An $O(\log\log n)$-approximate budget feasible mechanism for subadditive valuations*

Rian Neogi† Kanstantsin Pashkovich† Chaitanya Swamy†

Abstract

In *budget-feasible mechanism design*, there is a set of items U, each owned by a distinct seller. The seller of item e incurs a private cost $\bar{c}_e$ for supplying her item. A buyer wishes to procure a set of items from the sellers of maximum value, where the value of a set $S \subseteq U$ of items is given by a valuation function $v : 2^U \to \mathbb{R}_+$. The buyer has a budget of $B \in \mathbb{R}_+$ for the total payments made to the sellers. We wish to design a mechanism that is *truthful*, that is, sellers are incentivized to report their true costs, *budget-feasible*, that is, the sum of the payments made to the sellers is at most the budget B, and that outputs a set whose value is large compared to $\text{OPT} := \max\{v(S) : \bar{c}(S) \leq B, S \subseteq U\}$.

Budget-feasible mechanism design has been extensively studied, with the literature focusing on (classes of) *subadditive* (or complement-free) valuation functions, and various polytime, budget-feasible mechanisms, achieving constant-factor approximation to OPT, have been devised for the special cases of additive, submodular, and XOS (or fractionally subadditive) valuations. However, for general subadditive valuations, the best-known approximation factor achievable by a polytime budget-feasible mechanism (given access to demand oracles) was only $O(\log n/\log\log n)$, where $n = |U|$ is the number of items.

We improve this state-of-the-art significantly by designing a randomized budget-feasible mechanism for subadditive valuations that *achieves a substantially-improved approximation factor of* $O(\log\log n)$ *and runs in polynomial time, given access to demand oracles.*

Our chief technical contribution is to show that, given any set $S \subseteq U$, one can construct in polynomial time a distribution $\mathcal{D}$ over posted-payment vectors $d \in \mathbb{R}_+^S$ satisfying $d(S) \leq B$ (where $d(S) = \sum_{e \in S} d_e$) such that, $\mathbb{E}_{d \sim \mathcal{D}}[v(\{e \in S : c_e \leq d_e\})] \geq \frac{1}{O(\log\log n)} \cdot v(S)$ for every cost-vector $c \in \mathbb{R}_+^S$ satisfying $c(S) \leq B/O(\log\log n)$. Using this distribution, we show how to construct a budget feasible mechanism for subadditive valuations that achieves an approximation factor of $O(\log\log n)$.

1 Introduction

In *budget-feasible mechanism design*, there is a ground set U of n elements or items, and a valuation function $v : 2^U \to \mathbb{R}_+$, where $v(S)$ specifies the value of a subset $S \subseteq U$. We are also given a budget $B \in \mathbb{R}_+$. The mechanism designer (or buyer) seeks to procure a set of elements of maximum value. Each element e is held by a strategic seller who incurs a private cost $\bar{c}_e \geq 0$ for supplying item e; we will often identify sellers with the items they hold. The central difficulty here is that the private costs $\{\bar{c}_e\}$ are known only to the sellers, and not to the mechanism designer. Each seller e reports a cost c_e to the mechanism, which may not be equal to her true cost. In order to incentivize sellers

*An extended abstract is to appear in the Proceedings of the 26th ACM Conference on Economics and Computation (EC) 2025.

†{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.

to reveal their true costs, the mechanism designer makes payments to the sellers. The utility of a seller is equal to the payment it receives minus the cost incurred by potentially supplying the item. Our goal is to design a *truthful* mechanism, where each seller maximizes their utility by reporting her true cost. Additionally, the mechanism designer also has a budget B and we need to ensure that the sum of the payments given out to the sellers do not exceed this budget.

We want to design a mechanism that outputs a set that is a good approximation to the best set one can get subject to the budget constraint. That is to say, we compare the value of our solution to the benchmark $\mathrm{OPT} := \max\{v(S) : c(S) \leq B\}$. Note that for a truthful mechanism, it does not matter whether we consider the true costs or the reported costs for the budget constraint. For general valuation functions as input, the work of [Sin10] shows that no non-trivial approximation is possible for any budget feasible mechanism. Hence work has been focussed on special classes of valuation functions, such as, typically (in increasing order of generality) additive, submodular, XOS, or subadditive valuations.

For subadditive valuations, [BCGL17] made an important contribution showing that the problem admits a randomized $O(\log n/\log\log n)$-approximate budget feasible mechanism. (Recall that $n = |U|$). They also provide an $O(1)$-factor approximate mechanism in the Bayesian setting, where costs are sampled from a prior distribution that is known to the mechanism designer. They use Yao's minimax principle to show that their $O(1)$-approximate Bayesian mechanism implies the *existence* of an $O(1)$-factor approximate mechanism in the prior-free setting. This work was later improved in two ways: [BGG$^+$22] achieved a *deterministic* $O(\log n/\log\log n)$ approximation, and [NPS24] provide a *constructive* $O(1)$-factor approximate mechanism in the prior-free setting, that runs in exponential time.

1.1 Our contributions

We design a budget-feasible mechanism for subadditive valuations that achieves an $O(\log\log n)$-approximation, and runs in polynomial time given access to demand oracles. This is a substantial improvement over the previous-best approximation factor of $O(\log n/\log\log n)$ achievable in poly-time (also given demand oracle) [BCGL17, BGG$^+$22].

Theorem 1.1. *There is a truthful budget feasible mechanism for subadditive valuations that achieves an approximation factor of $O(\log\log n)$, and runs in polynomial time given a demand oracle for the valuation v.*

Our mechanism is randomized: it satisfies truthfulness and the budget constraint with probability 1, and obtains expected value at least $\mathrm{OPT}/O(\log\log n)$. To state our key technical contribution, it will be useful to discuss some snippets from prior work on budget-feasible mechanism design for subadditive valuations. An important insight to emerge from the work of [BCGL17, NPS24] is the following structural property of subadditive functions, stated explicitly in [NPS24]. (This is quite implicit in [BCGL17] hiding behind their use of Yao's principle.)

Theorem 1.2 ([NPS24])**.** *Let $v : 2^U \to \mathbb{R}_+$ be a subadditive function and $S^* \subseteq U$. Let $K \subseteq \mathbb{R}_+^{S^*}$ be a finite set. There exists a distribution $\mathcal{D}^*$ over vectors in K such that, for any $c \in K$, we have $\mathbb{E}_{d\sim\mathcal{D}^*}[v(\{e \in S^* : c_e \leq d_e\})] \geq \frac{1}{2}v(S^*)$.*

For completeness, we include the proof of the above theorem in the appendix. [NPS24] show that this leads to an $O(1)$-approximate budget-feasible mechanism by computing a suitable set S^* of large value, and taking K to a discretized version of the simplex $\Delta^{S^*}(B) := \{x \in \mathbb{R}_+^{S^*} : \sum_{e\in S^*} x_e \leq B\}$: a vector d sampled from the distribution $\mathcal{D}^*$ yields take-it-or-leave-it posted payments for the sellers (that sum to at most B), so that if we select seller e only if $c_e \leq d_e$ we obtain expected value at least $\frac{1}{2}v(S^*)$. However, computing the distribution in Theorem 1.2 entails solving an exponentially large LP, and so, this only yields an *exponential-time mechanism*.

Our *chief technical contribution* is to prove the following relaxed version of Theorem 1.2, which shows that, for the simplex $\Delta^{S^*}(B)$, a distribution satisfying a certain weaker guarantee *can be computed in polytime*. For a vector $\alpha \in \mathbb{R}^U$ an $S \subseteq U$, let $\alpha(S)$ denote $\sum_{e \in S} \alpha_e$.

Theorem 1.3 (Informal version of Theorem 3.1)**.** *Let $v : 2^U \to \mathbb{R}_+^U$ be a subadditive function, $S^* \subseteq U$, and $B \in \mathbb{R}_+$. Using demand oracles, one can compute in polynomial time a distribution $\mathcal{D}$ over vectors in $\Delta^{S^*}(B)$, such that for every $c \in \mathbb{R}_+^{S^*}$ satisfying $c(S^*) \leq \frac{B}{O(\log \log n)}$, we have the guarantee $\mathbb{E}_{d \sim \mathcal{D}}[v(\{e \in S^* : c_e \leq d_e\})] \geq \frac{v(S^*)}{O(\log \log n)}$.*[1]

To utilize Theorem 1.3 to obtain our $O(\log \log n)$-approximate mechanism, our plan, roughly speaking, is to use Theorem 1.3 in the same fashion as Theorem 1.2: that is, we use a vector d sampled from the distribution $\mathcal{D}$ in Theorem 1.3 as posted payments for the sellers. But implementing this plan takes some work in order to deal with the case where the cost-vector c is not handled by Theorem 1.3 (i.e., we have $c(S^*) > B/O(\log \log n)$). We cannot simply use the mechanisms of [NPS24, BCGL17], replacing Theorem 1.2 with Theorem 1.3, because these mechanisms use oracles, such as knapsack-cover oracle [NPS24], approximate XOS oracle [BCGL17], that are different from (and somewhat more-powerful than) a demand oracle. (In [NPS24], the set S^* is obtained using an oracle they call a knapsack-cover oracle; whereas in [BCGL17], the case where $c(S^*)$ is large is dealt with by assuming that one has an XOS oracle for an XOS-approximation of v.) We assume only demand-oracle access in Theorem 1.1, so we need to come up with some novel ideas to deal with the case where $c(S^*)$ is large. Combining these with Theorem 1.3 leads to Theorem 1.1.

1.2 Technical overview

Theorem 1.3 constitutes the technical core of our mechanism, and we begin by discussing the key ideas that go toward proving this result.

Fix a subadditive valuation $v : 2^U \mapsto \mathbb{R}_+$. For $B \in \mathbb{R}_+$ and $S \subseteq U$, let $\Delta^S(B) := \{x \in \mathbb{R}_+^S : \sum_{e \in S} x_e \leq B\}$. We use $\Delta(B)$ as a shorthand for $\Delta^U(B)$. Fix $\gamma \in (0, 1]$. We focus on the case $S^* = U$ in Theorem 1.3, which will allow us to convey the main ideas.

The statements of Theorem 1.2 and Theorem 1.3 can be viewed as statements about optimal strategies in a two-player zero-sum item bidding game. We refer to the two players as the d-player and the c-player. The d-player chooses a bid vector $d \in \Delta(B)$. For each $e \in U$, we interpret d_e as the d-player's bid for the item e. The restriction $d \in \Delta(B)$ ensures that the sum of the bids of the d-player is within the budget B. The c-player chooses another vector $c \in \Delta(\gamma B)$, representing the c-player's bids[2]. The payoff of the game for the d-player is $v(\{e \in U : c_e \leq d_e\})$, that is, it is the value of the set of items where the d-player's bid is higher than the c-player's bid. The goal of the d-player is to maximize this payoff, and goal of the c-player is to minimize this payoff.

By von Neumann's minimax theorem, this zero-sum game has a Nash Equilibrium over mixed-strategies for the d-player and c-player, and the payoff to the d-player under a mixed Nash equilibrium is equal to the max-min value of the game.[3] Note that a mixed-strategy for the d-player is a

[1]The set K in Theorem 1.2 is finite, but here the guarantee holds for all c satisfying the stated condition.

[2]The use of notation c here is suggestive. We will later take this vector to be the vector of reported costs $\{c_e\}_{e \in U}$ from the budget-feasible mechanism design problem.

[3]von Neumann's minimax theorem requires a finite space of pure strategies, and so, more precisely, we need to work with a discretization K of the strategy space $\Delta(B)$; accordingly, in (1), the $\min_{c \in \Delta(B)}$ should really be $\min_{c \in K}$. Under the assumption that the sellers use some polynomial number s of bits to describe their reported cost, such a

distribution over vectors in $\Delta(B)$, and for the c-player, is a distribution over vectors in $\Delta(\gamma B)$. The max-min value corresponds to the d-player going first, followed by the c-player, so we may assume that the c-player plays pure strategies only. Taking K to be (more precisely, a discretization of) $\Delta(B)$, Theorem 1.2 shows that, for any $\gamma \leq 1$, this max-min value is at least $\frac{1}{2}v(U)$. That is,

$$\max_{\substack{\text{distribution } \mathcal{D} \\ \text{over vectors in } \Delta(B)}} \min_{c \in \Delta(B)} \mathbb{E}_{d \sim \mathcal{D}}[v(\{e \in U : c_e \leq d_e\})] \geq \frac{1}{2}v(U). \tag{1}$$

Theorem 1.3 can also be interpreted as a statement about finding mixed-strategies for the item bidding game. It says that, when $\gamma = O(\frac{1}{\log \log n})$, *we can compute in polynomial time*, a mixed strategy for the d-player (that is, a distribution over vectors in $\Delta(B)$) that achieves a payoff that is a $O(\log \log n)$-approximation to the optimal payoff. That is,

$$\min_{c \in \Delta(B/O(\log \log n))} \mathbb{E}_{d \sim \mathcal{D}}[v(\{e \in U : c_e \leq d_e\})] \geq \frac{v(U)}{O(\log \log n)} \tag{2}$$

With this two-player-game view in mind, we next describe how to efficiently compute a good mixed-strategy for the d-player in the item bidding game, which leads to the proof of Theorem 1.3.

Finding good strategies. As mentioned earlier, computing the distribution in Theorem 1.2 entails solving an exponentially-large LP, with variables and constraints for the players' pure strategies, which we cannot hope to do in polynomial time even with demand oracles. Nonetheless, let us attempt to find such a distribution even if the guarantee on its payoff is suboptimal. When $\gamma = 1$ both players pick a vector in $\Delta(B)$. In this setting, one can show that any pure strategy of the d-player fails to give payoff better than $\max_{e \in U} v(e)$ in the worst case (see Theorem A.1). Thus, the d-player must resort to using mixed strategies in order to get a decent payoff.

How should the d-player randomize? To gain intuition, suppose that there we have k disjoint sets $S_1, \ldots, S_k \subseteq U$ with the property that each set S_i has large value, that is, $v(S_i) \geq \alpha v(U)$ for all $i = 1, \ldots, k$, for some fixed $\alpha \in (0, 1)$. The d-player can utilize this partition to gain an advantage over the c-player. The d-player selects a set S from $S_1, \ldots, S_k$ uniformly at random and focuses their entire budget B on elements in S, giving bids of zero to elements in $U \setminus S$. The c-player does not know the random choice of the d-player. No matter what bids are chosen by c-player, we will always have that $\mathbb{E}[c(S)] \leq \frac{B}{k}$. This is because, for any item e, the probability that $e \in S$ is at most $\frac{1}{k}$.

Herein lies the advantage of using randomization: the d-player gets to use their entire budget of B on the set of items in S, whereas the c-player has a budget of only $\frac{B}{k}$ (in expectation) to use for items in S. Thus, the d-player can utilize the sets $S_1, \ldots, S_k$ to effectively boost his own budget by a factor of k. As we will show, the diminished bidding power for the c-player over the items in S will make it possible for the d-player to find a *pure strategy* over S that achieves a large payoff compared to $v(S)$. This is in contrast to the situation where both the d-player and the c-player have the same budget B, for which no pure strategy provides a good payoff for the d-player (Theorem A.1). Since $v(S) \geq \alpha v(U)$, a strategy that attains a large payoff compared to $v(S)$ also attains a large payoff compared to $v(U)$, albeit with an α-factor loss.

This is, in essence, the idea underlying how we compute the distribution $\mathcal{D}$ in Theorem 1.3. However, there are a few things left unspecified and/or are oversimplified in the above exposition. (1) For what values of α and k do there exist such disjoint sets $S_1, \ldots, S_k$ of U such that $v(S_i) \geq \alpha v(U)$ for all $i = 1, \ldots, k$, and how can we find these sets in polynomial time? (2) Once a set

discretization is always possible since we may limit ourselves to 2^s possibilities for each entry of d and c.

$S \in \{S_1, \ldots, S_k\}$ is sampled, how can the d-player find a pure strategy over the items in S that guarantees good payoff? (3) We have $\mathbb{E}[c(S)] \leq \frac{B}{k}$, but it need not be true that the sampled set satisfies $c(S) \leq \frac{B}{k}$ always.

Dealing with (3) is straightforward. We simply use Markov's inequality to bound $c(S)$. For any $\epsilon > 0$, Markov's inequality guarantees that $c(S) \leq \frac{B}{\epsilon k}$ with probability at least $1 - \epsilon$. This has the downside of boosting the effective budget of the c-player by a factor of $\frac{1}{\epsilon}$. We remedy this by asserting that the c-player pick a vector $c \in \mathbb{R}_+^U$ such that $c(U) \leq \epsilon B$ in the first place, that is, we assume that the c-player is already restricted to a limited budget of ϵB. With this assumption, using Markov's inequality, we get that $c(S) \leq \frac{B}{k}$ with probability at least $1 - \epsilon$. We end up using $\epsilon = \frac{1}{O(\log \log n)}$, which is why we require that $c(U) \leq \frac{B}{O(\log \log n)}$ in the statement of Theorem 1.3.

Dealing with issues (1) and (2) is more involved. To address (1), we observe that we do not really need disjoint sets $S_1, \ldots, S_k$ with $v(S_i) \geq \alpha v(U)$ for all $i = 1, \ldots, k$. Sampling a set S uniformly from $S_1, \ldots, S_k$ induces a distribution over sets with the properties (i) $\mathbb{E}_S[v(S)] \geq \alpha v(U)$ and (ii) $\Pr[e \in S] \leq \frac{1}{k}$. *We really only need a distribution $\mathcal{S}$ over subsets $S \subseteq U$ satisfying (i) and (ii) for our argument.* Property (i) ensures that the d-player retains a large fraction of the total value when sampling a set S from $\mathcal{S}$, and property (ii) ensures that $\mathbb{E}[c(S)] \leq \frac{B}{k}$ when sampling S from $\mathcal{S}$. This naturally leads us to the following LP, which is parameterized by a subset $S \subseteq U$ and value $\kappa \in (0, 1)$, which finds the distribution of largest expected value with marginals at most κ.

$$
\begin{aligned}
\text{maximize} \quad & \sum_{T \subseteq S} v(T) x_T && \qquad \big(\text{LP}\,(\kappa, S)\big) \\
\text{subject to} \quad & \sum_{T : e \in T} x_T \leq \kappa \qquad \forall e \in S \\
& \sum_{T \subseteq S} x_T \leq 1 \\
& x \geq 0.
\end{aligned}
$$

A feasible solution x to $\big(\text{LP}\,(\kappa, S)\big)$ yields a distribution over subsets T of S, where we sample a set $T \neq \emptyset$ with probability x_T, and sample $\emptyset$ with probability $1 - \sum_{T' \subseteq S} x_{T'}$. (which is non-negative due to the second constraint). The constraint $\sum_{T : e \in T} x_T \leq \kappa$ for all $e \in S$ ensures that $\Pr_T[e \in T] \leq \kappa$. Moreover, the objective function $\sum_{T \subseteq S} v(T) x_T$ is simply $\mathbb{E}_T[v(T)]$.

It is not hard to show that we can solve $\big(\text{LP}\,(\kappa, S)\big)$ in polynomial time using demand oracles, since the separation problem for the dual LP (see $\big(\text{DLP}(\kappa, S)\big)$) amounts to precisely a demand-oracle computation. Let $\text{OPT}_{\text{LP}}(\kappa, S)$ denote the optimal value of $\big(\text{LP}\,(\kappa, S)\big)$. The distribution corresponding to this optimal solution now replaces the partition $S_1, \ldots, S_k$. The d-player now samples a set S from this distribution and restricts their bids to elements in S. In the special case when there exists a partition $S_1, \ldots, S_k$ satisfying $v(S_i) \geq \alpha v(U)$ for all $i = 1, \ldots, k$, the uniform distribution over $S_1, \ldots, S_k$ is a feasible solution to $\big(\text{LP}\,(\kappa, U)\big)$, showing that $\text{OPT}_{\text{LP}}(\kappa, U) \geq \alpha v(U)$. However, the LP optimizes over a much richer family of distributions, and so the value of the LP can be high even when such a partition does not exist.

Finally, to address (2), we show that the *optimal solution to the dual of* $\big(\text{LP}\,(\kappa, S)\big)$ (see $\big(\text{DLP}(\kappa, S)\big)$) can be used to design a pure strategy (that is, a vector of bids) for the d-player. This is captured by the following lemma.

Lemma 1.4 (Informal version of Lemma 3.2). *Fix $B \in \mathbb{R}_+$, $\kappa \in (0, 1)$ and $S \subseteq U$. Using demand oracles, we can compute in polynomial time, a vector $d \in \mathbb{R}_+^S$ such that $d(S) = B$, and $v(\{e \in S : c_e \leq d_e\}) \geq v(S) - \text{OPT}_{\text{LP}}(\kappa, S)$ holds for all $c \in \Delta^S(\kappa B)$.*

In other words, the d-player can play a pure strategy over the items in S that achieves good payoff as long as $\mathrm{OPT}_{\mathrm{LP}}(\kappa, S)$ is small.

This discussion leads to the following modified approach for the d-player.

1. First, sample a set $S \subseteq U$ from the distribution given by the optimal solution of $\big(\mathrm{LP}\,(\kappa, U)\big)$ for some κ to be decided later.

2. Then, choose a good pure strategy for the d-player over the items in S using Theorem 1.4. Extend it to a pure strategy for all the items U by setting zeroes for entries corresponding to elements in $U \setminus S$.

The above two step process leads to a mixed strategy for the d-player, where the randomness comes from the random choice of S.

The only thing remaining is to decide what value of κ to pick. This is a balancing act. If κ is too small, it could be that $\mathrm{OPT}_{\mathrm{LP}}(\kappa, U)$ is too small, which means that the d-player has no hope of achieving a good payoff, even if he wins all the items in S. If κ is too large, then Theorem 1.4 might fail to guarantee a good payoff (to see this, consider the extreme case of $\kappa = 1$, in which case $\mathrm{OPT}_{\mathrm{LP}}(\kappa, S) = v(S)$ and so Theorem 1.4 provides no guarantee on the payoff).

For a fixed subset $S \subseteq U$, Theorem 1.4 guarantees a payoff of at least $v(S) - \mathrm{OPT}_{\mathrm{LP}}(\kappa, S)$. When this set S is sampled from the distribution $\mathcal{S}$ coming from the optimal solution of $\big(\mathrm{LP}\,(\kappa, U)\big)$, the expected payoff is at least $\mathbb{E}_{S \sim \mathcal{S}}[v(S) - \mathrm{OPT}_{\mathrm{LP}}(\kappa, S)]$. The first term $\mathbb{E}_{S \sim \mathcal{S}}[v(S)]$ is simply $\mathrm{OPT}_{\mathrm{LP}}(\kappa, U)$. It turns out that the second term $\mathbb{E}_{S \sim \mathcal{S}}[\mathrm{OPT}_{\mathrm{LP}}(\kappa, S)]$ can be upper bounded by $\mathrm{OPT}_{\mathrm{LP}}(\kappa^2, U)$. Thus, the expected payoff for the d-player using this strategy is at least $\mathrm{OPT}_{\mathrm{LP}}(\kappa, U) - \mathrm{OPT}_{\mathrm{LP}}(\kappa^2, U)$. A result of [DKL20] shows that there exists $\kappa \in (0, 1)$ for which $\mathrm{OPT}_{\mathrm{LP}}(\kappa, U) - \mathrm{OPT}_{\mathrm{LP}}(\kappa^2, U)$ is at least $\frac{1}{O(\log\log n)} \cdot v(U)$. This is the value of κ that we will use for step (1). Combining everything, we obtain a distribution $\mathcal{D}$ over vectors in $\Delta(B)$, for which $\mathbb{E}[v(\{e \in U : c_e \le d_e\})] \ge \frac{1}{O(\log\log n)} v(U)$ for all $c \in \Delta(\frac{B}{O(\log\log n)})$.

Constructing the mechanism. To complement Theorem 1.3, we need to show how one can use the distribution (or equivalently, mixed-strategy for the d-player) provided by Theorem 1.3 to design a budget-feasible mechanism achieving a good approximation. Our mechanism (see Algorithm 1 in Section 4) builds upon the framework used by [BCGL17] (also later used by [NPS24]). We randomly partition the set of sellers into two groups U_1, U_2, get an estimate of OPT from the sellers in U_1 who are then discarded, and work over the sellers in U_2. The set S^* we use in Theorem 1.3 is obtained by computing a demand set over U_2 with appropriately chosen prices. This ensures that $v(S^*) = \Omega(\mathrm{OPT})$. As discussed earlier, we sample a posted-payment vector d from the distribution $\mathcal{D}$ in Theorem 1.3, which yields the set of elements $R = \{e \in S^* : c_e \le d_e\}$. The value d_e can be thought of as the *threshold* for seller e, in the sense that e will not be picked into the solution if it declares a cost higher than d_e. So we often refer to the vector d as a *threshold vector* for the elements in S^*. The resulting mechanism is truthful (due to Myerson's characterization of truthfulness, Theorem 2.3) and has payments bounded by d_e, and since $d(U) \le B$, we obtain budget-feasibility. If $c(S^*) \le \frac{B}{O(\log\log n)}$, the approximation factor follows from the guarantee in Theorem 1.3. But we also need to deal with the case where this does not hold. This presents some technical challenges, and here, we need to proceed significantly differently from the mechanisms of [NPS24, BCGL17]. [BCGL17] show in the analysis of their mechanism for subadditive valuations in the Bayesian setting, that when $c(S^*) > \frac{B}{\alpha}$, the subadditive valuation v is actually "α-close to an XOS function", and so one can use a mechanism for XOS valuations in this case. However, this requires one to have a suitable oracle for this XOS-approximation v^{xos} of v: either an XOS-oracle for v^{xos}, or a demand oracle for v^{xos}, as required by the mechanisms for XOS valuations in [BCGL17] and [NPS24] respectively. We only have a demand oracle for v, so this presents a significant challenge. In the mechanism of [NPS24] for subadditive valuations, S^* is obtained using what they call a knapsack-cover oracle for v, which ensures that $c(S^*) \le B$, so they can always resort to Theorem 1.2. In our case, we do not have a knapsack-oracle for v, and we are using Theorem 1.3, so even if we had $c(S^*) \le B$, we still need to deal with the case where $c(S^*) > B/O(\log\log n)$.

We proceed instead as follows. We show that when v is α-close to an XOS function—which happens when $c(S^*) > B/\alpha$—that we can solve another LP using a demand oracle to obtain another suitable threshold-vector d' (see steps 8, 9 of Algorithm 1). We utilize a pruning operation to obtain a set R' such that: $c_e \le d'_e$ for all $e \in R'$, and $d'(R') \le B$. We show that these two properties imply that $v(R') = \Omega(\mathrm{OPT})$, and that it can be procured with payments at most the budget.

1.3 Related work

Budget-feasible mechanism design was introduced by Singer [Sin13], who also gave $O(1)$-factor randomized mechanisms for submodular valuations. This was followed by a sequence of work [BGG$^+$22, BCGL17, ABM17, ABM16, AKS19, JT18, LMSZ21, GJLZ20, DPS11, CGL11] improving approximation ratios for budget feasible mechanisms under additive, submodular, XOS, or subadditive valuations. Variants of the problem were also considered, such as incorporating additional downward monotone set family constraints [ABM16, LMSZ21, NPS24, HHCT23], considering large market assumptions [AGN14], considering multi-unit generalizations [CC14, AKM$^+$23, WZQ$^+$19, KS22, QWC$^+$20], multidimensional generalizations [NPS25], or considering more general budget constraints [NPS24]. Variants incorporating experimental design problems were also looked at [HIM14].

For additive valuations, [GJLZ20] achieved an optimal approximation ratio of 2. For submodular valuations, [BGG$^+$22] provide a deterministic budget feasible mechanism with an approximation ratio of 4.75, which is the current best known approximation ratio. For XOS and subadditive valuations, it is known that no approximation better than $\sqrt{n}$ is possible using only value oracle queries [Sin10], so work on these valuation classes has always assumed access to at least demand oracles. For XOS valuations, an $O(1)$-approximation was known using demand oracles and XOS oracles [BCGL17, ABM17], but only recently a polytime $O(1)$-approximation was obtained using only demand oracles [NPS24]; the latter work also achieves the current-best approximation for XOS valuations. For subadditive valuations, there is an exponential-time $O(1)$-approximate budget feasible mechanism [NPS24], and also polytime mechanisms that achieve an $O(\log n/\log\log n)$-approximation [BCGL17, BGG$^+$22]. A polynomial-time $O(1)$-approximation for subadditive functions using demand oracles still eludes our current understanding. Our work makes progress towards this goal.

The use of distributions with bounded marginals, as in $\big(\mathrm{LP}\,(\kappa, S)\big)$, has been considered before in the context of subadditive valuations [DKL20, BW23]. Importantly, [DKL20] use this LP to construct an $O(\log\log n)$-prophet inequality for online subadditive combinatorial auctions. The existence of a Nash equilibrium for our item-bidding zero-sum game that achieves payoff $\frac{1}{2}v(U)$ was implicit in the work of [BCGL17, NPS24]. Variants of our item-bidding game, in the context of first-price and second-price auctions over subadditive valuations, have been analyzed in [BR11, HKMN11, FFGL15]. Our item-bidding game can also be seen as a generalization of the Colonel Blotto game to subadditive valuations [Bor53]. The Colonel Blotto game has attracted recent attention in theoretical computer science [Rob06, VLS18, BAEJ20, BDD$^+$22].

Future directions

Finding a polynomial time constant-factor approximation for subadditive budget feasible mechanisms remains an interesting open question. As our proof shows, to do so, it would suffice to find a value of κ such that $\mathrm{OPT}_{\mathrm{LP}}(\kappa, U) - \mathrm{OPT}_{\mathrm{LP}}(\kappa^2, U) \geq \frac{1}{O(1)} v(U)$. (Recall that $\mathrm{OPT}_{\mathrm{LP}}(\kappa, U)$ is the maximum expected value of any distribution over subsets of U with marginals at most κ.) Finding such a κ would also imply a constant factor prophet inequality for subadditive combinatorial auctions via static posted prices by the work of [DKL20]. However, [DKL20] provide a counter-example showing that there is a subadditive valuation v for which no such κ exists, and that their guarantee of $\mathrm{OPT}_{\mathrm{LP}}(\kappa, U) - \mathrm{OPT}_{\mathrm{LP}}(\kappa^2, U) \geq v(U)/O(\log \log n)$ is in fact tight. Instead, they suggest considering *non-uniform* marginal bounds, where each element $e \in U$ has a separate value κ_e, and we optimize over distributions over subsets $S \subseteq U$ satisfying $\Pr[e \in S] \leq \kappa_e$. If we are able to find a non-uniform $\kappa \in (0,1)^U$ vector for which $\mathrm{OPT}_{\mathrm{LP}}(\kappa, U) - \mathrm{OPT}_{\mathrm{LP}}(\{\kappa_e^2\}_e, U) \geq \frac{1}{O(1)} v(U)$, this would lead to an $O(1)$-approximation for both subadditive budget-feasible mechanisms and online subadditive combinatorial auctions.

The key role played by bounded-marginal distributions with large expected value, both in the design of budget-feasible mechanisms for subadditive valuations, and in the design of a prophet-inequality for subadditive combinatorial auctions in [DKL20], suggests two pertinent lines of inquiry. First, it brings into further focus the need for understanding the power of this tool, in particular, how one may analyze and exploit distributions with non-uniform marginals, as discussed above. Second, it raises the question of whether there is a deeper connection between the two, seemingly disparate problem domains of budget-feasible mechanism design and prophet inequalities for combinatorial auctions. In particular, is there a connection between the structural properties of subadditive functions given by Theorems 1.2 and 1.3, and the existence of good distributions with non-uniform marginals?

2 Preliminaries

Let U be the set of elements, and let $n := |U|$. Let $v : 2^U \to \mathbb{R}_+^U$ be the valuation function. In budget-feasible mechanism design, the valuation function is always assumed to be normalized, that is, $v(\emptyset) = 0$. A valuation function $v : 2^U \to \mathbb{R}_+$ is said to be *subadditive* if $v(S) + v(T) \geq v(S \cup T)$ holds for all $S, T \subseteq U$. We will also assume that the valuation function is monotone, that is, $v(S) \leq v(T)$ for all $S \subseteq T \subseteq U$. This assumption is without loss of generality as one can obtain a mechanism for non-monotone subadditive valuations if one has a mechanism for monotone subadditive valuations with the same approximation ratio (see [BCGL17]).

For an element $e \in U$ and subset $S \subseteq U$, we use $v(e)$ as a shorthand for $v(\{e\})$, and $S + e$, $S - e$ as shorthands for $S \cup \{e\}$ and $S \setminus \{e\}$ respectively. For a set $S \subseteq U$ and a vector $p \in \mathbb{R}_+^U$, we use $p(S)$ as a shorthand to denote $\sum_{e \in S} p_e$. For $B \in \mathbb{R}_+$ and $S \subseteq U$, let $\Delta^S(B) = \{x \in \mathbb{R}_+^S : x(S) \leq B\}$. We use $\Delta(B)$ as a shorthand for $\Delta^U(B)$. For a cost vector $c \in \mathbb{R}^U$, we denote by c_{-e} the vector in $\mathbb{R}^{U-e}$ obtained by dropping the coordinate corresponding to e. Moreover, for $c'_e \in \mathbb{R}$, we denote by (c'_e, c_{-e}) the vector x in $\mathbb{R}^U$ where $x_{e'} = c_{e'}$ for all $e' \in U - e$, and $x_e = c'_e$.

Throughout, $\mathrm{OPT} := \max\{v(S) : c(S) \leq B\}$ denotes the optimal value. We may assume, using standard preprocessing, that every element $e \in U$ satisfies $c_e \leq B$, as discarding elements that do not satisfy this affects neither the approximation factor nor truthfulness. Let e^* denote $\mathrm{argmax}_{e \in U}\, v(e)$. Note that the preprocessing guarantees that $\mathrm{OPT} \geq v(e^*)$.

Our mechanism will use a random-partitioning step to compute a good estimate for OPT. Let U_1, U_2 be a random partition of U obtained by placing each element of U independently with probability $\frac{1}{2}$ in U_1 or U_2. Let $V_1^* = \max\{v(S) : c(S) \leq B, S \subseteq U_1\}$ be the optimum over U_1, and let $V_2^* = \max\{v(S) : c(S) \leq B, S \subseteq U_2\}$ be the optimum over U_2. The subadditivity of v can be used to show that both V_1^*, V_2^* are large with high probability.

Lemma 2.1 ([NPS24]). *Let $v : 2^U \to \mathbb{R}_+$ be a subadditive valuation function. Then, we have* $\Pr[V_2^* \geq \frac{\mathrm{OPT}}{2}, V_2^* \geq V_1^* \geq \frac{\mathrm{OPT} - v(e^*)}{4}] \geq \frac{1}{4}$.

Oracles for accessing the valuation. Explicitly representing a subadditive valuation function would require space exponential in n, so in order to meaningfully talk about computational efficiency, we assume that the valuation function is specified via a suitable oracle. The most natural oracle one could consider is a *value oracle* which takes as input a subset $S \subseteq U$ and returns its value $v(S)$. However, [Sin10] showed that one needs exponentially many value oracle queries to achieve approximation ratio better than $\sqrt{n}$ for subadditive valuations; in fact, this lower bound holds even for the subclass of XOS valuations. Therefore, in keeping with prior work on budget-feasible mechanism design (when considering valuations more general than submodular functions), we assume access to a *demand oracle*, which takes as input prices p_e for each $e \in U$ and returns a subset $T \in \operatorname{argmax}_{S \subseteq U}\{v(S) - p(S)\}$. Demand oracles are often used in the field of mechanism design, and they have a simple economic interpretation: the oracle returns the subset of items in U with maximum utility, when the cost of purchasing an item e is p_e. A demand oracle can also be used to compute an approximation to OPT.

Lemma 2.2 ([BDO18]). *Let $v : 2^U \to \mathbb{R}_+$ be subadditive. For any $\epsilon > 0$, one can compute a $(2 + \epsilon)$-approximate solution to* OPT *in polynomial time using demand oracles.*

It is important that the demand oracle uses a consistent tie-breaking rule in case there are multiple sets that attain the maximum $\max_{S \subseteq U}\big(v(S) - p(S)\big)$. This can always be achieved, since fixing some ordering of the elements of U, one can always perturb the prices p to obtain a set that is lexicographically smallest (according to the ordering) among all sets that attain the maximum (see [NPS24]).

Mechanism design. A mechanism consists of an allocation rule f along with a payment scheme p. In the setting of budget-feasible mechanism design, we are given as input the publicly-known information of the valuation function $v : 2^U \to \mathbb{R}_+$, the budget $B \in \mathbb{R}_+$, and the seller's reported costs $\{c_e\}_{e \in U}$. The allocation rule, or algorithm, f outputs a subset $S \subseteq U$ of sellers whose items are purchased by the mechanism designer. We refer to the set S returned by f as the set of *winners*. The payment scheme p then assigns to each seller a payment p_e. The utility of seller e, when her true private cost is $\overline{c}_e \geq 0$ is $u_e = p_e - \overline{c}_e$ if $e \in S$ and $u_e = p_e$ if $e \notin S$. Each seller reports a cost that maximizes her own utility. Note that $f, \{p_e\}_{e \in U}$ are functions of v, B and c. For ease of notation, we treat v and B as fixed and treat $f, \{p_e\}_{e \in U}$ as functions of c only. Additionally, u_e is a function of v, B, the reported costs c and the true cost $\overline{c}_e$ of seller e. For ease of notation, we tread v, B as fixed and refer to $u_e(\overline{c}_e; c_e, c_{-e})$ as the utility of seller e when the reported costs are c and the true cost of seller e is $\overline{c}_e$. We seek a mechanism $\mathcal{M} = (f, p)$ satisfying the following properties.

- **Truthfulness.** Each seller e maximizes her utility by reporting her true private cost. That is, for every $\overline{c}_e, c_e, c_{-e}$, we have $u_e(\overline{c}_e; \overline{c}_e, c_{-e}) \geq u_e(\overline{c}_e; c_e, c_{-e})$.
- **Individual Rationality.** $u_e(\overline{c}_e; \overline{c}_e, c_{-e}) \geq 0$ for every e and every $\overline{c}_e, c_{-e}$. Note that this implies that $p_e(c) \geq 0$ for all c.
- **No positive transfers.** We do not pay sellers whose items we do not purchase, that is, $p_e(c) = 0$ whenever $e \notin f(c)$.

- **Budget Feasibility.** We have $\sum_{e \in U} p_e(c) \le B$ for all c. Assuming no positive transfers, this is equivalent to $\sum_{e \in f(c)} p_e(c) \le B$. Note that if $\mathcal{M}$ is individually rational, this implies $c(f(c)) \le B$.

A *budget feasible mechanism* is a mechanism that satisfies the above properties. Our mechanism is randomized, and will satisfy the above properties with probability 1, that is, under all realizations of its random bits. A randomized mechanism that is truthful with probability 1 is also called a *universally-truthful* mechanism.

In addition, we want the mechanism to return a solution that has value close to $\mathrm{OPT} := \max\{v(S) : c(S) \le B, S \subseteq U\}$. We say that $\mathcal{M}$ achieves an α-approximation if f is a α-approximate algorithm, that is, $v(f(c)) \ge \frac{\mathrm{OPT}}{\alpha}$, where $\alpha \ge 1$. When $\mathcal{M}$ is randomized, we say that $\mathcal{M}$ achieves an α-approximation if the expected value obtained is at least OPT/α.

Budget-feasible mechanism design is a single-parameter setting. In single-parameter settings, Myerson's Lemma gives a powerful characterization truthfulness. We say that a deterministic algorithm f is *monotone* if for every seller $e \in U$, and every $c_e \ge c'_e \in \mathbb{R}_+$, and every $c_{-e} \in \mathbb{R}_+^{U-e}$, if $e \in f(c_e, c_{-e})$ then $e \in f(c'_e, c_{-e})$. That is, f is monotone if a winner remains a winner upon decreasing its reported cost.

Theorem 2.3 (Truthfulness in single-parameter domains [Mye81])**.** *Given an algorithm f for budget-feasible mechanism design (i.e., for approximately computing* OPT*), there exists payment functions $\{p_e\}_{e \in U}$ such that (f, p) is a truthful mechanism if and only if f is monotone.*

Moreover, suppose that f is monotone, and $\tau_e = \tau_e(c_{-e}) := \sup\{c_e \ge 0 : e \in f(c_e, c_{-e})\}$ is finite for every $e \in U$ and $c_{-e} \in \mathbb{R}_+^{U-e}$. Then setting $p_e(c) = \tau_e(c_{-e})$ if e is a winner and 0 otherwise, is a payment scheme that yields a truthful, individually-rational mechanism with no positive transfers. We refer to τ_e as the threshold price *or* threshold payment *for seller e.*

Optimizing over distributions with bounded marginals. Any feasible solution x to $\big(\mathrm{LP}\,(\kappa, S)\big)$ can be viewed as a distribution over subsets T of S. We abuse notation and denote by $T \sim x$ as a random set T which takes value T' with probability $x_{T'}$, and $\emptyset$ with probability $1 - \sum_{T' \subseteq S} x_{T'}$ (which is non-negative due to the second constraint). We use the notation $\mathrm{OPT}_{\mathrm{LP}}(\kappa, S)$ to denote the optimal value of $\big(\mathrm{LP}\,(\kappa, S)\big)$. The dual of $\big(\mathrm{LP}\,(\kappa, S)\big)$ is

$$
\begin{array}{lll}
\text{minimize} & \kappa p(S) + \mu & \\
\text{subject to} & p(T) + \mu \ge v(T) & \forall T \subseteq S \\
 & p, \mu \ge 0. &
\end{array}
\qquad \big(\mathrm{DLP}(\kappa, S)\big)
$$

$\big(\mathrm{LP}\,(\kappa, S)\big)$ and its dual can be solved using a demand oracle. This can be done by the Ellipsoid method, where a demand oracle can be used to construct a separation oracle for the dual of $\big(\mathrm{LP}\,(\kappa, S)\big)$. This is done in the following manner: Given a vector (p, μ), in order to check if its feasible, we need to check whether $v(T) - p(T) \le \mu$ holds for all $T \subseteq S$. This can be done by calling a demand oracle on prices p, and checking if $\max\{v(T) - p(T) : T \subseteq U\} \le \mu$. If it does, then (p, μ) is feasible, otherwise the maximum demand set T^* satisfies $v(T^*) - p(T^*) > \mu$ which serves as the violated constraint. Given this separation oracle for the dual, the Ellipsoid method can then be used to find an optimal solution for both the LP and its dual.

Just as in the case for demand oracles, we will assume that our LP is solved with a consistent tie-breaking rule that selects a canonical optimal solution in the case that there are multiple optimal solutions. That is, the solver returns a fixed optimal primal solution x and optimal dual solution (p, μ) once S and κ are fixed.

3 Finding a good distribution over threshold vectors

The main result of this section is to prove Theorem 1.3, which is restated more precisely below.

Theorem 3.1. *Let $B \in \mathbb{R}_+$ and $S^* \subseteq U$. Let $n := |U| \geq 8$. Using demand oracles, one can compute in polynomial time a distribution $\mathcal{D}[S^*]$ over vectors $d \in \Delta^{S^*}(B)$ such that*

$$\mathbb{E}_{d \sim \mathcal{D}[S^*]}[v(\{e \in S^* : c_e \leq d_e\})] \geq \frac{v(S^*)}{16 \log \log n} \qquad \text{for all vectors } c \in \Delta^{S^*}(B/16 \log \log n).$$

We use the notation $\mathcal{D}[S^*]$ to emphasize that the distribution depends on the set $S^* \subseteq U$. We begin by proving the following lemma, which is a formal statement of Theorem 1.4. The lemma shows that the dual variables of $\big(\mathrm{LP}\,(\kappa, S)\big)$ can be used to find a pure strategy d for the d-player that achieves a payoff of at least $v(S) - \mathrm{OPT}_{\mathrm{LP}}(\kappa, S)$ against all vectors $c \in \mathbb{R}_+^S$ satisfying $c(S) \leq \kappa B$. Fix $S \subseteq U$ and $\kappa \in (0,1)$. Fix an optimal solution (p, μ) to $\big(\mathrm{DLP}(\kappa, S)\big)$. We define $d^{S,\kappa}$ to be the vector in $\mathbb{R}_+^U$ given by $d_e^{S,\kappa} = p_e \cdot \frac{B}{p(S)}$ for $e \in S$ and $d_e^{S,\kappa} = 0$ for $e \in U \setminus S$,

Lemma 3.2. *Let $B \in \mathbb{R}_+$, $\kappa \in (0,1)$, and $S \subseteq U$. Then the vector $d := d^{S,\kappa}$ satisfies the following.*

- *$d(S) = B$.*
- *For all $c \in \mathbb{R}_+^S$ with $c(S) \leq \kappa B$, we have $v(\{e \in S : c_e \leq d_e\}) \geq v(S) - \mathrm{OPT}_{\mathrm{LP}}(\kappa, S)$.*

Proof. Let (p, μ) be the optimal solution to $\big(\mathrm{DLP}(\kappa, S)\big)$ used to obtain d. The first property holds by construction: $d(S) = \sum_{e \in S} p_e \cdot \frac{B}{p(S)} = B$. Now fix $c \in \mathbb{R}_+^S$ such that $c(S) \leq \kappa B$. Let $T = \{e \in S : c_e > d_e\}$. Note that $v(\{e \in S : c_e \leq d_e\}) = v(S \setminus T)$. We will show that $v(S \setminus T) \geq v(S) - \mathrm{OPT}_{\mathrm{LP}}(\kappa, S)$. We have that

$$v(T) \leq p(T) + \mu = d(T) \cdot \frac{p(S)}{B} + \mu \leq c(T) \cdot \frac{p(S)}{B} + \mu \leq \kappa p(S) + \mu = \mathrm{OPT}_{\mathrm{LP}}(\kappa, S)$$

where the first inequality is because $d_e < c_e$ for all $e \in T$, and the second inequality holds since $c(T) \leq \kappa B$. Thus, $v(S \setminus T) \geq v(S) - v(T) \geq v(S) - \mathrm{OPT}_{\mathrm{LP}}(\kappa, S)$. ☐

Note that, given S and κ, one can compute the vector $d^{S,\kappa}$ in polynomial time using demand oracles since the optimal solution to the dual of $\big(\mathrm{LP}\,(\kappa, S)\big)$ can be computed in polynomial time using demand oracles.

We will construct a strategy for the d-player that provably obtains an expected payoff of $\mathrm{OPT}_{\mathrm{LP}}(\kappa, S^*) - \mathrm{OPT}_{\mathrm{LP}}(\kappa^2, S^*)$, for some $\kappa \in (0,1)$. It turns out that there always exists $\kappa \in (0,1)$, computable in polynomial time, such that $\mathrm{OPT}_{\mathrm{LP}}(\kappa, S^*) - \mathrm{OPT}_{\mathrm{LP}}(\kappa^2, S^*) \geq \frac{1}{8 \log \log n} v(S^*)$. This was shown by [DKL20]; to keep exposition self-contained, we include a proof at the end of this section. This is the value of κ that we will use when solving the LP.

Lemma 3.3 ([DKL20])**.** *Let $S^* \subseteq U$. Suppose $n := |U| \geq 8$. There exists $\kappa \in (0,1)$ such that $\mathrm{OPT}_{\mathrm{LP}}(\kappa, S^*) - \mathrm{OPT}_{\mathrm{LP}}(\kappa^2, S^*) \geq \frac{1}{8 \log \log n} v(S^*)$. Moreover, this value of κ can be found in polynomial time using demand oracles.*

Now we are ready to prove Theorem 3.1.

Proof of Theorem 3.1. Let $\kappa \in (0,1)$ be such that $\mathrm{OPT}_{\mathrm{LP}}(\kappa, S^*) - \mathrm{OPT}_{\mathrm{LP}}(\kappa^2, S^*) \geq \frac{v(S^*)}{8 \log \log n}$ as guaranteed by Theorem 3.3. The distribution $\mathcal{D}[S^*]$ is obtained by sampling a set $S \sim x$ where x is the optimal solution for $\big(\mathrm{LP}\,(\kappa, S^*)\big)$ and then taking the vector $d^{S,\kappa}$. By Theorem 3.2, we have that $\mathcal{D}[S^*]$ is a distribution of vectors in $\Delta^{S^*}(B)$. Additionally, note that x can be computed in polynomial time, and also has polynomial sized support, since its the output of the Ellipsoid algorithm. Moreover, given S and κ, the vector $d^{S,\kappa}$ can also be computed in polynomial using demand oracles. It follows that a complete description of $\mathcal{D}[S^*]$ (that is, a list of vectors in the support of $\mathcal{D}[S^*]$ and their probabilities) can be computed in polynomial time.

Now we will show that sampling a vector from $\mathcal{D}[S^*]$ provides a good payoff for the d-player. We will first show that $\mathbb{E}_S[v(S) - \mathrm{OPT}_{\mathrm{LP}}(\kappa, S)] \geq \frac{v(S^*)}{O(\log\log n)}$. Note that $\mathbb{E}_S[v(S)] = \mathrm{OPT}_{\mathrm{LP}}(\kappa, S^*)$. We claim that $\mathbb{E}_S[\mathrm{OPT}_{\mathrm{LP}}(\kappa, S)] \leq \mathrm{OPT}_{\mathrm{LP}}(\kappa^2, S^*)$. Consider the distribution over subsets T of S^*, where we first sample a set S from $(x_S)_{S\subseteq S^*}$ where x is the optimal solution to $\big(\mathrm{LP}\,(\kappa, S^*)\big)$ and then sample a set T from $(y_T)_{T\subseteq S}$ where y is the optimal solution to $\big(\mathrm{LP}\,(\kappa, S)\big)$. For any element $e \in S^*$, we have that $\Pr[e \in S] \leq \kappa$ and $\Pr[e \in T \mid e \in S] \leq \kappa$. Thus $\Pr[e \in T] \leq \kappa^2$. Hence the distribution over T has marginals at most κ^2, implying that $\mathbb{E}_T[v(T)] \leq \mathrm{OPT}_{\mathrm{LP}}(\kappa^2, S^*)$. Finally, observe that $\mathbb{E}_T[v(T)] = \mathbb{E}_S[\mathrm{OPT}_{\mathrm{LP}}(\kappa, S)]$ by construction. Thus, we have that $\mathbb{E}_S[\mathrm{OPT}_{\mathrm{LP}}(\kappa, S)] \leq \mathrm{OPT}_{\mathrm{LP}}(\kappa^2, S^*)$. This shows that $\mathbb{E}_S[v(S) - \mathrm{OPT}_{\mathrm{LP}}(\kappa, S)] \geq \mathrm{OPT}_{\mathrm{LP}}(\kappa, S^*) - \mathrm{OPT}_{\mathrm{LP}}(\kappa^2, S^*) \geq \frac{v(S^*)}{8\log\log n}$, by our choice of κ.

Now, we are almost done. We would like to invoke Theorem 3.2 to show that $\mathbb{E}[v(\{e \in S^* : c_e \leq d_e\})] \geq \mathbb{E}[v(S) - \mathrm{OPT}_{\mathrm{LP}}(\kappa, S)] \geq \frac{v(S^*)}{8\log\log n}$. However, this only works when $c(S) \leq \kappa B$. We need to show that the cost vector satisfies this with high probability.

Fix $c \in \mathbb{R}_+^{S^*}$ satisfying $c(S^*) \leq \frac{B}{16\log\log n}$. Recall that S is sampled from the distribution $(x_S)_{S\subseteq S^*}$. Since this distribution has marginals at most κ, we have that $\mathbb{E}[c(S)] \leq \kappa c(S^*) = \frac{\kappa B}{16\log\log n}$. By Markov's inequality, with probability at least $1 - \frac{1}{16\log\log n}$, we have that $c(S) \leq (16\log\log n)\mathbb{E}[c(S)] \leq \kappa B$. Let Ω be the event that $c(S) \leq \kappa B$, and let $\overline{\Omega}$ be the event that $c(S) > \kappa B$. We have,

$$\begin{aligned}
\frac{1}{8\log\log n} v(S^*) &\leq \mathbb{E}[v(S) - \mathrm{OPT}_{\mathrm{LP}}(\kappa, S)] \\
&= \mathbb{E}[v(S) - \mathrm{OPT}_{\mathrm{LP}}(\kappa, S) \mid \Omega]\Pr[\Omega] + \mathbb{E}[v(S) - \mathrm{OPT}_{\mathrm{LP}}(\kappa, S) \mid \overline{\Omega}]\Pr[\overline{\Omega}] \\
&\leq \mathbb{E}[v(S) - \mathrm{OPT}_{\mathrm{LP}}(\kappa, S) \mid \Omega]\Pr[\Omega] + \frac{1}{16\log\log n} v(S^*)
\end{aligned}$$

where we used that fact that $\Pr[\overline{\Omega}] \leq \frac{1}{16\log\log n}$ and $\mathbb{E}[v(S) - \mathrm{OPT}_{\mathrm{LP}}(\kappa, S) \mid \overline{\Omega}] \leq v(S^*)$. This implies that

$$\mathbb{E}[v(S) - \mathrm{OPT}_{\mathrm{LP}}(\kappa, S) \mid \Omega]\Pr[\Omega] \geq \frac{1}{16\log\log n} v(S^*).$$

Let $Q = \{e \in S : c_e \leq d_e\}$. Under the event Ω, by Theorem 3.2, we have that $v(Q) \geq v(S) - \mathrm{OPT}_{\mathrm{LP}}(\kappa, S)$. Thus,

$$\mathbb{E}_S[v(Q) \mid \Omega]\Pr[\Omega] \geq \mathbb{E}[v(S) - \mathrm{OPT}_{\mathrm{LP}}(\kappa, S) \mid \Omega]\Pr[\Omega] \geq \frac{v(S^*)}{16\log\log n}.$$

Hence, the payoff of the d-player, which is $v(Q)$, has expected value at least $\frac{v(S^*)}{16\log\log n}$. This concludes the proof. ☐

Proof of Lemma 3.3. Let $Z = \{2^{-2^i} : 1 \leq i \leq \log\log n\}$. We will find $\kappa \in Z$ satisfying $\mathrm{OPT}_{\mathrm{LP}}(\kappa, S^*) - \mathrm{OPT}_{\mathrm{LP}}(\kappa^2, S^*) \geq \frac{1}{4\log\log n} v(S^*)$. By a telescoping argument, we have,

$$\sum_{z\in Z}\Big(\mathrm{OPT}_{\mathrm{LP}}(z, S^*) - \mathrm{OPT}_{\mathrm{LP}}(z^2, S^*)\Big) = \mathrm{OPT}_{\mathrm{LP}}\Big(\tfrac{1}{4}, S^*\Big) - \mathrm{OPT}_{\mathrm{LP}}\Big(\tfrac{1}{n^2}, S^*\Big)$$

Setting $x_{S^*} = \frac{1}{4}$ and $x_S = 0$ for all other sets $S \subset S^*$ gives rise to a feasible solution for $\left(\text{LP}\left(\frac{1}{4}, S^*\right)\right)$ with objective $\frac{1}{4}v(S^*)$. Hence $\text{OPT}_{\text{LP}}\left(\frac{1}{4}, S^*\right) \geq \frac{1}{4}v(S^*)$. Setting $p_e = v(S^*)$ for all $e \in S^*$ and $\mu = 0$ gives a feasible solution to $\left(\text{DLP}(\frac{1}{n^2}, S^*)\right)$ with objective value $\frac{1}{n^2}p(S^*) + \mu = \frac{1}{n}v(S^*)$. This shows that $\text{OPT}_{\text{LP}}\left(\frac{1}{n^2}, S^*\right) \leq \frac{1}{n}v(S^*)$. Hence we have

$$\sum_{z \in Z}\Big(\text{OPT}_{\text{LP}}(z, S^*) - \text{OPT}_{\text{LP}}(z^2, S^*)\Big) \geq \left(\frac{1}{4} - \frac{1}{n}\right)v(S^*) \geq \frac{1}{8}v(S^*)$$

for $n \geq 8$. Since $|Z| = \log\log n$, there exists some $\kappa \in Z$ for which $\text{OPT}_{\text{LP}}(\kappa, S^*) - \text{OPT}_{\text{LP}}(\kappa^2, S^*) \geq \frac{1}{8\log\log n}v(S^*)$.

To find κ in polynomial time using demand oracles, we may simply compute $\text{OPT}_{\text{LP}}(z, S^*) - \text{OPT}_{\text{LP}}(z^2, S^*)$ for each value of $z \in Z$ and take the maximum. This can be done in polynomial time as the LP can be solved in polynomial time using demand oracles. $\square$

4 The budget feasible mechanism

We now use the result in Section 3 to design a randomized budget feasible mechanism for subadditive valuations that achieves an approximation factor of $O(\log\log n)$, thereby proving Theorem 1.1. As mentioned in Section 1.2, we first randomly partition the set of sellers into two groups U_1, U_2, and get an estimate of OPT from the sellers in U_1 and discard them. We then work over the sellers in U_2. We compute a demand set over U_2, with appropriately chosen prices. This results in a set $S^* \subseteq U_2$ that has good value compared to OPT. When $c(S^*) \leq \frac{B}{O(\log\log n)}$, we can utilize Theorem 3.1 to prove a good approximation ratio by sampling a vector $d \sim \mathcal{D}[S^*]$ and return $R = \{e \in S^* : c_e \leq d_e\}$. However, we also need to deal with the case when $c(S^*) > \frac{B}{O(\log\log n)}$. In this scenario, we will show that one can construct a single vector d' that provides suitable thresholds. We show that a suitable choice for d' is to set $d'_e = q_e \cdot \frac{O(\log\log n)B}{\text{OPT}}$, where $q \in \mathbb{R}_+^{S^*}$ is an optimal solution to the LP: max $q(S^*)$ subject to $q(S) \leq v(S^*) - v(S^* \setminus S)$ for all $S \subseteq S^*$. We select the set of elements A for which $c_e \leq d'_e$, and *prune* this set to construct a subset $R' \subseteq A$ satisfying $\Omega(\text{OPT}) - v(e^*) \leq v(R') \leq O(\text{OPT})$. We show that constructing R' this way admits budget-feasible threshold payments. Returning either R' or e^* then gives a good approximation, since if $v(e^*)$ is small then $v(R') \geq \Omega(\text{OPT}) - v(e^*)$ would be large. One caveat here is that we cannot really branch on whether $c(S^*) \leq \frac{B}{O(\log\log n)}$ as this causes problems with truthfulness. So instead, we return one of R, R', or e^*, each chosen with a suitable probability. The algorithm is described in detail below. We show that Algorithm 1 is a distribution over monotone algorithms, and so the corresponding payments can be obtained using Theorem 2.3.

For step 8, it is important that the vector q is computed with a consistent tie-breaking rule. That is, q is fixed once S^* and v are fixed.

We assume that $n \geq 8$, as otherwise, the guarantees follow from [NPS24]. Theorem 1.1 follows from Lemmas 4.1, which shows polynomial running time, Lemma 4.3, which proves the approximation guarantee, and Lemma 4.4, which proves truthfulness and budget feasibility.

Lemma 4.1. *Algorithm 1 can be implemented in polynomial time using at most polynomial many demand oracle queries.*

Proof. Step 3 and step 5 can be implemented using demand oracles by Theorem 2.2 and Theorem 3.1 respectively. We show how to implement step 8. To solve (P) in step 8, it suffices to give a separation oracle for the constraints of the LP. One can then use the Ellipsoid method to solve it. Let $q \in \mathbb{R}_+^{S^*}$.

ALGORITHM 1: Budget feasible mechanism for subadditive valuations

Input : A valuation function $v : 2^U \to \mathbb{R}_+$, budget $B \in \mathbb{R}_+$, and reported costs $\{c_e\}_{e\in U}$
Output : A subset $S \subseteq U$

If $n := |U| < 8$, run the exponential-time mechanism from [NPS24] for subadditive valuations. Otherwise, proceed as below.
Construct a partition U_1, U_2 of U, where each item $e \in U$ is put into U_1 or U_2 with probability $\frac{1}{2}$.
Let V_1 be a $(2+\epsilon)$-approximation of V_1^*, the optimum value over U_2, obtained using Theorem 2.2. Let $\alpha = \frac{1}{4(2+\epsilon)+1}$ and $\beta = 1 - \alpha$.
Let $S^* \leftarrow \operatorname{argmax}_{S\subseteq U_2} \left\{ v(S) - \frac{V_1}{2B} \cdot c(S) \right\}$.
Use Theorem 3.1 to get a distribution $\mathcal{D}[S^*]$ over vectors $d \in \Delta^{S^*}(B)$ such that $\mathbb{E}_{d\sim\mathcal{D}[S^*]}[v(\{e \in S^* : c_e \le d_e\})] \ge \frac{v(S^*)}{16 \log\log n}$ for all $c \in \Delta^{S^*}\left(\frac{B}{16\log\log n}\right)$.
Sample $d \sim \mathcal{D}[S^*]$.
Let $R \leftarrow \{e \in S^* : c_e \le d_e\}$.
Compute an optimal solution $q \in \mathbb{R}^{S^*}$ to the following LP:

$$\max \quad q(S^*) \qquad \text{s.t.} \qquad q(S) \le v(S^*) - v(S^* \setminus S) \quad \forall S \subseteq S^*, \qquad q \ge 0. \tag{P}$$

Let $A \leftarrow \{e \in S^* : c_e \le q_e \cdot \frac{4B}{V_1}\}$.
Let R' be a maximal prefix of A satisfying $q(R') \le \frac{1}{4}V_1$.
return R with probability 0.8α, R' with probability 0.8β, and e^* with probability 0.2.

Observe that

$$\begin{aligned} q(S) \le v(S^*) - v(S^* \setminus S) \quad \forall S \subseteq S^* &\iff q(S^*) - q(S^* \setminus S) \le v(S^*) - v(S^* \setminus S) \quad \forall S \subseteq S^* \\ &\iff v(T) - q(T) \le v(S^*) - q(S^*) \quad \forall T \subseteq S^*. \end{aligned}$$

The final condition above can be easily checked using a demand-oracle query with prices q.[4] If the oracle returns some $T \subseteq S^*$ with $v(T) - q(T) > v(S^*) - q(S^*)$, then the LP constraint for $S^* \setminus T$ is violated, and otherwise q is feasible. □

Before analyzing the approximation ratio, we prove a lemma regarding the cost of the demand set.

Lemma 4.2. *For every $S \subseteq S^*$, we have that $\frac{V_1}{2B}c(S) \le v(S^*) - v(S^* \setminus S) \le v(S)$.*

Proof. Let $S \subseteq S^*$. By construction of S^*, we have that $v(S^*) - \frac{V_1}{2B}c(S^*) \ge v(S^* \setminus S) - \frac{V_1}{2B}c(S^* \setminus S)$. Rearranging, we get that $\frac{V_1}{2B}c(S) = \frac{V_1}{2B}c(S^*) - \frac{V_1}{2B}c(S^* \setminus S) \le v(S^*) - v(S^* \setminus S)$, showing the first inequality. The second inequality follows by subadditivity of v. □

Now we are ready to prove the approximation ratio.

Lemma 4.3. *Algorithm 1 returns a solution T such that $\mathbb{E}[v(T)] \ge \frac{1}{O(\log\log n)} \cdot \mathrm{OPT}$.*

Proof. Recall that $V_i^* = \max\{v(S) : c(S) \le B, S \subseteq U_i\}$ for each $i \in \{1,2\}$, and $v(e^*) = \max_{e\in U} v(e)$. Let Γ be the event that $V_2^* \ge V_1^* \ge \frac{\mathrm{OPT} - v(e^*)}{4}$ and $V_2^* \ge \frac{\mathrm{OPT}}{2}$. By Theorem 2.1,

[4] More precisely, we set the price of an element $e \in S^*$ to be q_e, and set the prices of elements outside of S^* to be ∞, or some very large value. Then the demand-oracle query effectively amounts to a demand-oracle query over the ground set S^*.

Γ holds with probability $\geq \frac{1}{4}$. Assume that the event Γ happens. So we have $V_2^* \geq \frac{\mathrm{OPT}}{2}$ and $V_1 \geq \frac{V_1^*}{2+\epsilon} \geq \frac{\mathrm{OPT}-v(e^*)}{4(2+\epsilon)} = \frac{\alpha}{\beta} \cdot \big(\mathrm{OPT}-v(e^*)\big)$.

First, let us show that $v(S^*) \geq \frac{\mathrm{OPT}}{4}$. To see this, let $O_2^* = \operatorname{argmax}\{v(S) : c(S) \leq B, S \subseteq U_2\}$, so $v(O_2^*) = V_2^*$. So we have

$$v(S^*) \geq v(S^*) - \frac{V_1}{2B} \cdot c(S^*) \geq v(O_2^*) - \frac{V_1}{2B} \cdot c(O_2^*) \geq V_2^* - \frac{V_1}{2} \geq \frac{V_2^*}{2} \geq \frac{\mathrm{OPT}}{4}.$$

If $c(S^*) \leq \frac{B}{16 \log\log n}$, then by Theorem 3.1, it follows that $\mathbb{E}_{d\sim\mathcal{D}[S^*]}[v(R)] \geq \frac{v(S^*)}{16\log\log n} \geq \frac{\mathrm{OPT}}{64\log\log n}$. So if $c(S^*) \leq \frac{B}{16\log\log n}$, then the expected value obtained is at least

$$\Pr[\Gamma] \cdot 0.8\alpha \cdot \frac{\mathrm{OPT}}{64\log\log n} \geq \frac{\mathrm{OPT}}{(2880+1280\epsilon)\log\log n}.$$

Now suppose $c(S^*) > \frac{B}{16\log\log n}$. In this case, we argue that R' has good value. By Theorem 4.2, the vector $\frac{V_1}{2B}c$ is feasible for (P). Hence the optimal solution q to (P) satisfies $q(S^*) \geq \frac{V_1}{2B} \cdot c(S^*)$. We have

$$q(A) = q(S^*) - q(S^* \setminus A) \geq \frac{V_1}{2B} \cdot c(S^*) - \frac{V_1}{4B} \cdot c(S^* \setminus A) \geq \frac{V_1}{4B} \cdot c(S^*) > \frac{V_1}{64\log\log n}$$

Thus, $v(A) \geq v(S^*) - v(S^* \setminus A) \geq q(A) \geq \frac{V_1}{64\log\log n}$. This implies that

$$\begin{aligned} v(R') &\geq \min\Big\{\tfrac{V_1}{4} - v(e^*),\ \tfrac{V_1}{64\log\log n}\Big\} \geq \min\Big\{\tfrac{\alpha}{\beta} \cdot \big(\mathrm{OPT}-v(e^*)\big) - v(e^*),\ \tfrac{\alpha}{\beta} \cdot \tfrac{\mathrm{OPT}-v(e^*)}{64\log\log n}\Big\} \\ &= \frac{\alpha}{\beta} \cdot \min\Big\{\mathrm{OPT} - \tfrac{v(e^*)}{\alpha},\ \tfrac{\mathrm{OPT}-v(e^*)}{64\log\log n}\Big\} \end{aligned}$$

by the construction of R'. So if $c(S^*) > \frac{B}{16\log\log n}$, the expected value returned is at least

$$\begin{aligned} 0.2 \cdot v(e^*) + \frac{1}{4} \cdot 0.8\beta \cdot \tfrac{\alpha}{\beta} \cdot \min\Big\{\mathrm{OPT} - \tfrac{v(e^*)}{\alpha},\ \tfrac{\mathrm{OPT}-v(e^*)}{64\log\log n}\Big\} &\geq \min\Big\{0.2\alpha \cdot \mathrm{OPT},\ 0.2\alpha \cdot \tfrac{\mathrm{OPT}}{64\log\log n}\Big\} \\ &= 0.2\alpha \cdot \frac{\mathrm{OPT}}{64\log\log n} = \frac{\mathrm{OPT}}{(2880+1280\epsilon)\log\log n}. \end{aligned}$$

So in all cases, the expected value returned is $\mathrm{OPT}/O(\log\log n)$. □

Finally, we prove truthfulness and budget feasibility.

Lemma 4.4. *Algorithm 1 can be combined with suitable payments to obtain a mechanism that is truthful, individually rational, has no positive transfers, and is budget feasible, with probability* 1.

Proof. We argue that Algorithm 1 is monotone under all realizations of its random bits, which immediately implies by Theorem 2.3 that using threshold prices as payments yields truthfulness, individual rationality, and no positive transfers, with probability 1.

Truthfulness, individual rationality, and no positive transfers. Fix a realization of the random bits, so that the partition U_1, U_2 of U is fixed, and the set returned (either R, R' or e^*) is fixed. Fixing the random bits also fixes the vector d sampled from $\mathcal{D}[S]$ for any given set $S \subseteq U$. We need to show that if $e \in U$ is a winner, and e decreases their cost from c_e to $c'_e < c_e$, then e remains a winner.

First, suppose that the set returned is $e^* = \mathrm{argmax}_{e \in U}\, v(e)$. Note that e^* does not depend on the cost of any element in U, thus e^* decreasing their cost does not affect whether e^* remains a winner or not.

Suppose that the set returned is R. Let $e \in R$ with cost c_e, and suppose that e decreases their reported cost to $c'_e < c_e$. We claim that the set S^* computed in step 4 is the same for both the inputs $c := (c_e, c_{-e})$ and $c' := (c'_e, c_{-e})$. Let $\mathcal{S}(c)$ denote the set of optimal solutions to $\max\{v(S) - \frac{V_1}{2B} \cdot c(S) : S \subseteq U_2\}$ and $\mathcal{S}(c')$ denote the set of optimal solutions to $\max\{v(S) - \frac{V_1}{2B} \cdot c'(S) : S \subseteq U_2\}$ (note that U_2 is the same for both c and c' since the random bits are fixed). Let $S^*(c) \in \mathcal{S}(c)$ and $S^*(c') \in \mathcal{S}(c')$ denote the sets computed in step 4 for the inputs c and c' respectively. We argue that $S^*(c) \in \mathcal{S}(c') \subseteq \mathcal{S}(c)$. Since the demand oracle uses a consistent tie-breaking rule, this implies that it must return the set $S^*(c)$ also under the input c', i.e., $S^*(c) = S^*(c')$.

For a set $H \subseteq U$ and any $\tilde{c} \in \mathbb{R}^U_+$, define $\mathsf{demd}(H, \tilde{c}) = v(H) - \frac{V_1}{2B} \cdot \tilde{c}(H)$. Consider any $H \in \mathcal{S}(c')$. We have $\mathsf{demd}(S^*(c), c) \geq \mathsf{demd}(H, c)$ and $\mathsf{demd}(H, c') \geq \mathsf{demd}(S^*(c), c')$, by definition. Adding these inequalities and simplifying, we obtain that $c_{e'} - c_e \geq c(H) - c'(H)$. This can only happen if $e \in H$, which implies that this inequality is in fact an equality, and hence the two inequalities that were added to yield this must also be equalities. So we have $\mathsf{demd}(S^*(c), c) = \mathsf{demd}(H, c)$ and $\mathsf{demd}(H, c') = \mathsf{demd}(S^*(c), c')$. The former implies that $H \in \mathcal{S}(c)$, and so $\mathcal{S}(c') \subseteq \mathcal{S}(c)$, and the latter implies that $S^*(c) \in \mathcal{S}(c')$. Thus, $S^*(c) = S^*(c')$.

From this it follows that the distributions $\mathcal{D}[S^*(c)]$ and $\mathcal{D}[S^*(c')]$ are the same. Since the random bits used by the mechanism are fixed, this implies that the vector d sampled from $\mathcal{D}[S^*]$ is the same under both c and c'. It follows that R is the same set under both the inputs c and c'. Hence, since $e \in R$ under input c, we also have $e \in R$ under input c'.

Now, suppose that the set returned is R'. Let $e \in R'$ and suppose that e decreases their reported cost to $c'_e < c_e$. As before, let $c = (c_e, c_{-e})$ and $c' = (c'_e, c_{-e})$. We claim that the sets A and R' are the same under both inputs c and c'. As argued above, the set S^* is the same under both inputs c and c'. This implies that the vector q computed in step 8 is also the same under c and c' (since the computation of q is fixed once S^* and v are fixed). This then also implies that A is the same under both inputs: since $e \in R' \subseteq A$ under input c, we have $c'_e < c_e \leq q_e \cdot \frac{4B}{V_1}$; for any other element $f \neq e$ belonging to A under input c, we have $c'_f = c_f \leq q_f \cdot \frac{4B}{V_1}$. Finally, since the computation of R' does not consider the costs, it follows that R' is also the same under both reported costs c and c'. In particular, since $e \in R'$ under input c, we also have $e \in R'$ under input c'.

Hence we have shown that, regardless of the realizations of the random bits, we have a monotone algorithm.

Budget feasibility. Now we show that the threshold payments yield budget feasibility, regardless of the realizations of the random bits used by the algorithm. Again, fix a realization of the random bits. Let τ_e be the threshold price for $e \in U$ as defined in Theorem 2.3. If the mechanism returns e^*, then $\tau_{e^*} = B$ and $\tau_e = 0$ for all $e \neq e^*$, so we obtain budget feasibility.

Otherwise, the mechanism returns either R or R'. Note that we have argued above that if a winner e changes her reported cost and remains a winner, then the set returned, as also the "intermediate" sets S^*, A, also do not change. This property, which is called no-bossiness in [NPS24], makes it quite convenient to reason about threshold payments.

Suppose that the mechanism returns R. Fix $e \in U$ that is a winner under reported costs c. Let $S^* = S^*(c)$ be the set computed in step 4, and d be the vector obtained by sampling from $\mathcal{D}[S^*]$ using the fixed random bits. Since S^* and d do not change if e remains a winner, it is easy to see that, by design, step 7 ensures that $\tau_e \leq d_e$. It follows that the total payment is at most $d(S^*) \leq B$, since $d \in \Delta^{S^*}(B)$.

Now, suppose that the mechanism returns R', and consider a winner $e \in R'$ under input c.

Since the vector q and the sets A, R computed in steps 8–10 do not change if e remains a winner, by design, step 9 ensures that $\tau_e \le q_e \cdot \frac{4B}{V_1}$. So since $q(R') \le \frac{V_1}{4}$, the total payment is at most B.

Note that the above discussion also shows that the threshold for a player e can be computed in polytime. The threshold value τ'_e for e to belong to the set S^* can be obtained using a demand oracle. If the set output is R, then the threshold for e is $\min\{\tau'_e, d_e\}$. If the set output is R', then the threshold for e to lie in A is $\tau^A_e = \min\{\tau'_e, q_e \cdot \frac{4B}{V_1}\}$. Note that fixing A also fixes the set R'. Let $\overline{A}$ be the A-set obtained under input (c'_e, c_{-e}) when e belongs to it, i.e., when $c'_e < \tau^A_e$; let $\overline{R}$ be the corresponding R'-set. Note $\overline{A}$ and $\overline{R}$ depend only on c_{-e}. One can infer that $\tau_e = \tau^A_e$, if $e \in \overline{R}$ and is 0 otherwise. □

References

[ABM16] Georgios Amanatidis, Georgios Birmpas, and Evangelos Markakis. Coverage, matching, and beyond: new results on budgeted mechanism design. In *Proceedings of 12th WINE*, pages 414–428. Springer, 2016. 7

[ABM17] Georgios Amanatidis, Georgios Birmpas, and Evangelos Markakis. On budget-feasible mechanism design for symmetric submodular objectives. In *Proceedings of 13th WINE*, pages 1–15, 2017. 7

[AGN14] Nima Anari, Gagan Goel, and Afshin 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. 7

[AKM+23] Georgios Amanatidis, Sophie Klumper, Evangelos Markakis, Guido Schäfer, and Artem Tsikiridis. Partial allocations in budget-feasible mechanism design: bridging multiple levels of service and divisible agents. In *Proceedings of 19th WINE*, pages 41–58. Springer, 2023. 7

[AKS19] Georgios Amanatidis, Pieter Kleer, and Guido Schäfer. Budget-feasible mechanism design for non-monotone submodular objectives: Offline and online. In *Proceedings of 20th EC*, pages 901–919, 2019. 7

[BAEJ20] Enric Boix-Adserà, Benjamin L. Edelman, and Siddhartha Visveswara Jayanti. The multiplayer colonel blotto game. *Proceedings of 21st EC*, 2020. 7

[BCGL17] Xiaohui Bei, Ning Chen, Nick Gravin, and Pinyan Lu. Worst-case mechanism design via bayesian analysis. *SIAM Journal on Computing*, 46(4):1428–1448, 2017. 2, 3, 6, 7, 8

[BDD+22] Soheil Behnezhad, Sina Dehghani, Mahsa Derakhshan, Mohammad Taghi Hajiaghayi, and Saeed Seddighin. Fast and simple solutions of blotto games. *Oper. Res.*, 71:506–516, 2022. 7

[BDO18] Ashwinkumar Badanidiyuru, Shahar Dobzinski, and Sigal Oren. Optimization with demand oracles. *Algorithmica*, 81:2244 – 2269, 2018. 9

[BGG+22] Eric Balkanski, Pranav Garimidi, Vasilis Gkatzelis, Daniel Schoepflin, and Xizhi Tan. Deterministic budget-feasible clock auctions. In *Proceedings of 33rd SODA*, pages 2940–2963, 2022. 2, 7

[Bor53] Emile Borel. The theory of play and integral equations with skew symmetric kernels. *Econometrica: journal of the Econometric Society*, pages 97–100, 1953. 7

[BR11] Kshipra Bhawalkar and Tim Roughgarden. Welfare guarantees for combinatorial auctions with item bidding. In *Proceedings of 22nd SODA*, 2011. 7

[BW23] Kiril Bangachev and S. Matthew Weinberg. q-partitioning valuations: Exploring the space between subadditive and fractionally subadditive valuations. *ArXiv*, abs/2304.01451, 2023. 7

[CC14] Hau Chan and Jing Chen. Truthful multi-unit procurements with budgets. In *Proceedings of 10th WINE*, pages 89–105. Springer, 2014. 7

[CGL11] Ning Chen, Nick Gravin, and Pinyan Lu. On the approximability of budget feasible mechanisms. In *Proceedings of 22nd SODA*, pages 685–699, 2011. 7

[DKL20] Paul Dütting, Thomas Kesselheim, and Brendan Lucier. An o(log log m) prophet inequality for subadditive combinatorial auctions. *Proceedings of 61st FOCS*, pages 306–317, 2020. 6, 7, 8, 11

[DPS11] Shahar Dobzinski, Christos H. Papadimitriou, and Yaron Singer. Mechanisms for complement-free procurement. In *Proceedings of 12th EC*, pages 273–282, 2011. 7

[FFGL15] Michal Feldman, Hu Fu, Nick Gravin, and Brendan Lucier. Simultaneous auctions without complements are (almost) efficient. *Games Econ. Behav.*, 123:327–341, 2015. 7

[GJLZ20] Nick Gravin, Yaonan Jin, Pinyan Lu, and Chenhao Zhang. Optimal budget-feasible mechanisms for additive valuations. *ACM Trans. Econ. Comput.*, 8(4), 2020. 7

[HHCT23] He Huang, Kai Han, Shuang Cui, and Jing Tang. Randomized pricing with deferred acceptance for revenue maximization with submodular objectives. In *Proceedings of 32nd WWW*, page 3530–3540, 2023. 7

[HIM14] Thibaut Horel, Stratis Ioannidis, and S. Muthukrishnan. Budget feasible mechanisms for experimental design. In *Proceedings of 11th LATIN*, pages 719–730, 2014. 7

[HKMN11] Avinatan Hassidim, Haim Kaplan, Y. Mansour, and Noam Nisan. Non-price equilibria in markets of discrete goods. In *Proceedings of 12th EC*, 2011. 7

[JT18] Pooya Jalaly and Éva Tardos. Simple and efficient budget feasible mechanisms for monotone submodular valuations. In *Proceedings of 14th WINE*, pages 246–263, 2018. 7

[KS22] Sophie Klumper and Guido Schäfer. Budget feasible mechanisms for procurement auctions with divisible agents. In *Proceedings of 15th SAGT*, pages 78–93, 2022. 7

[LMSZ21] Stefano Leonardi, Gianpiero Monaco, Piotr Sankowski, and Qiang Zhang. Budget feasible mechanisms on matroids. *Algorithmica*, 83(5):1222–1237, 2021. 7

[Mye81] Roger B. Myerson. Optimal auction design. *Math. Oper. Res.*, 6(1):58–73, 1981. 10

[NPS24] Rian Neogi, Kanstantsin Pashkovich, and Chaitanya Swamy. Budget-Feasible Mechanism Design: Simpler, Better Mechanisms and General Payment Constraints. In *Proceedings of 15th ITCS*, pages 84:1–84:22, 2024. 2, 3, 6, 7, 8, 9, 13, 14, 16

[NPS25] Rian Neogi, Kanstantsin Pashkovich, and Chaitanya Swamy. Multidimensional budget-feasible mechanism design. *arXiv preprint arXiv:2508.09367*, 2025. 7

[QWC+20] Yu Qiao, Jun Wu, Hao Cheng, Zilan Huang, Qiangqiang He, and Chong-Jun Wang. Truthful mechanism design for multiregion mobile crowdsensing. *Wirel. Commun. Mob. Comput.*, 2020:8834983:1–8834983:15, 2020. 7

[Rob06] Brian Roberson. The colonel blotto game. *Economic Theory*, 29:1–24, 2006. 7

[Sin10] Yaron Singer. Budget feasible mechanisms. In *Proceedings of 51st FOCS*, pages 765–774, 2010. 2, 7, 9

[Sin13] Yaron Singer. Budget feasible mechanism design. *ACM SIGecom Exchanges*, 12(2):24–31, 2013. 7

[VLS18] Dong Quan Vu, Patrick Loiseau, and Alonso Silva. Efficient computation of approximate equilibria in discrete colonel blotto games. In *International Joint Conference on Artificial Intelligence*, 2018. 7

[WZQ+19] Jun Wu, Yuan Zhang, Yu Qiao, Lei Zhang, Chong-Jun Wang, and Junyuan Xie. Multi-unit budget feasible mechanisms for cellular traffic offloading. In *Adaptive Agents and Multi-Agent Systems*, 2019. 7

A Proofs omitted from the main body

Proof of Theorem 1.2. Consider the following LP

$$\begin{aligned} \text{maximize} \quad & t && \text{(P')} \\ \text{subject to} \quad & \sum_{d \in K} \frac{v(\{e \in U : c_e \le d_e\})}{v(U)} \cdot x_d \ge t \qquad \forall c \in K \\ & \sum_{d \in K} x_d = 1 \\ & x \ge 0. \end{aligned}$$

Let $M \in \mathbb{R}^{K \times K}$ be a matrix with rows and columns indexed by elements in K defined as $M_{c,d} = \frac{v(\{e \in U : c_e \le d_e\})}{v(U)}$ for every $c, d \in K$. Let $\vec{1}$ be the all-ones vector and let J be the all-ones matrix of dimension $|K| \times |K|$. Then (P') can be written succinctly as

$$\text{maximize} \quad t \qquad \text{subject to} \qquad Mx \ge t\vec{1}, \quad \vec{1}^\intercal x = 1, \quad x \ge 0.$$

The dual LP is

$$\text{minimize} \quad \rho \qquad \text{subject to} \qquad M^\intercal y \le \rho\vec{1}, \quad \vec{1}^\intercal y = 1, \quad y \ge 0. \qquad \text{(D')}$$

Let (y, ρ) be a feasible solution to (D'). By subadditivity of v, for every $c, d \in K$, we have that $v(\{e \in U : c_e \le d_e\}) + v(\{e \in U : d_e \le c_e\}) \ge v(U)$. Hence, we get that $M_{c,d} + M_{d,c} =$

$\frac{v(\{e \in U : c_e \leq d_e\}) + v(\{e \in U : d_e \leq c_e\})}{v(U)} \geq 1$ for every $c, d \in K$. In other words, $M + M^\intercal \geq J$, that is, every entry of $M + M^\intercal$ is at least the corresponding entry of J. Now,

$$\rho = \rho(y^\intercal \vec{1}) = y^\intercal(\rho\vec{1}) \geq y^\intercal M y = \frac{1}{2} y^\intercal (M + M) y = \frac{1}{2} y^\intercal (M + M^\intercal) y \geq \frac{1}{2} y^\intercal J y = \frac{1}{2},$$

where we used the fact that $\vec{1}^\intercal y = 1$ and the fact that $y^\intercal M y = y^\intercal M^\intercal y$. Hence we conclude that $\rho \geq \frac{1}{2}$ for every feasible solution of (D'). In particular, this implies that the optimal value of (D'), and hence (P'), is at least $\frac{1}{2}$.

Now let (x, t) be an optimal solution for (P'). Consider the distribution $\mathcal{D}^*$ over vectors in K, where the probability of sampling a vector $d \in K$ is x_d. Then, for all $c \in K$, we have $\mathbb{E}_{d \sim \mathcal{D}^*}[v(\{e \in U : c_e \leq d_e\})] = \sum_{d \in K} v(\{e \in U : c_e \leq d_e\}) x_d \geq t \cdot v(U) \geq \frac{1}{2} v(U)$. This proves the lemma. □

Lemma A.1. *For every $d \in \mathbb{R}_+^U$ with $d(U) \leq B$, there exists $c \in \mathbb{R}_+^U$ with $c(U) \leq B$ such that $v(\{e \in U : c_e \leq d_e\}) \leq v(e^*)$.*

Proof. Let $Q = \{e \in U : d_e > 0\}$. Let e' be an arbitrary element in Q. Let $\epsilon = \frac{1}{n} \min_{e \in Q}\{d_e\} > 0$. Construct the vector c by setting $c_e = d_e + \epsilon$ for all $e \in U - e'$, and set $c_{e'} = d_{e'} - (n-1)\epsilon$. Then note that $c(U) = d(U) \leq B$. Moreover, $c_e \geq 0$ for all $e \in U$ by our choice of ϵ. Now, $v(\{e \in U : c_e \leq d_e\}) = v(e') \leq \max_{e \in U}\{v(e)\} = v(e^*)$. □