# Online Pricing for Multi Type of Items

Yong Zhang[1,2]⋆, Francis Y.L. Chin[2]⋆⋆, and Hing-Fung Ting[2]⋆⋆⋆

[1] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China.
[2] Department of Computer Science, The University of Hong Kong, Hong Kong
{yzhang,chin,hfting}@cs.hku.hk

**Abstract.** In this paper, we study the problem of online pricing for multi-type items. Given a seller with $k$ types of items where the amount of each type is $m$, a sequence of users $\{u_1, u_2, ...\}$ arrive one by one. Each user is single-minded, i.e., each user is only interested in a particular bundle of items. The seller must set the unit price and assign some amount of bundles to each user upon his/her arrival. Bundles can be sold fractionally. Each $u_i$ has his/her value function $v_i(\cdot)$ such that $v_i(x)$ is the highest unit price $u_i$ is willing to pay for $x$ bundles. The objective is to maximize the revenue of the seller by setting the price and amount of bundles for each user. In this paper, we first show that the lower bound of the competitive ratio for this problem is $O(\log h + \log k)$, where $h$ is the highest unit price to be paid among all users. We then give a deterministic online algorithm Pricing, whose competitive ratio is $O(\sqrt{k} \cdot \log h \log k)$. The lower and upper bounds match with the optimal result $O(\log h)$ asymptotically when $k = 1$.

## 1 Introduction

Economy, a very important facet in the world, has received deep-and-wide studies by scientists from economy, mathematics, and computer science for many years. In computer science, researchers often build theoretical models for some economic events, then solve the problems by using techniques derived from algorithm design, combinatorial optimization, randomness, etc.

In this paper, we study the problem of item pricing, which is one of the most important problems in computational economy. Item pricing contains two kinds of participators: the seller and the user. The seller has some items, which will be sold to the users at some designated prices; the user will buy the items at an acceptable price. The objective is to maximize the total revenue of the seller by assigning items to the users. To achieve this target, the prices of the items must be sold dynamically, i.e., the prices of items are different for different users, at different times, in different locations, with different amounts, ... If the designated

⋆ Research supported by NSFC (No. 11171086) and Shanghai Key Laboratory of Intelligent Information Processing, China. Grant No. IIPL-2010-010
⋆⋆ Research supported by HK RGC grant HKU 7117/09E
⋆⋆⋆ Research supported by HK RGC grant HKU-7171/08E

price is higher than the expected price of an user, this user will reject the item; otherwise, this user will accept the item.

Formally speaking, given a seller with $k$ types of items, $i_1$, $i_2$, ..., $i_k$, the amount of each type is $m$, thus, the total amount of items is $m \cdot k$. A sequence of users $\{u_1, u_2, ...\}$ come one by one, each user is *single-minded*, i.e., each user is only interested in a particular bundle of items. For example, user $u$'s interested bundle is $I_u = \{i_1, i_2\}$ (or $I_u = \{1, 2\}$). The bundles can be sold fractionally, but the amount of each item in the sold bundle must be the same. Still considering the above example on user $u$, the seller may sell half bundle to $u$, i.e., half $i_1$ and half $i_2$. The seller must set the unit price and sell a certain number of bundles to each user on his/her arrival. In this paper, easy of computation and comparison, the unit price is defined on items, not on bundles, even though, one can convert to the other easily. For example, the seller sells 1.5 bundles $I_u = \{1, 2\}$ to $u$ at price 3, then the unit price is 1. Actually, if we define the unit price on bundle, the results in this paper still hold, because unit price on bundle $I$ can be regarded as $|I|$ times the unit price on item. However, define the unit price on item is more convenient since different bundles can be compared easily. Each $u_i$ has his/her value function $v_i(\cdot)$ such that $v_i(x)$ is the highest unit price $u_i$ is willing to pay for $x$ bundles. Generally, the more bundles an user buys, the lower unit price he expects. Thus, in this paper, we assume that $v_i(x)$ is non-increasing. Let $h$ be the highest value among all $v_i(x)$, i.e., $v_i(x) \leq h$ for all $i$ and $x$. When user $u$ comes with his interested bundle $I_u$, assuming that the seller sets unit price $p$ and assigns $\ell$ bundles to $u$. If $p > v_i(\ell)$, user $u$ cannot accept this price, thus, no bundle is bought by $u$. Otherwise, $p \leq v_i(\ell)$, $u$ will accept this price and pay $p \cdot \ell \cdot |I_u|$ to the seller.

To understand this model clearly, consider the example as shown in Figure 1. The seller has $k = 3$ types of items, and each type contains $m = 2$ items. There are three single-minded users who want to buy these items. User 1's interested bundle is $I_1 = \{1, 2\}$; the unit prices at which user 1 is willing to buy his interested bundle are 5 and 5 for buying 1 and 2 bundles respectively, i.e., $v_1(1) = v_2(1) = 5$. User 2's interested bundle is $I_2 = \{2, 3\}$; the unit prices at which user 2 is willing to buy his interested bundle are 6 and 4 for buying one and two bundles respectively, i.e., $v_2(1) = 6$ and $v_2(2) = 4$. User 3's interested bundle is $I_3 = \{1, 3\}$; the unit prices at which user 3 is willing to buy his interested bundle are 7 and 4 for buying one and two bundles respectively, i.e., $v_3(1) = 7$ and $v_3(2) = 4$.

When user 1 comes, to maximize the seller's revenue on this user, the seller will assign 2 bundles of $I_1 = \{1, 2\}$ at unit price 5 to him. When user 2 and user 3 come, there is no item 1 and item 2 left. In this case, user 2 and user 3 cannot buy anything and the total revenue of the seller is 20. However, the optimal strategy can achieve a total revenue of 36 by assigning one bundle of $I_1$ at unit price 5 to user 1, one bundle of $I_2 = \{2, 3\}$ at unit price 6 to user 2, and one bundle of $I_3 = \{1, 3\}$ at unit price 7 to user 3.

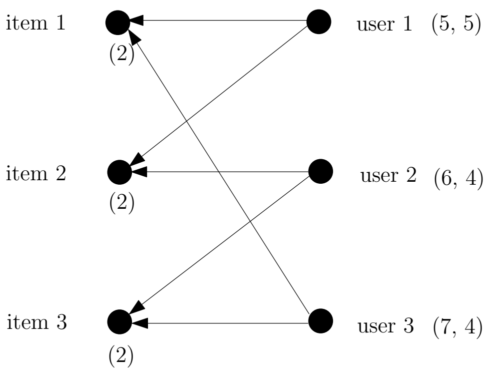


**Fig. 1.** An example of online pricing for multi items.

We consider the online version of this problem, i.e., before the $i$-th user comes, the seller has no information of the $j$-th user for $j \geq i$. To measure the performance of online algorithms, competitive analysis is generally used, i.e., to compare the outputs between the online algorithm and the optimal offline algorithm (which assumes all information is known in advance). Given the seller with item set $B$ and an user sequence $\sigma$, let $A(B, \sigma)$ and $O(B, \sigma)$ denote the total revenue received by the seller according to the online algorithm $A$ and the optimal offline algorithm $O$, respectively. The competitive ratio of the algorithm $A$ is

$$R_A = \sup_{B,\sigma} \frac{O(B, \sigma)}{A(B, \sigma)}.$$

The pricing problem for items has been well studied in the past few years. Both multi-type and single-type items have been considered. Previous work has been focused on two supply models: the unlimited supply model [1, 2, 6, 10, 12] where the number of each type of item is unbounded and the limited supply model [1, 3–5, 7, 11, 13, 14] where the number of each type of item is bounded by some value. As for the users, there are several users' behaviors studied, including single-minded [7–10, 12, 14] (each user is only interested in a particular set of items), unit-demand [2–6, 12, 14] (each user will buy at most one item in total) and envy free [1, 5, 7, 10, 12] (after the assignment, no user would prefer to be assigned a different set of items with the designated prices, loosely speaking, each user is happy with his/her purchase). Most of the previous studies have considered a combination of the above scenarios (e.g. envy-free pricing for single-minded users when there is unlimited supply). In [15], Zhang et al. considered a more practical and realistic model in the sense that the seller has a finite number of items (one type of item with limited supply) and users can demand more than one items and arrive online. They proved that the lower bound of the competitive ratio is $O(\log h)$, moreover, they gave a deterministic online algorithm with competitive ratio $O(\log h)$, where $h$ is the highest unit price.

In this paper, besides generalizing the problem to more than one type of items and bundles of items required by users, the idea used in our proposed online pricing algorithm is rooted by considering the amount of remaining items in additional to the user's value function in determining the price and amount of items to be sold. The proof in establishing the upper bound is more complicated by employing a fine price partition. The result in this paper can also match the optimal result $O(\log h)$ [15] asymptotically when there is only one type of item ($k = 1$) in the model.

This paper is organized as follows: Section 2 proves the lower bound of the competitive ratio for this variant to be $O(\log h + \log k)$; in Section 3, a deterministic online algorithm whose competitive ratio is $O(\sqrt{k} \log h \log k)$ is given.

## 2 Lower Bound of the Competitive Ratio

In this part, our target is to show the lower bound of the competitive ratio of the online pricing problem for multi type of items is $O(\log h + \log k)$, where $h$ is the highest unit price and $k$ is the number of types.

To easily analyze the lower bound, we assume that $h = 2^\ell$ and $k = 2^j$, i.e., $\log h$ and $\log k$ are both integers. The lower bound is proved step by step. In each step, the adversary sends an user to the seller. In this proof, all value functions are flat, particularly, the value function $v(x)$ is some power of 2 for all $x$.

In the first $\log k$ steps, the value function $v(x) = 1$.

*Step 1:*

The adversary sends user $u_1$ to the seller, and $u_1$'s interested bundle is $\{1\}$.

If the seller assigns $x_1$ bundles to $u_1$ such that $x_1 \leq m/(\log h + \log k)$, the adversary stops. In this case, the revenue of the seller is at most $m/(\log h+\log k)$, while the maximal revenue is $m$ by assigning all $m$ bundles to $u_1$. Thus, the ratio in this case is at least $O(\log h + \log k)$.

Otherwise, the seller assigns more than $m/(\log h + \log k)$ bundles to $u_1$. The adversary will send the next user to the seller.

*Step 2:*

The adversary sends user $u_2$ to the seller, $u_2$'s interested bundle is $\{1, 2\}$.

If the seller assigns $x_2$ bundles to $u_2$ such that $x_1 + x_2 \leq 2m/(\log h + \log k)$, the adversary stops. In this case, the revenue of the seller is at most $x_1 + 2x_2 \leq 3m/(\log h + \log k)$, while the optimal revenue is $2m$ by assigning all $i_1$ and $i_2$ to user $u_2$. Thus, the ratio in this case is at least $O(\log h + \log k)$.

Otherwise, the seller assigns $x_2$ bundles to $u_2$ such that $x_1+x_2 > 2m/(\log h+ \log k)$.

...

*Step $\ell$: ($1 < \ell \leq \log k$)*

The adversary sends user $u_\ell$ to the seller such that $u_\ell$'s interested bundle is $\{1, 2^{\ell-2}+1--2^{\ell-1}\}$.

If the seller assigns $x_\ell$ bundles to $u_\ell$ such that $\sum_{t=1}^{\ell} x_t \leq \ell \cdot m/(\log h + \log k)$, the adversary stops. In this case, the revenue achieved by the seller is

$$x_1 + 2x_2 + ... + (2^{\ell-2}+1)x_\ell \tag{1}$$

**Lemma 1.** *If the adversary stops at step $\ell$, the total revenue is at most*

$$\frac{(2^{\ell-1}+\ell-1)m}{\log h + \log k}.$$

*Proof.* From previous steps, we have

$$\sum_{p=1}^{t} x_p \geq \frac{tm}{\log h + \log k} \qquad (1 \leq t < \ell),$$

thus,

$$\sum_{p=t+1}^{\ell} x_p \leq \frac{(\ell-t)m}{\log h + \log k} \qquad (1 \leq t < \ell).$$

Therefore, Equation (1) achieves the maximal value $\frac{(2^{\ell-1}+\ell-1)m}{\log h+\log k}$ when each $x_p$ equals to $m/(\log h + \log k)$. □

The optimal revenue is $(2^{\ell-2}+1)m$ by assigning all $m$ bundles to $u_\ell$. Therefore, in this case, the competitive ratio is still bounded by $O(\log h + \log k)$.

Otherwise, the adversary sends the next user to the seller.

...

In the following $\log h$ steps (step $\log k + 1 \leq \ell \leq \log k + \log h$), the interested bundles are $\{1, k/2+1--k\}$, and the value functions are $v(x) = 2^{\ell - \log k}$ at step $\ell$.

*Step* $\log k + 1$*:*

The adversary sends user $u_{\log k+1}$ to the seller.

If the seller assigns $x_{\log k+1}$ bundles to $u_{\log k+1}$ such that $\sum_{\ell=1}^{\log k+1} x_\ell \leq (\log k+1)m/(\log h+\log k)$, the adversary stops. In this case, the revenue achieved by the seller is

$$x_1 + 2x_2 + ... + (k/2+1)x_{\log k} + 2 \cdot (k/2+1)x_{\log k+1} \tag{2}$$

Similar to the proof in Lemma 1, we can find that the revenue achieved is at most

$$\frac{(3k/2 + \log k + 1)m}{\log k + \log h}.$$

The optimal revenue is $(k+2)m$ by assigning all $m$ bundles to $u_{\log k+1}$. Therefore, in this case, the competitive ratio is still bounded by $O(\log h + \log k)$.

Otherwise, the adversary sends the next user to the seller.

The analysis on Steps until Step $\log k + \log h - 1$ are similar to the above one.

*Step* $\log k + \log h$*:*
The adversary sends user $u_{\log k + \log h}$ to the seller and the seller assigns $x_{\log k + \log h}$ bundles to the user. Since all interested bundles include $i_1$, thus, the total number of all assigned bundles is no more than $m$. Thus, the adversary must stop at this step. Similar to the previous analysis, we can say that the ratio between the optimal solution and the revenue achieve by the online algorithm is at least $O(\log k + \log h)$.

Therefore, we have the following conclusion.

**Theorem 1** *For the online pricing for multi type of items, the lower bound of the competitive ratio is* $O(\log h + \log k)$.

## 3 Online Algorithm

To maximize the revenue of the seller on a particular user $u$ with interested bundle $I$, a straightforward idea is finding unit price $p$ and amount of bundles $b$ such that $p$ is acceptable when buying $b$ bundles, and $b \cdot p$ is maximized. If we assign $b$ bundles with unit price $p$ to $u$, the revenue is $b \cdot p \cdot |I|$, which is maximal.

In our algorithm, we assign unit price $2^\ell$ ($\ell \geq 0$) to each user. In this way, we have no need to consider all possible prices, and we will show that the performance doesn't be affected too much. Let $(b, p)$ be the assignment such that $b \cdot p \cdot |I|$ is maximal. W.l.o.g., suppose $2^i \leq p < 2^{i+1}$, note that $v(x)$ is non-increasing, we have $v^{-1}(2^{i+1}) \leq b \leq v^{-1}(2^i)$. Thus,

$$b \cdot p \leq v^{-1}(2^i) \cdot p \leq v^{-1}(2^i) \cdot 2^{i+1} = 2 \cdot v^{-1}(2^i) \cdot 2^i.$$

If we choose the unit price equals to some power of 2, $(v^{-1}(2^i), 2^i)$ is a candidate of the assignment, which is at least half of the maximal value.

In our algorithm, we partition the amount of each type of item into $\lfloor \log h \rfloor$ stages, from stage 1 to stage $\lfloor \log h \rfloor$. The amount of items in stage $i$ can be only assigned with unit price $2^{i-1}$. Furthermore, partition the items in each stage into $\lfloor \log k \rfloor + 1$ levels, from level 0 to level $\lfloor \log k \rfloor$. For type $i$, items in level $\ell$ can be only assigned to users such that type $i$ is in the user's interested bundle and the size of the bundle is within $[2^{\ell-1}+1, 2^\ell]$. For example, an user $u$'s interested bundle is $\{1, 2, 3, 4\}$, thus, in our algorithm, we choose items from level 2 in some stage to satisfy this user.

Let $\delta_i^{s,t}$ denote the available amount of items of type $i$ in stage $s$ level $t$. Initially, $\delta_i^{s,t} = m/(\lfloor \log h \rfloor(\lfloor \log k \rfloor + 1)$.

Next, we will formally describe the pricing algorithm for multi type of items.

According to the algorithm Pricing, if an user $u$ with interested bundle $I$ cannot be satisfied, that means in each acceptable stage $s$, at least one type of item in $I$ at level $\lceil \log |I| \rceil$ are all assigned to other users.

**Algorithm 1 Pricing**

Let $I$ be the interested bundle of the coming user $u$.
Let $\ell = \lceil \log |I| \rceil$ $\quad\triangleright$ $\ell$ denotes the level which may assign items to user $i$.
Let $x_j$ be the largest amount of bundles that user $i$ is willing to buy given unit price $2^j$ and satisfying $x_j \leq m$.
Let $y_j = \min\{x_j, \min_{i \in I}\{\delta_i^{j+1,\ell}\}\}$.
Let $s = \arg\max_j y_j \cdot 2^j$ such that $y_j > 0$.
**if** no such $s$ exists **then**
    Assign 0 bundles to user $u$.
**else**
    Set the unit price $p = 2^s$.
    Assign $y_s$ bundles to user $u$.
    $\delta_i^{s+1,\ell} = \delta_i^{s+1,\ell} - y_s$ for all $i \in I$.
**end if**

For an user sequence $\{u_1, u_2, ...\}$, let $ALG$ denote the total revenue received from the algorithm Pricing, let $OPT$ be the revenue achieved by the optimal algorithm. Next, we give the competitive ratio of the algorithm Pricing, i.e., prove the upper bound of the ratio between $OPT$ and $ALG$.

After the processing of Pricing, for each type of item, some levels in some stages are full, the others still contain some available items. Classify all levels into two classes: $L_i^f$ denotes the levels of type $i$ which are full, i.e., if $\delta_i^{s,t} = 0$, level $t$ in stage $s$ of type $i$ belongs to $L_i^f$; $L_i^n$ denotes the levels of type $i$ which contain available items, i.e., $\delta_i^{s,t} > 0$ for the corresponding levels.

Compare the assignments from the optimal algorithm and Pricing, we also partition the assignments from the optimal algorithm into two classes according to $L_i^f$ and $L_i^n$. In the optimal solution, consider the assignment to an user with interested bundle $I$, suppose the unit price $p \in [2^\ell, 2^{\ell+1})$. In the assignment from Pricing, if there exist $i \in I$ such that $\delta_i^{\ell+1,\lceil \log |I| \rceil} = 0$, i.e., this level belongs to $L_i^f$, we say the revenue of this assignment from the optimal algorithm belongs to $O_f$. Otherwise, if for any $i \in I$, $\delta_i^{\ell+1,\lceil \log |I| \rceil} > 0$ in the assignment w.r.t. Pricing, the revenue of this assignment belongs to $O_n$.

Moreover, we partition the assignment according to the size of the bundle. If $|I| \leq \sqrt{k}$, we say the size of the bundle is *small*, otherwise, the size is *large*. The revenue of the assignment from the optimal algorithm is further partitioned into four classes: $O_f^s$, $O_f^l$, $O_n^s$, and $O_n^l$, where $O_f^s$ and $O_n^s$ denote the revenue from small bundles, $O_f^l$ and $O_n^l$ denote the revenue from large bundles. Note that $O_f^s + O_f^l = O_f$ and $O_n^s + O_n^l = O_n$. Next, we compare these four classes with $ALG$ respectively.

**Lemma 2.** $\frac{O_f^s}{ALG} \leq O(\sqrt{k} \cdot \log h \log k)$.

*Proof.* By the optimal algorithm, consider an assignment $A$ of a bundle $I$ with unit price $p \in [2^\ell, 2^{\ell+1})$. Assume that $|I| \leq \sqrt{k}$ and in the assignment from

Pricing, at least one type $i \in I$ at level $\lceil \log |I| \rceil$ in stage $\ell+1$ is full. The optimal revenue of the assignment $A$ is

$$p \cdot m \cdot |I| \tag{3}$$

From the algorithm Pricing, the revenue at level $\lceil \log |I| \rceil$ in stage $\ell+1$ of type $i$ is

$$2^\ell \cdot m/(\lfloor \log h \rfloor (\lfloor \log k \rfloor + 1)) \tag{4}$$

The ratio between (3) and (4) is

$$O(\sqrt{k} \cdot \log h \log k).$$

Suppose by the optimal algorithm, more than one assigned small bundles, $I_1$, $I_2$, ..., $I_j$ with unit prices $p_1$, $p_2$, ..., $p_j$ satisfy $p_{j'} \in [2^\ell, 2^{\ell+1})$ for $1 \leq j' \leq j$ and all $\lceil \log |I_{j'}| \rceil$ $(1 \leq j' \leq j)$ are equal, if these bundles share an item $i$ such that in the assignment from Pricing, level $\lceil \log |I_1| \rceil$ in stage $\ell+1$ of item $i$ is full, the total revenue for these bundles in the optimal solution is at most

$$\max\{p_{j'}\} \cdot m \cdot \max\{|I_{j'}|\} \quad 1 \leq j' \leq j \tag{5}$$

Compare the value in Equation (5) with the revenue from Pricing at level $\lceil \log |I| \rceil$ in stage $\ell+1$ of type $i$ (Equation (4)), the ratio is also

$$O(\sqrt{k} \cdot \log h \log k).$$

From the optimal pricing, all assignments in $O_f^s$ can be partitioned into parts, each part contains the assignments mentioned above. Since the total revenues on such full levels is a lower bound of $ALG$, we have

$$\frac{O_f^s}{ALG} \leq O(\sqrt{k} \cdot \log h \log k).$$

□

**Lemma 3.** $\frac{O_f^l}{ALG} \leq O(\sqrt{k} \cdot \log h \log k)$.

*Proof.* By the optimal algorithm, consider some amount of bundle $I$ with unit price $p \in [2^\ell, 2^{\ell+1})$ is assigned to an user. Assume that $|I| > \sqrt{k}$ and there exist an item $i \in I$ such that in the assignment from Pricing, level $\lceil \log |I| \rceil$ in stage $\ell+1$ of item $i$ is full. Note that in Pricing, items in this level can be only assigned to bundles with size in between $(2^{\lceil \log |I| \rceil - 1}, 2^{\lceil \log |I| \rceil}]$, thus, the total revenue on level $\lceil \log |I| \rceil$ in stage $\ell+1$ is at least

$$\frac{2^\ell \cdot m \cdot 2^{\lceil \log |I| \rceil - 1}}{\lfloor \log h \rfloor (\lfloor \log k \rfloor + 1)} \tag{6}$$

Note that the optimal revenue for bundles with unit price $p \in [2^\ell, 2^{\ell+1})$ is at most

$$2^{\ell+1} \cdot m \cdot k \tag{7}$$

The ratio between the above two equations is $O(\sqrt{k} \cdot \log h \log k)$. Combine all revenues in $O_f^l$, we can say that this lemma is true. □

**Lemma 4.** $\frac{O_n^s}{ALG} \leq O(\sqrt{k} \cdot \log h \log k)$.

*Proof.* Again, consider an assigned bundle $I$ from the optimal algorithm, such that the revenue on $I$ belongs to $O_n^s$ and the unit price is $p \in [2^\ell, 2^{\ell+1})$. The algorithm Pricing chooses the unit price $2^j$ such that $2^j \cdot y_j$ is maximized. Note that $(2^\ell, y_\ell)$ is also a candidate for satisfying bundle $I$.

- If Pricing assigns $(2^\ell, y_\ell)$, since after the assignment, $\delta_i^{\ell+1,\lceil \log I \rceil} > 0$ for all $i \in I$, the revenue achieved on $I$ by Pricing is at least half of the optimal revenue on this bundle.
- Otherwise, Pricing assigns $(2^j, y_j)$ such that $j \neq \ell$. From the choosing criteria, $2^j \cdot y_j \geq 2^\ell \cdot y_\ell$.
  - If $\delta_i^{\ell+1,\lceil \log I \rceil} > y_\ell$ for all $i \in I$, from above analysis, we can say the revenue on $I$ by Pricing is at least half of the optimal revenue on this bundle.
  - Otherwise, $\delta_i^{\ell+1,\lceil \log I \rceil} = y_\ell$ for some $i \in I$. Since $2^j \cdot y_j \geq 2^\ell \cdot y_\ell$, the revenue achieved on $I$ by Pricing plus the current revenue on level $\lceil \log I \rceil$ in stage $\ell + 1$ of type $i$ is at least

$$\frac{2^\ell \cdot m}{\lfloor \log h \rfloor (\lfloor \log k \rfloor + 1)} \tag{8}$$

    This is because if we assign $(2^\ell, y_\ell)$ for this bundle, level $\lceil \log I \rceil$ in stage $\ell + 1$ of type $i$ is full.
    Similar to the analysis in Lemma 2, suppose more than one assigned small bundles $I_1$, $I_2$, ... with unit price $p_1$, $p_2$, ... within $[2^\ell, 2^{\ell+1})$ and the sizes of these bundles are all within $(2^{\lceil \log |I_1| \rceil - 1}, 2^{\lceil \log |I_1| \rceil}]$, if all these bundles share type $i$, the total revenue on such bundles by the optimal scheme is at most

$$2^{\ell+1} \cdot m \cdot 2^{\lceil \log |I_1| \rceil} \tag{9}$$

    The ratio between the above two terms is $O(\sqrt{k} \cdot \log h \log k)$.

Mapping the assignments of $O_n^s$ to the corresponding assignments of Pricing described above, each assignment by Pricing is counted at most TWICE. Combining the above analysis, we can say that

$$\frac{O_n^s}{ALG} \leq O(\sqrt{k} \cdot \log h \log k).$$

□

**Lemma 5.** $\frac{O_n^l}{ALG} \leq O(\sqrt{k} \cdot \log h \log k)$.

*Proof.* The proof of this lemma is similar to the proofs in Lemma 3 and Lemma 4. Consider an assigned bundle $I$ from the optimal algorithm, such that the revenue on $I$ belongs to $O_n^l$ and the unit price is $p \in [2^\ell, 2^{\ell+1})$. The algorithm Pricing chooses the unit price $2^j$ such that $2^j \cdot y_j$ is maximized. Since any level $\lceil \log |I| \rceil$ in stage $\ell + 1$ of item $i \in I$ is not full, $(2^\ell, y_\ell)$ is a candidate.

– If Pricing assigns $(2^\ell, y_\ell)$, since after the assignment, $\delta_i^{\ell+1,\lceil \log I\rceil} > 0$ for all $i \in I$, the revenue achieved on $I$ is at least half of the optimal revenue on this bundle.
– Otherwise, Pricing assigns $(2^j, y_j)$ such that $j \neq \ell$. From the choosing criteria, $2^j \cdot y_j \geq 2^\ell \cdot y_\ell$.
  - If $\delta_i^{\ell+1,\lceil \log I\rceil} > y_\ell$ for all $i \in I$, from above analysis, we can say the revenue on $I$ by Pricing is at least half of the optimal revenue on this bundle.
  - Otherwise, $\delta_i^{\ell+1,\lceil \log I\rceil} = y_\ell$ for some $i \in I$. Since $2^j \cdot y_j \geq 2^\ell \cdot y_\ell$, the revenue achieved on $I$ by Pricing plus the current revenue on level $\lceil \log I\rceil$ in stage $\ell + 1$ is at least

$$\frac{2^\ell \cdot m \cdot 2^{\lceil \log |I|\rceil - 1}}{\lfloor \log h\rfloor(\lfloor \log k\rfloor + 1)} \quad (10)$$

This is because if we assign $(2^\ell, y_\ell)$ for this bundle, level $\lceil \log I\rceil$ in stage $\ell + 1$ of type $i$ is full, and the size of each bundle on this level is at least $2^{\lceil \log |I|\rceil - 1}$.

From the optimal scheme, the total revenue on assignments from unit price in between $[2^\ell, 2^{\ell+1})$ and bundle size in between $(2^{\lceil \log |I|\rceil - 1}, 2^{\lceil \log |I|\rceil}]$ is at most

$$2^{\ell+1} \cdot m \cdot k \quad (11)$$

The ratio between the above two terms is $O(\sqrt{k} \cdot \log h \log k)$

Mapping the revenue of assignments in $O_n^l$ to the assignment by Pricing, from the above analysis, each assignment is counted at most twice. Thus,

$$\frac{O_n^l}{ALG} \leq O(\sqrt{k} \cdot \log h \log k).$$

□

Now we give the main conclusion of this paper.

**Theorem 2** *The competitive ratio of the algorithm Pricing is at most*

$$O(\sqrt{k} \cdot \log h \log k).$$

*Proof.* From the definition of $O_f^s$, $O_f^l$, $O_n^s$, and $O_n^l$, these four classes are disjoint. Note that if a requested bundle cannot be satisfied, it must belongs to $O_f^s$ or $O_f^l$. Combining Lemma 2 until Lemma 5, we can say that the competitive ratio of the algorithm Pricing is $O(\sqrt{k} \cdot \log h \log k)$. □

## References


1. N. Balcan, A. Blum, and Y. Mansour. Item pricing for revenue maximization. In Proc. of the 9th ACM Conference on Electronic Commerce (EC'08), pp. 50-59.
2. Nikhil Bansal, Ning Chen, Neva Cherniavsky, Atri Rurda, Baruch Schieber, Maxim Sviridenko. Dynamic pricing for impatient bidders. ACM Transactions on Algorithms Vol. 6 Issue 2, March 2010.
3. Patrick Briest. Uniform Budgets and the Envy-Free Pricing Problem. Proceedings of the 35th international colloquium on Automata, Languages and Programming, Part I, July 07-11, 2008, Reykjavik, Iceland
4. Patrick Briest , Piotr Krysta. Buying cheap is expensive: hardness of non-parametric multi-product pricing. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, p.716-725, January 07-09, 2007, New Orleans, Louisiana
5. Ning Chen and Xiaotie Deng. Envy-Free Pricing in Multi-item Markets In Proc. of the 35th international colloquium on Automata, Languages and Programming (ICALP 2010), pp. 418-429.
6. Ning Chen, Arpita Ghosh, Sergei Vassilvitskii. Optimal envy-free pricing with metric substitutability. In Proc. of the 9th ACM conference on Electronic commerce (EC'08), pp. 60-69.
7. Maurice Cheung and Chaitanya Swamy. Approximation Algorithms for Single-minded Envy-free Profit-maximization Problems with Limited Supply. In Proc. of 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008), pp. 35-44
8. Khaled Elbassioni, Rajiv Raman , Saurabh Ray , René Sitters. On Profit-Maximizing Pricing for the Highway and Tollbooth Problems. Proceedings of the 2nd International Symposium on Algorithmic Game Theory, p.275-286, October 18-20, 2009, Paphos, Cyprus
9. Khaled Elbassioni, René Sitters, and Yan Zhang. A quasi-PTAS for profit-maximizing pricing on line graphs. In Proc. of the 15th annual European conference on Algorithms (ESA'07), pp. 451-462.
10. Amos Fiat and Amiram Wingarten Envy, Multi Envy, and Revenue Maximization. In Proc. of the 5th International Workshop on Internet and Network Economics (WINE 2009), pp. 498-504.
11. Alexander Grigoriev, Joyce van Loon, René Sitters and Marc Uetz How to Sell a Graph: Guidelines for Graph Retailers. In Proc. of the 32nd International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2006), pp. 125-136.
12. V. Guruswami, J. Hartline, A. Karlin, D. Kempe, C. Kenyon, and F. McSherry. On Profit-Maximizing Envy-Free Pricing. In Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 05), pp. 1164-1173, 2005.
13. Sungjin Im, Pinyan Lu and Yajun Wang Envy-Free Pricing with General Supply Constraints. In Proc. of the 6th International Workshop on Internet and Network Economics (WINE 2010), pp. 483-491.
14. Robert Krauthgamer, Aranyak Mehta , Atri Rudra. Pricing commodities. Theoretical Computer Science, v.412 n.7, p.602-613, February, 2011
15. Yong Zhang, Francis Chin, Hing-Fung Ting. Competitive Algorithms for Online Pricing. In Proc. of the 17th Annual International Computing and Combinatorics Conference (COCOON 2011), LNCS 6842, pp. 391-401.