

Revenue Maximizing Envy-free Fixed-price Auctions with Budgets

Riccardo Colini-Baldeschi¹, Stefano Leonardi¹,
Piotr Sankowski², and Qiang Zhang²

¹ Sapienza University of Rome, Italy

² University of Warsaw, Poland

1 Preliminaries

An auction with budgets is denoted by $\mathcal{A} = \langle n, m, \mathbf{M}, \mathbf{S}, \mathbf{v}, \mathbf{b} \rangle$. There are n bidders and m different goods. $\mathbf{M} = \langle m_1, \dots, m_m \rangle$ indicates the amount of supply of each good. For every $j \in \{1, \dots, m\}$, there are m_j copies of good j . $\mathbf{S} = \langle S_1, \dots, S_n \rangle$ indicates the *preference set* of each bidder. For every $i \in \{1, \dots, n\}$, S_i is the subset of goods in which bidder i is interested. Finally, $\mathbf{v} = \langle v_1, \dots, v_n \rangle$ indicates the valuation of each bidder. For every $i \in \{1, \dots, n\}$, bidder i has a valuation of v_i for each copy of goods in his preference set S_i . Given price p , the demand of a bidder i is defined as:

$$D_i(p) = \begin{cases} \min(\lfloor \frac{b_i}{p} \rfloor, \sum_{j \in S_i} m_j), & \text{if } v_i \geq p \\ 0, & \text{if } v_i < p \end{cases}$$

When there is no ambiguity on the price, we simply use D_i instead of $D_i(p)$.

A mechanism maps every auction \mathcal{A} to an outcome $\langle \mathbf{X}, p \rangle$, where $\mathbf{X} = \langle \mathbf{X}_1, \dots, \mathbf{X}_n \rangle$ is an allocation mapping goods to bidders and $p \in R$ is a fixed price for every goods. Note that for every $i \in \{1, \dots, n\}$, $\mathbf{X}_i = \langle x_i^1, \dots, x_i^m \rangle$ where bidder i gets x_i^j copies of good j . An outcome $\langle \mathbf{X}, p \rangle$ must satisfy all of the following conditions:

1. *limited supply*: for every $j \in \{1, \dots, m\}$, it holds $\sum_{i \in \{1, \dots, n\}} x_i^j \leq m_j$;
2. *bidder rationality*: for every $i \in \{1, \dots, n\}$, and every $j \in \{1, \dots, m\}$, if $x_i^j > 0$ then $j \in S_i$ and $v_i \geq p$;
3. *budget constraint*: for every $i \in \{1, \dots, n\}$, it holds $p \sum_{j=1}^m x_i^j \leq b_i$.

Given outcome $\langle \mathbf{X}, p \rangle$, the utility of bidder i is defined as:

$$u_i(X_i, p) = (v_i - p) \sum_{j \in S_i} x_i^j.$$

The revenue $r(\mathbf{X}, p)$ of the auction is the total payments of bidders, i.e., $r(\mathbf{X}, p) = p \sum_{i \in \{1, \dots, n\}, j \in S_i} x_i^j$. When the price and allocation are clear in the context, the revenue is simply denoted by r .

Given price p , we partition bidders as follows. Bidders with $v_i = p$ are called *value-limited* bidders, denoted by $VL_p = \{i \in \{1, \dots, n\} | v_i = p\}$. Similarly, bidders with $v_i > p$ are called *non-value-limited* bidders, denoted by $NVL_p = \{i \in \{1, \dots, n\} | v_i > p\}$. Finally, bidders with $v_i < p$ are called *exited* bidders, denoted by $E_p = \{i \in \{1, \dots, n\} | v_i < p\}$. When there is no ambiguity on the price, we will remove the subscript from the sets.

We consider the following notion of envy-freeness. Given an outcome $\langle \mathbf{X}, p \rangle$, we say bidder i *envies* bidder i' if the following conditions are satisfied:

1. allocating $X_{i'}$ to i would be *rational* for i , i.e., $X_{i'} \subseteq S_i$;
2. $b_i \geq p \sum_{j \in S_{i'}} x_{i'}^j$;
3. $u_i(X_{i'}, p) > u_i(X_i, p)$.

An outcome is *envy-free* if for every pair of bidders $i, i' \in \{1, \dots, n\}$, bidder i does not envy bidder i' .

Definition 1. *The envy-free fixed-price auction: Given $\langle n, m, \mathbf{M}, \mathbf{S}, \mathbf{v}, \mathbf{b} \rangle$, design a mechanism to compute an envy-free outcome $\langle \mathbf{X}, p \rangle$ that maximizes auctioneer's revenue $r(\mathbf{X}, p)$.*

2 Unit-demand Bidders

In this section, we consider envy-free fixed-price auction when all bidders have unit demands and budgets, that is, bidders are only interested in obtaining at most one copy of a good in their preference sets. Here, envy-freeness implies that bidder i will be allocated one item if another bidder i' gets some item $j \in S_i$ and the price is not higher than $\min\{v_i, b_i\}$. Guruswami et al. [2] considered non-fixed-price schemes in this setting. They showed that the problem is APX-hard and gave a logarithmic approximation algorithm. A logarithmic in-approximability result was given by Chalermsook et al. [1] for unlimited supply. This logarithmic hardness translates to our setting as unit-demand min-buying pricing they defined induces an envy-free allocation. On the contrary, we present an algorithm optimizing the revenue among all fixed-price allocations. Despite the fact that our auction is fixed-price, we prove that it is able to extract $O(\log n)$ of the optimum envy-free auction. We assume in this section for all $i \in [1, \dots, m]$ we have $m_i \leq n$, as higher supply of goods is not needed in the unit demand case.

We model auctions for the unit-demand case as bipartite graphs. Specifically, given a price p , the demand graph G_p is a bipartite graph $(V = \{1, \dots, n\}, U = \{1, \dots, m'\}, E)$, where $\{1, \dots, m'\}$ contains m_j identical vertices for good j and E contains an edge (i, j) if and only if $j \in S_i$ and $\min\{v_i, b_i\} \geq p$. We identify vertices in V with bidders. We define non-value-limited subgraph G'_p of G_p to contain only bidders that are non-value-limited.

Denote for a set $S \subseteq V(G)$, $\nu_G(S) = \{v : (u, v) \in E(G) \text{ and } u \in S\}$. A set of vertices $S \subseteq V$ is said to be *tight* in G if and only if $|\nu_G(S)| \geq |S|$. We observe

that we cannot sell items to all bidders in S if S is not tight in G'_p , because we will not be able to satisfy the demand of everybody in S . On the other hand, it is not necessarily true that we can satisfy all bidders in a given tight set S , because S can contain subsets that are not tight. Using the tools from matching theory we will prove the existence of canonical tight sets in G'_p , that can be fully satisfied. We define *surplus* of $S \subseteq V \cup U$ in G as $\min_{T \subseteq S} |\nu_G(T)| - |T|$. By Hall's marriage theorem a set $S \subseteq V$ that has positive surplus in G has a perfect matching onto $\nu_G(S)$. The following is a consequence of Gallai-Edmonds decomposition theorem in the case of bipartite graphs.

Theorem 1 (Theorem 3.2.4 from [4]). *Let $G = (V, U, E)$ be a bipartite graph. V can be partitioned into three subsets C_V, A_V, D_V and U can be partitioned into three subsets C_U, A_U, D_U such that:*

- $\nu_G(D_V) = A_U$ and $\nu_G(D_U) = A_V$,
- surplus of C_V in $G[C_V \cap C_U]$ is 0,
- surplus of A_V in $G[A_V \cap D_U]$ is positive,
- surplus of A_U in $G[A_U \cap D_V]$ is positive.

We observe the following.

Corollary 1. *Let A_U be given by Theorem 1 for G'_p . No item from A_U can be sold in any envy-free allocation at price p .*

By Hall's marriage theorem we know that the matching M that is needed in the following observation always exists.

Corollary 2. *Let C_V and A_V be given by Theorem 1 for G'_p . Let M be a matching in G'_p that matches C_V with C_U and A_V with D_U then M induces an envy-free allocation at price p .*

In other words M is the maximum size matching in $G'_p \setminus A_U$. So far we have considered only non-value-limited bidders in G'_p . Adding value-limited bidders can increase the number of items we can sell and lead to higher revenue. Still no item in A_U can be sold. Potentially, in $G_p \setminus A_U$ there can be a maximum size matching such that it does not induce an envy-free allocation. However, this can be countered using the following well known fact.

Corollary 3 (Corollary 3.1.6 from [4]). *If a set of vertices is covered by some matching then it is also covered by some maximum matching.*

By applying this corollary to M in G_p , we know that there exists a maximum size matching M' in $G'_p \setminus A_U$ that matches all vertices in C_V and A_V . Hence, M' induces the maximum revenue envy-free allocation at price p . We note that the constructions of Theorem 1 and Corollary 3 can be executed in polynomial time. Now, we can observe that the set of candidate fixed-prices is small.

Lemma 1. *The set of candidate fixed-prices for the unit-demand case is given by $\cup_{i \in \{1, \dots, m\}} \min(v_i, b_i)$.*

Algorithm 1: Optimal Revenue Maximizing Envy-free Fixed-price for Unit-Demanded Bidders

Input: $\langle \{1, \dots, n\}, \{1, \dots, m\}, \mathbf{M}, \mathbf{S}, \mathbf{v}, \mathbf{b} \rangle$.
Output: a price p , an allocation $\mathbf{X} = \langle \mathbf{X}_1, \dots, \mathbf{X}_n \rangle$, the optimal revenue r .

- 1 Let \mathbf{p} be the set containing $\{\min(v_i, b_i)\}$, $r = 0$, and $S = \emptyset$;
- 2 **while** $\mathbf{p} \neq \emptyset$ **do**
- 3 Choose some $y \in \mathbf{p}$, $\mathbf{p} = \mathbf{p} \setminus \{y\}$;
- 4 Construct the demand graphs G_y and G'_y with respect to y ;
- 5 Use Theorem 1 to find A_U in G'_y ;
- 6 Find maximum size matching M of $G'_y \setminus A_U$;
- 7 Extend M to maximum matching M' in $G_y \setminus A_U$ using Corollary 3;
- 8 **if** $y|M| > r$ **then**
- 9 Set $p = y$, $r = y|M|$;
- 10 **for** $i \in \{1, \dots, n\}$ **do**
- 11 **if** $j \in \mathcal{M}$ **then**
- 12 Set $x_i^j = 1$ if $(i, j) \in M$;

The main result in this section is the following theorem, which directly follows lemmas and corollaries above.

Theorem 2. *Algorithm 1 outputs, in polynomial time, an envy-free outcome which optimizes auctioneer's revenue among all envy-free fixed-price outcomes.*

Next theorem shows that our auction is $O(\log n)$ -approximate with respect to the general envy-free auction.

Theorem 3. *Algorithm 1 outputs, in polynomial time, an envy-free outcome which is $O(\log n)$ -approximate with respect to the optimum envy-free action.*

3 Envy-free Fixed-price Multi-unit Auctions with Budgets

We now turn to envy-free fixed-price auction when bidders are not unit-demand and have budgets, that is, bidders are interested in obtaining items in their preference sets as much as possible within their budgets. In this section, we present a Fully Polynomial Time Approximation Scheme (FPTAS) for the case that one single type of good is in limited supply of m copies. This setting is often referred to as *multi-unit auction*. Note that in the indivisible environment the easier approaches used with divisible items are no more useful. The presence of discontinuity points that produce discrete jumps in the demand functions force us to carefully analyze the prices, in particular when the number of items is small and a rough analysis can blemish the approximation ratio. We denote by $\langle \mathbf{X}^{\text{OPT}}, p^{\text{OPT}} \rangle$ the revenue maximizing envy-free outcome. Note that $\mathbf{X}^{\text{OPT}} = \langle x_1^{\text{OPT}}, \dots, x_n^{\text{OPT}} \rangle$. Bidder i gets x_i^{OPT} copies of the good and pays

$p^{OPT} x_i^{OPT}$. For the optimal revenue we have $r^{OPT} = p^{OPT} \sum_{i=1}^n x_i^{OPT}$. The proposed FPTAS considers the following two cases regarding \mathbf{X}^{OPT} :

1. More than $\frac{n^2}{\epsilon}$ goods are allocated in the revenue maximizing envy-free outcome, that is, $m \geq \sum_{i=1}^n x_i^{OPT} > \frac{n^2}{\epsilon}$;
2. At most $\frac{n^2}{\epsilon}$ copies are allocated in the revenue maximizing envy-free outcome, that is, $\sum_{i=1}^n x_i^{OPT} \leq \frac{n^2}{\epsilon}$.

The distinction between these two cases is important as follows. In the former case that the number of copies is not polynomial in n , the proposed FPTAS achieves a revenue at least $(1 - \epsilon)r^{OPT}$ by restricting to a polynomial number of prices. On the other hand, a dynamic programming is design to extract the optimal r^{OPT} through the enumeration of all the possible optimal outcomes.

3.1 The case of $m \geq \sum_{i=1}^n x_i^{OPT} > \frac{n^2}{\epsilon}$

In this case, there are a non-polynomial number of copies. Hence, one cannot enumerate all the possible sensible prices. The propose algorithm overcome this problem by finding a polynomial number of prices that could extract an $(1 - \epsilon)$ fraction of the optimal revenue. We assume for convenience $0 \leq v_1 \leq v_2 \leq \dots \leq v_n$. OPT will also set $p^{OPT} \in [0, v_n]$. Set $v_0 = 0$, and let us define $\Pi_i = (v_{i-1}, v_i]$. There exists exactly one Π_i such that $p^{OPT} \in \Pi_i$. Denote by $A_i = \{i' : v_{i'} > v_{i-1}\}$ the set of bidders with values higher than v_{i-1} .

Furthermore, for each interval Π_i , we define a price

$$\bar{p}_i = \begin{cases} \min\{v_i, \frac{\epsilon \cdot \sum_{j \in A_i} b_j}{n^2}\}, & \text{if } \frac{\epsilon \cdot \sum_{j \in A_i} b_j}{n^2} > v_{i-1} \\ v_i, & \text{if } \frac{\epsilon \cdot \sum_{j \in A_i} b_j}{n^2} \leq v_{i-1} \end{cases}$$

Note that price $\bar{p}_i \in \Pi_i$ and $\bar{p}_i = v_i$ if $v_{i-1} = v_i$. Algorithm 2 computes the revenue obtained from each price in $\{\bar{p}_i\}_{i=1}^n$ and outputs the maximum one.

We show that the revenue r produced by Algorithm 2 is an $(1 - \epsilon)$ approximation to the optimal revenue r^{OPT} . The proof is split into two parts: i) We first consider the case of $p^{OPT} \in \Pi_i$ and $\bar{p}_i = \frac{\epsilon \cdot \sum_{j \in A_i} b_j}{n^2}$; ii) We later address the case of $p^{OPT} \in \Pi_i$ and $\bar{p}_i = v_i$.

We use the following lemma as a tool to prove the first case.

Lemma 2. *If $\bar{p}_i = \frac{\epsilon \cdot \sum_{j \in A_i} b_j}{n^2}$, then $r^{OPT} \geq \frac{\sum_{j \in A_i} b_j}{n}$.*

Proof. If $\bar{p}_i = \frac{\epsilon \cdot \sum_{j \in A_i} b_j}{n^2}$ then there is enough copies to satisfy the demand of all bidders in A_i , those with $v_{i'} > \bar{p}_i$:

$$\sum_{i \in A_i} D_i(\bar{p}_i) = \sum_{i \in A_i} \lfloor \frac{b_i}{\bar{p}_i} \rfloor \leq \sum_{i \in A_i} \frac{b_i}{\bar{p}_i} = \frac{n^2}{\epsilon}.$$

Note that essentially those bidders are the ones in A_i . Thus, we have $r^{OPT} \geq \sum_{j \in A_i} \lfloor \frac{b_j}{\bar{p}_i} \rfloor \cdot \bar{p}_i \geq \sum_{j \in A_i} (\frac{b_j}{\bar{p}_i} - 1) \cdot \bar{p}_i \geq \sum_{j \in A_i} b_j - n \cdot \bar{p}_i = \sum_{j \in A_i} b_j - \frac{\epsilon \cdot \sum_{j \in A_i} b_j}{n} = \sum_{j \in A_i} \frac{b_j}{n} (n - \epsilon)$. From $n \geq 2$ we have $n - \epsilon > 1$ and the lemma follows. \square

Algorithm 2: Envy-free Fixed-price Multi-unit Auction with Budgets
 $(\sum_{i=1}^n x_i^{OPT} > \frac{n^2}{\epsilon})$

Input: $\langle A = \{1, \dots, n\}, m, \mathbf{v}, \mathbf{b} \rangle$.

Output: a price p^{ALG} , an allocation $\mathbf{X}^{ALG} = \langle x_1^{ALG}, \dots, x_n^{ALG} \rangle$, the revenue r^{ALG} .

```

1 Set  $r = 0$ , Let  $\bar{p} = \{\bar{p}_1, \dots, \bar{p}_n\}$  be the set of candidate prices;
2 while  $\bar{p} \neq \emptyset$  do
3   Pick a price  $\bar{p}_i \in \bar{p}$ ;
4   Compute  $\langle p', \mathbf{X}', r' \rangle = \text{Sell}(\bar{p}_i, A_i, m, \mathbf{v}, \mathbf{b})$ ;
5   if  $r' > r$  then
6     Set  $\langle p^{ALG}, \mathbf{X}^{ALG}, r^{ALG} \rangle = \langle p', \mathbf{X}', r' \rangle$ ;
7     Set  $\bar{p} = \bar{p} \setminus \{\bar{p}_i\}$ ;
1 Sell( $p, A, \mathbf{v}, \mathbf{b}$ ):
2 Set  $\mathbf{X} = \langle x_1 = 0, \dots, x_n = 0 \rangle, r = 0$ ;
3 if  $\sum_{i \in A} D_i(p) \leq m$  then
4   For each  $i \in A$ , set  $x_i = D_i(p)$ , and set  $r = p \sum_{i \in A} D_i(p)$ ;
5 return  $\langle p, \mathbf{X}, r \rangle$ 

```

Lemma 3. If $p^{OPT} \in \Pi_i$ and $\bar{p}_i = \frac{\epsilon \cdot \sum_{j \in A_i} b_j}{n^2}$, then $r^{ALG} \geq (1 - \epsilon)r^{OPT}$.

Proof. $r^{OPT} \leq \sum_{j \in A_i} b_j = \sum_{j \in A_i} \frac{b_j}{\bar{p}_i} \cdot \bar{p}_i \leq \sum_{j \in A_i} (\lfloor \frac{b_j}{\bar{p}_i} \rfloor + 1) \cdot \bar{p}_i \leq \sum_{j \in A_i} x_j^{ALG} \cdot \bar{p}_i + \sum_{j \in A_i} \bar{p}_i = r^{ALG} + \sum_{j \in A_i} \bar{p}_i = r^{ALG} + n \cdot \frac{\epsilon \cdot \sum_{j \in A_i} b_j}{n^2} \leq r^{ALG} + \epsilon \cdot r^{OPT}$.
The first inequality comes from the assumption that $p^{OPT} \in \Pi_i$, and the last inequality is implied by Lemma 2. Hence, we have $r^{ALG} \geq (1 - \epsilon)r^{OPT}$. \square

Now we move to the second part of the proof. This is the case that $p^{OPT} \in \Pi_i$ and $\bar{p}_i = v_i$. We first observe the following:

Lemma 4. The optimal price $p^{OPT} \notin \Pi_i$ if $\frac{\epsilon \cdot \sum_{j \in A_i} b_j}{n^2} < v_{i-1}$.

Proof. Assume that $\frac{\epsilon \cdot \sum_{j \in A_i} b_j}{n^2} < v_{i-1} < p^{OPT}$, it implies that there is less than $\frac{n^2}{\epsilon}$ copies sold in the optimal allocation. It contradicts the assumption that at least $\frac{n^2}{\epsilon}$ goods allocated to the optimal revenue maximizing envy-free outcome. \square

Lemma 4 allows us to only prove the revenue guarantee of Algorithm 2 on the case that $p^{OPT} \in \Pi_i$, $\bar{p}_i = v_i$ and $\frac{\epsilon \cdot \sum_{j \in A_i} b_j}{n^2} \geq v_i$. In this case, it could be that the total demand of bidders in A_i exceeds the number of copies, that is, $\sum_{j \in A_i} D_j(\bar{p}_i) \geq \frac{n^2}{\epsilon} > m$. Denote by $t_k^{\bar{p}} = \{i \in \{1, \dots, n\} | D_i(\bar{p}) \geq k\}$ the set of bidders with demand at least k and $l = \max\{z | t_z > 0, \sum_{j=1}^z |t_j^{\bar{p}}| \leq m\}$. Algorithm 2 allocates to each bidder $i \in A_i$ a supply of $\min\{l, D_i(\bar{p})\}$ copies at price \bar{p} .

Lemma 5. *If $p^{OPT} \in \Pi_i$ and $\bar{p}_i = v_i$, $r^{ALG} > (1 - \epsilon)r^{OPT}$.*

Proof. By Lemma 4 and the assumption that $m \geq \frac{n^2}{\epsilon}$, it is sufficient to consider the case that $\sum_{i \in N} D_i(v_i) \leq m$ only. In this case, we have enough copies to satisfy the demand of all bidders at price v_i . Algorithm 2 sells at least $\frac{n^2}{\epsilon} - n$ copies at price v_i . Since the maximum revenue any algorithm can get is no more than the budgets of the bidders, it is easy to see the optimal revenue is at most $\frac{n^2}{\epsilon}v_i$. Hence, Algorithm 2 achieves a $(1 - \epsilon)$ -approximation to the optimal revenue. \square

Besides a good approximation to the optimal revenue, the following lemma shows outcomes produced by Algorithm 2 are envy-free.

Lemma 6. *$\langle X^{ALG}, p^{ALG} \rangle$ is envy-free.*

Proof. When price \bar{p}_i produce a total demand less than the available copies, i.e., $\sum_{i \in A_i} D_i(\bar{p}_i) \leq m$, we satisfy all the demands. Hence, no bidder envies others. When $\sum_{i \in A_i} D_i(\bar{p}_i) > m$, for every $i \in A_i$, it holds either bidder i obtains $D_i(\bar{p}_i)$ copies or bidder i receive the maximum number of copies among all bidders. It concludes that allocation $\langle X^{ALG}, p^{ALG} \rangle$ is envy-free.

Now we present the main theorem in this section, as stated below.

Theorem 4. *When there are more than $\frac{n^2}{\epsilon}$ copies sold in the revenue maximizing envy-free outcome in multi-unit auctions, Algorithm 2 outputs, in polynomial time, an envy-free outcome that achieves a $(1 - \epsilon)$ -approximation to the optimal envy-free revenue.*

3.2 The case of $\sum_{i=1}^n x_i^{OPT} \leq \frac{n^2}{\epsilon}$

Now we consider the other case that at most $\frac{n^2}{\epsilon}$ copies of the good are allocated in the revenue maximizing envy-free outcome. We first present a dynamic programming that maximizes the revenue when a price is given. Since there is only a polynomial number of possible prices for p^{OPT} , i.e., $p^{OPT} \in \bigcup_{i=1}^n (\{\frac{b_i}{j}\}_{j=1}^{\frac{n^2}{\epsilon}} \cup \{v_i\})$, one can simply run the dynamic programming for every possible price and output the optimal revenue. Now let us concentrate on computing the maximum revenue when a price is given. We first sort bidders by their demands. Without loss of generality, we assume $D_1(p) \leq \dots \leq D_n(p)$. Recall that VL is the set of value-limited bidders, and NVL is the set of non-value-limited bidders. Let $VL(i)$ be the i th value-limited bidder in the sorted order. Similarly, let $NVL(i)$ be the i th non-value-limited bidder in the sorted order. Observe that both sets VL and NVL are also sorted by non-decreasing demand at price p . The correctness of the dynamic programming relies on the following claim.

Claim 1 *There exists a revenue maximizing envy-free allocation X^{OPT} such that the following conditions hold: i) $x_1^{OPT} \leq x_2^{OPT} \leq \dots \leq x_n^{OPT}$; ii) $x_{VL^1}^{OPT} \leq x_{VL^2}^{OPT} \leq \dots \leq x_{VL^{|VL|}}^{OPT}$; iii) $x_{NVL^1}^{OPT} \leq x_{NVL^2}^{OPT} \leq \dots \leq x_{NVL^{|NVL|}}^{OPT}$.*

The dynamic programming fills a 4-dimensional table $t[i][j][k][h]$ of boolean values representing the existence of an envy-free allocation of exactly h copies among the first i value-limited bidders and the first j non-value-limited bidders such that the maximum number of copies allocated to a bidder is k . The dynamic programming proceeds in rounds. In the current round either the next value-limited or the next non-value-limited bidder in the order is considered:

- The bidder considered in the current round is value-limited. If the maximum number of copies already allocated to value limited bidders $[1, \dots, i-1]$ and non-value-limited bidders $[1, \dots, j]$ is k and $t[i-1][j][k][h-k]$ is true, then we can allocate k copies to value-limited bidder j . The resulting allocation is still envy-free. On the other hand, if the maximum number of copies allocated is less than k then we must ensure that giving k copies to value-limited bidder j does not violate envy-freeness. That implies that all non-value-limited bidders $[1, \dots, j]$ have budgets less than kp .
- The bidder considered in the current round is non-value-limited. Similar to the previous case, if the maximum number of copies already allocated to a bidder is k and $t[i][j-i][k][h-k]$ is true, then we can allocate k copies to this bidder. The resulting allocation is still envy-free. On the other hand, if the maximum number of copies already allocated to a bidder is less than k , we must ensure that giving k copies to this value-limited bidder does not affect envy-freeness. This essentially means that the already allocated non-value-limited bidders have budgets less than kp .

Finally, the revenue maximizing envy-free allocation is the entry of the table with *TRUE* value in $t[VL][NVL][\cdot][h]$ that maximizes the total number of allocated copies of the good $h \leq m$.

Now we are ready to present Algorithm 3 that computes revenue maximizing outcomes when the number of copies allocated is at most $\frac{n^2}{\epsilon}$. The core of Algorithm 3 is a dynamic programming that computes revenue maximizing outcomes on every possible optimal price.

Theorem 5. *When there are at most $\frac{n^2}{\epsilon}$ copies sold in the revenue maximizing envy-free outcome in multi-unit auctions, Algorithm 3 outputs, in polynomial time in terms of n and ϵ , an envy-free outcome that achieves the optimal envy-free revenue.*

Proof. Given a price, the size of table in the dynamic programming is $\mathcal{O}(n^2 \frac{n^2}{\epsilon})$. A calculation of a value of a single entry requires $\mathcal{O}(\frac{n^2}{\epsilon})$ time, which gives a total $\mathcal{O}(\frac{n^6}{\epsilon^2})$ for computing a revenue maximizing outcome for a given price. Since there are $\mathcal{O}(\frac{n^3}{\epsilon})$ possible price, the total complexity of the dynamic programming is $\mathcal{O}(\frac{n^9}{\epsilon^3})$. The envy-freeness of outcomes produced by Algorithm 3 is guaranteed by the recursive formula. The recursive formula produces all possible non-decreasing allocations that satisfy Claim 1. Hence, the maximum revenue achieved by this dynamic programming gives us the optimal envy-free revenue. \square

Algorithm 3: Dynamic Programming for Envy-free Fixed-price Multi-unit Auction with Budgets ($\sum_{i=1}^n x_i^{OPT} \leq \frac{n^2}{\epsilon}$)

Input: $\langle \{1, \dots, n\}, m, \mathbf{v}, \mathbf{b} \rangle$.

Output: a price p^{DP} and the revenue r^{DP} .

- 1 Set $P = \bigcup_{i=1}^n (\{\frac{b_i}{j}\}_{j=1}^{\frac{n^2}{\epsilon}} \cup \{v_i\})$;
- 2 **while** $P \neq \emptyset$ **do**
- 3 Pick a $p \in P$;
- 4 Initial a 4-dimensional table, set $t[0][0][0][0] = T$, and
 $t[0][0][k][h] = F, \forall 0 < k \leq m, 0 < h \leq m$;
- 5 Compute every entry in the table by the following recursive formula:

$$t[i][j][k][h] = \left[(i > 0) \wedge (k \leq D_{VL(i)}) \wedge (h > k) \right. \\
 \wedge \left(t[i-1][j][k][h-k] \right. \\
 \left. \vee \left[\bigvee_{k' < k} t[i-1][j][k'][h-k] \wedge (k > D_{NVL(j)}) \right] \right) \\
 \vee \left[(j > 0) \wedge (k \leq D_{NVL(j)}) \wedge (h > k) \right. \\
 \left. \wedge \left(t[i][j-1][k][h-k] \right. \right. \\
 \left. \left. \vee \left[\bigvee_{k' < k} t[i][j-1][k'][h-k] \wedge (k > D_{NVL(j-1)}) \right] \right) \right] \\
 \left. \right]$$
- 6 Let $l = \max\{h \in \{1, \dots, m\} \mid \exists i, j, k, t[i][j][k][h] = T\}$;
- 7 **if** $lp > r^{DP}$ **then**
- 8 Set $p^{DP} = p$ and $r^{DP} = lp$;
- 9 Set $P = P \setminus \{p\}$;

4 Envy-free Revenue Maximizing Fixed-price Multi-good Auctions with Budgets and Matching Preferences

In previous section we presented encouraging results when there is one single type of good in the auction. However, the results do not apply to the setting when there are different types of goods and bidders have *matching preferences*, i.e., bidder i values v_i all copies of goods in $S_i \subseteq \{1, \dots, m\}$ and 0 for items not in S_i . We start with a hardness result as shown below.

Theorem 6. *When there are non-constant number of types of goods, the optimal revenue in any envy-free multi-good auctions with budgets cannot be approximated within $O(\min(n, m)^{1/2-\epsilon})$ for any $\epsilon > 0$ unless $P = NP$.*

Corollary 1 *When there are non-constant number of types of goods, the optimal revenue in any envy-free fixed-price multi-good auctions with budgets cannot be approximated within $O(\min(n, m)^{1/2-\epsilon})$ for any $\epsilon > 0$ unless $P = NP$.*

4.1 A FPTAS for Constant Number of Types of Goods

The main result in this section is the following theorem.

Theorem 7. *When the number of types of goods is a constant, there exists an algorithm outputs, in polynomial time in terms of n and ϵ , an envy-free fixed-price outcomes that achieves a $(1 - \epsilon)$ -approximation to the optimal envy-free fixed-price revenue.*

The algorithm could be seen a generalization of the results in Section 3. Due to the space limit, we only discuss the ideas of how to extend our techniques in Section 3 to obtain a FPTAS in this setting. The full version of this section is available in Appendix B. As before, the proposed FPTAS considers two cases regarding \mathbf{X}^{OPT} .

OPT allocates more than $\frac{n^2}{\epsilon}$ copies of goods An important difference with respect to the multi-unit case is the following. When \bar{p}_i equals to $\frac{\epsilon \cdot \sum_{j \in A_i} b_j}{n^2}$, it is possible that there is no feasible allocation that sells $\frac{n^2}{\epsilon}$ copies of items at price \bar{p}_i . This is because the cumulative demand for a subset of items could be greater than the number of copies of the items. Hence, in order to find a price at which $\frac{n^2}{\epsilon}$ can be sold and to keep the envy-freeness of the allocation, we transfer the problem to a min-cost max-flow problem. When the maximum flow is not equal to $\frac{n^2}{\epsilon}$, the solution of min-cost max-flow identifies the subset of bidders who do not obtain $\frac{b_{i'}}{\bar{p}_i}$ copies of items. Then, the algorithm iteratively updates \bar{p}_i by decreasing a particular amount until \bar{p}_i is out of the range of I_i or $\frac{n^2}{\epsilon}$ copies of items can be sold.

OPT allocates at most $\frac{n^2}{\epsilon}$ copies of goods In order to cooperate this setting, the dynamic programming needs to ensure that envy-freeness between different types of bidders. Bidders belongs to the same type if their preference sets are the same. To guarantee the envy-freeness of the allocation, when bidder i is allocated to a copy of good $j \in S_i$, bidder i' must also be allocated a copies of $j' \in S_{i'}$ if $j \in S_{i'}$. We solve this problem in the dynamic programming by consider together different type bidders whose preference sets intersect. Similar to Section 3, bidders are divided into two groups, valued-limited and non-valued-limited bidders. Bidders are sorted by their budgets in each group and considered in this order. Besides satisfying the limited supply and non-decreasing numbers of goods assigned to bidders in each group, to guarantee that the allocation is envy-free, when bidder i is assigned a bundle of goods, the dynamic programming checks envy-freeness for all those bidders with preference set that includes the bundle, i.e., preference sets are the supersets of the bundle, prefer their bundles to the bundle assigned to bidder i or they cannot afford to buy bidder i 's bundle. Therefore, the dynamic programming has $2^c - 1$ parameters to indicate the largest bundles assigned to each type of bidders.

Bibliography

- [1] Chalermsook P, Chuzhoy J, Kannan S, Khanna S (2012) Improved hardness results for profit maximization pricing problems with unlimited supply. In: Gupta A, Jansen K, Rolim JDP, Servedio RA (eds) APPROX-RANDOM, Springer, Lecture Notes in Computer Science, vol 7408, pp 73–84
- [2] Guruswami V, Hartline JD, Karlin AR, Kempe D, Kenyon C, McSherry F (2005) On profit-maximizing envy-free pricing. In: Proceedings of the 16th annual ACM-SIAM symposium on Discrete algorithms, SODA '05, pp 1164–1173
- [3] Håstad J (1997) Clique is hard to approximate within $n^{1-\epsilon}$. Electronic Colloquium on Computational Complexity (ECCC) 4(38)
- [4] Lovász L, Plummer D (2009) Matching Theory. AMS Chelsea Publishing Series, AMS Chelsea Pub., URL <http://books.google.pl/books?id=yW3WSVq8ygcC>
- [5] Végh LA (2012) Strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives. In: Proceedings of the 44th symposium on Theory of Computing, ACM, pp 27–40

A Appendix - Proofs

A.1 Proof of Corollary 1

Proof. By positive surplus of A_U we have that for every subset of items $T \subseteq A_U$ there are more bidders than $|T|$ interested in buying items from T . Hence, for every allocation of T there is at least one bidder that gets nothing and hence envies the others.

A.2 Proof of Corollary 2

Proof. Observe that only bidders in D_V could potentially envy someone, but D_V is incident only with A_U which is left unsold.

A.3 Proof of Lemma 1

Proof. In the unit-demand case the price paid by the bidders can always be increased without losing any revenue till the minimum between the valuation and the budget of a bidder. Therefore, we need to consider at most $2n$ different prices.

A.4 Proof of Lemma 3

Proof. Consider the optimum allocation \bar{X}_i and optimum pricing \bar{p}_i . Assume that in this allocation bidders are sorted in descending order by \bar{p}_i . Consider a random number l that is selected uniformly at random from the set $[1, \dots, \log(2n)]$, and consider price $p_l = \frac{\bar{p}_l}{2}$. Let us denote by $\bar{n}(p)$ the number of bidders who pay in optimum allocation no less than p . We observe that

$$OPT = \sum_{i=1}^n \bar{p}_i = \sum_{i=1}^n \bar{n}(\bar{p}_i)(\bar{p}_i - \bar{p}_{i+1}) \leq 2 \sum_{l=0}^{\log(2n)} \bar{n}(p_l) p_l + n \frac{p_l}{2^{\log(2n)}} = 2 \sum_{l=0}^{\log(2n)} \bar{n}(p_l) p_l + \frac{OPT}{n}.$$

Additionally note that for each p_l selling items to bidders with $p_i \geq p_l$ at fixed price p_l is envy-free. Hence, choosing random p_l gives fixed-price $O(\log n)$ -approximation in expectation with respect to the optimum envy-free pricing. By the optimality of our algorithm this bound in expectation is turned into worst-case bound. \square

A.5 Proof of Lemma 4

Proof. $\frac{\epsilon \cdot \sum_{j \in A_i} b_j}{n^2} < v_{i-1}$ and $p^{OPT} > v_{i-1}$ imply that there is less than $\frac{n^2}{\epsilon}$ items sold in the optimal allocation thus a contradiction since we are in the case of at least $\frac{n^2}{\epsilon}$ goods allocated to the optimal revenue maximizing envy-free outcome.

A.6 Proof of Lemma 6

Proof. When price \bar{p}_i produce a total demand less than the available items, i.e., $\sum_{i \in A_i} D_i(\bar{p}_i) \leq m$, we satisfy all the demands. Hence, no bidder envies others. When $\sum_{i \in A_i} D_i(\bar{p}_i) > m$, for every $i \in A_i$, it holds either bidder i obtains $D_i(\bar{p}_i)$ items or bidder i receive the maximum number of items among all bidders. It concludes that allocation $\langle X^{ALG}, p^{ALG} \rangle$ is envy-free.

A.7 Proof of Claim 1

Proof. Since the second and third conditions directly follow from the first condition, it is sufficient to prove the first condition. Suppose there exists another revenue maximizing envy-free allocation X' where $x'_i > x'_j$ for a pair i and $j > i$. Since X' is envy-free, it must be the case that both i and j are value-limited bidders. Thus, we can swap the allocation between bidders i and j , which results in a new envy-free allocation and maintains the same revenue. Recursively applying these, we could find out a revenue maximizing envy-free allocation satisfying the above conditions.

A.8 Proof of Theorem 5

Proof. Given a price, the size of table in the dynamic programming is $\mathcal{O}(n^2 \frac{n^2}{\epsilon})$. A calculation of a value of a single entry requires $\mathcal{O}(\frac{n^2}{\epsilon})$ time, which gives a total $\mathcal{O}(\frac{n^6}{\epsilon^2})$ for computing a revenue maximizing outcome for a given price. Since there are $\mathcal{O}(\frac{n^3}{\epsilon})$ possible price, the total complexity of the dynamic programming is $\mathcal{O}(\frac{n^9}{\epsilon^3})$.

The envy-freeness of outcomes produced by Algorithm 3 is guaranteed by the recursive formula. The recursive formula produces all possible non-decreasing allocations that satisfy Claim 1. Hence, the maximum revenue achieved by this dynamic programming gives us the optimal envy-free revenue.

A.9 Proof of Theorem 6

Proof. We give a reduction from the Maximum Independent Set (MIS) problem on graphs. It is well-known that MIS cannot be approximated within $|V|^{1-\epsilon}$ where V is the set of vertices. Let $G = (V, E)$ be an instance of MIS, and d is the maximum degree in G . The good in the auction consist of all edges E , and a set of dummy items. Each edge or each dummy item corresponds to a distinct type of good with one copy. We create the following two sets of bidders:

A: contains bidder v for each vertex $v \in V$. The preference set S_v of bidder v is equal to the set of edges incident to v in G . If the number of edges is smaller than d , add some distinct dummy items to S_v such that $|S_v| = d$. The budget b_v of bidder v is equal to d and his valuation v_v for each copy of good is 1.

B : contains another $d + 1$ identical bidders for each bidder $i \in A$. Their preference sets are equal to S_i . Their budgets are $d - 1$ and their valuations for each copy of good in S_i are $d - 1 + \epsilon$, where $\epsilon > 0$.

By the construction above, there are $n = |V| + |V|(d + 1)$ bidders and $m = d|V| - 2|E|$ different types of good. The supply of each type of good is 1. Due to the limited supply, any envy-free allocation cannot sell items to bidders in B . It leaves that the possible envy-free allocations are the ones which sell some bidders in A their entire preference sets and charge them more than $d - 1$ for the set. Note that the optimal revenue is to charge bidder v at d for the set S_v . Since the revenue obtained from each bidder in A is the same, the revenue maximizing envy-free allocation corresponds to the solution in the maximum independent set. The fact that MIS cannot be approximated with $|V|^{1-\epsilon}$ [3] implies that the optimal revenue cannot be approximated within $O(\min(n, m)^{1/2-\epsilon})$.

One could easily see that, even the optimal fixed price is given, the optimal revenue can not be approximated within $O(\min(n, m)^{1/2-\epsilon})$ for any $\epsilon > 0$ unless $P = NP$.

B Appendix - A FPTAS for Constant Number of Types of Goods

Here we devise a FPTAS for envy-free fixed-price auction in which there are a constant number c of different types of goods and m_j copies available for goods of type j . Since there are different types of goods and bidders have preference sets, the size of the allocated bundles are no more the only important parameter to guarantee the feasibility and the envy-freeness of the outcome. We extend the algorithm described in Section 3 with a few essential changes. The section follows the structure of Section 3. First, we show an algorithm that achieves a $(1 - \epsilon)$ fraction of the optimum revenue maximizing envy-free fixed-price outcome when the optimum revenue maximizing envy-free fixed-price outcome OPT allocates more than $\frac{n^2}{\epsilon}$ copies of goods. Next, we provide a dynamic programming that reaches the optimal envy-free fixed-price outcome if OPT allocates at most $\frac{n^2}{\epsilon}$ copies of goods.

B.1 OPT allocates more than $\frac{n^2}{\epsilon}$ copies of goods

We assume $0 \leq v_1 \leq v_2 \leq \dots \leq v_n$ and $p^{OPT} \in [0, v_n]$. Set $v_0 = 0$, and let us define $\Pi_i = (v_{i-1}, v_i]$. We know that there exists exactly one Π_i such that $p^{OPT} \in \Pi_i$. Let $A_i = \{j : v_j > v_{i-1}\}$, that is, the set of bidders whose values higher than the price if the price is set to v_{i-1} .

An important difference with respect to the multi-unit setting is as follows. Even if $\bar{p}_i = \frac{\epsilon \sum_{j \in A_i} b_j}{n^2}$ makes the cumulative demand less than $\frac{n^2}{\epsilon}$, it could be that we cannot satisfy the demand of all the bidders. This is because we don't know if the cumulative demand for a specific good is greater than the availability

of that good, even if the total demand is less than $\frac{n^2}{\epsilon}$ and the copies of different goods is greater than $\frac{n^2}{\epsilon}$,

Hence, in order to keep the feasibility and the envy-freeness we impose (different) bounds of the maximum number of goods on subsets of bidders whose demands are more than the current availability of goods in their preference sets. To detect such subsets of bidders, we use a max-flow min-cost algorithm. Formally, given a price p we construct a flow network $G^p = (V^p, E^p)$. The set of vertices are $V^p = \{\{s\} \cup \{1, \dots, n\} \cup \{1, \dots, c\} \cup \{t\}\}$. The source vertex s is linked to all the bidders ($\forall i \in \{1, \dots, n\} \exists (s, i) \in E$) and the capacity of each edge (s, i) is the demand of bidder i for the current price p , i.e. $D_i(p)$. Each bidder i is linked to all types of goods in his preference set S_i , i.e. $\forall j \in S_i \exists (i, j) \in E^p$. The capacity of each edge that is an outgoing edge from a bidder $i \in \{1, \dots, n\}$ is equal to infinity. All the goods are then linked to the sink vertex t and the capacity of all those edges are equal m_j (availability of each type of good). The cost function for each edge $e \in E^p$ is $c^e(f) = f^2$, where f is the amount of flow on e .

Now, we can construct a min-cost max-flow problem $MinCostMaxFlow^p(I, J)$ as described above for a given price p , a set of goods J , set of bidders I . The solution of such a problem is the max-flow that minimizes the sum of the costs in the network. Since the cost function is quadratic, minimize the cost is equal to distribute goods as even as possible amongst the bidders. The solution of the problem is a vector $\mathbf{x} = \{x_1, \dots, x_n\}$ that represents fraction numbers of copies of goods allocated to each bidder, $x_i \in \mathbb{R}^{\geq 0}$.

Now given a price p , a set of active bidders A_i and a set of goods $\bar{M} \subseteq \{1, \dots, c\}$ we run the flow problem $MinCostMaxFlow^p(A_i, \bar{M})$. Given the solution of the flow problem \mathbf{X} , we can partition the set of active bidders A_i . $A_i^{NB} = \{j \in A_i | x_j = D_j(p)\}$ is the set of bidders that exhausted their demand. $A_i^B = \{j \in A_i | x_j < D_j(p)\}$ is the set of bidders that cannot obtain their full demand.

Note that price $\bar{p}_i \in \Pi_i$ and $\bar{p}_i = v_i$ if $v_{i-1} = v_i$. Initially the price is set to $\bar{p}_i = \frac{\epsilon \cdot \sum_{j \in A_i} b_j}{n^2}$ and the $MinCostMaxFlow^{\bar{p}_i}(A_i, M)$ detects A_i^{NB} and A_i^B . If the maximum flow is less than $\frac{n^2}{\epsilon}$, we lower the price to $\bar{p}_i = \frac{\sum_{j \in A_i^{NB}} b_j}{\frac{n^2}{\epsilon} - \sum_{j \in A_i^B} f_j}$.

Thus the demand of all the bidders increase. As a consequence, the bidders that were detected as bounded remain bounded. The bidders that were not detected as bounded in the previous round now, with an increased demand, could become bounded. So we need to execute the $MinCostMaxFlow^{\bar{p}_i}(A_i, M)$ with the updated price and add to the set A_i^B the bidders that become bounded. Iteratively doing it, the algorithm ends up with one of the following conditions:

- a maximum flow of $\frac{n^2}{\epsilon}$ and $\bar{p}_i \in [v_{i-1}, v_i]$ are found;
- a maximum flow is less than $\frac{n^2}{\epsilon}$ and $A_i^{NB} = \emptyset$. It implies that it is impossible to find a feasible price such that more than $\frac{n^2}{\epsilon}$ copies of goods are sold to bidders in A_i . It further implies that the optimal price is not between v_{i-1} and v_i .

– $\bar{p}_i = v_i$ until \bar{p}_i is less than v_{i-1} .

The algorithm for detecting A_i^{NB} , A_i^B and compute \bar{p}_i for each interval Π_i is given as Algorithm 4.

Algorithm 4: Compute price \bar{p}_i for interval Π_i

Input: $\langle A_i, M = \{1, \dots, c\}, \{m_1, \dots, m_c\} \rangle$.

Output: A price \bar{p}_i , a set A_i^B of bidders and a flow result $\mathbf{X} = \{x_1, \dots, x_{|A_i|}\}$

```

1 Set  $\bar{p}_i = \frac{\epsilon \cdot \sum_{j \in A_i} b_j}{n^2}$ ;
2 Set  $A_i^{NB} = \emptyset$ ,  $A_i^B = \emptyset$ ;
3 Set  $\mathbf{f} = \text{MinCostMaxFlow}^{\bar{p}_i}(A_i, M)$ ;
4  $A_i^{NB} = \{j \in A_i \mid f_j = D_j(\bar{p}_i)\}$  where  $f_j$  the flow goes through bidder  $j$ ;
5  $A_i^B = A_i \setminus A_i^{NB}$ ;
6 while  $|f| \neq \frac{n^2}{\epsilon}$  do
7   if  $A_i^{NB} = \emptyset$  then
8     return  $\bar{p}_i = v_i$ ;
9   Set  $\bar{p}_i = \frac{\sum_{j \in A_i^{NB}} b_j}{\frac{n^2}{\epsilon} - \sum_{j \in A_i^B} f_j}$ ;
10  if  $\bar{p}_i < v_{i-1}$  then
11    return  $\bar{p}_i = v_i$ ;
12  Set  $\mathbf{f} = \text{MinCostMaxFlow}^{\bar{p}_i}(A_i, M)$ ;
13   $A_i^{NB} = \{j \in A_i \mid x_j = D_j(\bar{p}_i)\}$ ;
14   $A_i^B = A_i \setminus A_i^{NB}$ ;
15 return  $\min(\bar{p}_i, v_i)$  and  $X = \mathbf{f}$ ;

```

Lemma 7. For each interval Π_i , Algorithm 4 terminates in a polynomial time in n , and outputs a price \bar{p}_i such that the maximum flow is $\frac{n^2}{\epsilon}$ in $G^{\bar{p}_i}$ or setting $\bar{p} = v_i$.

Proof. Computing min-cost max-flow in a network with quadratic cost functions is strongly polynomial in the number of vertices and edges in the network [5]. Since the number of different types of goods is constant, the running time of computing min-cost max-flow in Algorithm 4 is polynomial in n . It is clear that, after computing a min-cost max-flow, Algorithm 4 terminates or at least one bidder is removed from A_i^{NB} . Therefore, the Algorithm 4 computes at most n times of min-cost max-flow.

At each round, Algorithm 4 decreases the price in the way that the demand of bidders in A_i^{NB} is the difference between $\frac{n^2}{\epsilon}$ and the number of copies assigned to bidders in A_i^B . Hence, Algorithm 4 will terminate with price \bar{p}_i such that the maximum flow is $\frac{n^2}{\epsilon}$, or setting $\bar{p}_i = v_i$ when the price decreases to a number less than v_{i-1} . \square

Now let us focus on a particular Π_i . Given A_i , Algorithm 4 computes \bar{p}_i , A_i^B , and \mathbf{X} . We set a virtual budget for each bidder in A_i as follows:

$$\bar{b}_{i'} = \begin{cases} b_{i'}, & \text{if } i' \notin A_i^B \\ x_j \cdot \bar{p}_i, & \text{if } i' \in A_i^B \end{cases}$$

Now consider the set of different types of goods in which bidders in A_i^B are interested, above lemma enable us to bound the maximum revenue we could extra from them is $\sum_{j \in A_i^B} x_j \bar{p}_i$. For other types of good, we bound it by the total budgets of bidders in A_i^{NB} .

Now, we prove that, by plugging in $\{\bar{b}_i\}$ and $\{\bar{p}_i\}$ computed by Algorithm 4 into Algorithm 2, it achieves a $(1 - \epsilon)$ approximation to the optimal revenue. The first part of the proof is that $p^{OPT} \in \Pi_i$ and $\bar{p}_i \in (v_{i-1}, v_i)$.

Lemma 8. *If $\bar{p}_i \in (v_{i-1}, v_i)$, then $r^{OPT} \geq \frac{\sum_{j \in A_i} \bar{b}_j}{n}$.*

Proof. In this case, Algorithm 4 finds a price in (v_{i-1}, v_i) such that we can have $\frac{n^2}{\epsilon}$ copies of goods to satisfy the demand of all bidders in A_i at price \bar{p}_i :

$$\sum_{i \in A_i} D_i(\bar{p}_i) = \sum_{i \in A_i} \lfloor \frac{\bar{b}_i}{\bar{p}_i} \rfloor \leq \sum_{i \in A_i} \frac{\bar{b}_i}{\bar{p}_i} = \frac{n^2}{\epsilon}.$$

Note that essentially those bidders are the ones in A_i . Thus, we have $r^{OPT} \geq \sum_{j \in A_i} \lfloor \frac{\bar{b}_j}{\bar{p}_i} \rfloor \cdot \bar{p}_i \geq \sum_{j \in A_i} (\frac{\bar{b}_j}{\bar{p}_i} - 1) \cdot \bar{p}_i \geq \sum_{j \in A_i} \bar{b}_j - n \cdot \bar{p}_i \geq \sum_{j \in A_i} \bar{b}_j - \frac{\epsilon \cdot \sum_{j \in A_i} \bar{b}_j}{n} = \sum_{j \in A_i} \frac{\bar{b}_j}{n} (n - \epsilon)$. From $n \geq 2$ we have $n - \epsilon > 1$ and the lemma follows. \square

Lemma 9. *If $p^{OPT} \in \Pi_i$ and $\bar{p}_i \neq v_i$, then $r^{ALG} \geq (1 - \epsilon)r^{OPT}$.*

Proof. $r^{OPT} \leq \sum_{j \in A_i} \bar{b}_j = \sum_{j \in A_i} \frac{\bar{b}_j}{\bar{p}_i} \cdot \bar{p}_i \leq \sum_{j \in A_i} (\lfloor \frac{\bar{b}_j}{\bar{p}_i} \rfloor + 1) \cdot \bar{p}_i \leq \sum_{j \in A_i} x_j^{ALG} \cdot \bar{p}_i + \sum_{j \in A_i} \bar{p}_i = r^{ALG} + \sum_{j \in A_i} \bar{p}_i \leq r^{ALG} + n \cdot \frac{\epsilon \cdot \sum_{j \in A_i} \bar{b}_j}{n^2} \leq r^{ALG} + \epsilon \cdot r^{OPT}$. The first inequality comes from the assumption that $p^{OPT} \in \Pi_i$, and the last inequality is implied by Lemma 2. Hence, we have $r^{ALG} \geq (1 - \epsilon)r^{OPT}$. \square

The second part is that $p^{OPT} \in \Pi_i$ and $\bar{p}_i = v_i$. We observe the following:

Lemma 10. *The optimal price $p^{OPT} \notin \Pi_i$, if $\frac{\sum_{j \in A_i^{NB}} b_j}{\frac{n^2}{\epsilon} - \sum_{j \in A_i^B} f_j} < v_{i-1}$.*

Proof. Algorithm 4 searches the maximum price such that it is possible that $\frac{n^2}{\epsilon}$ copies of goods are sold. If $\frac{\sum_{j \in A_i^{NB}} b_j}{\frac{n^2}{\epsilon} - \sum_{j \in A_i^B} f_j} < v_{i-1}$ and $p^{OPT} > v_{i-1}$, it implies that there is less than $\frac{n^2}{\epsilon}$ copies of goods sold in the optimal allocation thus a contradiction since we are in the case of at least $\frac{n^2}{\epsilon}$ goods allocated to the optimal revenue maximizing envy-free outcome. \square

Above lemma allows us to concentrate the case that $p^{OPT} \in \Pi_i$ and $\frac{\sum_{j \in A_i^{NB}} b_j}{\frac{n^2}{\epsilon} - \sum_{j \in A_i^B} f_j} \geq v_i$. The idea of the proof is essentially the same as Lemma 5.

Lemma 11. *If $p^{OPT} \in \Pi_i$ and $\bar{p}_i = v_i$, $r^{ALG} > (1 - \epsilon)r^{OPT}$.*

Proof. By Lemma 10, we only consider the cases that Algorithm 4 suggests that one could sell $\frac{n^2}{\epsilon}$ copies of goods at price $\bar{p}_i = v_i$. Since $m \geq \frac{n^2}{\epsilon}$, it is easy to see that Algorithm 2 sells at least $\frac{n^2}{\epsilon} - n$ copies at price v_i . Since the maximum revenue any algorithm can get is no more than the budgets of the bidders, it is easy to see the optimal revenue is at most $\frac{n^2}{\epsilon} v_i$. Hence, Algorithm 2 achieves a $(1 - \epsilon)$ -approximation to the optimal revenue. \square

Besides a good approximation to the optimal revenue, the following shows outcomes produced by Algorithm 2 are envy-free.

Lemma 12. *$\langle X^{ALG}, p^{ALG} \rangle$ is envy-free.*

Proof. When price $\bar{p}_i \neq v_i$, every bidder in A_i get his full demand $\lfloor \frac{\bar{b}_i}{\bar{p}_i} \rfloor$. When $\bar{p}_i = v_i$ and $\sum_{i \in A_i} D_i(\bar{p}_i) \leq m$, every bidder in A_i get their full demands. when $\bar{p}_i = v_i$ and $\sum_{i \in A_i} D_i(\bar{p}_i) > m$, for every bidder $i \in A_i$, it holds either bidder i obtains $D_i(\bar{p}_i)$ items or bidder i receive the maximum number of items among all bidders. It concludes that allocation $\langle X^{ALG}, p^{ALG} \rangle$ is envy-free. \square

Now we present the main theorem as stated below.

Theorem 8. *Given a price, when there are at least $\frac{n^2}{\epsilon}$ copies of different types of goods sold, computing \bar{p}_i as suggested in Algorithm 4, Algorithm 2 outputs a rational revenue maximizing envy-free allocation in polynomial time in n and $1/\epsilon$.*