Truthful Multi-unit Procurements with Budgets

Hau Chan and Jing Chen

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

Abstract. We study procurement games where each seller supplies multiple units of his item, with a cost per unit known only to him. The buyer can purchase any number of units from each seller, values different combinations of the items differently, and has a budget for his total payment. For a special class of procurement games, the *bounded knapsack* problem, we show that no universally truthful budget-feasible mechanism can approximate the optimal value of the buyer within $\ln n$, where n is the total number of units of all items available. We then construct a polynomial-time mechanism that gives a $4(1+\ln n)$-approximation for procurement games with *concave additive valuations*, which include bounded knapsack as a special case. Our mechanism is thus optimal up to a constant factor. Moreover, for the bounded knapsack problem, given the well-known FPTAS, our results imply there is a provable gap between the optimization domain and the mechanism design domain. Finally, for procurement games with *sub-additive valuations*, we construct a universally truthful budget-feasible mechanism that gives an $O(\frac{\log^2 n}{\log \log n})$-approximation in polynomial time with a demand oracle.

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

1 Introduction

In a procurement game/auction, m sellers compete for providing their items (referred to as products or services in some 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 buyer may purchase any number of units from each seller. For example, a local government may buy 50 displays from Dell, 20 laptops from Lenovo, and 30 printers from HP.[1] The buyer has a valuation function for possible combinations of the items, and a budget B for the total payment he can make. We consider universally truthful mechanisms that (approximately) maximize the buyer's value subject to the budget constraint.

Procurement games with budgets have been studied in the framework of budget-feasible mechanisms (see, e.g., [35, 19, 17, 10]). Yet most studies focus

[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. However, we allow cases where different sellers have the same item, just as one can buy the same laptops from Best Buy and/or Walmart.

on settings where each seller has only one unit of his item. Thus there are only two possible allocations for a seller: either his item is taken or it is not.[2] When a seller has multiple units 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. To the best of our knowledge, this is the first time where multi-unit budget-feasible mechanisms are systematically studied.

Multi-unit procurements with budgets can be used to model many interesting problems. For example, in the classic *bounded knapsack* problem 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 buys. In job scheduling, the planner may assign multiple jobs to a machine, with different values for different assignments. As another example, in the Provision-after-Wait problem in healthcare [11], the government needs to serve n patients at m hospitals. Each patient has his own value for being served at each hospital, and the value of the government is the social welfare.

1.1 Our main results

We present our main results in three parts. Due to lack of space, most of the proofs are provided in the full version of this paper [14].

An impossibility result. Although budget-feasible mechanisms with constant approximation ratios have been constructed for single-unit procurements [35, 17], our first result, formally stated and proved in Section 3, shows that this is impossible in multi-unit settings, even for the special case of bounded knapsack.

Theorem 1. (rephrased) *No universally truthful, budget-feasible mechanism can do better than a* $\ln n$*-approximation for bounded knapsack, where* n *is the total number of units of all items available.*

This theorem applies to all classes of multi-unit procurement games considered in this paper, since they all contain bounded knapsack as a special case.

An optimal mechanism for concave additive valuations. A concave additive valuation function is specified by the buyer's marginal values for getting the j-th unit of each item i, v_{ij}, which are non-increasing in j. The following theorem is formally stated in Section 4.

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

Our mechanism is very simple. The central part is a greedy algorithm, which yields a monotone allocation rule. However, one needs to be careful about how to compute the payments, and new ideas are needed for proving budget-feasibility.

Since bounded knapsack is a special case of concave 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.

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

Beyond concave additive valuations. We do not know how to use greedy algorithms to construct budget-feasible mechanisms for larger classes of valuations. The reason is that *they may not be monotone:* if a player lowers his cost, he might actually sell fewer units. This is demonstrated by our example in Section 5.1. Thus we turn to a different approach, *random sampling* [10, 24, 18, 16, 6, 7]. The following theorem is formally stated in Section 5.3.

THEOREM 5. (rephrased) *Given a demand oracle, there is a polynomial-time mechanism which is an $O(\frac{\log^2 n}{\log\log n})$-approximation for sub-additive valuations.*

A demand oracle is a standard assumption for handling sub-additive valuations [19, 10, 8], since such a valuation function takes exponentially many numbers to specify. Notice that for bounded knapsack and concave 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 consistent with the literature.

Our mechanism generalizes that of [10], which gives an $O(\frac{\log n}{\log\log n})$-approximation for single-unit sub-additive valuations. Several new issues arise in the multi-unit setting. For example, we must distinguish between an item and a unit of that item, and in both our mechanism and our analysis we need to be careful about which one to deal with. Also, we have constructed, as a sub-routine, a mechanism for approximating the optimal *single-item outcome*: namely, an outcome that only takes units from a single seller. We believe that this mechanism will be a useful building block for budget-feasible mechanisms in multi-unit settings.

1.2 Related work

Various procurement games have been studied [30, 33, 32, 21, 22], but without budget considerations. In particular, frugal mechanisms [5, 13, 20, 27, 36, 15, 28] aim at finding mechanisms that minimize the total payment. As a "dual" problem to procurement games, auctions where the buyers have budget constraints have also been studied [2, 23], but the models are very different from ours.

Single-unit budget-feasible mechanisms were introduced by [35], where the author achieved a constant approximation for sub-modular valuations. In [17] the approximation ratio was improved and variants of knapsack problems were studied, but still in single-unit settings. In [19] the authors considered 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. We notice that their randomized mechanism can be generalized to multi-unit settings, resulting in an $O(\log^3 n)$-approximation. In [10] the authors consider both prior-free and Bayesian models. For the former, they provide a constant approximation for XOS valuations and an $O(\frac{\log n}{\log\log n})$-approximation for sub-additive valuations; and for the latter, they provide a constant approximation for the sub-additive case. As mentioned we generalize their prior-free mechanism, but we need to give up a $\log n$ factor in the approximation ratio. It is nice to see that the framework of budget-feasible mechanism design generalizes to multi-unit settings.

In [25] the author considered settings where each seller has multiple items. Although it was discussed why such settings are harder than single-item settings, no explicit upper bound on the approximation ratio was given. Instead, the focus there was a different benchmark. The author provided a constant approximation of his benchmark for sub-modular valuations, but the mechanism does not run in polynomial time. Also, budget-feasible mechanisms where each seller has one unit of an infinitely divisible item have been considered in [3], under the *large-market* assumption: that is, the cost of buying each item completely is much smaller than the budget. The authors constructed a deterministic mechanism which is a $1 - 1/e$ approximation for additive valuations and which they also prove to be optimal. In our study we do not impose any assumption about the sellers costs, and the cost of buying all units of an item may or may not exceed the budget. Moreover, in [9] the authors studied online procurements and provided a randomized posted-price mechanism that is an $O(\log n)$-approximation for sub-modular valuations under the random ordering assumption.

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 the model is very different from ours and from those studied in the budget-feasibility framework in general.

1.3 Open problems

Many questions can be asked about multi-unit procurements with budgets and are worth studying in the future. Below we mention a few of them.

First, it would be interesting to close the gap between the upper bound in Theorem 1 and the lower bound in Theorem 5, even for subclasses such as sub-modular or diminishing-return valuations, as defined in Section 2. A related problem is whether the upper bound can be bypassed under other solution concepts. For example, is there a mechanism with price of anarchy [29, 34] better than $\ln n$? How about a mechanism with a *unique* equilibrium? Solution concepts that are not equilibrium-based are also worth considering, such as undominated strategies and iterated elimination of dominated strategies. Another problem is whether a better approximation can be achieved for other benchmarks, such as the one considered in [25], by truthful mechanisms that run in polynomial time.

Second, online procurements with budget constraints have been studied in both optimization settings [26] and strategic settings [9]. But only single-unit scenarios are considered in the latter. It is natural to ask, what if a seller with multiple units of the same item can show up at different time points, and the buyer needs to decide how many units he wants to buy each time.

Finally, the buyer may have different budgets for different sellers, a seller's cost for one unit of his item may decrease as he sells more, or the number of units each seller has may not be publicly known[3]. However, the last two cases are not single-parameter settings and presumably need very different approaches.

[3] In many real-life scenarios the numbers of available units are public information, including procurements of digital products, procurements of cars, some arms trades, etc. Here procurement auctions are powerful tools and may result in big differences in prices, just like in the car market. However, there are also scenarios where the sellers can hide the numbers of units they have, particularly in a seller's market. In such cases they may manipulate the supply level, hoping to affect the prices.

2 Procurement Games

Now let us define our model. In a procurement game there are m *sellers* who are the players, and one *buyer*. 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 number of units of all the items is $n \triangleq \sum_i n_i$. The *true cost* for providing one unit of item i is $c_i \geq 0$, and $c = (c_1, \ldots, c_m)$. The value of c_i is player i's private information. All other information is public.

An *allocation* A is a profile of integers, $A = (a_1, \ldots, a_m)$. For each $i \in [m]$, $a_i \in \{0, 1, \ldots, n_i\}$ and a_i denotes the number of units bought from player i. An *outcome* ω is 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 payment to player i. Player i's *utility* at ω is $u_i(\omega) = P_i - a_i c_i$.

The buyer has a *valuation function* V, mapping allocations to non-negative reals, such that $V(0, \ldots, 0) = 0$. For allocations $A = (a_1, \ldots, a_m)$ and $A' = (a'_1, \ldots, a'_m)$ with $a_i \leq a'_i$ for each i, $V(A) \leq V(A')$ —namely, V is *monotone*.[4] The buyer has a *budget* B and wants to implement an *optimal allocation*,

$$A^* \in \operatorname*{argmax}_{A: \sum_{i \in [m]} c_i a_i \leq B} V(A),$$

while keeping the *total payment* within the budget. An outcome $\omega = (A, P)$ is *budget-feasible* if $\sum_{i \in [m]} P_i \leq B$.

The solution concept. A deterministic revealing mechanism is *dominant-strategy truthful (DST)* if for each player i, announcing c_i is a dominant strategy:

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

A deterministic mechanism is *individually rational* if $u_i(c) \geq 0$ for each i. A randomized mechanism is *universally truthful* (respectively, *individually rational*) if it is a probabilistic distribution over deterministic mechanisms that are DST (respectively, individually rational).

A deterministic DST mechanism is *budget-feasible* if its outcome under c is budget-feasible. A universally truthful mechanism is *budget-feasible (in expectation)* if the expected payment under c is at most B.

Definition 1. *Let $\mathcal{C}$ be a class of procurement games and $f(n) \geq 0$. A universally truthful mechanism is an* $f(n)$-approximation *for class $\mathcal{C}$ if, for any game in $\mathcal{C}$, the mechanism is individually rational and budget-feasible, and the outcome under the true cost profile c has expected value at least $\frac{V(A^*)}{f(n)}$.*

Remark 1. One can trade truthfulness for budget-feasibility: given a universally truthful budget-feasible mechanism, by paying each player the expected payment he would have received, we get a mechanism that is *truthful in expectation* and meets the budget constraint *with probability 1*. As implied by Theorem 1, no universally truthful mechanism that meets the budget constraint with probability 1 can do better than a $\ln n$-approximation. Thus there has to be some trade-off.

[4] Monotonicity is a standard assumption for single-unit budget-feasible mechanisms.

Remark 2. We allow different players to have identical items, just like different dealers may carry the same products, with or without the same cost. But we require the same player's units have the same cost. In the future, one may consider cases where one player has units of different items with different costs: that is, a multi-parameter setting instead of single-parameter.

Below we define several classes of valuation functions for procurement games.

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

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

Sub-additive valuations. A much larger class is the *sub-additive* valuations. Here a valuation V is such that, for any $A = (a_1, \ldots, a_m)$ and $A' = (a'_1, \ldots, a'_m)$,

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

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

Notice that the requirement of sub-additivity is imposed only across different players, and values can change arbitrarily across units of the same player. Indeed, when A and A' differ at a single player, sub-additivity *does not impose any constraint* on $V(A)$ and $V(A')$, not even that there are decreasing margins. Thus this definition is more general than requiring sub-additivity also across units of the same player. Following the literature, we stick to the more general notion.

Between concave additivity and sub-additivity, two classes of valuations have been defined, as recalled below.[5] To the best of our knowledge, no budget-feasible mechanisms were considered for either of them in multi-unit settings.

- *Diminishing return*: for any 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.[6]

[5] The literature of multi-unit procurements has been particularly interested in valuations with some forms of "non-increasing margins", thus has considered classes that contain all concave additive valuations but not necessarily all additive ones.

[6] The item-wise max and min operations when defining sub-additivity and submodularity follow directly from the set-union and set-intersection operations when defining them in general settings, and have been widely adopted in the literature (see, e.g., [12] and [26]). One may consider alternative definitions where, for example, $\vee$ represents item-wise sum rather than item-wise max. However, we are not aware of existing studies where the alternative definitions are used.

Diminishing return implies sub-modularity, and both collapse to sub-modularity in single-unit settings. The reason for diminishing return to be considered separately is that multi-unit sub-modularity is a very weak condition: when A and A' differ at a single player, it does not impose any constraint, as sub-additivity. Diminishing return better reflects the idea behind single-unit sub-modularity: the buyer's value for one extra unit of any item gets smaller as he buys more.

Since the valuation classes defined above are nested:

$$\begin{aligned}&\text{bounded knapsack } \subseteq \text{concave additivity } \subseteq \text{ diminishing return}\\ \subseteq\ &\text{sub-modularity } \subseteq \text{ sub-additivity,}\end{aligned}$$

any impossibility result for one class applies to all classes above it, and any positive result for one class applies to all classes below it. Moreover, since sub-additivity contains additivity, any positive result for the former also applies to the latter.

Demand oracle. A sub-additive valuation function V may take exponentially many numbers to specify. Thus following the studies of single-unit sub-additive valuations [35, 10], we consider a *demand oracle*, which takes as input a set of players $\{1, \ldots, m\}$, a profile of costs $(p_1, \ldots, p_m)$ and a profile of numbers of units $(n_1, \ldots, n_m)$,[7] and returns, regardless of the budget, an allocation

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

It is well known that a demand oracle can simulate in polynomial time a value oracle, which returns $V(A)$ given A. Thus we also have access to a value oracle.

Our goal. We shall construct universally truthful mechanisms that are individually rational, budget-feasible, and approximate the optimal value of the buyer. Our mechanisms run in polynomial time for concave 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 [4]. Following Myerson's lemma [31] or the characterization in [4], the only truthful mechanisms are those with a monotone allocation rule and 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 of his units bought by the mechanism.

With a budget constraint, 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 also (2) the unique total payment that it induces must satisfy the budget constraint. Therefore, similar to 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.

[7] In single-unit settings a demand oracle takes as input a set of players and the costs. For multi-unit settings it is natural to also include the numbers of units.

3 Impossibility results for bounded knapsack

The following observation for bounded knapsack is immediate.

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

Proof. When $m = 1$, $n_1 = n$, $v_1 = 1$ and $c_1 = B$, a DST mechanism, being an n-approximation, must buy 1 unit and pay the player B. When $c_1 = B/n$, the mechanism must still buy 1 unit and pay B, otherwise the player will bid B instead. Thus the mechanism's value is 1, while the optimal value is n. □

Clearly, buying 1 unit from a player $i \in \mathrm{argmax}_j\, v_j$ and paying him B is an n-approximation. For randomized mechanisms we have the following.

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

Proof. 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 the player's expected utility by bidding b when $c_1 = c$. For each $k \in [n]$, consider the bid $\frac{B}{k}$: let P^k be the expected payment and, for each $j \in [n]$, let p^k_j be the probability for the mechanism to buy j units. When $c_1 = \frac{B}{k}$, the optimal value is k and

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

as the mechanism is an $f(n)$-approximation. By universal truthfulness and individual rationality, $u_1(\frac{B}{k}; \frac{B}{k}) \geq u_1(\frac{B}{k-1}; \frac{B}{k})$ $\forall k > 1$ and $u_1(B; B) \geq 0$. Namely,

$$P^k - \frac{B}{k}\sum_{j\in[n]} p^k_j \cdot j \geq P^{k-1} - \frac{B}{k}\sum_{j\in[n]} p^{k-1}_j \cdot j \quad \forall k > 1, \quad \text{and}$$

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

Summing up these n inequalities, we have

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

which implies

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

By Equation 1, we have

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

By budget-feasibility, $P^n \leq B$. Thus $f(n) \geq \ln n$, implying Theorem 1. □

Remark 3. Notice that as long as the mechanism is truthful *in expectation* and individually rational *in expectation* (namely, with respect to the players' expected utilities), the analysis of Theorem 1 implies that it cannot do better than a $\ln n$-approximation. Also notice that the impossibility result does not impose any constraint on the running time of the mechanism.

Remark 4. When there is a single player, that player has a monopoly and it is not too surprising that no mechanism can do better than a $\ln n$-approximation. For example, in frugality mechanism design in procurement games, it has been explicitly assumed that there is no monopoly. However, when monopoly might actually exist, it is interesting to see that there is a tight bound (by Theorems 1 and 2) on the power of budget-feasible mechanisms in multi-unit settings.

4 An optimal mechanism for concave additive valuations

We construct a polynomial-time universally truthful mechanism M_{Add} that is a $4(1 + \ln n)$-approximation for procurement games with concave additive valuations. Our mechanism is very simple, and the basic idea is a greedy algorithm with proportional cost sharing, as has been used for single-unit settings [17, 35]. 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 the units with value 0 from consideration. Because we shall show that M_{Add} is universally truthful, we describe it only with respect to the truthful bid $(c_1, \ldots, c_n)$. Also, we describe the allocation rule only, since it uniquely determines the threshold payments. An algorithm for computing the thresholds will be given in the analysis. Finally, let $i^* \in \operatorname{argmax}_i v_{i1}$ be the player with the highest marginal value, e_{i^*} be the allocation with 1 unit of item i^* and 0 unit of others, and $A_\perp = (0, \ldots, 0)$ be the allocation where nothing is bought. We have the following.

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

Combining Theorems 1 and 2 we immediately have the following.

Corollary 1. *Mechanism M_{Add} is optimal up to a constant factor among all universally truthful, individually rational, and budget-feasible mechanisms for multi-unit procurement games with concave additive valuations.*

Mechanism M_{Add} for Concave 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 $\frac{c_{i_k}}{v_{i_k j_k}} \le \frac{B}{\sum_{\ell \le k} v_{i_\ell j_\ell}}$.
   - (c) Pick up the first k pairs in the list: that is, output allocation $A = (a_1, \ldots, a_n)$ where $a_i = |\{\ell : \ell \le k \text{ and } i_\ell = i\}|$.

Remark 5. 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, and our mechanism provides such an approximation.

Furthermore, for bounded knapsack, without strategic considerations there is an FPTAS, while with strategic considerations the best is a $\ln n$-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.

Finally, it would be interesting to see how the constant gap between Theorems 1 and 2 can be closed, and whether there is a mechanism that meets the budget constraint with probability 1 and achieves an $O(\log n)$-approximation.

An optimal mechanism for symmetric valuations. A closely related class of valuations are the *symmetric* ones: there exists $v_1, \ldots, v_n$ such that, for any allocation A with k units, $V(A) = \sum_{\ell \le k} v_\ell$. In general, symmetric valuations are not concave additive, nor are concave additive valuations necessarily symmetric. But they are equivalent with a single seller. Thus the proof of Theorem 1 implies no mechanism can do better than a $\ln n$-approximation for symmetric valuations, as stated in the first part of the theorem below. Similar to our analysis of Theorem 2, one can verify that the following mechanism is a $4(1+\ln n)$-approximation for symmetric valuations: it is the same as M_{Add} except in Step 2, where k is set to be the largest number in $[n]$ satisfying $c_{i_k} \le \frac{B}{k}$. We omit the analysis since it is very similar to that of M_{Add}, and only present the following theorem.

Theorem 3. *For symmetric valuations, no universally truthful mechanism can be an $f(n)$-approximation with $f(n) < \ln n$, and there exists a polynomial-time universally truthful 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 with proportional cost-sharing played an important role in budget-feasible mechanisms, we do not know how to use it for multi-unit sub-additive valuations, since *it is not monotone.* Indeed, by lowering his cost, a player i will still sell his first unit as in the old 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. Under the new ordering there is no guarantee whether player i will sell any of his remaining units. Below we give an example demonstrating this phenomenon 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, we work through the algorithm and define the marginal values on the way. The valuation function will be defined accordingly.

Given any allocation A and player i, denote by $V(i|A)$ the marginal value of item i, 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 unavailable now), $V(2|A_1) = 5+5\epsilon$, 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, the final allocation is A_3, and player 2 sells 2 units.

Now let $c_2' = 1-\epsilon < c_2$. 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 sells 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'. But player 2 only sells one unit, violating monotonicity.

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

$$V(0,0,0) = 0, V(1,0,0) = 10, V(0,1,0) = 10+\epsilon, V(0,0,1) = 10-\epsilon,$$
$$V(1,1,0) = 15+5\epsilon, V(1,0,1) = 15-\epsilon, V(0,2,0) = 15-4\epsilon, V(0,1,1) = 15+6\epsilon,$$
$$V(0,0,2) = 15,$$
$$V(1,2,0) = 16+6\epsilon, V(1,1,1) = 16+4\epsilon, V(1,0,2) = 16, V(0,2,1) = 16+5\epsilon,$$
$$V(0,1,2) = 16+7\epsilon,$$
$$V(0,2,2) = V(1,2,1) = V(1,1,2) = 16+7\epsilon, V(1,2,2) = 16+7\epsilon.$$

One can verify that V is consistent with the marginal values and has diminishing returns. Indeed, for any allocation with k units for k from 0 to 4, the marginal value of adding 1 more unit is roughly $10, 5, 1, \epsilon, 0$, and thus diminishing.

Given the non-monotonicity of the greedy algorithm, we turn to another approach for constructing truthful mechanisms, namely, *random sampling.* We provide our main mechanism in Section 5.3. In Section 5.2 we first construct a mechanism that will be used as a subroutine.

5.2 Approximating the optimal single-item allocation

From the analysis of Theorem 1, we notice that part of the hardness in designing mechanisms for multi-unit settings comes from cases where a single player's item contributes a lot to the optimal solution. In order to obtain a good approximation, we need to identify such a player and buy *as many units as possible from him.* More precisely, 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 buy $\lambda^{**} \triangleq \min\{n_{i^{**}}, \lfloor \frac{B}{c_{i^{**}}} \rfloor\}$ units from i^{**} and pay him (at most) B. We shall refer to (i^{**}, λ^{**}) as the *optimal single-item allocation.*

Notice that a similar scenario occurs in single-unit settings: part of the value approximation comes from a single player i^* with the highest marginal value. 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 their bids. Below we construct a universally truthful mechanism, M_{One}, which is budget-feasible and approximates $V(\lambda^{**} e_{i^{**}})$ within a $1 + \ln n$ factor. We have the following theorem.

Mechanism M_{One} for Approximating the Optimal Single-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 $\ell \leq k$ and $\theta_\ell = \frac{B}{\ell}$ for each $k + 1 \leq \ell \leq \lambda^{**}$.
4. Output allocation $\lambda^{**} e_{i^{**}}$ and pay $\sum_{\ell \leq \lambda^{**}} \theta_\ell$ to player i^{**}.

Theorem 4. *Mechanism M_{One} is universally truthful, individually rational, budget-feasible, and 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 universally truthful, individually rational, and budget-feasible mechanisms.*

Remark 6. As it will become clear from the analysis, M_{One} does not require the valuation to be sub-additive. The only thing it requires is that, for each player i, $V(\lambda e_i)$ is non-decreasing in λ. Thus it can be used for valuations that are not even monotone, as long as they are monotone across units of the same item.

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 single-unit 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 A truthful mechanism for sub-additive valuations

Our mechanism for sub-additive valuations generalizes that of [10]. In particular, the algorithm A_{Max} and the mechanism M_{Rand} below are respectively variants of their algorithm SA-ALG-MAX and mechanism SA-RANDOM-SAMPLE. Several new issues arise in multi-unit settings. For example, we must now distinguish between an item and a unit of that item. In the mechanism and its analysis, we sometimes deal with an item —thus all of its units at the same time— and sometimes deal with a single unit. Also, as discussed in Section 5.2, the role of player i^* with the highest marginal value is replaced by player i^{**}, and the way i^{**} contributes to the value approximation has changed a lot —this is where the extra $\log n$ factor comes. Indeed, to construct and analyze our mechanism one need good understanding of the problem in multi-unit settings. Our mechanism M_{Sub} is a uniform distribution between M_{Rand} and the mechanism M_{One} of Section 5.2. We have the following theorem.

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 $i$, $i^{**} = \mathrm{argmax}_i V(n'_i e_i)$, $v^* = V(n'_{i^{**}} e_{i^{**}})$, and
   $\mathcal{V} = \{v^*, 2v^*, \ldots, mv^*\}$.
2. For $v \in \mathcal{V}$ from $mv^*$ to $v^*$,
   (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
       $S = (s_1, \ldots, s_m) \in \arg\max_{A=(a_1,\ldots,a_m): a_i \leq n'_i \forall i} V(A) - \sum_{i \in [m]} a_i p_i$.
       (When there are multiple optimal solutions, the oracle always returns the same
       one whenever queried with the same instance.)
   (b) Set allocation $S_v = A_\perp$. (Recall $A_\perp = (0, \ldots, 0)$ represents buying nothing.)
   (c) If $V(S) < \frac{v}{2}$, then continue to the next $v$.
   (d) Else, order the players according to $s_i c_i$ decreasingly with ties broken lexico-
       graphically, and denote them by $i_1, \ldots, i_m$.
       Let $k$ be the largest number in $[m]$ satisfying $\sum_{\ell \leq k} s_{i_\ell} c_{i_\ell} \leq B$, and let $S_v$ be
       $S$ projected on $\{i_1, \ldots, i_k\}$: $S_v = \bigvee_{\ell \leq k} s_{i_\ell} e_{i_\ell}$, namely, $S_v$ consists of taking $s_{i_\ell}$
       units of item $i_\ell$ for each $\ell \leq k$, and taking 0 unit of others.
3. Output $S_{Max} \in \mathrm{argmax}_{v \in \mathcal{V}} V(S_v)$.
   (When there are several choices, the algorithm chooses one arbitrarily, but always
   outputs the same one when executed multiple times with the same input.)
```

Mechanism M_{Rand}

1. Put 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.

Theorem 5. *Mechanism M_{Sub} runs in polynomial time, is universally truthful, and is an $O(\frac{(\log n)^2}{\log \log n})$-approximation for procurement games with sub-additive valuations.*

Since diminishing return, sub-modularity, and additivity all imply sub-additivity, we immediately have the following.

Corollary 3. *M_{Sub} is an $O(\frac{(\log n)^2}{\log \log n})$-approximation for procurement games with diminishing returns, those with sub-modular valuations, and those with additive valuations.*

Remark 7. The worst case of the approximation above comes from cases where $V(\lambda^{**} e_{i^{**}})$ (and thus M_{One}) is the main contribution to the final value. Unlike single-unit settings, we need an additional $\log n$ factor because the optimal approximation ratio for $V(\lambda^{**} e_{i^{**}})$ is $O(\log n)$. For scenarios where the players' costs are very small, 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. In such a *small-cost* setting, which is very similar to the *large-market* setting considered by [3] except that the items here are not infinitely divisible, the subroutine M_{One} in M_{Sub} can be replaced by "allocating $n_{i^{**}}$ units of item i^{**} and paying him B", and the $\log n$ factor is avoided, resulting in an $O(\frac{\log n}{\log \log n})$-approximation.

A small-cost setting is possible in some markets, but it is not realistic in many others. For example, in the Provision-after-Wait problem in healthcare [11], it is very unlikely that all patients can be served at the most expensive hospital within the government's budget. Also, in many procurement games, a seller, as the manufacture of his product, can be considered as having infinite supply, and the total cost of all units he has will always exceed the buyer's budget. Thus one need to be careful about where the small-cost condition applies.

Acknowledgements

We thank several anonymous reviewers for their comments. The first author is supported by the NSF Graduate Research Fellowship.

Bibliography

[1] Aggarwal, G., Hartline, J.D.: Knapsack auctions. In: SODA. pp. 1083–1092 (2006)

[2] Aggarwal, G., Muthukrishnan, S., Pál, D., Pál, M.: General auction mechanism for search advertising. In: Proceedings of the 18th international conference on World Wide Web. pp. 241–250 (2009)

[3] Anari, N., Goel, G., Nikzad, A.: Mechanisms design for crowdsourcing: An optimal $1 - 1/e$ approximate budget-feasible mechanism for large markets. In: FOCS (2014)

[4] Archer, A., Tardos, E.: Truthful mechanisms for one-parameter agents. In: Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science. pp. 482–491 (2001)

[5] Archer, A., Tardos, E.: Frugal path mechanisms. ACM Trans. Algorithms 3(1), 1–22 (2007)

[6] Babaioff, M., Dinitz, M., Gupta, A., Immorlica, N., Talwar, K.: Secretary problems: Weights and discounts. In: Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 1245–1254 (2009)

[7] Babaioff, M., Immorlica, N., Kleinberg, R.: Matroids, secretary problems, and online mechanisms. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 434–443 (2007)

[8] Badanidiyuru, A., Dobzinski, S., Oren, S.: Optimization with demand oracles. In: EC. pp. 110–127 (2012)

[9] Badanidiyuru, A., Kleinberg, R., Singer, Y.: Learning on a budget: Posted price mechanisms for online procurement. In: Proceedings of the 13th ACM Conference on Electronic Commerce. pp. 128–145 (2012)

[10] Bei, X., Chen, N., Gravin, N., Lu, P.: Budget feasible mechanism design: From prior-free to bayesian. In: Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing. pp. 449–458 (2012)

[11] Braverman, M., Chen, J., Kannan, S.: Optimal provision-after-wait in healthcare. In: Proceedings of the 5th Conference on Innovations in Theoretical Computer Science. pp. 541–542 (2014)

[12] Calinescu, G., Chekuri, C., Pál, M., Vondrák, J.: Maximizing a monotone submodular function subject to a matroid constraint. SICOMP, special section on STOC'08 40(6), 1740–1766 (2011)

[13] Cary, M.C., Flaxman, A.D., Hartline, J.D., Karlin, A.R.: Auctions for structured procurement. In: Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 304–313 (2008)

[14] Chan, H., Chen, J.: Truthful multi-unit procurements with budgets (2014), full version, avaiable at `http://arxiv.org/abs/1409.7595`

[15] Chen, N., Elkind, E., Gravin, N., Petrov, F.: Frugal mechanism design via spectral techniques. In: FOCS. pp. 755–764 (2010)

[16] Chen, N., Gravin, N., Lu, P.: Mechanism design without money via stable matching. CoRR abs/1104.2872 (2011)

[17] Chen, N., Gravin, N., Lu, P.: On the approximability of budget feasible mechanisms. In: Proceedings of the Twenty-second Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 685–699 (2011)

[18] Dobzinski, S.: Two randomized mechanisms for combinatorial auctions. In: Proceedings of APPROX'07/RANDOM'07. pp. 89–103 (2007)

[19] Dobzinski, S., Papadimitriou, C.H., Singer, Y.: Mechanisms for complement-free procurement. In: Proceedings of the 12th ACM Conference on Electronic Commerce. pp. 273–282 (2011)

[20] Elkind, E., Sahai, A., Steiglitz, K.: Frugality in path auctions. In: Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 701–709 (2004)

[21] Feigenbaum, J., Papadimitriou, C.H., Sami, R., Shenker, S.: A bgp-based mechanism for lowest-cost routing. In: PODC. pp. 173–182 (2002)

[22] Feigenbaum, J., Sami, R., Shenker, S.: Mechanism design for policy routing. In: PODC. pp. 11–20 (2004)

[23] Goel, G., Mirrokni, V., Leme, R.P.: Clinching auctions beyond hard budget constraints. In: EC (2014)

[24] Goldberg, A.V., Hartline, J.D., Karlin, A.R., Wright, A., Saks, M.: Competitive auctions. In: Games and Economic Behavior. pp. 72–81 (2002)

[25] Gravin, N.: Incentive Compatible Design of Reverse Auctions, *Section 5.2.* Ph.D. thesis, Nanyang Technological University (2013)

[26] Kapralov, M., Post, I., Vondrák, J.: Online submodular welfare maximization: Greedy is optimal. In: Proceedings of SODA'13. pp. 1216–1225 (2013)

[27] Karlin, A.R., Kempe, D., Tamir, T.: Beyond vcg: Frugality of truthful mechanisms. In: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science. pp. 615–626 (2005)

[28] Kempe, D., Salek, M., Moore, C.: Frugal and truthful auctions for vertex covers, flows and cuts. In: FOCS. pp. 745–754 (2010)

[29] Koutsoupias, E., Papadimitriou, C.H.: Worst-case equilibria. In: Proceedings of the 16th annual conference on Theoretical Aspects of Computer Science (STACS'99). pp. 404–413 (1999)

[30] Mishra, D., Veeramani, D.: Vickrey-dutch procurement auction for multiple items. European Journal of Operational Research 180, 617–629 (2006)

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

[32] Nisan, N., Ronen, A.: Algorithmic mechanism design. Games and Economic Behavior 35, 166–196 (2001)

[33] Parkes, D.C., Kalagnanam, J.: Models for iterative multiattribute procurement auctions. Management Science (Special Issue on Electronic Markets) 51, 435–451 (2005)

[34] Roughgarden, T., Tardos, E.: How bad is selfish routing? Journal of the ACM 49(2), 236–259 (2002)

[35] Singer, Y.: Budget feasible mechanisms. In: Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science. pp. 765–774 (2010)

[36] Talwar, K.: The price of truth: Frugality in truthful mechanisms. In: Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS'03). pp. 608–619 (2003)