Multi-unit Procurements with Budgets, and An Optimal Truthful Mechanism for Bounded Knapsack

Hau Chan Jing Chen
Department of Computer Science, Stony Brook University
{hauchan, jingchen}@cs.stonybrook.edu

Abstract

We study procurement games where each seller has multiple units of the item he supplies, and the buyer can purchase any number of units for each seller's item. Each seller has a cost for one unit of his item, which is his private information. The buyer has a budget B and the total payment he makes to the sellers cannot exceed B.

Procurement games have been studied in the framework of budget feasible mechanisms [27]. However, all the studies of budget feasible mechanisms so far have focused on settings where, each seller has only one unit of his item, and the buyer decides from whom to buy instead of how many to buy for each item. This is the first time where budget feasible mechanisms are studied in multi-unit settings.

For a special class of procurement games, namely, the *bounded knapsack* problem, we show that no (randomized) dominant-strategy truthful (DST) budget feasible mechanism can approximate the value of the optimal allocation within better than $\ln n$, where n is the total number of units of all items available. This is very different from single-unit settings, where constant approximation is known for many scenarios, including knapsack.

We then construct a polynomial-time randomized DST budget feasible mechanism that gives a $4(1+\ln n)$ approximation for procurement games with *additive valuations*, which include bounded knapsack as a special case. Our impossibility result implies that our mechanism is optimal up to a constant factor. Moreover, for the bounded knapsack problem, given the well known FPTAS, our results imply that there is provably a gap between the optimization domain and the mechanism design domain.

Finally, for a much broader class of procurement games, those with *sub-additive valuations*, we construct a randomized DST budget feasible mechanism that gives an $O(\frac{\log^2 n}{\log \log n})$ approximation. The mechanism runs in polynomial time given a demand oracle —a standard oracle for dealing with sub-additive valuations.

Keywords: procurement auction, budget feasible mechanism, optimal mechanism, approximation

1 Introduction

In a procurement auction, m sellers compete for providing their items (which are referred to as products or services in particular scenarios) to the buyer. Each seller i has one item and can provide at most n_i units of it, with a fixed cost c_i per unit which is known only to him. The items need not be substitutions of each other and the buyer may purchase any combination of them with any number of units each. For example, a local government may decide to buy 50 displays from Dell, 20 laptops from Lenovo, and 30 printers from HP.[1] The buyer has his own value function for possible ways of combining the items, which is publicly known. He also has a budget B, publicly known

[1]In reality Dell also sells laptops and Lenovo also sells displays. But for the purpose of this paper we consider settings where each seller has one item to supply, but has many units of it.

as well, and the total amount spent in the procurement cannot exceed B. The buyer's goal is to maximize his value from the items bought, subject to the budget constraint. Since the true cost of each seller is not known by anybody else, as mechanism designers we look for dominant strategy truthful (DST) mechanisms which elicit these costs. The mechanisms should produce outcomes (namely, how many units to buy from and how much to pay to each seller) that meet the budget constraint and are approximately optimal for the buyer.

Procurement auctions with budgets have been studied in the framework of budget feasible mechanisms (see, e.g., [27, 14, 12, 7]). Yet, all the studies so far have focused on settings where each seller has only one unit of his item. In such settings there are only two possibilities from a seller's point of view: either his item is taken and he gets paid, or it is not taken.[2] When a seller has multiple units of the same item and may benefit from selling any number of them, there are more possibilities for him to deviate into, and it becomes harder to provide incentives for him to be truthful, as demonstrated by our impossibility result. However, many interesting mechanisms can be constructed for multi-unit settings, and in some cases they are actually optimal, as demonstrated by our positive results. To the best of our knowledge, we are the first to explore budget feasible mechanisms in the more general multi-unit settings.

Multi-unit procurement auctions with budgets can be used to model many interesting problems. For example, the classic *bounded knapsack* problem can be considered as a special case of such an auction, where the buyer has a value v_i for one unit of item i, and his total value is the sum of his value for each unit he has bought. Here each item's cost is its private information and has to be elicited by a DST mechanism. Even for this very special case of multi-unit procurement auctions, we are not aware of any study in the mechanism design literature.

Also, in a job scheduling problem the central planner may want to assign multiple jobs to a machine, and his value not only depends on which machines are used but also on how many jobs each machine gets. Here a machine has a private cost for one unit of the workload, and the number of units available at a machine may be decided by the machine's own constraints or may be taken as the total number of jobs to be scheduled.

As another example, in the Provision-after-Wait problem in healthcare as introduced by [8], the government needs to serve n patients at m hospitals in each time unit. Each patient has his own value for being served at a particular hospital, and the value of the government is the social welfare generated from the patients. Again here a hospital has a private cost for serving one patient. Each hospital may have its own physical capacity on how many patients it can serve in one time unit, or this capacity can be taken to be the total number of patients. But the study in [8] focused on the structure of the optimal solution and the dynamics between hospitals and patients, instead of eliciting the hospitals' true costs by DST budget feasible mechanisms.

1.1 Our main results

An impossibility result. Although DST budget feasible mechanisms with constant approximation ratios have been constructed in single-unit procurements for additive valuations (namely, the knapsack problem) and sub-modular valuations [27, 12], our first result shows this is impossible for multi-unit settings, even for the special case of bounded knapsack, even for randomized mechanisms:

THEOREM 1. (rephrased) *No randomized mechanism can do better than a $\ln n$ approximation for bounded knapsack, where n is the total number of units of all items.*

[2] In the coverage problem a player has a set of elements, but still the outcome is binary for him: either his whole set is taken or none of the elements is taken.

This theorem applies to all classes of multi-unit procurements we consider in this paper because they all contain bounded knapsack as a special case.

An optimal mechanism for additive valuations. Theorem 1 gives us a guide on what to expect when constructing DST budget feasible mechanisms for multi-unit settings. In fact, for *additive valuations* —where the buyer has a value v_{ij} for each item i's j-th unit and his total value is the sum of his value for each unit he gets—, it is surprising that we can construct a simple mechanism which is almost optimal.

THEOREM 2. (rephrased) *There is a polynomial-time mechanism which is a* $4(1 + \ln n)$ *approximation for additive valuations.*

Since bounded knapsack is a special case of additive valuations, our mechanism is optimal within a constant factor. More interestingly, given that bounded knapsack has an FPTAS when there is no strategic considerations, our results show that there is a gap between the optimization domain and the mechanism design domain for what one can expect when solving bounded knapsack.

The main idea of our mechanism is a greedy algorithm based on each unit's marginal value-cost rate. Indeed, greedy algorithms are natural choices when constructing budget feasible mechanisms in single-unit settings, because it is obviously monotone: when a seller lowers his cost his item gets ranked better, thus if it was taken by the greedy algorithm before it will still be taken now. In single parameter settings (which cover both single-unit and multi-unit procurements), monotonicity is the first thing one need for a DST mechanism. For multi-unit additive valuations, the greedy algorithm is still monotone and thus can be used as a building block for DST mechanisms. Yet we need to be careful about how to compute the payments to the sellers, and proving that the total payment is budget feasible also need new ideas, as shown by the analysis in Section 4.

Beyond additive valuations. We do not know how to use greedy algorithms to construct DST budget feasible mechanisms for more general classes of valuations. The fundamental reason is that *they are not monotone anymore.* Intuitively, after lowering his cost a seller i's first unit may be picked up by the algorithm in an earlier step, say in step 2 instead of the previous step 3. But now the first two units picked up by the algorithm are different from before, and the marginal values of all remaining units of all sellers, including those of seller i's, have to be re-calculated with respected to these new units.[3] Depending on how the buyer's valuation function is defined, the re-calculated marginal values may completely change how the sellers' units are ranked. Accordingly, other sellers' remaining units may become more preferable than seller i's second unit, and seller i may not get any other unit he got before. Notice that this does not happen in single-unit settings: seller i has only one unit, and once this unit is picked up by the algorithm, how the other sellers' marginal values are re-calculated and re-ordered with respect to it does not affect seller i at all.

To demonstrate the non-monotonicity of greedy algorithms clearer, in Section 5.1 for valuations with *diminishing return* (which is a natural generalization of single-unit sub-modular values and is defined in Section 2), we give a concrete example where the greedy algorithm on marginal value-cost rates is not monotone: by lowering his cost a seller actually gets fewer units in the final outcome. Thus for more general settings, such as multi-unit *sub-modular* and *sub-additive* valuations (also defined in Section 2), the greedy algorithm cannot be used either.

Given this intrinsic limitation of greedy algorithms, we turn to a different approach, *random sampling* [7, 18, 13, 11, 4, 5], for constructing DST budget feasible mechanisms in settings beyond additive valuations. In particular we have the following theorem.

[3] The marginal values do not change for additive valuations, and that is why the greedy algorithm is still monotone there.

THEOREM 4. (rephrased) *There is a polynomial-time mechanism which is an $O(\frac{\log^2 n}{\log\log n})$ approximation for sub-additive valuations.*

Notice that for bounded knapsack and additive valuations our results are presented using the natural logarithm, since those are the precise bounds we achieve. While for sub-additive valuations we present our asymptotic bound under base-2 logarithm, to be more consistent with the literature.

In Theorem 4 the running time is measured when given access to a demand oracle, which is a standard assumption for handling sub-additive valuations [14, 7], since such a valuation takes exponentially many numbers to specify. Since sub-additivity includes sub-modularity and diminishing return as special cases, Theorem 4 applies to those settings as well.

Our mechanism is a generalization of the budget feasible mechanism constructed by [7], which gives an $O(\frac{\log n}{\log\log n})$ approximation for single-unit sub-additive valuations. But finding the correct way of bringing their mechanism to multi-unit settings requires a good understanding of the new problem and new ideas to deal with scenarios that did not occur before. In particular, before an item and the unique unit of it are the same thing, but in multi-unit settings they are totally different. Thus in both our mechanism and our analysis we need to be very careful about when we are dealing with an item (thus all units of it) and when we are dealing with a unit.

Moreover, to obtain the desired approximation ratio we need to first construct a mechanism for approximating the optimal *single-item outcome*: namely, an outcome that only takes units from a single seller. We believe this mechanism is of its own interest, and will be a useful building block for constructing budget feasible mechanisms in multi-unit settings.

1.2 Open problems

There are many problems in multi-unit procurements that are not the focus of this paper but worth studying in the future. For example, for sub-additive valuations there is a gap between the upperbound in Theorem 1 and the lowerbound in Theorem 4. It would be interesting to see how this gap can be closed: either by a better upperbound for sub-additive valuations or by a better mechanism, or both. As another example, online procurement with budget has been studied, both in strategic settings [6] and in optimization settings [19]. But in the strategic setting again only single-unit scenarios are considered. It is natural to ask what if a seller with multiple units of the same item can show up at different time points in the game and the buyer needs to decide how many units he wants to buy each time. Furthermore, various constraints can be imposed to the original setting, such as the buyer having different budgets for different sellers. Finally, one can ask what if a seller's cost for one unit of his item is not fixed but decreases as he produces more units. But this setting is not a single-parameter setting anymore, and presumably very different approaches will be needed there.

1.3 Related work

Various procurement auctions have been studied in the literature, but without budget considerations (see, e.g., [23, 26] and the references there). In particular, [25, 16, 17] study the minimization of total *cost* of the sellers for the items bought, instead of the total *payment* actually made by the buyer. The framework of frugality as introduced by [3] aims at finding mechanisms that minimize the payment in procurement auctions, and has been extensively studied by [9, 15, 20, 28, 10, 21]. But the problems studied there are very different from multi-unit procurement auctions, and there is no hard budget constraints that the outcomes must meet.

As mentioned in our discussion above, the problems we study generalize the framework of budget feasible mechanisms for single-unit procurements as introduced by [27]. In [27] DST budget feasible mechanisms that achieves constant approximation ratio for sub-modular valuations were constructed, and the author also showed impossibility results for fractional sub-additive valuations and for the problem of hiring a team of agents. Later [12] improved the approximation ratio of [27], for both randomized mechanisms and deterministic mechanisms. Variants of knapsack problems were also studied there, but the setting is still such that each seller only has a single item. In [14] the authors studied budget feasible mechanisms for single-unit sub-additive valuations, and constructed a randomized mechanism that is an $O(\log^2 n)$ approximation and a deterministic mechanism that is an $O(\log^3 n)$ approximation. It is interesting to notice that their randomized mechanism can also be generalized to multi-unit settings, resulting in a mechanism that is an $O(\log^3 n)$ approximation. For deterministic mechanisms, as will be discussed in our Section 3, it is impossible to obtain any approximation ratio better than n in multi-unit settings, and an n-approximation is trivial. Thus we focus on randomized mechanisms. Moreover, in [7] the authors study budget feasible mechanisms in both prior-free models and Bayesian models. For the former, they provide a constant approximation for XOS valuations and a $O(\frac{\log n}{\log \log n})$ approximation for sub-additive valuations; and for the latter, they provide a constant approximation for the sub-additive case. As we have discussed before, we generalize their mechanism for the prior-free worst case mechanism to multi-unit settings, but we need to give up a $\log n$ factor in the approximation ratio. It would also be interesting to study multi-unit budget feasible mechanisms in the Bayesian model.

Finally, knapsack auctions have been studied by [1], where the underlying optimization problem is the knapsack problem, but a seller's private information is the value of his item, instead of the cost. Thus this is a very different problem from ours or those studied in the budget feasibility framework in general.

2 The Procurement Game

Let us now specify the parameters of a procurement environment[4] and the goal we want to achieve.

There is one *buyer* who is the central planner or the mechanism designer, and m sellers who are the *players* of the procurement game. There are m *items* and they may or may not be different. Each player i can provide n_i *units* of item i, where each unit is indivisible. The total units of all the items that can be provided is $n \triangleq \sum_i n_i$. The *true cost* for providing one unit of item i is $c_i \geq 0$ and the *true cost profile* is $c = (c_1, \ldots, c_m)$. The value of c_i is player i's private information. All other information in a procurement environment is publicly known by both the players and the buyer.

An *allocation* A is a profile of integers, $A = (a_1, \ldots, a_m)$, where $a_i \in \{0, 1, \ldots, n_i\}$ for each $i \in [m]$ and it denotes the number of units (of item i) bought from player i. An *outcome* ω is specified by a pair, $\omega = (A, P)$, where A is an allocation and P is the *payment profile*, a profile of non-negative reals with P_i being the amount of money the buyer pays to player i. The *utility* of player i at outcome ω is $u_i(\omega) = P_i - a_i c_i$. As usual, the players are risk-neutral and will act to maximize their expected utilities.

The *value* of an allocation to the buyer is specified by his *valuation function* V, mapping allocations to non-negative reals, such that $V(0, \ldots, 0) = 0$. We shall consider *monotone* valuation functions: namely, for any two allocations $A = (a_1, \ldots, a_m)$ and $A' = (a'_1, \ldots, a'_m)$ such that $a_i \leq a'_i$ for each i, $V(A) \leq V(A')$.

The buyer has a *budget* B and he wants to maximize his value while keeping the total payment within the budget. An allocation A is *(budget) feasible* if its total cost, $\sum_{i \in [m]} c_i a_i$, does not exceed

[4] An environment specifies what the designer knows in order to design an mechanism: the players, the items, the possible outcomes, which parameters are publicly known, which are the players' private information, etc. Also, from now on we shall use the more general term "procurement games" instead of "procurement auctions".

B. We denote by

$$A^* \in \underset{A \text{ is a feasible allocation}}{\operatorname{argmax}} V(A)$$

an optimal feasible allocation, whose value we want to approximate. From the buyer's point of view, it is not enough for the allocation in the final outcome to be budget feasible: it is the total payment that has to meet the budget constraint. In particular, an outcome $\omega = (A, P)$ is *(budget) feasible* if $\sum_{i \in [m]} P_i \leq B$.

A mechanism is *dominant-strategy truthful (DST)* if it asks each player to reveal his cost and, for each player i, announcing his true cost c_i is a dominant strategy: namely,

$$u_i(c_i, c'_{-i}) \geq u_i(c'_i, c'_{-i}) \ \forall c'_i, c'_{-i}.$$

Throughout this paper, we only consider DST mechanisms that are also *individually rational*, namely, $u_i(c) \geq 0$ for each i.

A DST mechanism is *budget feasible* if its outcome under c is budget feasible. For randomized mechanisms, dominant-strategy truthfulness and individual rationality are defined with respect to the players' expected utilities, and budget feasibility is defined with respect to the expected payment.[5] We are interested in DST mechanisms that approximate the optimal value for the buyer while meeting the budget constraint:

Definition 1. *Let $\mathcal{C}$ be a class of procurement environments and f be a non-negative function defined on the total number of units n. A DST mechanism is an* $f(n)$-approximation *for class $\mathcal{C}$ if, for any environment in $\mathcal{C}$, the mechanism is budget feasible and individually rational, and the outcome under the true cost profile c has (expected) value at least $\frac{V(A^*)}{f(n)}$.*

Below we define several classes of procurement environments, based on the structure of the valuation functions.

Additive valuations and the bounded knapsack problem. An important class of valuation functions in the procurement game are the *additive* ones. For such a valuation function V, there exists a value v_{ik} for each item i and each unit $k \in [n_i]$, such that $v_{i1} \geq v_{i2} \geq \cdots \geq v_{in_i}$ and, given an allocation $A = (a_1, \ldots, a_m)$, $V(A) = \sum_{i \in [m]} \sum_{k \in [a_i]} v_{ik}$. Indeed, v_{ik} is the marginal value the buyer receives from an extra k-th unit of item i given that he has already gotten $k-1$ units, no matter how many units he has gotten for other items.

A special case of additive valuations is the *bounded knapsack* problem, one of the most classical problems in computational complexity. Here for each item i, all the units have the same value v_i: that is, $v_{i1} = v_{i2} = \cdots = v_{in_i} = v_i$.

Sub-additive valuations. A much more general class of valuation functions are the *sub-additive* ones. A valuation function V is sub-additive if for any two allocations $A = (a_1, \ldots, a_m)$ and $A' = (a'_1, \ldots, a'_m)$,

$$V(A \vee A') \leq V(A) + V(A'),$$

where $\vee$ is the item-wise max operation: $A \vee A' = (\max\{a_1, a'_1\}, \ldots, \max\{a_m, a'_m\})$.

[5] One can also define ex-post budget feasibility, requiring that, under a randomized outcome, with probability 1 the total payment does not exceed the budget. But one can convert a randomized outcome into another one with the same randomized allocation and a deterministic payment profile where the payment for a player is the expected payment in the former. Since the players are expected-utility maximizers, these two outcomes are equivalent for them. Thus considering ex-post budget feasibility is the same as considering budget-feasibility in expectation.

Notice that in multi-unit procurement settings, being sub-additive is a very weak assumption. Indeed, when there is only one item, sub-additivity does not impose any constraint on the valuation, not even decreasing marginal.

Sub-additivity clearly implies additivity. Between them, two other classes of valuation functions have been defined in multi-unit settings [19], but to the best of our knowledge no budget-feasible mechanisms have been considered for them:

- *Diminishing return*: for any two allocations A and A' such that $a_i \leq a'_i$ for each i, and for any item j, $V(A+e_j) - V(A) \geq V(A'+e_j) - V(A')$, where $A+e_j$ means adding one extra unit of item j to A unless $a_j = n_j$, in which case $A + e_j = A$.

- *Sub-modularity*: for any A and A', $V(A \vee A') + V(A \wedge A') \leq V(A) + V(A')$, where $\wedge$ is the item-wise min operation.

Diminishing return implies sub-modularity, and both collapse to sub-modularity in single-unit settings. The reason why diminishing return has been considered when generalizing single-unit sub-modularity to multi-unit settings is that, multi-unit sub-modularity as we just introduced is a very weak condition: when there is only one item it does not impose any constraint, just like sub-additivity. While diminishing return reflects the basic idea behind single-unit sub-modularity more precisely: the buyer's value for getting one extra unit of any item becomes smaller as he gets more.

Since the valuation classes defined above are nested:

$$\text{bounded knapsack} \subseteq \text{additivity} \subseteq \text{diminishing return} \subseteq \text{sub-modularity} \subseteq \text{sub-additivity},$$

any impossibility result for one class automatically applies to all classes above it, and any positive result for one class applies to all classes below it.

Demand oracle. In general, a sub-additive valuation function V takes exponentially many numbers to specify, and one has to assume the existence of some query oracles to access V. As in the studies of single-unit sub-additive valuations [27, 7], we consider a *demand oracle*, which takes as input a set of players $\{1, \ldots, m\}$, a profile of numbers of units $(n_1, \ldots, n_m)$, and a profile of costs $(p_1, \ldots, p_m)$, and returns an allocation

$$\hat{A} \in \operatorname*{argmax}_{A=(a_1,\ldots,a_m):a_i \leq n_i \forall i} V(A) - \sum_{i\in[m]} a_i p_i,$$

regardless of the budget.[6]

Our goal. We shall construct (randomized) DST mechanisms that are individually rational, budget feasible, and approximately achieve the optimal value for the buyer. The mechanisms run in polynomial time for additive valuations, and in polynomial time given the demand oracle for sub-additive valuations.

Single-parameter settings with budgets. Since the cost c_i is player i's only private information, we are considering single-parameter settings [2]. Following Myerson's lemma [24] or the characterization in [2], the only DST mechanisms are those with a monotone allocation rule and

[6]In single-unit settings a demand oracle takes as input a set of players and a cost profile. For multi-unit settings it is natural to add the number of units for each item as part of the input as well.

threshold payments. In multi-unit settings, each unit of an item i has its own threshold and the total payment to i will be the sum of the thresholds for all units of his item bought by the mechanism.

With budget, this characterization still holds, but the problem becomes harder: the monotone allocation rule must be such that, not only (1) it provides good approximation to the optimal value, but (2) the unique total payment induced by the allocation rule satisfies the budget constraint. Therefore, as for single-unit budget-feasible mechanisms, we shall construct monotone allocation rules while keeping an eye on the structure of the threshold payments. We need to make sure that when the two are combined, both (1) and (2) are satisfied.

3 Impossibility results for bounded knapsack

For deterministic mechanisms, the following observation for bounded knapsack is immediate.

Observation 1. *No deterministic DST mechanism can be an n-approximation for bounded knapsack.*

Proof. Consider the case where $m = 1$, $n_1 = n$, and $v_1 = 1$. When $c_1 = B$, the optimal allocation is to buy one unit from player 1. Thus any DST mechanism that is an n-approximation must buy at least 1 unit. Given the budget constraint, it must buy exactly 1 unit and pay B to player 1 (by individual rationality). Therefore when $c_1 = B/n$, the mechanism, seeing bid B/n, must also buy one unit and pay B: otherwise player 1 can get a better utility by bidding B, contradicting the fact that the mechanism is DST. Thus with true cost B/n the value generated by the mechanism is 1 while the optimal value is n. □

Of course there is an n-approximation, which buys 1 unit from a player $i \in \text{argmax}_j v_j$ and pays B to player i, without looking at the players' bids.

For randomized mechanisms, we have the following theorem.

Theorem 1. *No randomized DST mechanism can be an $f(n)$-approximation for bounded knapsack with $f(n) < \ln n$.*

Proof. For the sake of contradiction, assume there exists a function $f(n) < \ln n$ and a randomized DST mechanism which is an $f(n)$-approximation. Again consider the case where $m = 1$, $n_1 = n$, and $v_1 = 1$.

For any $b, c \in [0, B]$, let $u_1(b; c)$ be player 1's expected utility by bidding b while the true cost is c. For each $k \in [n]$, consider the outcome under bid $\frac{B}{k}$: denote by P^k the expected payment, and by p_j^k the probability that the mechanism buys j units from player 1 for each $j \in [n]$.

Notice that when the true cost is $\frac{B}{k}$, the optimal value is k. Since the mechanism is an $f(n)$-approximation, we have

$$\sum_{j \in [n]} p_j^k \cdot j \geq \frac{k}{f(n)} \quad \forall k \in [n]. \tag{1}$$

Because the mechanism is DST and individually rational, we have $u_1(\frac{B}{k}; \frac{B}{k}) \geq u_1(\frac{B}{k-1}; \frac{B}{k})$ for each $1 < k \leq n$, and $u_1(B; B) \geq 0$. That is,

$$P^k - \frac{B}{k} \sum_{j \in [n]} p_j^k \cdot j \geq P^{k-1} - \frac{B}{k} \sum_{j \in [n]} p_j^{k-1} \cdot j \quad \forall 1 < k \leq n,$$

and

$$P^1 - B \sum_{j \in [n]} p_j^1 \cdot j \geq 0.$$

Summing up these n inequalities, we have

$$\sum_{k\in[n]} P^k - \sum_{k\in[n]} \frac{B}{k} \sum_{j\in[n]} p_j^k \cdot j \geq \sum_{1\leq k<n} P^k - \sum_{1\leq k<n} \frac{B}{k+1} \sum_{j\in[n]} p_j^k \cdot j,$$

which implies

$$P^n \geq \frac{B}{n} \sum_{j\in[n]} p_j^n \cdot j + \sum_{1\leq k<n} \frac{B}{k(k+1)} \sum_{j\in[n]} p_j^k \cdot j.$$

By Equation 1, we have

$$P^n \geq \frac{B}{f(n)} + \sum_{1\leq k<n} \frac{B}{(k+1)f(n)} = \frac{B}{f(n)} \sum_{k\in[n]} \frac{1}{k} \geq \frac{B\ln n}{f(n)}.$$

Since $f(n) < \ln n$, we have $P^n > B$, contradicting the fact that the mechanism is budget feasible. Thus Theorem 1 holds. $\square$

4 An optimal mechanism for additive valuations

Let us now construct a polynomial-time DST mechanism M_{Add} that is a $4(1 + \ln n)$-approximation for procurement games with additive valuations. Our mechanism is very simple, and the basic idea is a greedy algorithm with proportional cost sharing, as have been used in budget feasible mechanisms for single-unit settings [12, 27]. However, the key here is to understand the structure of the threshold payments and to show that the mechanism is budget feasible, which requires ideas not seen before. Moreover, given our impossibility result, this mechanism is optimal (up to a constant factor). In particular, it achieves the optimal approximation ratio for bounded knapsack. The simplicity and the optimality of our mechanism make it attractive to be actually implemented in real-life scenarios.

Notations and Conventions. Without loss of generality, we assume $v_{ij} > 0$ for each item i and $j \in [n_i]$, since otherwise the mechanism can first remove all units with value 0 from consideration. Since we shall show the mechanism is DST, we shall describe it only with respect to the truthful bid $(c_1, \ldots, c_n)$. Since the threshold payments are uniquely determined by the allocation rule, we shall only describe the allocation rule. An algorithm for computing the thresholds will be given in the analysis. Finally, let $i^* \in \text{argmax}_i\, v_{i1}$, e_{i^*} be the allocation with 1 unit of item i^* and 0 unit of other items, and $A_\perp = (0, \ldots, 0)$ be the allocation where nothing is bought.

The mechanism works as follows.

Mechanism M_{Add} for Additive Valuations

1. With probability $\frac{1}{2(1+\ln n)}$, go to Step 2; with probability $\frac{1}{2}$, output e_{i^*} and stop; and with the remaining probability, output $A_\perp$ and stop.
2. For each $i \in [m]$ and $j \in [n_i]$, let the *value-rate* $r_{ij} = v_{ij}/c_i$.
   (a) Order the n pairs (i, j) according to r_{ij} decreasingly, with ties broken lexicographically, first by i and then by j.
   For any $\ell \in [n]$, denote by (i_ℓ, j_ℓ) the ℓ-th pair in the ordered list.
   (b) Let k be the largest number in $[n]$ satisfying $\frac{c_{i_k}}{v_{i_k j_k}} \leq \frac{B}{\sum_{\ell\leq k} v_{i_\ell j_\ell}}$.
   (c) Pick up the first k pairs in the list: that is, output allocation $A = (a_1, \ldots, a_n)$ where $a_i = |\{\ell : \ell \leq k \text{ and } i_\ell = i\}|$.

Theorem 2. *Mechanism M_{Add} runs in polynomial time, is DST and individually rational, and is a $4(1+\ln n)$-approximation for procurement games with additive valuations.*

Proof. We break the proof of Theorem 2 into the following lemmas.

Lemma 1. *Mechanism M_{Add} is DST and individually rational.*

Proof. This mechanism is a convex combination of three deterministic sub-mechanisms: the one outputs $A_\perp$, the one outputs e_{i^*}, and the one in Step 2. Obviously the first two are monotone.

For the one in Step 2, notice that for any $k' \leq k$ we have

$$\frac{c_{i_{k'}}}{v_{i_{k'}j_{k'}}} \leq \frac{c_{i_k}}{v_{i_k j_k}} \leq \frac{B}{\sum_{\ell \leq k} v_{i_\ell j_\ell}} \leq \frac{B}{\sum_{\ell \leq k'} v_{i_\ell j_\ell}},$$

and for any $k' \geq k+1$ we have

$$\frac{c_{i_{k'}}}{v_{i_{k'}j_{k'}}} \geq \frac{c_{i_{k+1}}}{v_{i_{k+1}j_{k+1}}} > \frac{B}{\sum_{\ell \leq k+1} v_{i_\ell j_\ell}} \geq \frac{B}{\sum_{\ell \leq k'} v_{i_\ell j_\ell}}.$$

Thus a pair $(i_{k'}, j_{k'})$ is picked up if and only if

$$\frac{c_{i_{k'}}}{v_{i_{k'}j_{k'}}} \leq \frac{B}{\sum_{\ell \leq k'} v_{i_\ell j_\ell}}. \tag{2}$$

For any player i and any pair (i,j) picked up by the mechanism, when c_i decreases, (i,j)'s rank is smaller than or equal to its previous rank. Since (i,j) satisfied Inequality 2 before, it continues to satisfy under the new ordering because the left-hand side gets smaller and the right-hand side can only gets bigger. Accordingly, (i,j) will still be picked up under the new ordering. Therefore the number of units of player i picked up by the mechanism will never decrease when i's cost decreases. Hence the mechanism is monotone.

Since each sub-mechanism pays the players according to the thresholds, each one of them is DST, and thus M_{Add} is DST.

It is easy to see that M_{Add} is individually rational. Indeed, the sub-mechanism outputting $A_\perp$ pays 0 to every player and the sub-mechanism outputting e_{i^*} pays B to player i^* and 0 to others. Thus the players get non-negative utilities in both of them. For the sub-mechanism in Step 2 and for any player i, even without giving the explicit formula of the threshold payments, by the monotonicity of the allocation rule one can see that, any pair (i,j) picked up under i's true cost c_i will still be picked up under any cost $c_i' < c_i$, and thus the threshold for picking up (i,j) is at least c_i. Accordingly, the total payment to i is at least $c_i a_i$, giving player i a non-negative utility. □

Lemma 2. *Mechanism M_{Add} runs in polynomial time.*

Proof. The only thing that is not clear from the mechanism's description is how to compute the threshold payments for the sub-mechanism in Step 2. For each player i and each $j \leq a_i$, letting θ_{ij} be the threshold for the j-th unit of i, we compute θ_{ij} using the following algorithm A_{Th}.

Algorithm A_{Th} for Computing the Threshold θ_{ij}

1. Order the $n - n_i$ pairs (i', j') with $i' \neq i$ according to the value-rate $r_{i'j'}$'s, with ties broken lexicographically. Denote by (i'_ℓ, j'_ℓ) the ℓ-th pair in the list.
2. Set $t'_{n-n_i+1} = +\infty$.
3. For α from $n - n_i$ to 0, compute $t_\alpha = \frac{v_{ij}B}{\sum_{\ell \leq j} v_{i\ell} + \sum_{\ell \leq \alpha} v_{i'_\ell j'_\ell}}$ and $t'_\alpha = \frac{v_{ij} c_{i'_\alpha}}{v_{i'_\alpha j'_\alpha}}$, except for $\alpha = 0$ where $t'_0 = 0$.
   (a) If $t_\alpha < t'_\alpha$, continue to the next round.
   (b) If $t'_\alpha \leq t_\alpha \leq t'_{\alpha+1}$, set $\theta_{ij} = t_\alpha$ and stop.
   (c) If $t_\alpha > t'_{\alpha+1}$, set $\theta_{ij} = t'_{\alpha+1}$ and stop.
4. Output $\perp$.

Claim 1. *The algorithm A_{Th} computes the correct threshold θ_{ij}.*

Proof. We first show the algorithm does not output $\perp$, namely, the condition in Step 3a does not hold in some round. Let (i, j) be the β-th pair in the ordered list produced by the mechanism, that is, $(i, j) = (i_\beta, j_\beta)$. We have $\beta \geq j$ as there are $j - 1$ pairs of player i before (i, j). Accordingly, there are $\beta - j$ pairs of other players before (i, j), denoted by (i'_1, j'_1) through $(i'_{\beta-j}, j'_{\beta-j})$ in the algorithm. By the construction of the mechanism and the algorithm,

$$\frac{t'_{\beta-j}}{v_{ij}} = \frac{c_{i'_{\beta-j}}}{v_{i'_{\beta-j} j'_{\beta-j}}} \leq \frac{c_i}{v_{ij}} \leq \frac{B}{\sum_{\ell \leq j} v_{i\ell} + \sum_{\ell \leq \beta-j} v_{i'_\ell j'_\ell}} = \frac{t_{\beta-j}}{v_{ij}}.$$

Thus $t'_{\beta-j} \leq t_{\beta-j}$ and in this round the condition in Step 3a does not hold. If the algorithm has not already stopped in some earlier round, it will stop here. Accordingly, the algorithm does not reach Step 4 and does not output $\perp$. Moreover, from this analysis we also know when the algorithm stops, $\alpha \geq \beta - j$.

We now show that if player i bids $c'_i > \theta_{ij}$ then the mechanism will not pick up pair (i, j). Indeed, if the algorithm stops in Step 3b, then $\theta_{ij} = t_\alpha$ and $c'_i > t_\alpha \geq t'_\alpha$. Accordingly,

$$\frac{c'_i}{v_{ij}} > \frac{t'_\alpha}{v_{ij}} = \frac{c_{i'_\alpha}}{v_{i'_\alpha j'_\alpha}},$$

and pair (i, j) is ranked after pair (i'_α, j'_α). Therefore Inequality 2 is violated for pair (i, j) because

$$\frac{c'_i}{v_{ij}} > \frac{t_\alpha}{v_{ij}} = \frac{B}{\sum_{\ell \leq j} v_{i\ell} + \sum_{\ell \leq \alpha} v_{i'_\ell j'_\ell}}.$$

Notice that pair (i, j) may be ranked after pair $(i'_{\alpha+1}, j'_{\alpha+1})$ and so on, but then Inequality 2 remains violated, since the right-hand side of the inequality will only get smaller, with more terms added to the denominator.

If the algorithm instead stops in Step 3c, then $c'_i > \theta_{ij} = t'_{\alpha+1}$, which implies

$$\frac{c'_i}{v_{ij}} > \frac{t'_{\alpha+1}}{v_{ij}} = \frac{c_{i'_{\alpha+1}}}{v_{i'_{\alpha+1} j'_{\alpha+1}}}.$$

Accordingly, pair (i,j) is ranked after pair $(i'_{\alpha+1}, j'_{\alpha+1})$. But the algorithm did not stop at $\alpha+1$, which means $t_{\alpha+1} < t'_{\alpha+1}$. Again Inequality 2 is violated for pair (i,j), since

$$\frac{c'_i}{v_{ij}} > \frac{t'_{\alpha+1}}{v_{ij}} > \frac{t_{\alpha+1}}{v_{ij}} = \frac{B}{\sum_{\ell \le j} v_{i\ell} + \sum_{\ell \le \alpha+1} v_{i'_\ell j'_\ell}}.$$

Notice that pair (i,j) may be ranked even further down, but then Inequality 2 remains violated, since the right-hand side of the inequality will only get smaller.

In sum, pair (i,j) will not be picked up by the mechanism for any $c'_i > \theta_{ij}$.

Finally, we show if player i bids $c'_i < \theta_{ij}$ then the mechanism will pick up pair (i,j). To do so, notice that no matter whether the algorithm stops in Step 3b or 3c, we have

$$c'_i < \theta_{ij} = \min\{t'_{\alpha+1}, t_\alpha\}.$$

Accordingly,

$$\frac{c'_i}{v_{ij}} < \frac{t'_{\alpha+1}}{v_{ij}} = \frac{c_{i'_{\alpha+1}}}{v_{i'_{\alpha+1} j'_{\alpha+1}}}$$

and

$$\frac{c'_i}{v_{ij}} < \frac{t_\alpha}{v_{ij}} = \frac{B}{\sum_{\ell \le j} v_{i\ell} + \sum_{\ell \le \alpha} v_{i'_\ell j'_\ell}}.$$

Thus (i,j) is ranked before $(i'_{\alpha+1}, j'_{\alpha+1})$, and Inequality 2 is satisfied for (i,j). Again, (i,j) may be ranked even earlier on, but then the right-hand side of the inequality will only get bigger, with some terms taken out from the denominator. Thus (i,j) will be picked up by the mechanism for any $c'_i < \theta_{ij}$.

Putting everything together, Claim 1 holds. □

The algorithm A_{Th} clearly runs in polynomial time, thus the mechanism M_{Add} runs in polynomial time as well, and Lemma 2 holds. □

Lemma 3. *Mechanism M_{Add} is budget feasible.*

Proof. We provide an upperbound for the total payment made by the sub-mechanism in Step 2 of M_{Add}, namely,

$$\sum_{i \le m} \sum_{j \le a_i} \theta_{ij} \le (1 + \ln n) B.$$

To do so, recall that the mechanism picks up the first k pairs in the ordered list according to the value-rates r_{ij}'s, denoted by (i_1, j_1) through (i_k, j_k). (By definition $k = \sum_{i \le m} a_i$.)

Re-order these k pairs according to v_{ij}'s decreasingly with ties broken lexicographically. We denote by $(\hat{i}_s, \hat{j}_s)$ the s-th pair in this new ordering. We have

$$v_{\hat{i}_1 \hat{j}_1} \ge v_{\hat{i}_2 \hat{j}_2} \ge \cdots \ge v_{\hat{i}_k \hat{j}_k} \tag{3}$$

and

$$\sum_{\ell \le k} v_{i_\ell j_\ell} = \sum_{\ell \le k} v_{\hat{i}_\ell \hat{j}_\ell}. \tag{4}$$

Below we show $\theta_{\hat{i}_s \hat{j}_s} \le \frac{B}{s}$ for any $s \le k$.

Assume for the sake of contradiction that there exists $s \le k$ such that $\theta_{\hat{i}_s \hat{j}_s} > \frac{B}{s}$. Consider the pair (i_k, j_k) in the mechanism's ordering. We have

$$\frac{c_{i_k}}{v_{i_k j_k}} \le \frac{B}{\sum_{\ell \le k} v_{i_\ell j_\ell}} = \frac{B}{\sum_{\ell \le k} v_{\hat{i}_\ell \hat{j}_\ell}} \le \frac{B}{\sum_{\ell \le s} v_{\hat{i}_\ell \hat{j}_\ell}} \le \frac{B}{s \cdot v_{\hat{i}_s \hat{j}_s}} < \frac{\theta_{\hat{i}_s \hat{j}_s}}{v_{\hat{i}_s \hat{j}_s}},$$

where the first inequality is by the construction of the mechanism, the equality is by Equation 4, the second inequality is because $s \leq k$, the third inequality is by Equation 3, and the last inequality is because $\theta_{\hat{i}_s \hat{j}_s} > \frac{B}{s}$. Therefore when player $\hat{i}_s$ bids some $c'_{\hat{i}_s} \in (\frac{B}{s}, \theta_{\hat{i}_s \hat{j}_s})$, we have

$$\frac{c'_{\hat{i}_s}}{v_{\hat{i}_s \hat{j}_s}} > \frac{B}{s \cdot v_{\hat{i}_s \hat{j}_s}} \geq \frac{c_{i_k}}{v_{i_k j_k}}.$$

Accordingly, for any $\ell < s$ such that $\hat{i}_\ell \neq \hat{i}_s$, the pair $(\hat{i}_\ell, \hat{j}_\ell)$ is ranked before $(\hat{i}_s, \hat{j}_s)$ by the mechanism under player $\hat{i}_s$'s new bid $c'_{\hat{i}_s}$, because $\frac{c_{\hat{i}_\ell}}{v_{\hat{i}_\ell \hat{j}_\ell}} \leq \frac{c_{i_k}}{v_{i_k j_k}}$.

Moreover, for any $\ell < s$ such that $\hat{i}_\ell = \hat{i}_s$, the pair $(\hat{i}_\ell, \hat{j}_\ell)$ is also ranked before $(\hat{i}_s, \hat{j}_s)$ by the mechanism under player $\hat{i}_s$'s new bid, because $v_{\hat{i}_\ell \hat{j}_\ell} \geq v_{\hat{i}_s \hat{j}_s}$ and the two pairs have the same cost. (When $v_{\hat{i}_\ell \hat{j}_\ell} = v_{\hat{i}_s \hat{j}_s}$, it must be $\hat{j}_\ell < \hat{j}_s$ due to the lexicographic tie-breaking rule, and thus $(\hat{i}_\ell, \hat{j}_\ell)$ is still ranked before $(\hat{i}_s, \hat{j}_s)$ by the mechanism under player $\hat{i}_s$'s new bid, due to the same tie-breaking rule.)

Accordingly, when player $\hat{i}_s$ bids $c'_{\hat{i}_s}$, all $s-1$ pairs $(\hat{i}_1, \hat{j}_1), \ldots, (\hat{i}_{s-1}, \hat{j}_{s-1})$ are ranked before $(\hat{i}_s, \hat{j}_s)$ by the mechanism, and the total value of all pairs before or equal to $(\hat{i}_s, \hat{j}_s)$ is at least $\sum_{\ell \leq s} v_{\hat{i}_\ell \hat{j}_\ell}$. Because

$$\frac{c'_{\hat{i}_s}}{v_{\hat{i}_s \hat{j}_s}} > \frac{B}{s \cdot v_{\hat{i}_s \hat{j}_s}} \geq \frac{B}{\sum_{\ell \leq s} v_{\hat{i}_\ell \hat{j}_\ell}},$$

the pair $(\hat{i}_s, \hat{j}_s)$ is not picked up by the mechanism under $c'_{\hat{i}_s}$, contradicting the fact that $c'_{\hat{i}_s} < \theta_{\hat{i}_s \hat{j}_s}$.

Therefore we have

$$\theta_{\hat{i}_s \hat{j}_s} \leq \frac{B}{s} \quad \text{for any } s \leq k,$$

which implies

$$\sum_{i \leq m} \sum_{j \leq a_i} \theta_{ij} = \sum_{s \leq k} \theta_{\hat{i}_s \hat{j}_s} \leq \sum_{s \leq k} \frac{B}{s} \leq B \sum_{s \leq n} \frac{1}{s} \leq (1 + \ln n) B,$$

as we wanted to show.

Since the sub-mechanism outputting e_{i^*} pays B to player i^* and 0 to others, the expected payment of mechanism M_{Add} is at most

$$\frac{1}{2(1 + \ln n)} \cdot (1 + \ln n) B + \frac{1}{2} \cdot B = B,$$

and Lemma 3 holds. □

Lemma 4. *Mechanism M_{Add} is a $4(1 + \ln n)$-approximation.*

Proof. Recall that A^* is an optimal allocation and i^* is a player whose first unit has the highest value among all units of all players. We shall show that $V(A) + V(e_{i^*})$ is a 2-approximation of $V(A^*)$.

To do so, notice that once the cost profile $(c_1, \ldots, c_n)$ is given, without strategic considerations, the optimization problem in our setting can be reduced to a 0-1 knapsack problem with n items and budget B. In particular, for each player $i \in [m]$ and each $j \in [n_i]$, there is an item (i, j) with value v_{ij} and cost $c_{ij} = c_i$. For any subset $S \subseteq \{(i, j) : i \in [m], j \in [n_i]\}$, its value is $V(S) = \sum_{(i,j) \in S} v_{ij}$, and its cost is $\sum_{(i,j) \in S} c_{ij} = \sum_{(i,j) \in S} c_i$. The allocation A^* naturally corresponds to an optimal set S^* in the 0-1 knapsack problem, and the pair $(i^*, 1)$ is the item with the highest value.

For 0-1 knapsack, it is well known (see, e.g., [22]) that the greedy algorithm under value-rate sorting gives constant approximation. In particular, letting $(i_1, j_1), \ldots, (i_n, j_n)$ be the ordered list with r_{ij}'s decreasing and ties broken lexicographically, and letting $\hat{k}$ be the largest number in $[n]$ satisfying $\sum_{\ell\leq\hat{k}} c_{i_\ell} \leq B$, we have

$$\sum_{\ell\leq\hat{k}} v_{i_\ell j_\ell} + v_{i^*1} \geq \sum_{\ell\leq\hat{k}+1} v_{i_\ell j_\ell} \geq V(S^*) = V(A^*), \tag{5}$$

where $v_{i_{\hat{k}+1}j_{\hat{k}+1}} = 0$ if $\hat{k} = n$, the first inequality is by the definition of i^*, and the second is because $\sum_{\ell\leq\hat{k}+1} v_{i_\ell j_\ell}$ is greater than or equal to the optimal fractional solution.

For the k pairs picked up by the mechanism, we have $\frac{c_{i_\ell}}{v_{i_\ell j_\ell}} \leq \frac{c_{i_k}}{v_{i_k j_k}}$ for any $\ell \leq k$, thus

$$\frac{\sum_{\ell\leq k} c_{i_\ell}}{\sum_{\ell\leq k} v_{i_\ell j_\ell}} \leq \frac{c_{i_k}}{v_{i_k j_k}} \leq \frac{B}{\sum_{\ell\leq k} v_{i_\ell j_\ell}},$$

which implies $\sum_{\ell\leq k} c_{i_\ell} \leq B$. Accordingly, $k \leq \hat{k}$.

If $k = \hat{k}$, by Inequality 5 we have

$$V(A) + V(e_{i^*}) = \sum_{\ell\leq\hat{k}} v_{i_\ell j_\ell} + v_{i^*1} \geq V(A^*). \tag{6}$$

If $k < \hat{k}$, then

$$\frac{\sum_{k+1\leq\ell\leq\hat{k}} c_{i_\ell}}{\sum_{k+1\leq\ell\leq\hat{k}} v_{i_\ell j_\ell}} \geq \frac{c_{i_{k+1}}}{v_{i_{k+1}j_{k+1}}} > \frac{B}{\sum_{\ell\leq k+1} v_{i_\ell j_\ell}},$$

where the first inequality is because $\frac{c_{i_\ell}}{v_{i_\ell j_\ell}} \geq \frac{c_{i_{k+1}}}{v_{i_{k+1}j_{k+1}}}$ for any $\ell \geq k+1$, and the second is by the construction of the mechanism. Accordingly,

$$\sum_{k+1\leq\ell\leq\hat{k}} v_{i_\ell j_\ell} < \frac{\sum_{k+1\leq\ell\leq\hat{k}} c_{i_\ell}}{B} \cdot \sum_{\ell\leq k+1} v_{i_\ell j_\ell} \leq \sum_{\ell\leq k+1} v_{i_\ell j_\ell} \leq \sum_{\ell\leq k} v_{i_\ell j_\ell} + v_{i^*1},$$

where the second inequality is because $\sum_{k+1\leq\ell\leq\hat{k}} c_{i_\ell} \leq \sum_{\ell\leq\hat{k}} c_{i_\ell} \leq B$. Thus

$$2V(A)+2V(e_{i^*}) = 2\sum_{\ell\leq k} v_{i_\ell j_\ell} + 2v_{i^*1} \geq \sum_{\ell\leq k} v_{i_\ell j_\ell} + v_{i^*1} + \sum_{k+1\leq\ell\leq\hat{k}} v_{i_\ell j_\ell} = \sum_{\ell\leq\hat{k}} v_{i_\ell j_\ell} + v_{i^*1} \geq V(A^*), \tag{7}$$

where the last inequality is by Inequality 5.

Combining Inequalities 6 and 7, we have that $V(A) + V(e_{i^*})$ is a 2-approximation for $V(A^*)$, and thus the expected value of the mechanism M_{Add}'s output is

$$\frac{1}{2(1+\ln n)} \cdot V(A) + \frac{1}{2} \cdot V(e_{i^*}) \geq \frac{V(A) + V(e_{i^*})}{2(1+\ln n)} \geq \frac{V(A^*)}{4(1+\ln n)}.$$

Since individually rationality and budget feasibility have been shown by Lemmas 1 and 3, mechanism M_{Add} is a $4(1+\ln n)$-approximation, and Lemma 4 holds. □

Putting everything together, Theorem 2 holds. □

Combining Theorems 1 and 2 we immediately have the following.

Corollary 1. *Mechanism M_{Add} is optimal up to a constant factor among all DST budget-feasible mechanisms for multi-unit procurement games with additive valuations.*

Remark 1. *Theorems 1 and 2 show that multi-unit settings are very different from single-unit settings. In single-unit settings various constant-approximation mechanisms have been constructed, while in multi-unit settings an $O(\log n)$ approximation is the best one can hope, and our mechanism provides such an approximation.*

Furthermore, for bounded knapsack, without strategic considerations there is an FPTAS, while with strategic considerations no mechanism can get better than an $\ln n$-approximation. Thus we have shown that bound knapsack is a problem for which ***provably*** *there is a gap between the optimization domain and the mechanism design domain.*

An optimal mechanism for symmetric valuations. A different but closely related class of valuation functions are the *symmetric* ones. In this case the buyer only cares about how many units an allocation has, not to whom they belong. Namely, there exists $v_1, \ldots, v_n$ such that, for any allocation A with k units in total, $V(A) = \sum_{\ell \leq k} v_\ell$.

In general, symmetric valuations are not additive since there is no unique value v_{ij} associated with player i's unit j, and additive is not symmetric either. But when there is only one item with n units, symmetric is equivalent to additive, and contains bounded knapsack as a special case. Thus the proof of Theorem 1 also implies that no mechanism can do better than $\ln n$-approximation for symmetric valuations. For general symmetric valuations, it is not hard to verify that the following mechanism is a $4(1 + \ln n)$-approximation: the same as M_{Add} except in Step 2, where k is set to be the largest number in $[n]$ satisfying $c_{i_k} \leq \frac{B}{k}$. We omit the analysis as it is very similar to that of M_{Add}, and only present the theorem below.

Theorem 3. *No randomized DST mechanism can be an $f(n)$-approximation with $f(n) < \ln n$ for symmetric valuations, and there exists a polynomial-time DST mechanism which is a $4(1 + \ln n)$-approximation.*

5 Truthful mechanisms for sub-additive valuations

5.1 The non-monotonicity of the greedy algorithm

Although the greedy algorithm based on marginal value-rates is an important building block for budget feasible mechanisms in single-unit settings and multi-unit additive settings, we do not know how to use it for multi-unit sub-additive settings. The fundamental reason is that, *it is not monotone anymore, even for valuations with diminishing returns, even without strategic considerations.* Indeed, by lowering his cost, a player i will still win his first unit in the previous allocation. But once the rank of his first unit changes, all units after that will be re-ranked according to their new marginal value-rates with respect to the updated allocation. Under the new ordering there is no guarantee whether player i will still get any of his remaining units. Re-ordering was not an issue in single-unit settings, because a player in such a setting need not care how other players' items are ordered after his own (unique) item. Below we give an example demonstrating the non-monotonicity of the greedy algorithm in settings with diminishing returns.

Example 1. *There are 3 players, $n_1 = 1$, $n_2 = n_3 = 2$, $c_1 = c_3 = 1$, $c_2 = 1 + \epsilon$ for some arbitrarily small $\epsilon > 0$, and $B = 3 + 2\epsilon$. To highlight the non-monotonicity of the greedy algorithm, instead of defining the valuation function directly, we work through the algorithm and define the marginal values on the way. The valuation function will be reconstructed afterward, based on the marginal values.*

For any allocation $A = (a_1, a_2, a_3)$ and player i, denote by $V(i|A)$ the marginal value of item i given A, namely, $V(A + e_i) - V(A)$. The greedy algorithm works as follows.

- *At the beginning, the allocation is* $A_0 = (0, 0, 0)$.
- $V(1|A_0) = 10$*,* $V(2|A_0) = 10 + \epsilon$*, and* $V(3|A_0) = 10 - \epsilon$*. Item 1 has the largest marginal value-rate, thus* $A_1 = (1, 0, 0)$.
- $V(1|A_1) = 0$ *(item 1 is not available anymore),* $V(2|A_1) = 5 + 5\epsilon$*, and* $V(3|A_1) = 5 - \epsilon$*. Item 2 has the largest marginal value-rate, thus* $A_2 = (1, 1, 0)$.
- $V(1|A_2) = 0$*,* $V(2|A_2) = 1 + \epsilon$*, and* $V(3|A_2) = 1 - \epsilon$*. Item 2 has the largest marginal value-rate, thus* $A_3 = (1, 2, 0)$.
- *The budget is used up, and the final allocation is* A_3*, where player 2 gets 2 units.*

Now let $c'_2 = 1 - \epsilon$*. The greedy algorithm works as follows.*

- $A_0 = (0, 0, 0)$.
- $V(1|A_0) = 10$*,* $V(2|A_0) = 10 + \epsilon$*, and* $V(3|A_0) = 10 - \epsilon$*. Item 2 has the largest marginal value-rate, thus* $A'_1 = (0, 1, 0)$*. (Notice that player 2 gets his first unit earlier than before.)*
- $V(1|A'_1) = 5 + 4\epsilon$*,* $V(2|A'_1) = 5 - 5\epsilon$*, and* $V(3|A'_1) = 5 + 5\epsilon$*. Item 3 has the largest marginal value-rate, thus* $A'_2 = (0, 1, 1)$.
- $V(1|A'_2) = 1 - 2\epsilon$*,* $V(2|A'_2) = 1 - \epsilon$*, and* $V(3|A'_2) = 1 + \epsilon$*. Thus* $A'_3 = (0, 1, 2)$.
- *The remaining budget is* 3ϵ*, no further unit can be added, and the final allocation is* A'_3*, where player 2 only gets one unit, violating monotonicity.*

Given the marginal values, the valuation function is defined as follows:

$$
\begin{aligned}
&V(0,0,0) = 0,\\
&V(1,0,0) = 10, V(0,1,0) = 10+\epsilon, V(0,0,1) = 10-\epsilon,\\
&V(1,1,0) = 15+5\epsilon, V(1,0,1) = 15-\epsilon, V(0,2,0) = 15-4\epsilon, V(0,1,1) = 15+6\epsilon, V(0,0,2) = 15,\\
&V(1,2,0) = 16+6\epsilon, V(1,1,1) = 16+4\epsilon, V(1,0,2) = 16, V(0,2,1) = 16+5\epsilon, V(0,1,2) = 16+7\epsilon,\\
&V(0,2,2) = V(1,2,1) = V(1,1,2) = 16+7\epsilon,\\
&V(1,2,2) = 16+7\epsilon.
\end{aligned}
$$

It is easy to verify that V *is consistent with the marginal values defined above. Diminishing return is easy to check as well. Indeed, notice that we put allocations with the same number of units in the same row, to ease the comparison. For any allocation with* k *units for* k *from* 0 *to* 4*, the marginal value of adding 1 more unit of any item is roughly* $10, 5, 1, \epsilon, 0$*, and thus diminishing.*

5.2 Approximating the optimal one-item allocation

Given the non-monotonicity of the greedy algorithm, we turn to another approach for constructing DST mechanisms, namely, *random sampling.* In particular, we use a variant of the mechanism for single-unit sub-additive valuations in [7] as a subroutine of our mechanism.

Yet a new problem arises in multi-unit settings and has to be solved first. Let us explain. In mechanisms for single-unit settings, part of the value approximation comes from choosing a single player i^* with the highest marginal value and paying him the budget B. Similarly, in our additive settings we choose player i^* whose first unit has the highest marginal value, take that unit and pay him B. But for multi-unit sub-additive settings, as will become clear in our analysis, in order to get a good approximation we need (as a subroutine) to choose a single player and pick up *as many units as possible from him*, under the budget constraint. Namely, given the true cost profile $(c_1, \ldots, c_n)$, let

$$i^{**} \in \operatorname*{argmax}_i V(\min\{n_i, \lfloor \frac{B}{c_i} \rfloor\} \cdot e_i),$$

where for any $\lambda \in [n_i]$, λe_i is the allocation with λ units of item i and 0 unit of others. Ideally we want to pick up $\lambda^{**} \triangleq \min\{n_{i^{**}}, \lfloor \frac{B}{c_{i^{**}}} \rfloor\}$ units from player i^{**} and pay him B. We shall refer to (i^{**}, λ^{**}) as the *optimal single-item allocation*.

The problem is, although the identity of player i^* is publicly known, both i^{**} and λ^{**} depend on the players' true costs, and have to be solved from the players' bids. It is easy to see the ideal solution does not correspond to a DST mechanism.

Below we construct a DST mechanism, M_{One}, which is budget feasible and approximate $V(\lambda^{**} e_{i^{**}})$ within a $(1 + \ln n)$ factor. Later we shall use it as another subroutine in our mechanism for sub-additive valuations.

Mechanism M_{One} for Approximating the Optimal One-item Allocation

With probability $\frac{1}{1+\ln n}$, do the following.

1. Let $v_i = V(\min\{n_i, \lfloor \frac{B}{c_i} \rfloor\} \cdot e_i)$ and order the players according to the v_i's decreasingly, with ties broken lexicographically.

   Let i^{**} be the first player in the list and $\lambda^{**} = \min\{n_{i^{**}}, \lfloor \frac{B}{c_{i^{**}}} \rfloor\}$.

2. Let $k \in [\lambda^{**}]$ be the smallest number such that player i^{**} is still ordered the first with cost $c'_{i^{**}} = \frac{B}{k}$.

3. Set $\theta_\ell = \frac{B}{k}$ for each $\ell \le k$ and $\theta_\ell = \frac{B}{\ell}$ for each $k + 1 \le \ell \le \lambda^{**}$.

4. Output allocation $\lambda^{**} e_{i^{**}}$ and pay $\sum_{\ell \le \lambda^{**}} \theta_\ell$ to player i^{**}.

Lemma 5. *Mechanism M_{One} is DST, individually rational, budget feasible, and is a $(1 + \ln n)$-approximation for $V(\lambda^{**} e_{i^{**}})$.*

Proof. It is easy to see that the allocation is monotone. Indeed, for any player $i \neq i^{**}$, increasing i's cost can only cause v_i to decrease. Thus i is still not the first in the list and gets 0 unit in the new allocation. Decreasing his cost can only cause him to get more units, since he gets 0 under c_i.

For player i^{**}, decreasing his cost to $c' < c_{i^{**}}$ can only cause $v_{i^{**}}$ to increase, thus he is still the first in the list, and the number of units he gets is $\min\{n_{i^{**}}, \lfloor \frac{B}{c'} \rfloor\} \ge \min\{n_{i^{**}}, \lfloor \frac{B}{c_{i^{**}}} \rfloor\}$. On the other hand, increasing his cost to $c' > c_{i^{**}}$ will either cause him to lose the first place and get 0 unit, or will have him still the first but with the number of units $\min\{n_{i^{**}}, \lfloor \frac{B}{c'} \rfloor\} \le \min\{n_{i^{**}}, \lfloor \frac{B}{c_{i^{**}}} \rfloor\}$.

Next, we show that for each $\ell \in [\lambda^{**}]$, θ_ℓ is the correct threshold for the ℓ-th unit of player i^{**}. We distinguish whether $\ell \le k$ or not.

If $\ell \le k$, then by bidding $c'_{i^{**}} > \theta_\ell = \frac{B}{k}$, we have $\frac{B}{c'_{i^{**}}} < k$, and thus $\lfloor \frac{B}{c'_{i^{**}}} \rfloor \le k - 1$. Accordingly,

$$V(\min\{n_{i^{**}}, \lfloor \frac{B}{c'_{i^{**}}} \rfloor\} \cdot e_i) \le V(\min\{n_{i^{**}}, k-1\} \cdot e_i) = V(\min\{n_{i^{**}}, \lfloor \frac{B}{B/(k-1)} \rfloor\} \cdot e_i).$$

By the definition of k, player i^{**} is not the first in the list by bidding $\frac{B}{k-1}$. Thus by bidding $c'_{i^{**}}$ he is not the first either, and does not get his ℓ-th unit.

By bidding $c'_{i^{**}} < \theta_\ell$, we have $\frac{B}{c'_{i^{**}}} \geq k$, and thus

$$V(\min\{n_{i^{**}}, \lfloor \frac{B}{c'_{i^{**}}} \rfloor\} \cdot e_i) \geq V(\min\{n_{i^{**}}, \lfloor \frac{B}{B/k} \rfloor\} \cdot e_i).$$

Since player i^{**} is the first by bidding $\frac{B}{k}$, he is still the first by bidding $c'_{i^{**}}$, and the number of units he gets is $\min\{n_{i^{**}}, \lfloor \frac{B}{c'_{i^{**}}} \rfloor\} \geq \min\{n_{i^{**}}, \lfloor \frac{B}{B/k} \rfloor\} = k \geq \ell$, where the equality is because $k \leq \lambda^{**} \leq n_{i^{**}}$. That is, by bidding $c'_{i^{**}}$ he still gets his ℓ-th unit.

If $k+1 \leq \ell \leq \lambda^{**}$, then by bidding $c'_{i^{**}} > \theta_\ell = \frac{B}{\ell}$ player i^{**} will not get his ℓ-th unit even if he remains to be the first in the list, since $\lfloor \frac{B}{c'_{i^{**}}} \rfloor \leq \ell - 1$. By bidding $c'_{i^{**}} < \theta_\ell$, we have $V(\min\{n_{i^{**}}, \lfloor \frac{B}{c'_{i^{**}}} \rfloor\} \cdot e_i) \geq V(\min\{n_{i^{**}}, \lfloor \frac{B}{B/\ell} \rfloor\} \cdot e_i)$. Again by the definition of k, by bidding $\frac{B}{\ell}$ player i is the first in the list, and thus by bidding $c'_{i^{**}}$ he is still the first. The number of units he gets is $\min\{n_{i^{**}}, \lfloor \frac{B}{c'_{i^{**}}} \rfloor\} \geq \min\{n_{i^{**}}, \lfloor \frac{B}{B/\ell} \rfloor\} = \ell$. That is, by bidding $c'_{i^{**}}$ he still gets his ℓ-th unit.

In sum, the θ_ℓ's are the correct thresholds, and the mechanism M_{One} is DST.

Individual rationality follows directly from the fact that for each $\ell \leq \lambda^{**}$ we have

$$\theta_\ell \geq \frac{B}{\lambda^{**}} \geq \frac{B}{\lfloor \frac{B}{c_{i^{**}}} \rfloor} \geq c_{i^{**}},$$

and thus $\sum_{\ell \leq \lambda^{**}} \theta_\ell \geq \lambda^{**} c_{i^{**}}$.

Furthermore, because for each $\ell \leq \lambda^{**}$ we have $\theta_\ell \leq \frac{B}{\ell}$, the total payment is

$$\sum_{\ell \leq \lambda^{**}} \theta_\ell \leq \sum_{\ell \leq \lambda^{**}} \frac{B}{\ell} \leq \sum_{\ell \leq n} \frac{B}{\ell} \leq (1 + \ln n)B.$$

Since this payment is made with probability $\frac{1}{1+\ln n}$, the mechanism is budget feasible.

Finally, under the true cost profile, the mechanism outputs $\lambda^{**} e_{i^{**}}$ with probability $\frac{1}{1+\ln n}$, and thus is a $(1 + \ln n)$-approximation for $V(\lambda^{**} e_{i^{**}})$. □

Since the impossibility result in Theorem 1 applies to settings with a single item, we have the following corollary.

Corollary 2. *Mechanism M_{One} is optimal for approximating $V(\lambda^{**} e_{i^{**}})$ among all DST budget-feasible mechanisms.*

Remark 2. *Notice that mechanism M_{One} does not even require the valuation function to be sub-additive. The only thing it requires is that, for each player i, $V(\lambda e_i)$ is a non-decreasing function for $\lambda \in [n_i]$. Thus it can also be used in many settings that are more general than sub-additivity, such as the class of all monotone valuations.*

Furthermore, given that (i^{**}, λ^{**}) is the multi-unit counterpart of player i^* in single-unit settings, and given the important role i^* has played in designing budget feasible mechanisms, we believe mechanism M_{One} will be a useful building block in the design of budget feasible mechanisms for multi-unit settings.

5.3 Our mechanism for sub-additive valuations

The random-sampling mechanism in [7] provides an $O(\frac{\log n}{\log \log n})$-approximation in single-unit sub-additive settings. In this subsection we generalize their result to multi-unit settings and provide an $O(\frac{(\log n)^2}{\log \log n})$-approximation. The algorithm A_{Max} and the mechanism M_{Rand} below are respectively variants of their algorithm SA-ALG-MAX and mechanism SA-RANDOM-SAMPLE. But the differences are also clear. Indeed, we need to carefully distinguish between an item and a unit of it. In the mechanism and the analysis, we sometimes deal with an item (and thus all of its units at the same time) and sometimes deal with a single unit. Also, the role of player i^* is replaced by i^{**}, and the way it contributes to the value approximation has changed in many places —this is where the extra $\log n$ factor comes. Our mechanism M_{Sub} is a uniform distribution between M_{Rand} and the mechanism M_{One} in Section 5.2.

Algorithm A_{Max}

Since this algorithm will be used multiple times with different inputs, we specify the inputs explicitly to avoid confusion. Given players $1, \ldots, m$, numbers of units $n_1, \ldots, n_m$, costs $c_1, \ldots, c_m$, budget B, and a demand oracle for the valuation function V, do the following.

1. Let $n'_i = \min\{n_i, \lfloor \frac{B}{c_i} \rfloor\}$ for each i, $i^{**} = \text{argmax}_i V(n'_i e_i)$, $v^* = V(n'_{i^{**}} e_{i^{**}})$, and $\mathcal{V} = \{v^*, 2v^*, \ldots, mv^*\}$.

2. For each $v \in \mathcal{V}$,

   (a) Set $p_i = \frac{v}{2B} \cdot c_i$ for each player i. Query the oracle with m players, number of units n'_i and cost p_i for each i, to find
   $S = (s_1, \ldots, s_m) \in \arg\max_{A=(a_1,\ldots,a_m): a_i \le n'_i \forall i} V(A) - \sum_{i \in [m]} a_i p_i$.
   (When there are multiple optimal solutions, the oracle always returns the same one whenever queried with the same instance.)

   (b) Set allocation $S_v = A_\perp$.

   (c) If $V(S) < \frac{v}{2}$, then continue to next v.

   (d) Else, order the players according to $s_i c_i$ decreasingly with ties broken lexicographically, and denote them by $i_1, \ldots, i_m$.
   Let k be the largest number in $[m]$ satisfying $\sum_{\ell \le k} s_{i_\ell} c_{i_\ell} \le B$, and let S_v be S projected on $\{i_1, \ldots, i_k\}$: namely, $S_v = \bigvee_{\ell \le k} s_{i_\ell} e_{i_\ell}$ (that is, take s_{i_ℓ} units of item i_ℓ for each $\ell \le k$, and take 0 unit of others).

3. Output $S_{Max} \in \text{argmax}_{v \in \mathcal{V}} V(S_v)$.

   (When there are several choices, the algorithm chooses one arbitrarily but always outputs the same one.)

We first prove a variant of Lemma 3.1 in [7].

Lemma 6. *For any input to A_{Max}, letting A^* be the optimal allocation under the same input, we have $V(S_{Max}) \ge \frac{V(A^*)}{8}$.*

Proof. Denoting A^* by $(a^*_1, \ldots, a^*_m)$, we have $A^* = \bigvee_{i \in [m]} a^*_i e_i$, and sub-additivity implies

$$V(A^*) \le \sum_{i \in [m]} V(a^*_i e_i).$$

Since $\sum_i a^*_i c_i \le B$, for each player i we have $a^*_i c_i \le B$, implying

$$a^*_i \le \min\{n_i, \lfloor \frac{B}{c_i} \rfloor\} = n'_i,$$

Mechanism M_{Rand}

1. Pick each player independently at random with probability 1/2 into group T, and let $T' = [m] \setminus T$.
2. Run A_{Max} with the set of players T, number of units n_i and cost c_i for each $i \in T$, budget B, and the demand oracle for valuation function V. Let v be the value of the returned allocation.
3. For k from 1 to $\sum_{i \in T'} n_i$,
   (a) Run A_{Max} with the set of players $T_k = \{i : i \in T', c_i \leq \frac{B}{k}\}$, number of units n_i and cost $\frac{B}{k}$ for each $i \in T_k$, budget B, and the demand oracle for V. Denote the returned allocation by $X = (x_1, \ldots, x_m)$, where $x_i = 0$ for each $i \notin T_k$.
   (b) If $V(X) \geq \frac{\log \log n}{64 \log n} \cdot v$, then output allocation X, pay $x_i \cdot \frac{B}{k}$ to each player i, and stop.
4. Output $A_\perp$ and pay 0 to each player.

and further implying $V(a_i^* e_i) \leq V(n_i' e_i) \leq V(n'_{i^{**}} e_{i^{**}})$. Accordingly,

$$V(A^*) \leq mV(n'_{i^{**}} e_{i^{**}}) = mv^*.$$

Since on the other hand we have $V(A^*) \geq V(n'_{i^{**}} e_{i^{**}}) = v^*$, there exists $v \in \mathcal{V}$ such that $\frac{V(A^*)}{2} \leq v \leq V(A^*)$.

Fixing such a v and letting $S = (s_1, \ldots, s_m)$ be the allocation returned by the demand oracle for v, we have

$$V(S) - \frac{v}{2B} \cdot \sum_{i \in [m]} s_i c_i \geq V(A^*) - \frac{v}{2B} \cdot \sum_{i \in [m]} a_i^* c_i \geq v - \frac{v}{2B} \cdot B = \frac{v}{2},$$

thus $V(S) \geq \frac{v}{2}$, and the algorithm will not output $A_\perp$. If $\sum_i s_i c_i \leq B$, then $S_v = S$, $V(S_{Max}) \geq V(S_v) \geq \frac{v}{2} \geq \frac{V(A^*)}{4}$, and we are done.

Assume now $\sum_i s_i c_i > B$. By the construction of S_v we have $\sum_{\ell \leq k} s_{i_\ell} c_{i_\ell} > \frac{B}{2}$. Letting $S'_v = \bigvee_{k+1 \leq \ell \leq m} s_{i_\ell} e_{i_\ell}$, we have $S = S_v \vee S'_v$, and thus $V(S) \leq V(S_v) + V(S'_v)$, which implies

$$V(S) - \frac{v}{2B} \cdot \sum_{i \in [m]} s_i c_i \leq V(S_v) - \frac{v}{2B} \cdot \sum_{\ell \leq k} s_{i_\ell} c_{i_\ell} + V(S'_v) - \frac{v}{2B} \cdot \sum_{k+1 \leq \ell \leq m} s_{i_\ell} c_{i_\ell}. \quad (8)$$

If $V(S_v) < \frac{v}{4}$, then

$$V(S) - \frac{v}{2B} \cdot \sum_{i \in [m]} s_i c_i < \frac{v}{4} - \frac{v}{2B} \cdot \frac{B}{2} + V(S'_v) - \frac{v}{2B} \cdot \sum_{k+1 \leq \ell \leq m} s_{i_\ell} c_{i_\ell} = V(S'_v) - \frac{v}{2B} \cdot \sum_{k+1 \leq \ell \leq m} s_{i_\ell} c_{i_\ell},$$

contradicting the fact that S is the optimal solution returned by the demand oracle. Thus

$$V(S_v) \geq \frac{v}{4} \geq \frac{V(A^*)}{8},$$

implying $V(S_{Max}) \geq \frac{V(A^*)}{8}$, as desired. $\square$

Below we prove our main theorem for sub-additive valuations. The main ideas follow from [7], but many new ideas are needed as well. In particular the proof of Lemma 8 requires novel ways of dealing with multi-unit allocations.

Theorem 4. *Mechanism M_{Sub} runs in polynomial time, is DST and individually rational, and is an $O(\frac{(\log n)^2}{\log\log n})$-approximation for procurement games with sub-additive valuations.*

Proof. The mechanism clearly runs in polynomial time. The proof that mechanism M_{Rand} is DST, individually rational, and budget feasible is almost the same as in [7], and thus we omit it here. By Lemma 5, mechanism M_{One} also satisfies all of those properties. Thus mechanism M_{Sub} is DST, individually rational, and budget feasible.

It remains to analyze the approximation ratio of M_{Sub}.

If $V(\lambda^{**}e_{i^{**}}) \geq \frac{V(A^*)}{2}$, then by Lemma 5 mechanism M_{One} already provides a $2(1+\ln n)$ approximation and we are done. From now on we assume $V(\lambda^{**}e_{i^{**}}) < \frac{V(A^*)}{2}$.

Consider the two player sets T and $T' = [m]\backslash T$ in Step 1 of M_{Rand}. Let $\hat{A}_T$ and $\hat{A}_{T'}$ respectively be the optimal allocation among the budget feasible ones that only take units from players in T and T'. We have the following lemma.

Lemma 7. *With probability at least* $1/4$*,*

$$V(\hat{A}_{T'}) \geq V(\hat{A}_T) \geq \frac{V(A^*)}{8}. \tag{9}$$

Proof. For any subset of players C, let A^*_C be A^* projected to C: letting $A^* = (a^*_1, \ldots, a^*_n)$, $A^*_C = \bigvee_{i \in C} a^*_i e_i$. If $C = \{i\}$, we write A^*_i instead of $A^*_{\{i\}}$. We show that there exists two disjoint player sets C_1, C_2 such that $C_1 \cup C_2 = [m]$,

$$V(A^*_{C_1}) \geq \frac{V(A^*)}{4}, \text{ and } V(A^*_{C_2}) \geq \frac{V(A^*)}{4}.$$

Similar to [7], we start with $C_1 = \emptyset$ and $C_2 = [m]$, and move players to C_1 one by one in an arbitrary order, until $V(A^*_{C_1}) \geq \frac{V(A^*)}{4}$. Letting i be the last player moved, we have $V(A^*_{C_1\backslash\{i\}}) < \frac{V(A^*)}{4}$. Since $C_1 \setminus \{i\}$ and $C_2 \cup \{i\}$ are two disjoint sets whose union is $[m]$, we have

$$A^* = A^*_{C_1\backslash\{i\}} \vee A^*_{C_2\cup\{i\}},$$

and sub-additivity implies $V(A^*) \leq V(A^*_{C_1\backslash\{i\}}) + V(A^*_{C_2\cup\{i\}})$.[7] Accordingly $V(A^*_{C_2\cup\{i\}}) > \frac{3V(A^*)}{4}$. Since A^* is budget feasible, the number of units i gets in A^* is at most $\min\{n_i, \lfloor\frac{B}{c_i}\rfloor\}$, and we have

$$V(A^*_i) \leq V(\min\{n_i, \lfloor\frac{B}{c_i}\rfloor\} \cdot e_i) \leq V(\lambda^{**}e_{i^{**}}) < \frac{V(A^*)}{2},$$

where the first inequality is because V is monotone and the second is by the definition of i^{**}. Since $i \notin C_2$, we have $A^*_{C_2\cup\{i\}} = A^*_{C_2} \vee A^*_i$, and again sub-additivity implies $V(A^*_{C_2\cup\{i\}}) \leq V(A^*_{C_2}) + V(A^*_i)$. Thus

$$V(A^*_{C_2}) \geq V(A^*_{C_2\cup\{i\}}) - V(A^*_i) > \frac{3V(A^*)}{4} - \frac{V(A^*)}{2} = \frac{V(A^*)}{4}.$$

[7] Notice that in single-unit settings partitioning the players is the same as partitioning the units, and given the optimal allocation S^* which is a subset of players, for any set $S \subseteq S^*$ we have $V(S^*) \leq V(S) + V(S^* \setminus S)$. But in our setting the partition has to be done in terms of players rather than units. Indeed, partitioning $A^* = (a^*_1, \ldots, a^*_m)$ into two arbitrary allocations $A = (a_1, \ldots, a_m)$ and $A' = (a'_1, \ldots, a'_m)$ with $a_i + a'_i = a^*_i$ for each i will not give us $A^* = A \vee A'$, since there may exist i such that both a_i and a'_i are strictly less than a^*_i. Only when A and A' are A^* projected to two disjoint player sets will one have $A^* = A \vee A'$ and thus $V(A^*) \leq V(A) + V(A')$.

Since $T \cap T' = \emptyset$ and $T \cup T' = [m]$, for each C_i we have $A^*_{C_i} = A^*_{C_i \cap T} \vee A^*_{C_i \cap T'}$, and thus $V(A^*_{C_i}) \leq V(A^*_{C_i \cap T}) + V(A^*_{C_i \cap T'})$. Accordingly, either $V(A^*_{C_i \cap T})$ or $V(A^*_{C_i \cap T'})$ is greater than $\frac{V(A^*)}{8}$. Since the players in C_1 and C_2 are partitioned to T and T' uniformly and independently, with probability at least 1/2 there exists i such that

$$V(A^*_{C_i \cap T}) \geq \frac{V(A^*)}{8} \text{ and } V(A^*_{C_{3-i} \cap T'}) \geq \frac{V(A^*)}{8}, \tag{10}$$

namely, the more valuable parts of C_1 and C_2 end up at different sides.

Since both allocations $A^*_{C_i \cap T}$ and $A^*_{C_{3-i} \cap T'}$ in Equation 10 are budget feasible, we have $V(\hat{A}_T) \geq V(A^*_{C_i \cap T})$ and $V(\hat{A}_{T'}) \geq V(A^*_{C_{3-i} \cap T'})$. Thus with probability at least 1/2, $V(\hat{A}_T) \geq \frac{V(A^*)}{8}$ and $V(\hat{A}_{T'}) \geq \frac{V(A^*)}{8}$. Because the role of T and T' can be switched, with probability 1/2 we have $V(\hat{A}_{T'}) \geq V(\hat{A}_T)$. Thus with probability at least 1/4 we have $V(\hat{A}_{T'}) \geq V(\hat{A}_T) \geq \frac{V(A^*)}{8}$, and Lemma 7 holds. $\square$

From now on we only consider scenarios where Inequality 9 holds. In this case, by Lemma 6, the value v computed in Step 2 of Mechanism M_{Rand} satisfies

$$v \geq \frac{V(\hat{A}_T)}{8} \geq \frac{V(A^*)}{64}. \tag{11}$$

Let A be the outcome of M_{Rand}. Different from [7], in our setting it is possible that $A = A_\perp$. However, we can prove the following lemma.

Lemma 8. *$V(A) + V(\lambda^{**} e_{i^{**}}) \geq \frac{\log \log n}{64 \log n} \cdot v$.*

Proof. To do so, we are going to partition $\hat{A}_{T'}$ into disjoint sets. But instead of partitioning according to the players as we have done in Lemma 7, this time we shall partition according to the units. Let $t = |T'|$, $\hat{a}_i$ be the number of units each player $i \in T'$ gets in $\hat{A}_{T'}$, and $n' = \sum_{i \in T'} \hat{a}_i$. Without loss of generality, assume $T' = \{1, 2, \ldots, t\}$ and $c_1 \geq c_2 \geq \cdots \geq c_t$.

Let L be the ordered list of player-unit pairs $(1,1), \ldots, (1, \hat{a}_1), (2,1), \ldots, (2, \hat{a}_2), \ldots, (t, 1), \ldots, (t, \hat{a}_t)$, and denote by (i_ℓ, j_ℓ) the ℓ-th pair in L, with $\ell \in [n']$. Similar to [7], we recursively partition the pairs in L into different groups as follows:

- Let α_1 be the largest integer such that $c_{i_1} \leq \frac{B}{\alpha_1}$. Put the first $\alpha'_1 = \min\{\alpha_1, n'\}$ pairs into group Z_1.
- Let $\beta_r = \alpha'_1 + \cdots + \alpha'_r$. If $\beta_r < n'$, then let α_{r+1} be the largest integer such that $c_{i_{\beta_r+1}} \leq \frac{B}{\alpha_{r+1}}$. Put the next $\alpha'_{r+1} = \min\{\alpha_{r+1}, n' - \beta_r\}$ pairs in group Z_{r+1}.

Let $x+1$ be the number of groups. For each $r \in [x+1]$, notice that Z_r naturally correspond to an allocation where each player i's number of units is the number of pairs of his in Z_r. Slightly abusing notation, we refer to this allocation as Z_r as well, and use $V(Z_r)$ to denote its value.

If $x = 0$, then there is only one group $Z_1 = L$. Thus $Z_1 = \hat{A}_{T'}$ and $V(Z_1) = V(\hat{A}_{T'})$. In round $k = \alpha_1$ of mechanism M_{Rand}, we have $T_k = T'$ since $c_i \leq c_1 \leq \frac{B}{\alpha_1}$ for each $i \in T'$. The optimal budget feasible allocation for T_k with unit-cost $\frac{B}{\alpha_1}$ has value at least $V(Z_1)$, because $|Z_1| \leq \alpha_1$, which makes Z_1 a budget feasible allocation under unit-cost $\frac{B}{\alpha_1}$. By Lemma 6, in this round we have

$$V(X) \geq \frac{V(Z_1)}{8} = \frac{V(\hat{A}_{T'})}{8} \geq \frac{\log \log n}{8 \log n} \cdot v,$$

where the last inequality is because v is the value of a budget feasible allocation for players in T, which implies $V(\hat{A}_T) \geq v$, and thus $V(\hat{A}_{T'}) \geq v$ by Inequality 9. Thus the mechanism, which may terminate before or at round α_1, will output an allocation A such that $V(A) \geq \frac{\log\log n}{64\log n} \cdot v$, and Lemma 8 holds.

If $x > 1$, notice that for any $1 \leq r < x$ there is at most one player whose pairs appear in both Z_r and Z_{r+1}: he is the last one picked up by Z_r and the first by Z_{r+1}. Denote this player by j_r.[8] We have

$$\hat{A}_{T'} = Z_1 \vee \hat{a}_{j_1} e_{j_1} \vee Z_2 \vee \hat{a}_{j_2} e_{j_2} \vee \cdots \vee \hat{a}_{j_x} e_{j_x} \vee Z_{x+1},$$

where $\hat{a}_{j_r} e_{j_r}$ is defined to be $A_\perp$ if there is no such a player j_r between some Z_r and Z_{r+1}. This is because, for any player in T', either all his units taken by $\hat{A}_{T'}$ appear in some Z_r, or he is player j_r for some r and all his units appear in $\hat{a}_{j_r} e_{j_r}$. By sub-additivity,

$$V(\hat{A}_{T'}) \leq \sum_{r\in[x+1]} V(Z_r) + \sum_{r\in[x]} V(\hat{a}_{j_r} e_{j_r}) \leq \sum_{r\in[x+1]} V(Z_r) + xV(\lambda^{**} e_{i^{**}}), \tag{12}$$

where the second inequality is because $\hat{A}_{T'}$ is budget feasible, and thus for each $i \in T'$ we have $\hat{a}_i c_i \leq B$, implying $\hat{a}_i \leq \min\{n_i, \lfloor \frac{B}{c_i} \rfloor\}$.

Letting $r^* \in \mathrm{argmax}_{r\in[x+1]} V(Z_r)$, by Inequality 12 we have

$$V(\hat{A}_{T'}) \leq xV(\lambda^{**} e_{i^{**}}) + (x+1)V(Z_{r^*}).$$

By a similar argument as in [7], we have that $n' \geq \left(\frac{x}{2}\right)^x$, which implies $n \geq \left(\frac{x}{2}\right)^x$, and thus $x \leq \frac{2\log n}{\log\log n}$. Accordingly,

$$V(\lambda^{**} e_{i^{**}}) + V(Z_{r^*}) \geq \frac{V(\hat{A}_{T'})}{x+1} \geq \frac{V(\hat{A}_{T'})}{2x} \geq \frac{\log\log n}{4\log n} \cdot V(\hat{A}_{T'}).$$

If $V(\lambda^{**} e_{i^{**}}) \geq \frac{\log\log n}{8\log n} \cdot V(\hat{A}_{T'})$, then Lemma 8 holds immediately, again because $V(\hat{A}_{T'}) \geq v$. Otherwise, we have $V(Z_{r^*}) \geq \frac{\log\log n}{8\log n} \cdot V(\hat{A}_{T'})$. In round $k = \alpha_{r^*}$ of mechanism M_{Rand}, T_k includes all players whose pairs appear in Z_{r^*}, and thus the optimal budget feasible allocation for T_k with unit-cost $\frac{B}{\alpha_{r^*}}$ is at least $V(Z_{r^*})$, because $|Z_{r^*}| \leq \alpha_{r^*}$, which makes Z_{r^*} a budget feasible allocation for T_k with unit-cost $\frac{B}{\alpha_{r^*}}$. By Lemma 6, the allocation X in this round satisfies

$$V(X) \geq \frac{V(Z_{r^*})}{8} \geq \frac{\log\log n}{64\log n} \cdot V(\hat{A}_{T'}) \geq \frac{\log\log n}{64\log n} \cdot v.$$

Thus the mechanism, which may terminate before or at round α_{r^*}, will output an allocation A such that $V(A) \geq \frac{\log\log n}{64\log n} \cdot v$, and Lemma 8 holds. □

Combining Lemma 7, Inequality 11 and Lemma 8, we have that the expected value of mechanism M_{Sub} is

$$\frac{1}{4} \cdot \left(\frac{V(A)}{2} + \frac{V(\lambda^{**} e_{i^{**}})}{2(1+\ln n)} \right) \geq \frac{V(A) + V(\lambda^{**} e_{i^{**}})}{8(1+\ln n)} \geq \frac{\log\log n}{8(1+\ln n)\cdot 64\log n \cdot 64} \cdot V(A^*) = \frac{V(A^*)}{O(\frac{(\log n)^2}{\log\log n})}.$$

Thus mechanism M_{Sub} is an $O(\frac{(\log n)^2}{\log\log n})$-approximation, and Theorem 4 holds. □

[8] In principle it is possible that a player's units spread among several consecutive groups $Z_r, Z_{r+1}, Z_{r+2}, \ldots$. In this case any group in the middle contains only pairs of this player, and $j_r = j_{r+1} = \ldots$. This will not affect our analysis.

5.4 Approximation when the players have small costs

Notice that the worst case of the approximation ratio above comes from the case where $V(\lambda^{**}e_{i^{**}})$ is the main contribution to the final value. An extra $\log n$ factor is introduced compared with single-unit settings since the optimal approximation ratio for $V(\lambda^{**}e_{i^{**}})$ is $O(\log n)$. For scenarios where the players' costs are very small compared with the budget, in particular, where $n_i c_i \leq B$ for each i, the optimal single-item allocation (i^{**}, λ^{**}) is publicly known, just as the player i^* in single-unit settings. Thus the mechanism M_{One} in our mechanism can be replaced by "allocating $n_{i^{**}}$ units of item i^{**} and paying him B", and the $\log n$ factor is avoided. Indeed, we have the following theorem, whose proof we omit.

Theorem 5. *For sub-additive valuations where $n_i c_i \leq B$ for each i, there exists a DST mechanism which is individually rational, budget feasible, and is an $O(\frac{\log n}{\log\log n})$ approximation.*

Such a *small-cost* setting is possible in some markets, but is not realistic in many others. For example, in the Provision-after-Wait problem in healthcare as discussed in our introduction, it is very unlikely that all patients can be served at the most expensive hospital without exceeding the government's budget. Also in many procurement auctions, a seller, as the manufacture of his product, can be considered as having infinite supply. Thus the total cost of all units he can provide will always exceed the buyer's budget. Thus one need to be careful about where the small-cost condition applies, and in our main results we focus on the general settings where we do not rely on such a condition.

References

[1] Gagan Aggarwal and Jason D. Hartline. Knapsack auctions. In *SODA*, pages 1083–1092, 2006.

[2] Aaron Archer and Éva Tardos. Truthful mechanisms for one-parameter agents. In *Proceedings of the 42Nd IEEE Symposium on Foundations of Computer Science*, FOCS '01, pages 482–, Washington, DC, USA, 2001. IEEE Computer Society.

[3] Aaron Archer and Éva Tardos. Frugal path mechanisms. *ACM Trans. Algorithms*, 3(1):3:1–3:22, February 2007.

[4] Moshe Babaioff, Michael Dinitz, Anupam Gupta, Nicole Immorlica, and Kunal Talwar. Secretary problems: Weights and discounts. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 1245–1254, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics.

[5] Moshe Babaioff, Nicole Immorlica, and Robert Kleinberg. Matroids, secretary problems, and online mechanisms. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 434–443, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[6] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Yaron Singer. Learning on a budget: Posted price mechanisms for online procurement. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, pages 128–145, New York, NY, USA, 2012. ACM.

[7] Xiaohui Bei, Ning Chen, Nick Gravin, and Pinyan Lu. Budget feasible mechanism design: From prior-free to bayesian. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 449–458, New York, NY, USA, 2012. ACM.

[8] Mark Braverman, Jing Chen, and Sampath Kannan. Optimal provision-after-wait in healthcare. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, ITCS'14, pages 541–542, New York, NY, USA, 2014. ACM.

[9] Matthew C. Cary, Abraham D. Flaxman, Jason D. Hartline, and Anna R. Karlin. Auctions for structured procurement. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, pages 304–313, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[10] Ning Chen, Edith Elkind, Nick Gravin, and Fedor Petrov. Frugal mechanism design via spectral techniques. In *FOCS*, pages 755–764, 2010.

[11] Ning Chen, Nick Gravin, and Pinyan Lu. Mechanism design without money via stable matching. *CoRR*, abs/1104.2872, 2011.

[12] Ning Chen, Nick Gravin, and Pinyan Lu. On the approximability of budget feasible mechanisms. In *Proceedings of the Twenty-second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 685–699. SIAM, 2011.

[13] Shahar Dobzinski. Two randomized mechanisms for combinatorial auctions. In *Proceedings of the 10th International Workshop on Approximation and the 11th International Workshop on Randomization, and Combinatorial Optimization. Algorithms and Techniques*, APPROX '07/RANDOM '07, pages 89–103, Berlin, Heidelberg, 2007. Springer-Verlag.

[14] Shahar Dobzinski, Christos H. Papadimitriou, and Yaron Singer. Mechanisms for complement-free procurement. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, EC '11, pages 273–282, New York, NY, USA, 2011. ACM.

[15] Edith Elkind, Amit Sahai, and Ken Steiglitz. Frugality in path auctions. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '04, pages 701–709, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.

[16] Joan Feigenbaum, Christos H. Papadimitriou, Rahul Sami, and Scott Shenker. A bgp-based mechanism for lowest-cost routing. In *PODC*, pages 173–182, 2002.

[17] Joan Feigenbaum, Rahul Sami, and Scott Shenker. Mechanism design for policy routing. In *PODC*, pages 11–20, 2004.

[18] Andrew V. Goldberg, Jason D. Hartline, Anna R. Karlin, Andrew Wright, and Michael Saks. Competitive auctions. In *Games and Economic Behavior*, pages 72–81, 2002.

[19] Michael Kapralov, Ian Post, and Jan Vondrák. Online submodular welfare maximization: Greedy is optimal. In *SODA*, pages 1216–1225, 2013.

[20] Anna R. Karlin, David Kempe, and Tami Tamir. Beyond vcg: Frugality of truthful mechanisms. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '05, pages 615–626, Washington, DC, USA, 2005. IEEE Computer Society.

[21] David Kempe, Mahyar Salek, and Cristopher Moore. Frugal and truthful auctions for vertex covers, flows and cuts. In *FOCS*, pages 745–754, 2010.

[22] Silvano Martello and Paolo Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Inc., New York, NY, USA, 1990.

[23] Debasis Mishra and Dharmaraj Veeramani. Vickrey-dutch procurement auction for multiple items. *European Journal of Operational Research*, 180:617–629, 2006.

[24] Roger B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[25] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.

[26] David C. Parkes and Jayant Kalagnanam. Models for iterative multiattribute procurement auctions. *Management Science (Special Issue on Electronic Markets)*, 51:435–451, 2005.

[27] Yaron Singer. Budget feasible mechanisms. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, FOCS '10, pages 765–774, Washington, DC, USA, 2010. IEEE Computer Society.

[28] Kunal Talwar. The price of truth: Frugality in truthful mechanisms. In *Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science*, STACS '03, pages 608–619, London, UK, UK, 2003. Springer-Verlag.