Overview*
On the Complexity of Simple and Optimal Deterministic Mechanisms for an Additive Buyer

SiNan Du†

November 23, 2021

contents

1 Introduction **3**
- 1.1 Structure . . . 3
- 1.2 Results . . . 3

2 Preliminaries **3**
- 2.1 Set-up . . . 3
- 2.2 Concepts . . . 4
- 2.3 Integer Linear Program . . . 5

3 IID with Support Size 2 **5**

4 #P-Hardness **8**
- 4.1 Construction . . . 9
- 4.2 Two Candidate Solution . . . 9
- 4.3 Proof Sketchy . . . 10

5 Constant Number of Items **11**
- 5.1 Set-up . . . 11
- 5.2 Solved in polynomial time . . . 11

6 Conclusions **12**

*This article is produced as a piece of overview of Chen, Xi and Matikas, George and Paparas, Dimitris and Yannakakis, Mihalis (2018). On the Complexity of Simple and Optimal Deterministic Mechanisms for an Additive Buyer.

†SiNan Du is with the Department of Computer Engineering, Beijing Institute of Technology, China, e-mail:1120193007@bit.edu.cn. Please contact me if any question in your convenience.

Abstract

I'm going to introduce their work [1] from two aspects: positive side and negative side. On the positive side, we show that when the distribution are i.i.d.with support size 2, the optimal revenue obtainable by any mechanism, even a randomized one, can be achieved by a simple solution of individual item pricing with a discounted price for the grand bundle and furthermore, it can be computed in polynomial time. Then we will demonstrate that the problem can be solved in polynomial time too when the number of items is constant. On the negative side, we show that the Revenue-Optimal Deterministic Mechanism Design problem for a single additive buyer is #P-hard, even when the distributions have support size 2 for each item and, more importantly, even when the optimal solution is guaranteed to be of a very simple solution of the above kind. They introduce to us that the following problems are also #P-hard, as immediate corollaries of the proof:

1. determining if individual item pricing is optimal for a given instance,
2. determining if grand bundle pricing is optimal, and
3. computing the optimal (deterministic) revenue.

1 Introduction

1.1 Structure

In this section, I will introduce you to the main structure of this report.

Section 1.2 summarizes the viewpoints of their work which are elaborated in the primitive paper and we will retell and demonstrate them in detail thus composing the backbone of this report. In Section 2, I intend to describe the set-up of the problem mainly in deterministic mechanisms and bring you back to some basic concepts of the problem. In Section 3, the detailed demonstration of Theorem 1.1 is exhibited along with many lemmas and proofs. In Section 4, I prove Theorem 1.2 and 1.3 with the plan to reduce from the #P-hard decision problem called COMP introduced in [2]. In Section 5, I will introduce you to the proof of Theorem 1.4. Last, we make a conclusion to the main components of their work in section 6.

1.2 Results

Theorem 1.1. *For IID distributions of support 2, the optimal revenue (even among randomized solutions) can be achieved by a discounted item pricing (i.e., single item prices and price for grand bundle), and it can be computed in polynomial time.*

Theorem 1.2. *The optimal deterministic pricing problem is #P-hard, even if all distributions have support 2, and if the optimal is guaranteed to have very simple form (we call it "discounted item pricing"): single item prices and price for grand bundle, buyer can buy any subset for sum of its item prices or the grand bundle at its price. It's also #P-hard to compute the optimal revenue.*

Theorem 1.3. *It is #P-hard to determine for a given instance.*
- if single item pricing is optimal,
- if grand bundle pricing is optimal.

Theorem 1.4. *For constant number of items (and any independent distributions), the problem can be also solved in polynomial time.*

2 Preliminaries

2.1 Set-up

In our problem, seller has n items for sale and buyer has private value for each item. Probability distribution of value for each item is known to seller thus valuation of buyer drawn randomly from $F = F_1 \times F_2 \times \cdots \times F_n$. Seller can assign a price to each subset or offer a menu of only some subsets (bundles), naturally, buyer buys subset S with maximum utility $u(S) = value(S) - price(S)$, if ≥ 0.

Optimal Pricing (Revenue Maximization) Problem is concerned with finding pricing that maximizes the expected revenue:

$$maxE[Revenue] = \sum_{v \sim F} Pr(v) \cdot price(S_v)$$

where S_v = bundle bought by buyer with valuation v.

2.2 Concepts

Single Item Pricing Scheme

In this pricing scheme, seller set a price for each item, then the price for each subset S: $\sum\{price(i)|i \in S\}$. Reflecting on the content learned in lecture, we know the optimal price for each item i: value p_i^* that maximizes $p_i^* \cdot Pr[values(i) \geq p_i^*]$ given by Myerson's Lemma [3].

Grand Bundle Pricing Scheme

In this pricing scheme, buyers can only buy the set of all items (the grand bundle) for a given price, or nothing at all. There are examples where it gets more revenue than single item pricing.

Partition Pricing Scheme

In this pricing scheme, seller partitions the items into groups and assign price to each group in partition thus buyers can buy any set of groups for sum of their prices. We can conclude that this scheme includes single item and grand bundle pricing as special cases and can get more revenue than both in some examples.

Randomized Scheme (Lottery Pricing)

Lottery = vector$(q_1, \cdots, q_n)$ of probabilities for the items if buyer buys the lottery then she gets each item i with probability q_i. Lottery pricing provides menu = set of (lottery, price) pairs and buyer buys lottery with maximum expected utility. There are examples where lottery pricing gives more revenue than the optimal deterministic pricing. However, deterministic mechanisms (bundle-pricings) are much more widely used in practice; they focus on deterministic mechanisms in their work.

Pricing Schemes $\longleftrightarrow$ Mechanism Design

- Buyer submits a bid for each item.
- Mechanism determines allocation the buyer receives and price she pays. Mechanism must be incentive compatible and individually rational.
- Bundle pricing $\longleftrightarrow$ deterministic mechanism

- Lottery pricing $\longleftrightarrow$ randomized mechanism

Thus, two central questions remain concerning the bundle-pricing (or optimal deterministic mechanism design) problem:

1. Is there an efficient algorithm that finds an optimal bundle-pricing?
2. If the problem above is hard in general, is there such an algorithm when the instance is promised to have a simple optimal bundle-pricing?

In their work, their results resolve both questions.

2.3 Integer Linear Program

- Let D_i = support of F_i and $D = D_1 \times \cdots \times D_n$ (exponential size)
- Variables: $x_{v,1}, \cdots, x_{v,n} \in \{0,1\}, \pi_v, \quad \forall v \in D$
- $(x_{v,1}, \cdots, x_{v,n})$ = characteristic vector of bundle bought for valuation v, π_v its price

$$max \sum_{v \in D} \pi_v \cdot Pr[v]$$

$$s.t. \quad x_{v,i} \in \{0,1\}, \quad \forall v \in D$$

$$\sum_{i \in [n]} v_i \cdot x_{v,i} - \pi_v \geq 0, \quad \forall v \in D$$

$$\sum_{i \in [n]} w_i \cdot x_{w,i} - \pi_w \geq \sum_{i \in [n]} w_i \cdot x_{v,i} - \pi_v \quad \forall w, v \in D$$

(w does not envy the bundle of v)

- The LP($x_{v,i} \in [0,1]$) models the optimal lottery problem.

3 IID with Support Size 2

We establish Theorem 1.1 in this section. Let $F_1, \cdots, F_n$ be i.i.d. distributions with support size 2. Without loss of generality we can assume that the support is $\{1, b\}$ with $b > 1$. (If the support is $\{0, b\}$ the problem is trivial: the optimal revenue can be achieved by offering every item at price b; if the support is $\{a, b\}$ with $0 < a < b$, then we can equivalently rescale it to$\{1, b/a\}$.) Let $p \in (0,1)$ be the probability that an item takes value b, and $1-p$ that it takes 1. We let P_i denote the probability of $v \sin F$ having i items at value b and $n-i$ at 1, for each $i \in [0:n]$. That is,

$$P_i = \binom{n}{i} \cdot p^i \cdot (1-p)^{n-i}$$

The following lemma for $P_i's$ is crucial. We prove it in proof (3).

Lemma 3.1. *There exists an integer $k \in [0:n]$ such that*

$$(n-i)P_i - (b-1)(P_{i+1} + \cdots + P_n) \tag{1}$$

is negative for all $i : 0 \le i < k$ and is nonnegative for all $i : k \le i \le n$.

Let $k \in [0:n]$ be in integer that satisfies Lemma 3.1, which is unique and can be computed in polynomial time. We use $S^*$to denote the following discounted item-pricing:

The grand bundle $[n]$is offered at $kb + n - k$ and each item is offered individually at b(the latter means that the buyer can buy any bundle $T \subseteq [n]$ at prices $|T|b$).
Given S^*, the behavior of the buyer is as follows. If a valuation vector has k or more items at b then the buyer buys the grand bundle at $kb+n-k$; otherwise it buys all the items that have value b. The expected revenue R^* of the discounted item-pricing S^* is then

$$R^* = \sum_{1 \le i \le k} b_i P_i + (kb + n - k) \sum_{k \le i \le n} P_i$$

It is clear that given k, R^* can be computed in polynomial time.

To finish the proof of Theorem 1.1, we show that S^* achieves the optimal revenue $Rev(F)$.

Lemma 3.2. *$R^* = Rev(F)$ when $F_1, \cdots, F_n$ are i.i.d. with support size 2 and k satisfies Lemma 3.1.*

We are now ready to prove Lemma 3.2.

Proof of Lemma 3.2. Since R^* is the expected revenue of S^*, it suffices to show $Rev(F) \le R^*$.
Expected revenue of S^* is $R^* = \sum_{1 \le i < k} bi \cdot P_i + (kb + n - k) \sum_{k \le i \le n} P_i$

- Since IID, the LP for the optimal lottery has a symmetric optimal solution [4], and the LP can be simplified to a more compact symmetric LP.
- Variables: $x_i, i = 1, \cdots, n$: probability of getting a value b item when the valuation i has i items at b.
  $y_i, i = 1, \cdots, n-1$: probability of getting a value 1 item when the valuation i has i items at b.
  $\pi_i, i = 0, \cdots, n$: prices of lottery for a valuation with i items at b.
- The symmetric LP maximizes $\sum_{i=0}^{n} \pi_i \cdot P_i$
  Relax the LP by keeping only some of the constraints:
  1. $0 \le x_i \le 1$ and $0 \le y_i \le 1$ for all i
  2. $\pi_0 \le ny_0$ (the utility of the all-1 valuation is ≥ 0)

3. For each $i \in [n]$, the valuation w with $w_j = b$ for $j \leq i$ and $w_j = 1$ for $j > i$ does not envy the lottery of the valuation v with $v_j = b$ for $j \leq i-1$ and $w_j = 1$ for $j > i-1$

$$bix_i + (n-i)y_i - \pi_i \geq b(i-1)x_{i-1} + (n-i+b)y_{i+1} - \pi_{i-1}$$

- Combine the inequalities to upper bound every π_i in terms of the x, y variables

$$\pi_0 \leq ny_0$$
$$\pi_i \leq bix_i + (n-i)y_i - (b-1)(y_{i-1} + y_{i-2} + \cdots + y_1 + y_0)$$

Replacing in the objective function every π_i by its upper bound

- Coefficient of x_i is $biP_i > 0 \Rightarrow$ expression maximized if $x_i > 1$
- Coefficient of y_i is $(n-i)P_i - (b-1)(P_{i+1} + \cdots + P_n)$, which is ¡ 0 if $i < k$, and ≥ 0 if $i > k \Rightarrow$ expression maximized if we set $y_i = 0$ for all $i < k$ and $y_i = 1$ for all $i \geq k$
- Substituting these values in the expression that upper bounds the objective function gives precisely R^*:

$$\begin{aligned}
&\sum_{1\leq i\leq n} bi \cdot P_i + \sum_{k\leq i\leq n-1} [(n-i)P_i - (b-1)(P_{i+1} + \cdots + P_n)] \\
&= \sum_{1\leq i\leq n} bi \cdot P_i + \sum_{k\leq i\leq n} [(n-i)P_i - (b-1)(P_{i+1} + \cdots + P_n)] \\
&= \sum_{1\leq i\leq n} bi \cdot P_i + \sum_{k\leq i\leq n} P_i[(n-i) - (b-1)(i-k)] \\
&= \sum_{1\leq i<k} bi \cdot P_i + \sum_{k\leq i\leq n} P_i \cdot [n + (b-1)k] \\
&= R^*
\end{aligned}$$

This finishes the proof of the Lemma 3.2. To prove the Lemma 3.1, we start with the following Lemma. □

Lemma 3.3. *For all $i = 1, \cdots, n-1$, $P_i(n-i) + \sum_{j\geq i+1} P_j(n - \frac{i}{p}) > 0$.*

Proof of Lemma 3.3. We use induction on $n - i$.
Basis: $n - i = 1$, i.e. $i = n - 1$. The left-hand side is

$$P_{n-1} + P_n(n - \frac{n-1}{p}) = n \cdot p^{n-1} \cdot (1-p) + p^n \cdot (n - \frac{n-1}{p}) = p^{n-1} > 0$$

Induction Step: We have $P_i(n-i) = P_{i+1} \cdot (i+1) \cdot \frac{1-p}{p}$ according the Binomial theorem expansion. Therefore, the left-hand side is equal to

$$\begin{aligned}
LHS &= P_{i+1} \cdot (i+1) \cdot \frac{1-p}{p} + P_{i+1} \cdot (n - \frac{i}{p}) + \sum_{j \geq i+2} P_j (n - \frac{i}{p}) \\
&= P_{i+1}(n-(i+1)) + P_{i+1} \cdot \frac{1}{p} + \sum_{j \geq i+2} P_j (n - \frac{i}{p}) \\
&> P_{i+1}(n-(i+1)) + \sum_{j \geq i+2} P_j (n - \frac{i+1}{p}) \\
&> 0
\end{aligned}$$

where the last inequality holds by the induction hypothesis. $\square$

Now we are ready to prove Lemma 3.1:

Proof of Lemma 3.1. We let k be the smallest $i \in [0:n]$ such that (1) is nonnegative. To prove that (1) is nonnegative for all $i \geq k$, it suffices to show that

$$\frac{(n-i)P_i}{\sum_{k \geq i+1} P_j}$$

is monotonically increasing for i from 0 to $n-1$. Fix an $i \in [n-1]$. Our goal is to show that

$$\frac{(n-i)P_i}{\sum_{j \geq i+1} P_j} - \frac{(n-(i-1))P_{i-1}}{\sum_{j \geq i} P_j} > 0$$

or equivalently, $(n-i)P_i \sum_{j \geq i} P_j - (n-(i-1))P_{i-1} \sum_{j \geq i+1} P_j > 0$. Since

$$(n-(i-1))P_{i-1} = iP_i \cdot \frac{1-p}{p},$$

we can rewrite the left-hand side as

$$P_i \left[(n-i) \sum_{j \geq i} P_j - i \frac{1-p}{p} \sum_{j \geq i+1} P_j \right] = P_i \left[(n-i)P_i + \sum_{j \geq i+1} P_j (n - (i/p)) \right],$$

which is positive Lemma by Lemma 3.3. Therefore, the desired inequality holds and k is unique according to the monotonicity. $\square$

4 #P-Hardness

We prove Theorem 1.2 and 1.3 in this section. We are going to reduce from the following #P-hard decision problem called COMP introduced in [5].
Input:

1. Set B of integers $0 < b_1 \leq b_2 \leq \cdots \leq b_n \leq 2^n$

2. Subset $W \subset [n]$ of size $|W| = n/2$. Let $w = \sum_{i \in W} b_i$
3. Integer t

Is the number of subsets $S \subset [n]$ of size $|S| = n/2$ such that $\sum_{i \in S} b_i \geq w$ at least t ? While COMP was shown to be #P-hard in [5], we need here the same problem with the following two extra conditions on the two input sets B and W, which we will refer to as COMP*. These extra conditions will come in handy in the reduction below.

4.1 Construction

We now present the reduction from COMP* to the optimal bundle-pricing problem.

- $n+1$ items: n items $\leftrightarrow$ b_i's + special
- First n items: almost iid with support{1, big}
  Item i:
  value 1 with probability $p = \frac{1}{2(h+1)}$, where $h = 2^{2n}$;
  value $h + 1 + b_i\delta$ with probability $1 - p$, where $\delta = \frac{1}{2^{3n}}$
- Item $n+1$: support $\{\sigma, \sigma + \alpha\}$, where $\sigma = \frac{1}{p^n}$, $\alpha = \frac{nh}{2} + w\delta (<< \sigma)$
  value σ with probability $\frac{\alpha}{\sigma+\alpha} + \epsilon$ for some $\epsilon = \epsilon(t) = o(\frac{1}{\sigma})$;
  value $\sigma + \alpha$ with probability $\frac{\sigma}{\sigma+\alpha} - \epsilon$ (= almost 1)

4.2 Two Candidate Solution

In this section, we define two simple bundle-pricings (Solution 1 and 2), as feasible solutions to the standard Integer Program for the optimal expected revenues achievable by a bundle-pricing, both of which are discounted item-pricings.

- Solution 1: Grand bundle at price $n+\sigma$ = sum of low values. Equivalently, single item pricing with all prices = low values.
- Solution 2: Discounted item pricing where all item prices = high values, and grand bundle price = $n + \sigma + \alpha$.

Recall Theorem 1.2 and 1.3: One of these two solutions is the unique optimal solution. #P-hard to tell which one of the two. Namely, we can acquire corollaries:

1. #P-hard to tell if single item pricing is optimal.
2. #P-hard to tell if grand bundle pricing is optimal.

4.3 Proof Sketchy

In this section, where we show that whenever $|\epsilon| = o(1/\sigma)$, one of these two solutions is the unique optimal solution to the standard IP (Integer Programming) and achieves the optimal expected revenues achievable by a bundle-pricing. This is done by relaxing the standard IP and showing that one of these two solutions is the unique optimal solution to the relaxed IP. As they are both feasible to the standard IP, we conclude that one of them is uniquely optimal for the standard IP. So we are ready to prove Theorem 1.2 and 1.3.

- Integer Linear Program, using the allocation variables $x_{v,i}$ and utility variables u_v instead of price variables π_v

$$(u_v = \sum_{i \in [n]} v_i \cdot x_{v,i} - \pi_v)$$

- Denote a valuation by (S, σ) (or $(S, \sigma + \alpha)$), for $S \subseteq [n]$ if S = set of first n items that have high value and $(n+1)_{th}$ item has value σ (or $\sigma + \alpha$)

- In solution 1, all variables $x_{v,i} = 1$

  For $v = (S, \sigma)$, $u_v = \sum_{i \in S} h_i$

  For $v = (S, \sigma + \alpha)$, $u_v = \alpha + \sum_{i \in S} h_i$

- In solution 2:

  1. if $v = (S, \sigma + \alpha)$, all $x_{v,i} = 1$, $u_v = \sum_{i \in S} h_i$
  2. if $v = (S, \sigma)$ and $\sum_{i \in S} h_i \geq \alpha$ then all $x_{v,i} = 1$, $u_v = \sum_{i \in S} h_i - \alpha$
  3. if $v = (S, \sigma)$ and $\sum_{i \in S} h_i < \alpha$ then $x_{v,i} = 1$ for all $i \in S$, $x_{v,i}$ for all $i \notin S$ and for $i = n + 1$, and $u_v = 0$

  Then We can conclude:

  - Every S with $|S| > n/2$ satisfies case 2,
  - every S with $|S| < n/2$ satisfies case 3,
  - a set S with $|S| = n/2$ satisfies case 2 if $\sum_{i \in S} b_i \geq w$ and case 3 otherwise

- Relaxed ILP - keep only a subset of the envy constraints

  - $(S, \sigma + \alpha)$ does not envy $(\emptyset, \sigma + \alpha)$, for all $S \neq \emptyset$,
  - $(\emptyset, \sigma + \alpha)$ does not envy (S, σ), and vice-versa, for all $S \subseteq [n]$,
  - for all $T \subset S \subset [n]$, (S, σ) does not envy (T, σ)

- Long sequence of lemmas shows that the optimal solution to the relaxed ILP must be either Solution 1 or Solution 2. For $v = (\emptyset, \sigma + \alpha)$,

– if $x_{v,n+1} = 0$ then it must be Solution 1,

– if $x_{v,n+1} = 1$ then it must be Solution 2

At this point, we have finished demonstrating the Theorem 1.2 and 1.3.

5 Constant Number of Items

In this section we prove Theorem 1.4.

5.1 Set-up

- number of items = k = constant, support size m for each item
- V = set of m^k possible valuation vectors (polynomial)
- $d = 2^k$ possible bundles (constant)

The standard IP in this case has a polynomial number of variables and constraints. However, Integer Programming is NP-hard, so we will use a different method to solve the problem in polynomial time. For the rest of this section, we assume two arbitrary orderings, one for the valuation vectors and one for the bundles, and we will use v_j to denote the ith valuation vector, and B_j to denote the jth bundle and p_j to denote its price.

We will argue that we can generate in polynomial time a set of price vectors p that includes an optimal one; we can compute the expected revenue for each of these vectors and pick the best one. To these end, we consider a partitioning of the d-dimensional space of possible price vectors p into cells, such that for all p in the same cell, the buyer has the same behavior for every v_i,i.e., buys the same bundle, if any. Consider the following set H of hyperplanes over p.

5.2 Solved in polynomial time

- Space R^d_+ of possible price vectors p for the bundles partitioned by hyperplanes into cells such that $\forall$ cell C, $\forall$ valuation v buys the same bundle for all $p \in C$.
- Hyperplanes:

  $\forall v \in V, \forall j \in [d]$: $\sum_{l \in B_j} v_l - p_j = 0$

  $\forall v \in V, \forall j, j's \in [d]$: $\sum_{l \in B_j} v_l - p_j = \sum_{l \in B_{j's}} v_l - p_j$

  $\forall j, j's \in [d]$: $p_j = p_{j'}$

- The supremum revenue for price vectors in C is given by an LP, and is achieved at a vertex of C
  $\Rightarrow$ Optimum overall is achieved at a vertex of the subdivision.

- Polynomial number of hyperplanes, constant dimension of d
  $\Rightarrow$ polynomial number of vertices.

- Try them all and pick the best.

6 Conclusions

In this work, we studied the optimal bundle-pricing problem (or equivalently, the Revenue-Optimal Deterministic Mechanism Design problem). We showed that the optimal (deterministic) pricing problem is hard, and this holds even when the optimal solution is very simple: simple item pricing + discount for grand bundle.

And we showed that there is no 'nice' (easy-to-check) characterization of when separate item pricing, or grand bundling extracts the maximum revenue achievable by any bundle-pricing.

On the positive side, we showed that i.i.d distributions with support size 2, the maximum revenue achievable by any lottery pricing can always be achieved by a discounted item-pricing, and we can compute it in polynomial time. The problem can be also solved in polynomial time for a constant number of items.

References

[1] Chen, Xi and Matikas, George and Paparas, Dimitris and Yannakakis, Mihalis (2018).On the Complexity of Simple and Optimal Deterministic Mechanisms for an Additive Buyer. In: Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA). Society for Industrial and Applied Mathematics. 2036–2049.

[2] X. Chen, I. Diakonikolas, A. Orfanou, D. Paparas, X. Sun, and M. Yannakakis, On the complexity of optimal lottery pricing and randomized mechanisms, Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Sciences, 2015.

[3] R.B. Myerson, Optimal auction design, Mathematics of Operations Research 6 (1981), no. 1, 58–73.

[4] C. Daskalakis and S.M. Weinberg, Symmetries and optimal multi-dimensional mechanism design, Proceedings of the 13th ACM Conference on Electronic Commerce, 2012, pp. 370–387.

[5] X. Chen, I. Diakonikolas, A. Orfanou, D. Paparas, X. Sun, and M. Yannakakis, On the complexity of optimal lottery pricing and randomized mechanisms, Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Sciences, 2015.