}(\overline{O}) \geq OPT_{\mathsf{Bench}} - \mathsf{opt}^{(2)}$ by construction. Let $V'_j = \max\{v(S) : S \subseteq U_j - G^*,\ c(S) \leq B\}$ for $j = 1, 2$. Let $T'_2 \subseteq U_2 - G^*$ be such that $v(T'_2) = V'_2$ and $c(T'_2) \leq B$.

Let Γ be the event that $v(\overline{O}_1), v(\overline{O}_2) \geq v_{-1}(\overline{O})/4$, and Ω be the event that $G^* \subseteq U_2$. Applying Lemma 2.4 (a) to $\overline{O}$, we obtain that $\Pr[\Gamma] \geq 0.5$. V'_1 and V'_2 are identically distributed, which also holds when we condition on Γ. So $\Pr[\{V'_2 \geq V'_1\} \cap \Gamma] \geq \frac{1}{4}$. Finally, event Ω is independent of events Γ and $\{V'_2 \geq V'_1\}$, since Ω depends only on the random choice for player i^*, and events Γ and the random variables V'_1, V'_2 depend only on the random choices for the other players. The upshot is that $\Pr[\{V'_2 \geq V'_1\} \cap \Gamma \cap \Omega] \geq \frac{1}{8}$.

We condition on this good event $\{V'_2 \geq V'_1\} \cap \Gamma \cap \Omega$. Since $G^* \cap U_1 = \emptyset$, we have $V'_1 = V^*_1$ and $V_1 \geq V^*_1/\beta$. Since $v(T'_2) \geq V'_2 \geq 8\lambda V'_1$, we have $w(T'_2) \geq 8n$. We now apply Lemma 2.8 with $\mathit{Val} = 4n$ to the additive valuation given by the $\{w_e\}_{e \in U_2}$ weights, to obtain $T \subseteq T'_2$ such that $c(T) \leq B/2$ and $4n - \max_{e \in T'_2} w_e < w(T) \leq 4n$. We have $\max_{e \in T'_2} w_e \leq \frac{n}{\lambda V_1} \cdot \max_{e \in T'_2} v(e) + 1 \leq \frac{n}{\lambda V_1} \cdot \mathsf{opt}^{(2)} + 1$, where the last inequality is because we have $T'_2 \subseteq U_2 - G^*$. Recall that the set S^* computed in step 3 of Algorithm Additive, is the optimal solution to the knapsack problem over U_2 with item values $u'_e = w_e - \frac{B^{\mathsf{new}}}{B} \cdot c_e$ for all $e \in U_2$, item weights $\{w_e\}$ and knapsack budget $B^{\mathsf{new}} = 4n$. So the above implies that

$$w(S^*) \geq u'(S^*) \geq u'(T) \geq w(T) - \frac{B^{\mathsf{new}}}{2} \geq 2n - \frac{n}{\lambda V_1} \cdot \mathsf{opt}^{(2)} - 1.$$

$$\text{Therefore} \quad v(S^*) \geq \frac{\lambda V_1}{n} \cdot \big(w(S^*) - n\big) \geq \lambda V_1 \Big(1 - \tfrac{1}{n}\Big) - \mathsf{opt}^{(2)} \geq \frac{\lambda V^*_1}{2\beta} - \mathsf{opt}^{(2)},$$

where the last inequality follows since $n \geq 2$. As in the proof of Theorem 5.6, we have $V^*_1 \geq \frac{OPT_{\mathsf{Bench}} - opt^{(2)}}{4}$. By Theorem 5.7, the expected value returned by Mechanism 2ndOpt-Poly is at least $\frac{1}{4} \cdot \frac{\mathsf{opt}^{(2)}}{2\gamma}$. So the expected value returned by Mechanism Additive is at least

$$p \cdot \frac{1}{8} \cdot \left(\frac{1}{64\beta} \cdot \big(OPT_{\mathsf{Bench}} - \mathsf{opt}^{(2)}\big) - \mathsf{opt}^{(2)}\right) + (1-p) \cdot \frac{\mathsf{opt}^{(2)}}{8\gamma} = \frac{p}{512\beta} \cdot OPT_{\mathsf{Bench}}. \qquad \square$$

Proof of Theorem 5.7. The mechanism runs in polytime because the knapsack-cover problem in step 3 takes polytime as the target value n is polynomially bounded, and the wt_e's are integers. Payment of at most B, and individual rationality are baked into the mechanism. We focus on proving truthfulness and the performance guarantee.

Clearly, players in $\mathcal{N}_1$ always get 0 utility, so again nothing by lying. For each $i \in \mathcal{N}_2$, the collection $\mathcal{S}_i = \{S \subseteq G_i : \mathsf{wt}(S) \geq n\}$ of feasible sets for player i does not depend on the cost-vector c. If the mechanism outputs $\emptyset$, then every set $S \in \bigcup_{i \in \mathcal{N}_2} \mathcal{S}_i$ has $c(S) > B$, so no player can achieve positive utility by lying. Suppose the mechanism outputs $\widehat{S} \subseteq G_{\widehat{i}}$. Player $\widehat{i}$ cannot benefit by lying, since $\widehat{S}$ already has minimum cost in $\mathcal{S}_{\widehat{i}}$ among her feasible sets. Consider a player $i \in \mathcal{N}_2$, $i \neq \widehat{i}$. If $i < \widehat{i}$, then by the choice of $\widehat{i}$, we have $c_i(T^*_i) > B$, so player i cannot achieve positive utility by lying. If $i > \widehat{i}$, then i will never be chosen in step 4, so again i cannot benefit by lying.

Let $\widetilde{i}$ and $\widetilde{j}$ be players in $\mathcal{N}$ having the largest and second-largest opt_i values respectively among all players in $\mathcal{N}$. With probability $\frac{1}{4}$, we have $\widetilde{j} \in \mathcal{N}_1$, $\widetilde{i} \in \mathcal{N}_2$. Assume that this event happens. Then, we have $Val \geq \frac{\mathsf{opt}_{\widetilde{j}}}{\gamma} = \frac{\mathsf{opt}^{(2)}}{\gamma}$, and there is some set $T_{\widetilde{i}} \subseteq G_{\widetilde{i}}$ such that $v(T_{\widetilde{i}}) = \mathsf{opt}_{\widetilde{i}} \geq Val$, $c_{\widetilde{i}}(T_{\widetilde{i}}) \leq B$. This implies that $\mathsf{wt}(T_{\widetilde{i}}) \geq n$, so we must have $c(S) \leq B$ for some $S \in \bigcup_{i \in \mathcal{N}_2} \mathcal{S}_i$. So the mechanism will output a set $\widehat{S}$ with $\mathsf{wt}(\widehat{S}) \geq n$, which implies that $v(\widehat{S}) \geq \frac{Val}{2} \geq \frac{\mathsf{opt}^{(2)}}{2\gamma}$. □

5.2.2 Superadditive costs: polytime mechanism using constrained demand oracle

For general XOS valuations and superadditive cost functions, we devise a polytime $O(1)$-approximation universally budget-feasible mechanism given access to a constrained demand oracle. This generalizes the result in Section 4.2.1, where we assumed no-overbidding. The only change to Algorithm XOS-Gen for general costs is that we run a new mechanism (Mechanism 2ndOpt-CDemd) in place of Mechanism 2ndOpt in step 4 of the algorithm. We first describe this new mechanism, and then show that the corresponding modified-version of Algorithm XOS-Gen yields an $O(1)$-approximation universally budget-feasible mechanism (Theorem 5.11).

Mechanism to replace Mechanism 2ndOpt. We extend the ideas underlying Mechanism 2ndOpt-Poly, which was used for additive valuations and costs. The key is to view Mechanism 2ndOpt-Poly as a means of combining single-player budget-feasible mechanisms—the mechanism for player $i \in \mathcal{N}_2$ outputs T_i^* computed in step 3 and payment B, if $c(T_i^*) \leq B$, and $\emptyset$ and zero payment, otherwise—while preserving truthfulness (and budget-feasibility).

We make this framework for combining single-player budget-feasible mechanisms explicit, and devise suitable single-player mechanisms that utilize a constrained demand oracle. To elaborate, we devise two budget-feasible mechanisms $\mathcal{M}_i^{(1)}, \mathcal{M}_i^{(2)}$ for each player i. Both mechanisms take a target value Val as a parameter, and their outputs have the property that if $\mathsf{opt}_i \geq Val$, then at least one of mechanisms returns expected value $\Omega(Val)$ (Theorem 5.9). Both mechanisms also output a "success bit," which if set to 0, indicates that player i receives zero utility under truthful reporting. We combine this collection of mechanisms to obtain expected value $\Omega\big(\mathsf{opt}^{(2)}\big)$ while maintaining budget-feasibility, by using random partitioning. We find, from one part $\mathcal{N}_1$, the right target Val to aim for. Then, we pick $j = 1$ or 2 with probability 0.5. We select the first player $i \in \mathcal{N}_2$ whose $\mathcal{M}_i^{(j)}$ mechanism's success bit is set to 1, and return the output of $\mathcal{M}_i^{(j)}$; if no bit is set to 1, we output $\emptyset$ and 0 payment. We now delve into the details.

Mechanisms: $\boldsymbol{\mathcal{M}_i^{(1)}}$ and $\boldsymbol{\mathcal{M}_i^{(2)}}$ for player $\boldsymbol{i \in \mathcal{N}}$

Input: Valuation $v : 2^U \mapsto \mathbb{R}_+$, budget B, item-set G_i, superadditive cost function $c_i : 2^{G_i} \mapsto \mathbb{R}_+$, $c_i \in \mathcal{C}_i$, target $Val \in \mathbb{R}_+$

Output: subset of G_i, payment to player i, success bit flag_i

Mechanism $\boldsymbol{\mathcal{M}_i^{(1)}}$

1 Compute $T_i^* \leftarrow \operatorname{argmax}_{S \subseteq G_i} \big\{v(S) - \frac{Val}{2B} \cdot c_i(S) : \ v(S) \leq \frac{Val}{2}\big\}$ using a constrained demand oracle.

2 If $v(T_i^*) - \frac{Val}{2B} \cdot c_i(T_i^*) \geq \frac{Val}{8}$, then **return** T_i^*, payment $\frac{2B}{Val} \cdot v(T_i^*) - \frac{B}{4}$, $\mathsf{flag}_i = 1$; otherwise **return** $\emptyset$, zero payment, $\mathsf{flag}_i = 0$

Mechanism $\boldsymbol{\mathcal{M}_i^{(2)}}$

1 Let $e_i^* = \operatorname{argmin}_{e \in G_i} \big\{c_i(e) : v(e) \geq \frac{Val}{8}\big\}$

2 If $c_i(e) \leq B$, **return** T_i^*, payment B, $\mathsf{flag}_i = 1$; otherwise **return** $\emptyset$, zero payment, $\mathsf{flag}_i = 0$

Mechanism 2NDOPT-CDEMD   // Combining the $\{\mathcal{M}_i^{(1)}, \mathcal{M}_i^{(2)}\}_{i\in\mathcal{N}}$ mechanisms

Input: Budget-feasible MD instance $\big(v : 2^U \mapsto \mathbb{R}_+, B, \{G_i, \mathcal{C}_i \subseteq \mathbb{R}_+^{2^{G_i}}\}, \{c_i \in \mathcal{C}_i\}\big)$ with superadditive costs

Output: subset of G_i for some player i, suitable payments

1 For every player i, define $\mathsf{opt}_i := \max_{S\subseteq G_i} \big\{v(S) : c_i(S) \le B\big\}$. Partition $\mathcal{N}$ into two sets $\mathcal{N}_1, \mathcal{N}_2$ by placing each player independently with probability $\frac{1}{2}$ in $\mathcal{N}_1$ or $\mathcal{N}_2$.

2 For every $i \in \mathcal{N}_1$, compute opt'_i, a γ-approximation to opt_i. We can achieve $\gamma = (2+\epsilon)$, for any $\epsilon > 0$, in polytime using a constrained demand oracle; see Section 8.1. Set $\mathit{Val} := \max_{i\in\mathcal{N}_1} \mathsf{opt}'_i$.

3 Set $j = 1$ or $j = 2$, each with probability $\frac{1}{2}$. If for every $i \in \mathcal{N}_2$, mechanism $\mathcal{M}_i^{(j)}$ sets $\mathsf{flag}_i = 0$, **return** $\emptyset$ and make no payment to any player. Otherwise, let $\widehat{i} \in \mathcal{N}_2$ be the smallest index $i \in \mathcal{N}_2$ for which $\mathcal{M}_i^{(j)}$ sets $\mathsf{flag}_i = 1$; **return** the output of $\mathcal{M}_i^{(j)}$ (i.e., subset of G_i, and payment).

Theorem 5.9. *Mechanism 2ndOpt-CDemd is universally budget-feasible, runs in polytime given a constrained demand oracle, and with superadditive cost functions, obtains expected value at least* $\frac{\mathsf{opt}^{(2)}}{32\gamma}$*, where* $\mathsf{opt}^{(2)}$ *is the second-largest* opt_i *value.*

Theorem 5.9 will follow fairly directly from the following result about the $\mathcal{M}_i^{(1)}, \mathcal{M}_i^{(2)}$ mechanisms.

Theorem 5.10. *For any player i, and any parameter Val, mechanisms* $\mathcal{M}_i^{(1)}$ *and* $\mathcal{M}_i^{(2)}$ *are budget-feasible, run in polytime given a constrained demand oracle, and satisfy the following properties.*

(a) If $\mathsf{opt}_i \ge \mathit{Val}$ *and* c_i *is superadditive, then at least one of* $\mathcal{M}_i^{(1)}$ *or* $\mathcal{M}_i^{(2)}$ *obtains value at least* $\frac{\mathit{Val}}{8}$.

(b) If flag_i *is set to* 0 *by* $\mathcal{M}_i^{(j)}$ *for any* $j = 1, 2$*, then player* i *cannot obtain positive utility by lying.*

Proof. It is clear that $\mathcal{M}_i^{(1)}, \mathcal{M}_i^{(2)}$ run in polytime given a constrained demand oracle. We first argue that both $\mathcal{M}_i^{(1)}$ and $\mathcal{M}_i^{(2)}$ are budget feasible. Consider $\mathcal{M}_i^{(1)}$. The payment made is always at most B and at least the cost incurred by i under truthful reporting, since if the mechanism outputs T_i^*, we have $v(T_i^*) \le \frac{\mathit{Val}}{2}$ and $v(T_i^*) - \frac{\mathit{Val}}{2B} \cdot c_i(T_i^*) \ge \frac{\mathit{Val}}{8}$. So the budget constraint is met, and individual rationality (IR) holds. To see truthfulness, suppose that i reports $c'_i \ne c_i$, and some other set $S \subseteq G_i$ is output by the constrained demand oracle. However, we have $v(T_i^*) - \frac{\mathit{Val}}{2B} \cdot c_i(T_i^*) \ge v(S) - \frac{\mathit{Val}}{2B} \cdot c_i(S)$, so the utility of i cannot increase under this mis-report. Moreover if $v(T_i^*) - \frac{\mathit{Val}}{2B} \cdot c_i(T_i^*) < \frac{\mathit{Val}}{8}$, then i obtains negative utility if mis-report causes flag_i to be set to 1.

For $\mathcal{M}_i^{(2)}$, things are even more straightforward. It is immediate that the payment is at most B and IR holds. Truthfulness is also immediate since we are outputting a min-cost element from a feasible set that is not affected by i's reported cost.

Part (a) follows, because by a now-routine analysis, we can show that if $\mathsf{opt}_i \ge \mathit{Val}$, then there is some $S \subseteq G_i$ with $c(S) \le B/2$ and $v(S) \in \big[\frac{\mathit{Val}}{2} - \max_{e\in G_i}\{v(e) : c_i(e) \le B\}, \frac{\mathit{Val}}{2}\big]$. This follows by applying Lemma 2.8 to the set S^* with $c_i(S^*) \le B$ that achieves value opt_i. So if $\max_{e\in G_i}\{v(e) : c_i(e) \le B\} \le \frac{\mathit{Val}}{8}$, then

$$v(T_1^*) - \frac{\mathit{Val}}{2B} \cdot c_i(T_i^*) \ge v(S) - \frac{\mathit{Val}}{2B} \cdot c_i(S) \ge \frac{\mathit{Val}}{2} - \frac{\mathit{Val}}{8} - \frac{\mathit{Val}}{4} = \frac{\mathit{Val}}{8};$$

otherwise $\mathcal{M}_i^{(2)}$ obtains value at least $\frac{\mathit{Val}}{8}$.

Given the truthfulness of $\mathcal{M}_i^{(1)}, \mathcal{M}_i^{(2)}$, part (b) follows from construction. □

Proof of Theorem 5.9. The polynomial running time follows from Theorem 5.10, and since the opt'_i estimates can be efficiently computed using a constrained demand oracle.

As in the proof of Theorem 5.7, we may assume that the largest opt_i corresponds to player in $\mathcal{N}_2$, and the second-largest opt_i value corresponds to a player in $\mathcal{N}_1$. This event happens with probability $\frac{1}{4}$, and assuming this, we have $Val \geq \frac{\mathsf{opt}^{(2)}}{\gamma}$ in Mechanism 2ndOpt-CDemd. So using Theorem 5.10 (a), the expected value returned is at least $\frac{1}{4} \cdot \frac{1}{8} \cdot \frac{\mathsf{opt}^{(2)}}{\gamma}$.

Fix $j \in \{1, 2\}$ as chosen by the mechanism. If Mechanism 2ndOpt-CDemd outputs $\emptyset$, then $\mathsf{flag}_i = 0$ for all $i \in \mathcal{N}_2$, which implies, by Theorem 5.10, that no player can obtain positive utility by lying. So suppose otherwise. Consider any $i \in \mathcal{N}_2$. Player $\widehat{i}$ cannot benefit by lying, since $\mathcal{M}_{\widehat{i}}^{(j)}$ is truthful. If $i > \widehat{i}$, then i will never be chosen in step 3, so i cannot benefit by lying. If $i < \widehat{i}$, then since $\mathsf{flag}_i = 0$, again by Theorem 5.10, player i cannot achieve positive utility by lying and causing flag_i to be set to 1. The budget constraint and IR hold with probability 1, because this holds for all the $\mathcal{M}_i^{(1)}, \mathcal{M}_i^{(2)}$ mechanisms. It follows that Mechanism 2ndOpt-CDemd is universally budget-feasible. $\square$

Algorithm XOS-Gen modified for superadditive costs: proof of Theorem 5.11. As mentioned earlier, the modified algorithm is simply Algorithm XOS-Gen, replacing Mechanism 2ndOpt with Mechanism 2ndOpt-CDemd in step 4 of the algorithm. Therefore taking $\lambda = 0.5$, the exact same analysis as in the proof of Theorem 5.6 leading up to (3) holds, and we obtain that the expected value returned is at least

$$\frac{p}{128} \cdot OPT_{\mathsf{Bench}} - \frac{17p}{128} \cdot \mathsf{opt}^{(2)} + (1-p) \cdot \frac{\mathsf{opt}^{(2)}}{32\gamma}$$

where we have utilized the guarantee in Theorem 5.9 about Mechanism 2ndOpt-CDemd. So taking $p = \frac{1}{1+17\gamma/4}$, we obtain expected value $\frac{1}{128+544\gamma} \cdot OPT_{\mathsf{Bench}}$. All steps run in polytime given a constrained demand oracle. So we obtain the following result.

Theorem 5.11. *Taking $\lambda = 0.5$ and suitable p in the above modified version of Algorithm XOS-Gen, together with suitable payments, we obtain a universally budget-feasible mechanism for superadditive costs that runs in polytime given a constrained demand oracle and achieves $O(1)$-approximation with respect to OPT_{Bench}.*

6 Submodular valuations

We now devise universally budget-feasible mechanisms for monotone submodular functions, which form a subclass of XOS valuations. Our mechanism works for arbitrary costs, without assuming no-overbidding, and runs in polytime given a demand oracle; but it yields a weaker approximation guarantee.

Let v be a monotone, submodular valuation. After obtaining an estimate of OPT_{Bench} via random partitioning, we depart from the template used for the algorithms in Section 4. We now use a greedy algorithm that iteratively considers the players in $\mathcal{N}_2$ in some fixed order, and picks a suitable set from $U_2 \cap G_i$ to add to the current set T using a demand oracle, until $v(T)$ becomes large enough. The analysis proceeds along the lines of the analysis of the standard greedy algorithm for monotone submodular-function maximization, to show that we obtain expected value $OPT_{\mathsf{Bench}}(v, B, c)/48 - \max_i v(G_i)/4$. Thus, under inputs (v, B, c) satisfying the large-market assumption $v(G_i) \leq \varepsilon \cdot OPT_{\mathsf{Alg}}(v, B, c)$, we obtain an $O(1)$-approximation with respect to $OPT_{\mathsf{Alg}}(v, B, c)$.

Our algorithm below uses demand queries of the form $\operatorname{argmax}_{S \subseteq G_i}\big(v(S|T) - \kappa \cdot c_i(S)\big)$, where $T \subseteq U$ and $v(S|T) := v(S \cup T) - v(T)$ is the incremental value of adding S to T. Such a demand-oracle query can be translated to a demand-oracle query for v over the set $T \cup G_i$ by setting by taking c_ℓ to be the identically-0 function over 2^{G_ℓ} for all $\ell \neq i$, and $c_i(A) = c_i(A - T)$ for all $A \subseteq G_i$. This amounts to "setting the prices for elements in T to be 0". (We assume that $\mathcal{C}_i$ is closed under this operation.)

Mechanism SUBMOD-UNIBF // universally-budget-feasible mechanism for monotone submodular valuations, general cost functions

Input: Budget-feasible MD instance $\big(v : 2^U \mapsto \mathbb{R}_+, B, \{G_i, \mathcal{C}_i \subseteq \mathbb{R}_+^{2^{G_i}}\}, \{c_i \in \mathcal{C}_i\}\big)$; parameter $\lambda \in [0, 1]$

Output: subset of U and payments

1 Partition $\mathcal{N}$ into two sets $\mathcal{N}_1, \mathcal{N}_2$ by placing each player independently with probability $\frac{1}{2}$ in $\mathcal{N}_1$ or $\mathcal{N}_2$. Let $U_j := \bigcup_{i \in \mathcal{N}_j} G_i$ be the induced partition of U. Compute LP_1^* using a demand oracle. Set $V_1 = \mathsf{LP}_1^*$.

2 Initialize $T \leftarrow \emptyset$. Considering players $i \in \mathcal{N}_2$ in some fixed order: if $v(T \cup G_i) \leq \lambda V_1$, compute $\mathrm{demd}_i \leftarrow \mathrm{argmax}_{S \subseteq G_i} \big\{v(S|T) - \frac{\lambda V_1}{B} \cdot c_i(S)\big\}$ and set $T_i = T$ and $T \leftarrow T \cup \mathrm{demd}_i$; otherwise, exit the loop.

3 **return** T. Pay $\frac{B}{\lambda V_1} \cdot v(\mathrm{demd}_i\,|T_i)$ to each player $i \in \mathcal{N}_2$ for which $T \cap G_i \neq \emptyset$, and 0 to the other players.

Theorem 6.1. *Taking $\lambda = 0.5$ in Mechanism Submod-UniBF, we obtain a universally budget-feasible mechanism that obtains expected value $OPT_{\mathsf{Bench}}(v, B, c)/48 - \max_{i \in [k]} v(G_i)/4$.*

Proof. Lemma 6.2 proves that the mechanism is universally budget feasible. We prove the approximation guarantee here. We drop (v, B, c) from the argument of OPT_{Bench}. Let $\kappa = \frac{\lambda V_1}{B}$. Let $O_2^* := \mathrm{argmax}_{S \subseteq U_2} \big\{v(S) - \kappa \cdot c(S)\big\}$. We first argue that $v(O_2^*) - \kappa \cdot c(O_2^*) \geq \mathsf{LP}_2^* - \kappa B$. Let x^* be an optimal solution to (BFLP(U_2)). Then, $v(O_2^*) - \kappa \cdot c(O_2^*) \geq \sum_S x_S^*\big(v(S) - \kappa \cdot c(S)\big) \geq \mathsf{LP}_2^* - \kappa B$.

If we exit the loop in step 2 prematurely without going over all the players in $\mathcal{N}_2$, then we clearly have $v(T) \geq \lambda V_1 - \max_{i \in \mathcal{N}_2} v(G_i)$. Suppose otherwise. Then we consider all players in $\mathcal{N}_2$, and if player i' is considered right after player i, we have $T_{i'} = T_i \cup \mathrm{demd}_i$. For any $i \in \mathcal{N}_2$, we have

$$v(\mathrm{demd}_i\,|T_i) - \kappa \cdot c_i(\mathrm{demd}_i) \geq v(O_2^* \cap G_i|T_i) - \kappa \cdot c(O_2^* \cap G_i) \geq v(O_2^* \cap G_i|T) - \kappa \cdot c(O_2^* \cap G_i) \quad (4)$$

where the last inequality follows from submodularity of v. Adding this for all $i \in \mathcal{N}_2$, since $v(\cdot|T)$ is also monotone and submodular, and hence subadditive, and $\{O_2^* \cap G_i\}_{i \in \mathcal{N}_2}$ partitions O_2^*, we obtain that $v(T) - \kappa \cdot c(T) \geq v(O_2^*|T) - \kappa \cdot c(O_2^*) = v(O_2^* \cup T) - v(T) - \kappa \cdot c(O_2^*)$. It follows that $2v(T) \geq v(O_2^*) - \kappa \cdot c(O_2^*) \geq \mathsf{LP}_2^* - \lambda V_1$. Therefore, $v(T) \geq \frac{\mathsf{LP}_2^* - \lambda V_1}{2}$.

By Lemma 2.6, we have that $\mathsf{LP}_2^* \geq V_1 = \mathsf{LP}_1^* \geq \frac{OPT_{\mathsf{Bench}}}{4}$ holds with probability at least $\frac{1}{4}$. Assuming that this event happens, plugging in $\lambda = \frac{1}{3}$, the above analysis shows that that $v(T) \geq \frac{V_1}{3} - \max_{i \in \mathcal{N}_2} v(G_i) \geq \frac{OPT_{\mathsf{Bench}}}{12} - \max_{i \in \mathcal{N}_2} v(G_2)$. So the expected value returned is at least $\frac{OPT_{\mathsf{Bench}}}{48} - \frac{\max_{i \in \mathcal{N}_2} v(G_2)}{4}$. □

Lemma 6.2. *Mechanism Submod-UniBF is universally budget feasible.*

Proof. We consider each possible random outcome of the mechanism, and show that the payments given in step 3 yield a budget-feasible mechanism. Let $(\mathcal{N}_1, U_1)$, $(\mathcal{N}_2, U_2)$ be the partition obtained in step 1. Let $\kappa = \frac{\lambda V_1}{B}$. Fix an input (v, B, c).

Consider a player i. If $i \in \mathcal{N}_1$, then her payment, cost incurred, and utility are always 0. So suppose $i \in \mathcal{N}_2$. We say that player i is *processed* in step 2, if we consider i in the loop in that step and compute a demand-set for i. The key observation is that whether player i is processed, and the set $T = T_i$ at the point when we consider i *do not depend on i's reported cost function*. If i is not processed, then i's payment, cost incurred, and utility are always 0. Otherwise, i's utility is $\frac{1}{\kappa} \cdot v(S|T_i) - c_i(S)$, where S is the demand-set computed for i when we process her. So by construction, this utility is maximized by reporting her true cost function, which yields the set $S = \mathrm{demd}_i$, and we obtain truthfulness.

The total payment made by the mechanism is $\frac{1}{\kappa} \cdot \sum_{i \text{ is processed}} v(\text{demd}_i \,|T_i) = v(T)/\kappa \leq B$, where the last inequality follows because $v(T) \leq \lambda V_1$ by construction. □

7 Subadditive valuations

Our mechanisms for XOS valuations can be utilized to obtain budget-feasible mechanisms for monotone subadditive valuations that achieve $O(\log k)$-approximation with respect to OPT_{Bench}, without assuming no-overbidding. Recall that k is the number of players. In particular, we obtain an $O(\log k)$-approximation budget-feasible-in-expectation mechanism that runs in polytime given a demand oracle, and an $O(\log k)$-approximation universally budget-feasible mechanism. Both mechanisms work for general costs and without assuming no-overbidding.

The idea here is to use an "XOS-like" pointwise-approximation of the subadditive valuation v. However, note that if $\tilde{v}$ is a pointwise approximation of v satisfying $v(S)/\gamma \leq \tilde{v}(S) \leq v(S)$ for all $S \subseteq U$, this *does not* imply that OPT_{Bench} inherits this approximation property: i.e., we do not necessarily have that $\frac{OPT_{\mathsf{Bench}}(v,B,c)}{OPT_{\mathsf{Bench}}(\tilde{v},B,c)} \in \big[1, O(\gamma)\big]$. So we need to proceed more carefully, and utilize $\tilde{v}$ internally in our mechanisms, in the appropriate demand-set computations of our mechanisms for XOS valuations.

It is well-known that v can be pointwise-approximated within an $O(\log n)$-factor (in the sense of $\tilde{v}$ above) by an XOS function, and that this is tight [16, 14]. To do better, we observe that the analysis of our mechanisms for XOS valuations relies on a weaker property than the function being XOS. An XOS function g satisfies the fractional-cover property (see Section 2) that for every set S, the value $\sum_{T\subseteq S} g(T)x_T$ of every fractional cover $\{x_T\}_{T\subseteq S}$ of S is at least $g(S)$. Our mechanisms for XOS valuations only require this property to hold for fractional covers supported on *player-respecting subsets of* S, i.e., on sets $T \subseteq S$, where $T \cap G_i \in \{\emptyset, S \cap G_i\}$ for all $i \in [k]$ (see Definition 7.1). This is because we only use the XOS property via Claim 2.3, to argue that $\sum_i \big(v(S) - v(S - G_i)\big) \leq v(S)$ (see, e.g., (2)), which follows from the fractional-cover property applied to a cover supported on $\{S - G_i\}_{i\in[k]}$, which are player-respecting subsets of S.

Definition 7.1. Given a set $S \subseteq U$, we say that $T \subseteq S$ is a *player-respecting subset* of S if $T \cap G_i \in \{\emptyset, S \cap G_i\}$ for all $i \in [k]$. Let $\mathcal{P}(S) \subseteq 2^S$ denote the collection of player-respecting subsets of S.

We say that a function $g : 2^U \mapsto \mathbb{R}_+$ is a *player-respecting XOS function*, if for every set $S \subseteq U$, the optimum value of the following LP, denoted $\mathsf{LP}^*_{\mathsf{PFC}}(S)$, is $g(S)$:

$$\min \sum_{T\in\mathcal{P}(S)} g(T)x_T \qquad \text{s.t.} \qquad \sum_{T\in\mathcal{P}(S): e\in T} x_T = 1 \quad \forall e \in S, \qquad x \geq 0. \tag{PFC(S)}$$

A feasible solution to the above LP is called a player-respecting fractional cover of S. When g is monotone, we can relax the equality constraints above to $\geq$-constraints.

The approximation factor we obtain is actually $O(\gamma)$, where $\gamma \geq 1$ is the best factor possible for a pointwise approximation of v by a player-respecting XOS function. One can show that the best such approximation comes from the family of LPs (PFC(S)) (just as the best pointwise XOS-approximation can be obtained by solving the fractional-cover LP [14]); γ corresponds to the integrality gap of this family, and we always have $\gamma = O(\log k)$.

Lemma 7.2. *Let $g : 2^U \mapsto \mathbb{R}_+$ be a monotone subadditive function. Define $g^{\mathsf{prxos}}(S) := \mathsf{LP}^*_{\mathsf{PFC}}(S)$ for all $S \subseteq U$. Then*

(a) g^{prxos} is a player-respecting XOS function; hence, $\sum_{i\in[k]} \big(g^{\mathsf{prxos}}(S) - g^{\mathsf{prxos}}(S - G_i)\big) \leq g^{\mathsf{prxos}}(S)$ for all $S \subseteq U$.

(b) if h is a player-respecting XOS function with $h(S) \le g(S)$ for all $S \subseteq U$, then we have $h(S) \le g^{\textsf{prxos}}(S)$ for all $S \subseteq U$;

(c) we have $g(S)/\gamma \le g^{\textsf{prxos}}(S) \le g(S)$ for all $S \subseteq U$, where γ is $\max_{S\subseteq U}$(integrality gap of (PFC(S)));

(d) $\gamma \le O(\log k)$.

We defer the proof of the above lemma to the end of the section. In the rest of this section, we fix the monotone, subadditive valuation v, and suppose that we have a player-respecting XOS function $v^{\textsf{prxos}}$ such that $v(S)/\gamma \le v^{\textsf{prxos}}(S) \le v(S)$ for all $S \subseteq U$, where $\gamma = O(\log k)$. We briefly describe the changes to the suitable mechanisms from Sections 4 and 5 below.

Budget-feasible-in-expectation mechanism for general costs. We modify Algorithm XOS-BFInExp by solving the following LP in step 2 to obtain x^*.

$$\max \sum_{S\subseteq U_2} \big(v^{\textsf{prxos}}(S) - \tfrac{\lambda V_1}{\gamma B}\cdot c(S)\big)x_S \quad \text{s.t.} \quad \sum_{S\subseteq U_2} v^{\textsf{prxos}}(S)x_S \le \frac{\lambda V_1}{\gamma}, \quad \sum_{S\subseteq U_2} x_S \le 1, \quad x \ge 0. \tag{5}$$

Since this is again a VCG computation and $v^{\textsf{prxos}}$ is a player-respecting XOS function by definition, as in Lemma 4.3, we obtain truthfulness and budget-feasibility in expectation. For the approximation, taking $\lambda = 0.5$, we argue as in Lemma 4.2 that assuming that $\textsf{LP}_2^* \ge \textsf{LP}_1^* \ge \frac{OPT_{\textsf{Bench}}(v,B,c)}{4}$, which happens with probability $\frac{1}{4}$, the optimal value of the above LP is $V_1/4\gamma$. It follows that the expected value returned is at least $\frac{OPT_{\textsf{Bench}}}{O(\gamma)}$. Let $\overline{x}$ be an optimal solution to (BFLP(U_2)), when we use the actual valuation v. We have $\sum_S v^{\textsf{prxos}}(S)\overline{x}_S \ge \textsf{LP}_2^*/\gamma$. Considering the solution $x' = \frac{\lambda V_1}{\gamma \sum_S v^{\textsf{prxos}}(S)\overline{x}_S}\cdot \overline{x}$, we then have that $x' \le \lambda\overline{x}$ is a feasible solution to (5) with objective value

$$\frac{\lambda V_1}{\gamma} - \frac{\lambda V_1}{\gamma B}\cdot\frac{\lambda V_1}{\gamma\sum_S v^{\textsf{prxos}}(S)\overline{x}_S}\cdot B \ge \frac{\lambda(1-\lambda)}{\gamma}\cdot V_1 = \frac{V_1}{4\gamma}.$$

Universally budget-feasible mechanism for general costs. Recall that $V_j^* := \max\{v(S) : S \subseteq U_j,\ c(S) \le B\}$ for $j = 1, 2$, where U_1, U_2 are identically-distributed player-respecting subsets of U that partition U. We proceed as in Algorithm XOS-Gen (in Section 5.2) for XOS valuations without assuming no-overbidding, with the only change being that in step 3 of the algorithm, we now compute $S^* \leftarrow \operatorname{argmax}_{S\subseteq U_2}\big\{v^{\textsf{prxos}}(S) - \frac{\lambda V_1}{\gamma B}\cdot c(S) :\ v^{\textsf{prxos}}(S) \le \frac{\lambda V_1}{\gamma}\big\}$. As before, we obtain universal budget-feasibility. For the approximation, we proceed as in the proof for XOS valuations.

Let $\textsf{opt}_i = \max_{S\subseteq G_i} \max\{v(S) : c_i(S) \le B\}$ for $i \in \mathcal{N}$, and $\textsf{opt}^{(2)}$ be the second-largest $\textsf{opt}_i$ value. Let $i^* \in \mathcal{N}$ be such that $\textsf{opt}_{i^*} = \max_{i\in\mathcal{N}} \textsf{opt}_i$, and let $G^* = G_{i^*}$. Let O^* be such that $c(O^*) \le B$ and $v_{-1}(O^*) = OPT_{\textsf{Bench}}$. Let $\overline{O} = O^* - G^*$, and $\overline{O}_j = \overline{O} \cap U_j$ for $j = 1, 2$. Note that for any $S \subseteq U - G^*$ with $c(S) \le B$, and any $i \in \mathcal{N}$, we have $v^{\textsf{prxos}}(S \cap G_i) \le v(S \cap G_i) \le \textsf{opt}^{(2)}$. Let $V_j' = \max\{v(S) : S \subseteq U_j - G^*,\ c(S) \le B\}$ for $j = 1, 2$. let $T_2' \subseteq U_2 - G^*$ be such that $v(T_2') = V_2'$ and $c(T_2') \le B$.

We may assume that $v(\overline{O}_1), v(\overline{O}_2) \ge v_{-1}(\overline{O})/4$, $V_2' \ge V_1'$, and $G^* \subseteq U_2$, an event that occurs with probability at least $\frac{1}{8}$. We take $\lambda = 0.5$, and $p = \frac{128\gamma}{144\gamma+1}$ so that $\frac{p}{8}\cdot\big(1 + \frac{1}{16\gamma}\big) = 1 - p$. We have $v^{\textsf{prxos}}(T_2') \ge \frac{V_2'}{\gamma} \ge \frac{2\lambda V_1'}{\gamma} = \frac{2\lambda V_1}{\gamma}$. So using Lemma 2.8 with $v^{\textsf{prxos}}$, we can find $T \subseteq T_2'$ with $c(T) \le B/2$ and $\min\big\{{v^{\textsf{prxos}}}_{-1}(T_2') - \frac{\lambda V_1}{\gamma}, \frac{\lambda V_1}{\gamma} - \max_{e\in T_2'} v(e)\big\} < v^{\textsf{prxos}}(T) \le \frac{\lambda V_1}{\gamma}$. This implies that $v^{\textsf{prxos}}(T) > \frac{\lambda V_1}{\gamma} - \textsf{opt}^{(2)}$, which can be used to show that $v^{\textsf{prxos}}(S^*) \ge \frac{\lambda V_1}{2\gamma} - \textsf{opt}^{(2)}$. So proceeding as in the rest of the proof of Theorem 5.6, the expected value returned is at least $\frac{p}{128\gamma}\cdot OPT_{\textsf{Bench}}(v, B, c) = \frac{1}{144\gamma+1}\cdot OPT_{\textsf{Bench}}(v, B, c)$.

Proof of Lemma 7.2

Part (a). Consider any $S \subseteq U$, and any player-respecting fractional cover $\{x_T\}_{T\in\mathcal{P}(S)}$ of S. For each $T \in \mathcal{P}(S)$, let y^T be an optimal solution to (PFC(T)), so that $g^{\mathsf{prxos}}(T) = \sum_{Z\in\mathcal{P}(T)} g(Z)y^T_Z$. Note that, for any $T \in \mathcal{P}(S)$, we have $\mathcal{P}(T) = \{Z \in \mathcal{P}(S) : Z \subseteq T\}$. Now consider the solution $\overline{x}$, where we set $\overline{x}_Z = \sum_{T\in\mathcal{P}(S):T\supseteq Z} x_T y^T_Z$ for every $Z \in \mathcal{P}(S)$. We argue that this is a feasible solution to (PFC(S)) of objective value $\sum_{T\in\mathcal{P}(S)} g^{\mathsf{prxos}}(T)x_T$, which implies that $g^{\mathsf{prxos}}(S) = \mathsf{LP}^*_{\mathsf{PFC}}(S) \le \sum_{T\in\mathcal{P}(S)} g^{\mathsf{prxos}}(T)x_T$, proving part (a).

For any $e \in S$, we have

$$\begin{aligned}\sum_{Z\in\mathcal{P}(S):e\in Z} \overline{x}_Z &= \sum_{Z\in\mathcal{P}(S):e\in Z}\ \sum_{T\in\mathcal{P}(S):T\supseteq Z} x_T y^T_Z = \sum_{T\in\mathcal{P}(S):e\in T} x_T \cdot \sum_{Z\in\mathcal{P}(S):Z\subseteq T,e\in Z} y^T_Z \\ &= \sum_{T\in\mathcal{P}(S):e\in T} x_T \cdot \sum_{Z\in\mathcal{P}(T):e\in Z} y^T_Z = \sum_{T\in\mathcal{P}(S):e\in T} x_T \cdot 1 = 1.\end{aligned}$$

The third equality is because $\mathcal{P}(T) = \mathcal{P}(S) \cap 2^T$, for $T \in \mathcal{P}(S)$, the fourth is because y^T is a feasible solution to (PFC(T)), and the final equality is because x_T is a feasible solution to (PFC(S)). This shows that $\overline{x}$ is a feasible solution to (PFC(S)). Its objective value is

$$\begin{aligned}\sum_{Z\in\mathcal{P}(S)} g(Z)\overline{x}_Z &= \sum_{Z\in\mathcal{P}(S)}\ \sum_{T\in\mathcal{P}(S):T\supseteq Z} g(Z)x_T y^T_Z = \sum_{T\in\mathcal{P}(S)} x_T \cdot \sum_{Z\in\mathcal{P}(T)} g(Z)y^T_Z \\ &= \sum_{T\in\mathcal{P}(S)} x_T \cdot g^{\mathsf{prxos}}(T)\end{aligned}$$

where the final equality is because $g^{\mathsf{prxos}}(T) = \mathsf{LP}^*_{\mathsf{PFC}}(T)$ and y^T is an optimal solution to (PFC(T)). The second statement follows from Claim 2.3.

Part (b). Consider any $S \subseteq U$. Let x^* be an optimal solution to (PFC(S)). Then,

$$g^{\mathsf{prxos}}(S) = \sum_{T\in\mathcal{P}(S)} g(T)x^*_T \ge \sum_{T\in\mathcal{P}(S)} h(T)x^*_T \ge h(S)$$

where the last inequality is because h is a player-respecting XOS function.

Part (c). Consider any $S \subseteq U$. It is clear that $g^{\mathsf{prxos}}(S) \le g(S)$ because setting $x_S = 1$ and all other $x_T = 0$ is a feasible solution to (PFC(S)). Moreover, the optimal value of an *integer* solution to (PFC(S)) is $g(S)$, since any integer solution corresponds to a partition $T_1, \dots, T_\ell$ of player-respecting subsets of S and has value $g(T_1) + g(T_2) + \ldots + g(T_\ell) \ge g(S)$, since g is subadditive. Therefore, we obtain that $g^{\mathsf{prxos}}(S) \ge g(S)/\gamma$.

Part (d). Consider any $S \subseteq U$. Relaxing the equality constraints of (PFC(S)) to $\ge$-inequalities, we see that (PFC(S)) consists of at most k distinct covering constraints, one for each $i \in [k]$ for which $S \cap G_i \ne \emptyset$. This is simply because if $e, e' \in S$ are such that $\{e, e'\} \subseteq S \cap G_i$ for some i, then $T \in \mathcal{P}(S)$ contains e iff it contains e', and so the constraints for e, e' are identical. Thus, by standard results on set cover, the integrality gap is $O(\log k)$. □

8 Guarantees relative to OPT_{Alg}

8.1 Algorithmic problem of approximately computing OPT_{Alg}

We now consider the algorithmic problem of developing polytime approximation algorithms for computing $OPT_{\mathsf{Alg}}(v, B, c) = \max_{S\subseteq U} \{v(S) : c(S) \le B\}$ given (v, B, c) as input, and a demand oracle. We exploit the ideas underlying our mechanisms to obtain a $(2+\varepsilon)$-approximation for subadditive v and superadditive c, i.e., c satisfies $c(S\cup T)\leq c(S)+c(T)$ for all $S,T\subseteq U$. Note that when the cost functions $c_1,\ldots,c_k$ comprising c are superadditive, then the function $c(S)=\sum_i c_i(S\cap G_i)$ is superadditive. Our guarantee is tight, since even for additive c, one cannot do better [11]. Also, it is known that value oracles are insufficient to obtain any approximation factor better than $\sqrt{n}$, even for XOS valuations and additive c.

Theorem 8.1. *We can obtain a $(2+\varepsilon)$-approximation with subadditive v and superadditive c, using demand oracles.*

Proof. The idea is similar to the algorithm in [11], which we simplify somewhat. Recall that LP^* is the optimal value of the following LP, which can be solved using a demand oracle. Here S ranges over subsets of U.

$$\max \sum_S v(S)x_S \qquad \text{s.t.} \qquad \sum_S c(S)x_S\leq B, \qquad \sum_S x_S\leq 1, \qquad x\geq 0. \tag{BFLP(U)}$$

We may assume that $c(e)\leq B$ for all $e\in U$. We first describe a simple 3-approximation algorithm. Let $\overline{x}$ be an optimal solution to the above LP. If there exists $e\in U$ with $v(e)\geq \mathsf{LP}^*/3$, then we simply return such an element. Otherwise, find $S^*=\operatorname{argmax}_{S\subseteq U}\bigl(v(S)-\kappa\cdot c(S)\bigr)$, where $\kappa=\frac{2\mathsf{LP}^*}{3B}$. We have $v(S^*)-\kappa\cdot c(S^*)\geq\sum_S\overline{x}_S\bigl(v(S)-\kappa\cdot c(S)\bigr)=\mathsf{LP}^*-\kappa B=\mathsf{LP}^*/3$. Now let T be a maximal subset of S^* with $c(T)\leq B$. If $T=S^*$, then we are done. Otherwise, for any $e\in S^*-T$, taking $T'=T\cup\{e\}$, we have $v(S^*)-\kappa\cdot c(S^*)\leq\bigl[v(T')-\kappa\cdot c(T')\bigr]+\bigl[v(S^*-T')-\kappa\cdot c(S^*-T')\bigr]$ since v is subadditive, and c is superadditive, and so $v(T')-\kappa\cdot c(T')\geq 0$. So $v(T')>\kappa B$, and $v(T)>\kappa B-v_{\max}\geq\mathsf{LP}^*/3$. Note that we do not actually need LP^* above: we can consider the largest V such that $\max_{S\subseteq U}\bigl(v(S)-\frac{2V}{3B}\cdot c(S)\bigr)$ is at least $V/3$.

To refine this to a $(2+\varepsilon)$-approximation, we use enumeration to enumerate all elements e that belong to some fixed optimal solution O^* with $c(e)\geq\varepsilon B$. Since c is superadditive, there are at most $\frac{1}{\varepsilon}$ such elements. So letting $H=\{e\in U: c(e)\geq\varepsilon B\}$ and $L=\{e\in U: c(e)<\varepsilon B\}$, we may assume that we know the set $A=O^*\cap H$. Note that $|A|\leq\frac{1}{\varepsilon}$ and $c(A)\leq B$. We now compute $S^*=\operatorname{argmax}_{S\subseteq A\cup L}\bigl(v(S)-\kappa'\cdot c(S)\bigr)$ where $\kappa'=\frac{\mathsf{LP}^*}{2B}$. As before, we may assume that $v_{\max}<\mathsf{LP}^*/2$. Now when we pick a maximal subset $T\subseteq S^*$ with $c(T)\leq B$, we first pick all the elements in $A\cap S^*$. This ensures that $c(T)>B-\varepsilon B$, and so $v(T)>(1-\varepsilon)\mathsf{LP}^*/2$. ☐

8.2 Tightness of the lower bounds in Theorem 1.1: mechanism-design guarantees

We now devise budget-feasible mechanisms obtaining approximation factors relative $OPT_{\mathsf{Alg}}(v,B,c)$, which will show that the lower bounds stated in Theorems 3.2 and 3.3 are tight for XOS valuations, and in some cases, also for subadditive valuations. Let $O^*=\operatorname{argmax}_{S\subseteq U}\{v(S): c(S)\leq B\}$ be an optimal solution to the algorithmic problem. Let $\zeta=\max_i|G_i|$. For every player i, order the elements in G_i arbitrarily; let $e^i_1,\ldots,e^i_{n_i}$ be this ordering, where $n_i\leq\zeta$. For an index $j\in[\zeta]$, let $\Psi^j:=\bigcup_{i\in[k]}\{e^i_j\}$, with the understanding that $\{e^i_j\}$ is the empty-set if $j>n_i$.

Without no-overbidding, consider the following randomized mechanism: we pick an element $e\in U$ uniformly at random, and return e if $c(e)\leq B$ along with a payment of B (to the player owning e), and the empty-set with 0 payments otherwise. This is clearly universally budget-feasible, and the expected value returned is at least $\sum_{e\in O^*}v(e)/n\geq OPT_{\mathsf{Alg}}/n$; this holds for any subadditive function. We can also obtain an $O(\zeta)$-approximation in this setting. We achieve this by reducing to the single-dimensional single-item setting losing an $O(\zeta)$-factor. We pick a random index $j\in[\zeta]$, and run a budget-feasible mechanism for the single-dimensional setting with universe Ψ^j, valuation v restricted to Ψ^j, and element-costs $\{c_i(e^i_j)\}_{i\in[k]}$. Letting OPT^j_{Alg} denote the optimum for this instance, by subadditivity of v, we have that $v(O^*) \leq \sum_{j=1}^{\zeta} v(O^* \cap \Psi^j) \leq \sum_{j=1}^{\zeta} OPT^j_{\mathsf{Alg}}$. So if the budget-feasible mechanism for the single-dimensional instance achieves an $O(1)$-approximation, we obtain $O(\zeta)$-approximation relative to OPT_{Alg} for the original instance. The mechanism inherits the budget-feasibility guarantee, i.e., universally budget-feasible or budget-feasible in expectation, from the budget-feasibility guarantee of the single-dimensional mechanisms.

Assuming no-overbidding, it is easy to see that returning $e^* = \text{argmax}_{e \in U} v(e)$ and making a payment of B (to the player owning e^*) yields a deterministic budget-feasible mechanism that achieves an n-approximation relative to OPT_{Alg} for any subadditive function, since $v(O^*) \leq nv(e^*)$ due to subadditivity.

The above mechanisms match the lower bounds in Theorems 3.2 and 3.3 (a). We next devise a universally budget-feasible mechanism that achieves an $O(\log \zeta)$-approximation relative to OPT_{Alg}, assuming no-overbidding.

Theorem 8.2. *Assuming no-overbidding, we can obtain polytime mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$ for XOS valuations that achieve an $O(\log n)$-approximation relative to OPT_{Alg}, such that:*
(a) $\mathcal{M}_1$ is budget-feasible in expectation, works with general costs, and uses a demand oracle;
(b) $\mathcal{M}_2$ is universally budget-feasible, works with superadditive costs, and uses a constrained demand oracle.
The approximation guarantees can be improved to $O(\log \max_i |G_i|)$ (relative to OPT_{Alg}), provided that we can solve the following algorithmic problem, for a given player-set $\mathcal{N}' \subseteq [k]$:

$$\max \quad v(S) \qquad \text{s.t.} \qquad S \subseteq \bigcup_{i \in \mathcal{N}'} G_i, \qquad c(S) \leq B, \qquad |S \cap G_i| \leq 1 \quad \forall i \in \mathcal{N}'. \qquad (\text{SD}(\mathcal{N}'))$$

Proof. This is a consequence of the fact that all our mechanisms work with a suitable estimate V_1 of our benchmarks. Their budget-feasibility guarantees hold regardless of the quality of this estimate, but their approximation guarantee relies on V_1 being a good-enough estimate. In our mechanisms, we obtain V_1 using random partitioning. But instead, since we know that $v_{\max} \geq OPT_{\mathsf{Alg}}/n$, we can take $V_1 = v_{\max} \cdot 2^{\ell}$, for a random exponent $\ell \in \{0\} \cup [\lceil \log_2 n \rceil]$. One of the choices for ℓ, will yield a good estimate of OPT_{Alg}, and this happens with probability at least $\Omega\Big(\frac{1}{\log_2 n}\Big)$. Recall that LP^*_j is the optimal value of (BFLP(U_j)) for $j = 1, 2$, and $V^*_j := \max\{v(S) : S \subseteq U_j,\ c(S) \leq B\}$ for $j = 1, 2$.

To elaborate, $\mathcal{M}_1$ simply runs steps 2, 3 of Algorithm XOS-BFInExp, where we now work with the entire universe U (so $\mathsf{LP}^*_2 = \mathsf{LP}^*$ now), with a random value V_1 obtained as above. The approximation factor of $\mathcal{M}_1$ follows from Lemma 4.2, which shows that for a given V_1, the expected value obtained is at least $\lambda V_1\big(1 - \frac{\lambda V_1}{\mathsf{LP}^*_2}\big)$. Since $\mathsf{LP}^*_2 \leq n v_{\max}$, for some V_1, we have $V_1 \leq \mathsf{LP}^*_2 \leq 2V_1$, we obtain value $\Omega(\mathsf{LP}^*_2)$ for this V_1.

Similarly, $\mathcal{M}_2$ runs steps 3, 4 of Algorithm XOS-UniBF taking $\lambda = p = 0.5$, again considering the entire universe U, with a random value V_1 obtained as above. Applying Lemma 2.8 to O^*, an optimal solution to the algorithmic problem, we can extract a set $T \subseteq O^*$ such that $c(T) \leq B/2$ and $\min\{OPT_{\mathsf{Alg}} - \lambda V_1, \lambda V_1\} - v_{\max} < v(T) \leq \lambda V_1$, so that $v(S^*) \geq \min\{OPT_{\mathsf{Alg}} - 1.5\lambda V_1, 0.5\lambda V_1\} - v_{\max}$. There is some V_1 with $V_1 \leq OPT_{\mathsf{Alg}} \leq 2V_1$, and for this V_1, we obtain expected value at least $\frac{pV_1}{4} = \Omega(OPT_{\mathsf{Alg}})$.

$O(\log \zeta)$**-approximation factors.** To obtain the improved guarantees, we refine our initial, coarse, estimate of OPT_{Alg}, from $v_{\max}$ above, to (an estimate of) $\mathsf{SDopt}(\mathcal{N})$, which is the optimal value of $(\text{SD}(\mathcal{N}))$. We have $\mathsf{SDopt}(\mathcal{N}) \geq OPT_{\mathsf{Alg}}/\zeta$, and so, roughly speaking, starting V_1 from $\mathsf{SDopt}(\mathcal{N})$ in place of $v_{\max}$, leads to an $O(\log \zeta)$-approximation.

To implement this, we revert to randomly partitioning the players into sets $\mathcal{N}_1$, $\mathcal{N}_2$, but this time to estimate $\mathsf{SDopt}(\mathcal{N})$. Let $\mathsf{SDopt}_j := \mathsf{SDopt}(\mathcal{N}_j)$, for $j = 1, 2$. We use $\max\{\mathsf{SDopt}_1, v_{\max}\}$ as our initial estimate. We then consider V_1 of the form $\max\{\mathsf{SDopt}_1, v_{\max}\} \cdot 2^\ell$, for a random exponent $\ell \in \{0\} \cup [\lceil \log \zeta \rceil]$, and run our earlier mechanisms with this V_1, now again doing the demand-set computation with item-set U_2. We now furnish the details.

We first argue that $\Pr\big[\max\{\mathsf{SDopt}_j, v_{\max}\} \geq \frac{OPT_{\mathsf{Alg}}}{5\zeta} \text{ for } j = 1, 2\big] \geq 0.5$. We call this the "good-estimate event." Let S be an optimal solution to (SD($\mathcal{N}$)). Note that $\max_i v(S \cap G_i) \leq v_{\max}$ since $|S \cap G_i| \leq 1$ for all $i \in [k]$. So applying Lemma 2.4 (a) to S, with probability 0.5, for $j = 1, 2$, we have $\mathsf{SDopt}_j \geq v(S \cap G_j) \geq \frac{\mathsf{SDopt}(\mathcal{N}) - v_{\max}}{4}$. This implies that $\max\{\mathsf{SDopt}_j, v_{\max}\} \geq \frac{4}{5} \cdot \mathsf{SDopt}_j + \frac{1}{5} \cdot v_{\max} \geq \frac{\mathsf{SDopt}(\mathcal{N})}{5}$.

Now the analogue of $\mathcal{M}_1$ runs steps 2, 3 of Algorithm XOS-BFInExp (on the item-set U_2) with V_1 of the form $\max\{\mathsf{SDopt}_1, v_{\max}\} \cdot 2^\ell$. We have that LP_1^*, LP_2^* are identically distributed, and this remains true even when we condition on the above good-estimate event. Hence, we have that the probability that the good-estimate event happens and $\mathsf{LP}_2^* \geq \mathsf{LP}_1^*$, and hence $\mathsf{LP}_2^* \geq \mathsf{LP}^*/2$, is at least 0.25. So the expected value obtained when we have V_1 satisfying $V_1 \leq \mathsf{LP}_2^* \leq 2V_1$ is $\Omega(\mathsf{LP}_2^*)$. Consequently, we obtain an $O(\log \zeta)$-approximation.

Similarly, the analogue of $\mathcal{M}_2$ runs steps 3, 4 of Algorithm XOS-UniBF taking $\lambda = p = 0.5$, again with a random value V_1 of the form $\max\{\mathsf{SDopt}_1, v_{\max}\} \cdot 2^\ell$. We have that V_1^*, V_2^* are identically distributed, so the probability that the good-estimate event happens and $V_2^* \geq V_1^*$, and hence $V_2^* \geq OPT_{\mathsf{Alg}}/2$, is at least 0.25. Applying Lemma 2.8 to $T_2^* \subseteq U_2$ such that $v(T_2^*) = V_2^*$, $c(T_2^*) \leq B$, we can extract $T \subseteq T_2^*$ such that $c(T) \leq B/2$ and $\min\{V_2^* - \lambda V_1, \lambda V_1\} - v_{\max} < v(T) \leq \lambda V_1$, so that $v(S^*) \geq \min\{V_2^* - 1.5\lambda V_1, 0.5\lambda V_1\} - v_{\max}$. There is some V_1 with $V_1 \leq OPT_{\mathsf{Alg}} \leq 2V_1$, and for this V_1, we obtain expected value at least $\frac{pV_1}{4} = \Omega(OPT_{\mathsf{Alg}})$. So the overall expected value obtained is $\Omega(OPT_{\mathsf{Alg}}/\log \zeta)$. □

Remark 8.3. Amanatidis et al. [3] make an "all-in" assumption in the single-dimensional LOS setting, which assumes that the budget B is large enough that the buyer can buy *all* levels of service from any player [3], which allows them to bypass the impossibility results in Theorem 3.3 and obtain an $O(1)$-approximation with respect to OPT_{Alg}.[16]

Under our notation, this translates to assuming that for every i, and $c \in \mathcal{C}_i$, we have $c_i(G_i) \leq B$. This is a rather strong assumption, but we note that under this assumption, it is quite easy to convert our results for XOS valuations to obtain $O(1)$-approximation relative to OPT_{Alg}. To see this, note that, under this assumption, choosing player $i^* = \operatorname{argmax}_i v(G_i)$, and returning G_{i^*} and paying B to player i^* yields a budget-feasible mechanism. Say we have a mechanism $\mathcal{M}$ that obtains expected value at least $OPT_{\mathsf{Bench}}(\ell; v, B, c)/\alpha$, for some $\alpha \geq 1$ and integer $\ell \geq 1$, where recall that $OPT_{\mathsf{Bench}}(\ell; v, B, c) := \max\{v_{-\ell}(S) : S \subseteq U,\ c(S) \leq B\}$, which is at least $OPT_{\mathsf{Alg}}(v, B, c) - \ell \cdot v(G_{i^*})$; all our mechanisms have this type of guarantee, for $\ell = O(1)$. Then, we can obtain $(\alpha + \ell)$-approximation relative to OPT_{Alg} by simply running $\mathcal{M}$ with probability $\frac{\alpha}{\alpha+\ell}$, and the mechanism that returns G_{i^*} with probability $\frac{\ell}{\alpha+\ell}$.

References

[1] G. Amanatidis, G. Birmpas, and E. Markakis. On budget-feasible mechanism design for symmetric submodular objectives. In *Proceedings of 13th WINE*, pages 1–15, 2017. 8

[2] G. Amanatidis, P. Kleer, and G. Schäfer. Budget-feasible mechanism design for non-monotone submodular objectives: Offline and online. In *Proceedings of 20th EC*, pages 901–919, 2019. 5, 8, 13

[16] The all-in assumption is closely related to the assumption that in the divisible-item setting (where one can buy a fraction of an item), the buyer can afford to buy an entire unit of item from any player [45]; this is a weaker version of the large-market assumption for divisible items [5] that allows [45] to again obtain $O(1)$-approximation relative to OPT_{Alg}.

[3] G. Amanatidis, S. Klumper, E. K. Markakis, G. Schafer, and A. Tsikiridis. Partial allocations in budget-feasible mechanism design: Bridging multiple levels of service and divisible agents. *ArXiv*, abs/2307.07385, 2023. 1, 2, 5, 8, 38

[4] N. Anari, G. Goel, and A. Nikzad. Mechanism design for crowdsourcing: An optimal 1-1/e competitive budget-feasible mechanism for large markets. In *Proceedings of 55th FOCS*, pages 266–275, 2014. 8, 17, 44

[5] N. Anari, G. Goel, and A. Nikzad. Budget feasible procurement auctions. *Oper. Res.*, 66:637–652, 2018. 1, 8, 38

[6] A. Archer and É. Tardos. Frugal path mechanisms. *ACM Transactions on Algorithms (TALG)*, 3(1):1–22, 2007.

[7] S. Assadi, T. Kesselheim, and S. Singla. Improved truthful mechanisms for subadditive combinatorial auctions: Breaking the logarithmic barrier. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 653–661, 2021. 9

[8] S. Assadi and S. Singla. Improved truthful mechanisms for combinatorial auctions with submodular bidders. In *FOCS*, pages 233–248, 2019. 9

[9] M. Babaioff, N. Immorlica, B. Lucier, and S. M. Weinberg. A simple and approximately optimal mechanism for an additive buyer. In *FOCS*, pages 21–30, 2014. 9

[10] M. Babaioff, R. Kleinberg, and A. Slivkins. Multi-parameter mechanisms with implicit payment computation. In *Proceedings of 14th EC*, pages 35–52, 2013. 9

[11] A. Badanidiyuru, S. Dobzinski, and S. Oren. Optimization with demand oracles. *Algorithmica*, 81:2244–2269, 2019. 5, 36

[12] E. Balkanski, P. Garimidi, V. Gkatzelis, D. Schoepflin, and X. Tan. Deterministic budget-feasible clock auctions. In *Proceedings of 33rd SODA*, pages 2940–2963, 2022. 1, 8

[13] E. Balkanski and J. Hartline. Bayesian budget feasibility with posted pricing. In *Proceedings of 25th WWW*, pages 189–203, 2016. 8

[14] X. Bei, N. Chen, N. Gravin, and P. Lu. Budget feasible mechanism design: from prior-free to bayesian. In *Proceedings of 44th STOC*, pages 449–458, 2012. 1, 3, 5, 8, 13, 33

[15] X. Bei, N. Chen, N. Gravin, and P. Lu. Worst-case mechanism design via bayesian analysis. *SIAM Journal on Computing*, 46(4):1428–1448, 2017. 8

[16] K. Bhawalkar and T. Roughgarden. Welfare guarantees for combinatorial auctions with item bidding. In *SODA*, pages 700–709, 2011. 5, 33

[17] G. Birmpas, E. Markakis, and G. Schäfer. Cost sharing over combinatorial domains. *ACM Transactions on Economics and Computation*, 10(1):1–26, 2022. 9

[18] Y. Cai, C. Daskalakis, and M. Weinberg. An algorithmic characterization of multi-dimensional mechanisms. In *Symposium on Theory of Computation*, pages 459–478, 2012. 9

[19] Y. Cai, C. Daskalakis, and M. Weinberg. Optimal multi-dimensional mechanism design: Reducing revenue to welfare maximization. In *Foundations of Computer Science*, pages 130–139, 2012. 9

[20] I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, and J. Wang. The unreasonable fairness of maximum nash welfare. *Proceedings of 17th EC*, 2016.

[21] H. Chan and J. Chen. Truthful multi-unit procurements with budgets. *ArXiv*, abs/1409.7595, 2014. 1, 2, 5, 8, 16, 42

[22] N. Chen, E. Elkind, N. Gravin, and F. Petrov. Frugal mechanism design via spectral techniques. In *Proceedings of 51st Foundations of Computer Science (FOCS)*, pages 755—764, 2010. 9

[23] N. Chen, N. Gravin, and P. Lu. On the approximability of budget feasible mechanisms. In *Proceedings of 22nd SODA*, pages 685–699, 2011. 8

[24] E. H. Clarke. Multipart pricing of public goods. *Public choice*, pages 17–33, 1971. 12

[25] S. Dobzinski and S. Dughmi. On the power of randomization in algorithmic mechanism design. *SIAM Journal on Computing*, 42(6):2287–2304, 2013. 9, 12

[26] S. Dobzinski, N. Nisan, and M. Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. In *Proceedings of 37th STOC*, pages 610–618, 2005. 9

[27] S. Dobzinski and S. Ovadia. Combinatorial cost sharing. In *Proceedings of 18th EC*, pages 387–404, 2017. 1, 3, 9

[28] S. Dobzinski, C. Papadimitriou, and Y. Singer. Mechanisms for complement-free procurement. In *Proceedings of 12th EC*, pages 273–282, 2011. 3, 8

[29] S. Dughmi, T. Roughgarden, and Q. Yan. Optimal mechanisms for combinatorial auctions and combinatorial public projects via convex rounding. *J. ACM*, 63(4):1–33, 2016. 9, 12

[30] P. Dütting, T. Kesselheim, and B. Lucier. An o(log log m) prophet inequality for subadditive combinatorial auctions. *ACM SIGecom Exchanges*, 18(2):32–37, 2020. 9

[31] A. Eden, M. Feldman, O. Friedler, I. Talgam-Cohen, and S. M. Weinberg. The competition complexity of auctions: A bulow-klemperer result for multi-dimensional bidders. In *ACM Conference on Economics and Computation*, 2017. 9

[32] A. Eden, M. Feldman, O. Friedler, I. Talgam-Cohen, and S. M. Weinberg. A simple and approximately optimal mechanism for a buyer with complements. *Operations Research*, 69(1):188–206, 2021. 9

[33] E. Elkind, L. A. Goldberg, and P. W. Goldberg. Frugality ratios and improved truthful mechanisms for vertex cover. In *Proceedings of 8th EC*, pages 336–345, 2007.

[34] M. Feldman, N. Gravin, and B. Lucier. Combinatorial auctions via posted prices. In *Proceedings of 26th SODA*, pages 123–135. SIAM, 2014. 9

[35] K. Georgiou and C. Swamy. Black-box reductions for cost-sharing mechanism design. *Games and Economic Behavior*, 113:17–37, 2019. 1, 9

[36] A. V. Goldberg, J. D. Hartline, A. R. Karlin, M. Saks, and A. Wright. Competitive auctions. *Games and Economic behavior*, 55(2):242–269, 2006. 9

[37] A. V. Goldberg, J. D. Hartline, and A. Wright. Competitive auctions and digital goods. In *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 735—744, 2001. 9

[38] T. Groves. Incentives in teams. *Econometrica: Journal of the Econometric Society*, pages 617–631, 1973. 12

[39] M. T. Hajiaghayi, M. R. Khani, and S. Seddighin. Frugal auction design for set systems: Vertex cover and knapsack. *Proceedings of 19th EC*, 2018.

[40] J. Hartline and A. Karlin. Profit maximization in mechanism design. *Algorithmic Game Theory*, 331, 2007. 1, 9

[41] H. Huang, K. Han, S. Cui, and J. Tang. Randomized pricing with deferred acceptance for revenue maximization with submodular objectives. In *Proceedings of 32nd WWW*, page 3530–3540, 2023. 8

[42] P. Jalaly and É. Tardos. Simple and efficient budget feasible mechanisms for monotone submodular valuations. In *Proceedings of 14th WINE*, pages 246–263, 2018. 8

[43] A. R. Karlin, D. Kempe, and T. Tamir. Beyond vcg: Frugality of truthful mechanisms. In *Proceedings of 46th FOCS*, pages 615–624. IEEE, 2005.

[44] D. Kempe, M. Salek, and C. Moore. Frugal and truthful auctions for vertex covers, flows and cuts. In *Proceedings of 51st Foundations of Computer Science (FOCS)*, pages 745—754, 2010. 9

[45] S. Klumper and G. Schäfer. Budget feasible mechanisms for procurement auctions with divisible agents. In *Algorithmic Game Theory*, 2022. 1, 8, 38

[46] R. Lavi. Computationally efficient approximation mechanisms. *Algorithmic Game Theory*, pages 301–329, 2007. 1, 9

[47] R. Lavi and C. Swamy. Truthful mechanism design for multi-dimensional scheduling via cycle monotonicity. In *Proceedings of 8th EC*, pages 252–261, 2007. 9

[48] R. Lavi and C. Swamy. Truthful and near-optimal mechanism design via linear programming. *J. ACM*, 58(6):1–24, 2011. 9, 12

[49] S. Leonardi, G. Monaco, P. Sankowski, and Q. Zhang. Budget feasible mechanisms on matroids. *Algorithmica*, 83(5):1222–1237, 2021. 1

[50] R. J. Lipton, E. K. Markakis, E. Mossel, and A. Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of 5th EC*, 2004.

[51] A. Mehta, T. Roughgarden, and M. Sundararajan. Beyond moulin mechanisms. *Games and Economic Behavior*, 67:125—155, 2009. 9

[52] H. Minooei and C. Swamy. Truthful mechanism design for multidimensional covering problems. In *Proceedings of the 8th International Workshop on Internet and Network Economics (WINE)*, pages 448–461, 2012. 9

[53] R. Neogi, K. Pashkovich, and C. Swamy. Budget-Feasible Mechanism Design: Simpler, Better Mechanisms and General Payment Constraints. In *Proceedings of 15th ITCS*, pages 84:1–84:22, 2024. 3, 5, 6, 8, 13, 19, 24

[54] T. Roughgarden and M. Sundararajan. Quantifying inefficiency in cost-sharing mechanisms. *Journal of the ACM*, 56(4), 2009. 9

[55] Y. Singer. Budget feasible mechanisms. In *Proceedings of 51st FOCS*, pages 765–774, 2010. 1, 8

[56] Y. Singer. Budget feasible mechanism design. *ACM SIGecom Exchanges*, 12(2):24–31, 2013. 1

[57] K. Talwar. The price of truth: Frugality in truthful mechanisms. In *Proceedings of the 21st Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 608—619, 2003. 9

[58] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961. 12

A Results and proofs omitted from Section 3

Proof of Theorem 3.3. As mentioned earlier, this result was proved by [21]. We include a proof here for completeness. For both results, we only require a single player. For $\theta \in \mathbb{R}_+$, let $c^{(\theta)}$ denote the additive cost function given by $c_e = \theta$ for all $e \in U$.

For part (a), consider the cost functions $c' = c^{(B)}$ and $c'' = c^{(1)}$. On input c', $\mathcal{M}$ must return at least one element, otherwise the statement holds for c'. Due to budget feasibility and IR, $\mathcal{M}$ can return at most one element, and so $\mathcal{M}$ returns exactly one element. Now consider input c''. We now have $OPT_{\mathsf{Alg}}(c'') = n$. But we claim that $\mathcal{M}$ must still output at most one element on input c'', which will prove part (a). If $\mathcal{M}$ outputs more than one item under input c'', the player's utility when c'' is her true cost, is less than $B - 1$, but her utility when she reports c' would be $B - 1$, contradicting truthfulness.

For part (b), let n be a power of 2. Let $I = \{0\} \cup [\log_2 n]$. For $\ell \in I$, let p_ℓ be the expected payment made by $\mathcal{M}$ and a_ℓ be the expected number of items procured by $\mathcal{M}$, when the player reports the cost-function $c^{(2^\ell)}$. Since $\mathcal{M}$ is truthful in expectation, the player's expected utility is maximized by reporting truthfully. So for any $\ell, r \in [\log_2 n]$, we have $p_\ell - 2^\ell a_\ell \geq p_r - 2^\ell a_r$. Define $p_{\log_2 n+1} = a_{\log_2 n+1} = 0$ for notational convenience. Then, we have

$$p_\ell - p_{\ell+1} \geq 2^\ell (a_\ell - a_{\ell+1}) \qquad \text{for all } \ell \in I.$$

where the inequality for $\ell = \log_2 n$ follows from individual rationality. Adding the above for all $\ell \in I$ we obtain that $p_0 \geq a_0 + \sum_{\ell=0}^{\log_2 n-1} (2^{\ell+1} - 2^\ell) a_{\ell+1} \geq \sum_{\ell=0}^{\log_2 n} 2^{\ell-1} a_\ell$. We have $p_0 \leq B = n$ since $\mathcal{M}$ is budget-feasible in expectation. So there is some $\ell \in I$ for which $2^\ell a_\ell \leq 2n/(1 + \log_2 n)$. But then on input $c^{(2^\ell)}$, since $OPT_{\mathsf{Alg}}(v, B, c^{(2^\ell)}) = n$, $\mathcal{M}$ obtains value at most $\frac{2}{1+\log_2 n} \cdot OPT_{\mathsf{Alg}}$. □

A.1 Bayesian budget-feasible mechanism design.

In the Bayesian setting, there is a publicly-known prior distribution $\mathcal{D}$ on $\mathcal{C}$ from which players' private cost functions are drawn. (Note that players are not necessarily independent.) A possibly randomized mechanism $\mathcal{M}$ is *Bayesian budget-feasible* if:

(a) IR holds with probability 1, where the probability is over both $\mathcal{D}$, and any random choices made by $\mathcal{M}$; and

(b) The expected payment made by $\mathcal{M}$ on any input $c \in \mathcal{C}$ drawn from $\mathcal{D}$ is at most the budget B; and

(c) It is *Bayesian incentive compatible* (BIC), which means that each player i maximizes her expected utility by reporting truthfully, where the expectation is both over the conditional distribution of other player's types given i's true type, and any random choices made by $\mathcal{M}$: for every i, every $\overline{c}_i, c'_i \in \mathcal{C}_i$, we have $\mathbb{E}_{(c_i,c_{-i})\sim\mathcal{D}}\big[u_i(\overline{c}_i; \overline{c}_i, c_{-i}) \,|\, c_i = \overline{c}_i\big] \geq \mathbb{E}_{(c_i,c_{-i})\sim\mathcal{D}}\big[u_i(\overline{c}_i; c'_i, c_{-i}) \,|\, c_i = \overline{c}_i\big]$.
Note that if there is only one player, then BIC coincides with truthfulness.

We say that $\mathcal{M}$ achieves an α-approximation relative to OPT_{Alg} for (v, B), if $\mathbb{E}_{c\sim\mathcal{D}}\big[v(f(c))\big] \geq \frac{1}{\alpha} \cdot \mathbb{E}_{c\sim\mathcal{D}}\big[OPT_{\mathsf{Alg}}(v, B, c)\big]$.

With these definitions in place, we now prove Corollary 3.4.

Proof of Corollary 3.4. We give two proofs. One is a black-box reduction showing that by Yao's minimax principle (or equivalently, the min-max formula for two-person zero-sum games), any approximation lower bound for budget-feasible-in-expectation mechanisms when there is only one player also applies to Bayesian budget-feasible mechanisms. Second, one can easily adapt the lower-bound constructions in Theorems 3.2 and 3.3 (b) to apply to the Bayesian setting.

Using Yao's minimax principle. The min-max formula for two-person zero-sum games states that for any $m \times n$ matrix A, we have $\max_{p\in\Delta_m} \min_{q\in\Delta_n} p^T Aq = \min_{q\in\Delta_n} \max_{p\in\Delta_m} p^T Aq$.

Suppose there is only one player, so that BIC coincides with truthfulness. Fix some (v, B). Consider the two-person game, where the row player chooses an input $c \in \mathcal{C}$, and the column player chooses a budget-feasible-in-expectation mechanism $\mathcal{M}$, and the payoff for $(c, \mathcal{M})$ is the approximation ratio achieved by $\mathcal{M}$ on input (v, B, c), i.e., $OPT_{\mathsf{Alg}}(v, B, c)/\mathbb{E}\big[\text{value returned by } \mathcal{M} \text{ on } (v, B, c)\big]$. To make the strategy-space finite, we consider inputs c and mechanisms $\mathcal{M}$ of some finite bit complexity. Since there is only one player, the strategy-space of the column player consists of all Bayesian budget-feasible mechanisms.

Suppose that for every distribution $\mathcal{D}$ on $\mathcal{C}$, there is a Bayesian budget-feasible mechanism that achieves approximation ratio α with respect to OPT_{Alg}. In terms of the above two-person game, this means that for every mixed-strategy of the row player, there is a column-player strategy that yields payoff at most α. By the min-max formula for two-person zero-sum games, this implies that there is a mixed strategy of the column player that guarantees that for every row-player (mixed) strategy, the payoff is at most α. That is, there is some distribution over budget-feasible-in-expectation mechanisms, which is another budget-feasible-in-expectation mechanism, that achieves approximation ratio at most α on every input. Hence, a lower bound on the approximation ratio achievable by budget-feasible-in-expectation mechanisms yields the same lower bound for Bayesian budget-feasible mechanisms.

Adapting the lower bounds in Theorems 3.2 and 3.3 (b). Recall the setup of the n-approximation lower bound in Theorem 3.2. We have $\gamma = 1 + \frac{n}{\epsilon}$, $U = [n]$, v is the additive valuation defined by $v(e) = \gamma^{n-e}$ for all $e \in [n]$, the budget B is 1, and for each $\ell \in [n]$, $c^{(\ell)}$ is the additive cost function defined by $c^{(\ell)}_e = M \geq (1 + n\gamma^n)B$ for all $e \in [\ell - 1]$, $c^{(\ell)}_\ell = 1$, and $c^{(\ell)}_e = 0$ for all $e \in \{\ell + 1, \ldots, n\}$. The distribution $\mathcal{D}$ chooses input $c^{(\ell)}$ with probability $K/OPT_{\mathsf{Alg}}(v, B, c^{(\ell)})$, where K is a normalization constant; so $\mathbb{E}_{c\sim\mathcal{D}}\big[OPT_{\mathsf{Alg}}(v, B, c)\big] = nK$.

The proof of Theorem 3.2 uses truthfulness, IR, and budget-feasibility to argue that, letting $q_\ell = \Pr\big[\mathcal{M} \text{ returns a set containing item } \ell \text{ on input } c^{(\ell)}\big]$, we have $\sum_{\ell\in[n]} q_\ell \leq 1$. We then obtain that letting Val_ℓ denote the expected value obtained by $\mathcal{M}$ on input $(v, B, c^{(\ell)})$, we have

$$Val_\ell \leq q_\ell \cdot OPT_{\mathsf{Alg}}(v, B, c^{(\ell)}) + (1 - q_\ell)v(\{\ell + 1, \ldots, n\}) + \tfrac{\gamma^{n-1}}{1+n\gamma^n} \cdot v(\ell).$$

Multiplying the above by $\frac{1}{OPT_{\mathsf{Alg}}(v,B,c^{(\ell)})} \leq \frac{1}{v(\ell)}$ and simplifying, we obtain that

$$\sum_{\ell\in[n]} \frac{Val_\ell}{OPT_{\mathsf{Alg}}(v, B, c^{(\ell)})} \leq \sum_{\ell\in[n]} \Big(q_\ell + \tfrac{(1-q_\ell)}{\gamma-1} + \tfrac{\gamma^{n-1}}{1+n\gamma^n}\Big) \leq 1 + \frac{n-1}{\gamma - 1} + \frac{1}{\gamma} \leq \frac{n}{\gamma - 1} \leq 1 + \epsilon.$$

Next consider the setup of Theorem 3.3 (b), which considers n to be a power of 2, the additive valuation v specified by $v(e) = 1$ for all $e \in U$, budget $B = n$, and additive cost functions $d^{(\ell)}$ specified by $d^{(\ell)}(e) = 2^\ell$ for all $e \in U$, where ℓ ranges in $I = \{0\} \cup [\log_2 n]$. The arguments therein show that if a_ℓ is the expected number of items procured by a Bayesian budget-feasible mechanism, then we have $\sum_{\ell=0}^{\log_2 n} 2^{\ell-1} a_\ell \leq n$. Consider the distribution where $d^{(\ell)}$ is chosen with probability $2^\ell/K$ for some normalization constant K. The expected value of the optimum is then $\frac{n}{K}(1 + \log_2 n)$, whereas the expected value returned by the mechanism is at most $\sum_{\ell=0}^{\log_2 n} 2^\ell a_\ell/K \leq \frac{2n}{K}$. □

A.2 Proof of Theorem 3.6: approximation-factor lower bounds relative to OPT_{Bench}

As mentioned in Section 3, we adapt the lower bound construction given by [4] to our setting. They describe an instance with an additive valuation in the *Bayesian* setting, that is, the costs for the items are drawn from a distribution. They show that there is no budget-feasible mechanism that achieves expected value strictly larger than $(1-\frac{1}{e})\mathbb{E}[OPT_{\mathsf{Alg}}]$, where the expectation is over the distribution of the costs and the randomness of the mechanism. (Recall that in our terminology, a budget-feasible mechanism is also truthful.)

We will use the same instance as theirs, where each seller holds a single item. In order to convert their lower bound into a lower bound for OPT_{Bench}, we wish to utilize the fact that $v_{\max} \ll OPT_{\mathsf{Alg}}$ in the large market setting. Since $OPT_{\mathsf{Alg}} \geq OPT_{\mathsf{Bench}} \geq OPT_{\mathsf{Alg}} - v_{\max}$, this implies that OPT_{Bench} essentially coincides with OPT_{Alg}, and thus any lower bound with respect to OPT_{Alg} is also a lower bound with respect to OPT_{Bench}.

Such an argument will show that there is no budget-feasible mechanism that achieves expected value strictly larger than $(1-\frac{1}{e})\mathbb{E}[OPT_{\mathsf{Bench}}]$, where the expectation is over the distribution of the costs and the randomness of the mechanism. However, one needs to be careful when arguing that this implies a lower bound in the worst-case setting. Typically, one argues that, if a mechanism $\mathcal{M}$ achieves approximation ratio less than α in the Bayesian setting, when given as input some distribution $\mathcal{D}$ of the costs, then there exists some $c \in \text{supp}(\mathcal{D})$ for which $\mathcal{M}$ achieves approximation ratio less than α in the worst-case setting, when given c as input. However, for the distribution $\mathcal{D}$ used in the lower bound instance of Anari et al., there exist cost vectors c drawn from $\mathcal{D}$) for which $OPT_{\mathsf{Alg}}(c)$ is not much larger than $v_{\max}$, and so OPT_{Bench} could be much smaller than OPT_{Alg}. Nonetheless, one can show that the distribution $\mathcal{D}$ satisfies $v_{\max} \ll OPT_{\mathsf{Alg}}(c)$ with very high probability, and this suffices to show a lower bound in the worst-case setting.

Now, we describe the lower bound instance of [4]. We are working in the single-dimensional setting, where each seller holds one item. So $n = |U|$ is the same as the number of players k. The valuation function is additive, with $v(e) = 1$ for every item e. Consider the Bayesian instance $\mathcal{I}$, where the cost of each item is drawn independently from the distribution $\mathcal{D}$ whose CDF $F(x)$ is given by

$$F(x) = \begin{cases} \frac{1}{e(1-x)}, & \text{for } 0 \leq x \leq 1-\frac{1}{e} \\ 1, & \text{for } x > 1-\frac{1}{e} \end{cases}$$

Note that the item costs i.i.d. Let $\mathcal{D}^k$ denote the joint distribution of costs over all the items. Let $\bar{c} = \mathbb{E}_{x\sim\mathcal{D}}[x]$. The budget B of the instance is set to be $B = \bar{c}\cdot n = \mathbb{E}_{c\sim\mathcal{D}^k}[c(U)]$. We use the following result proved in [4].

Lemma A.1 ([4]). *No budget-feasible mechanism can achieve value better than* $(1-\frac{1}{e})\cdot\mathbb{E}_{c\sim\mathcal{D}^k}[OPT_{\mathsf{Alg}}]$, *for the instance* $\mathcal{I}$.

We also need to make use of a Hoeffding bound, which is stated in the following lemma.

Lemma A.2. *Let* $x_1,\ldots,x_n$ *be i.i.d. random variables whose values always lie in* $[a,b]$. *Let* $\mu = \mathbb{E}[\sum_{i=1}^n x_i]$. *Then we have*

$$\Pr\left[\sum_{i=1}^n x_i \geq (1+\epsilon)\cdot\mu\right] \leq e^{-\frac{2\epsilon^2\mu^2}{n(b-a)^2}}$$

Finishing up the proof of Theorem 3.6. Suppose that there exists a budget-feasible mechanism $\mathcal{M}$ for this instance that achieves an α-approximation with respect to OPT_{Bench}. e run $\mathcal{M}$ on the instance $\mathcal{I}$. For $c \in \mathbb{R}_+^n$, let $\mathcal{M}(c)$ denote the (expected) value procured by $\mathcal{M}$ on instance $\mathcal{I}$ with costs c. Since the value of each item is 1, and since each seller holds one item, we always have $OPT_{\mathsf{Bench}} = OPT_{\mathsf{Alg}} - 1$ for the instance. So $\mathbb{E}_{c\sim\mathcal{D}^n}[\mathcal{M}(c)] \geq \alpha\mathbb{E}_{c\sim\mathcal{D}^n}[OPT_{\mathsf{Bench}}] = \alpha(\mathbb{E}_{c\sim\mathcal{D}^n}[OPT_{\mathsf{Alg}}] - 1)$. Since the cost of each item is drawn i.i.d., by Lemma A.2, we can show that the sum of the costs is concentrated around its mean, which is B. In particular, for any constant $\epsilon > 0$, we have,

$$\Pr[c(U) \geq (1+\epsilon)B] \leq e^{-\rho n}$$

where $\rho := \rho(\epsilon)$ is some constant depending only on ϵ. Thus, with probability $\geq 1 - e^{-\rho n}$, we have that $c(U) < (1+\epsilon)B$.

Suppose that the event $c(U) < (1+\epsilon)B$ occurs. Let $e_1, \ldots, e_k$ be the elements of U in ascending order of their costs. Pick the smallest index ℓ such that $S = \{e_1, \ldots, e_\ell\}$ satisfies $c(S) \geq B$. We claim that $v(S) \geq \frac{n}{1+\epsilon}$. Indeed, since S is the set of smallest total cost among all sets on ℓ elements, we have that the average cost of an element in S, which is $\frac{c(S)}{\ell}$, is at most the average cost, $\frac{c(U)}{n}$, over all the elements. Thus, $\frac{B}{\ell} \leq \frac{c(S)}{\ell} \leq \frac{c(U)}{n} < \frac{(1+\epsilon)B}{n}$ implying that $\ell > \frac{n}{1+\epsilon}$. Hence, the set $S' = S \setminus \{e_\ell\}$ satisfies $c(S') \leq B$ and $v(S') \geq \frac{n}{1+\epsilon} - 1$.

Thus, with probability $\geq 1 - e^{-\rho n}$, we have that $OPT_{\mathsf{Alg}} \geq \frac{n}{(1+\epsilon)} - 1$. Now the guarantee of $\mathcal{M}$ with respect to the algorithmic optimum is

$$\mathbb{E}_{c\sim\mathcal{D}^n}[\mathcal{M}(c)] \geq \alpha(\mathbb{E}_{c\sim\mathcal{D}^n}[OPT_{\mathsf{Alg}}] - 1) \geq \alpha(1 - e^{-\rho n})\left(\frac{n}{1+\epsilon} - 2\right)$$

For large enough n, we can pick $\epsilon' > 0$ small enough so that we have $(1 - e^{-\rho n})\left(\frac{n}{1+\epsilon} - 2\right) \geq (1-\epsilon')n$. Thus, $\mathbb{E}_{c\sim\mathcal{D}^n}[\mathcal{M}(c)] \geq (1-\epsilon')n \geq (1-\epsilon')\mathbb{E}_{c\sim\mathcal{D}^n}[OPT_{\mathsf{Alg}}]$. Hence $\mathcal{M}$ achieves a $(1-\epsilon')\alpha$ approximation with respect to OPT_{Alg}. By Lemma A.1, we must have that $(1-\epsilon')\alpha \leq 1 - \frac{1}{e}$, showing that we cannot get a better than $\frac{1}{1-\epsilon'}(1 - \frac{1}{e})$ approximation with respect to OPT_{Bench}. □

B Proof of Lemma 4.5

We mimic the proof of Lemma 2.4. Let I be a minimal prefix of $[k]$ such that, letting $A_1 = \bigcup_{i\in I}(S \cap G_i)$, we have $g_{-2\ell}(A_1) \geq \frac{g_{-(4\ell+1)}(S)}{2}$. Let $A_2 = S - A_1$. Let r be the last index in I. Then, we have $g_{-2\ell}(A_1 - G_r) < \frac{g_{-(4\ell+1)}(S)}{2}$. Let $J_1 \subseteq [k]$, $J_2 \subseteq [k]$ with $|J_1|, |J_2| \leq 2\ell$ be such that $g_{-2\ell}(A_1 - G_r) = g\big(A_1 - G_r - \bigcup_{i\in J_1} G_i\big)$ and $g_{-2\ell}(A_2) = g\big(A_2 - \bigcup_{i\in J_2} G_i\big)$ and Let $S' = S - \bigcup_{i\in J_1\cup J_2} G_i - G_r$. Since $|J_1 \cup J_2 \cup \{r\}| \leq 4\ell + 1$, we have $g(S') \geq g_{-(4\ell+1)}(S)$. Also, $A_2 - \bigcup_{i\in J_2} G_i = S' - (A_1 - G_r - \bigcup_{i\in J_1} G_i)$, and so by subadditivity of g, we have

$$g_{-2\ell}(A_2) = g\Big(A_2 - \bigcup_{i\in J_2} G_i\Big) \geq g(S') - g\Big(A_1 - G_r - \bigcup_{i\in J_1} G_i\Big) > \frac{g_{-(4\ell+1)}(S)}{2}.$$

Now fix a partition A_1', A_1'' of A_1, and a partition A_2', A_2'' of A_2. Observe that $g_{-\ell}(A_1') + g_{-\ell}(A_1'') \geq g_{-2\ell}(A_1)$. So some set $A_1^H \in \{A_1', A_1''\}$ satisfies $g_{-\ell}(A_1^H) \geq g_{-2\ell}(A_1)/2$; let A_1^L be the other set in $\{A_1', A_1''\}$. Similarly, one of A_2', A_2'', denoted A_2^H satisfies and $g_{-\ell}(A_2^H) \geq g_{-2\ell}(A_2)/2$; let A_2^L be the other set in $\{A_2', A_2''\}$.

Now we proceed exactly as in the proof of Lemma 2.4. The random partition U_1, U_2 induces random partitions of A_1 and A_2. Consider the event Γ that for both $\ell = 1, 2$, the random partition of A_ℓ induced by U_1, U_2 is the same as the partition A_ℓ^H, A_ℓ^L, up to permutations of the parts. For any $j = 1, 2$, we have $\Pr\big[U_j \cap A_1 = A_1^L,\ U_j \cap A_2 = A_2^L \,|\, \Gamma\big] = \frac{1}{4}$. So conditioned on Γ, with probability at least $\frac{1}{2}$, we have that both U_1, U_2 contain some big set. Removing the conditioning completes the proof. □