Arbitrage-free Pricing in User-based Markets

Chaolun Xia
Rutgers University
cx28@cs.rutgers.edu

S. Muthukrishnan
Rutgers University
muthu@cs.rutgers.edu

ABSTRACT

Users have various attributes, and in user-based markets there are buyers who wish to buy a target set of users with specific sets of attributes. The problem we address is that, given a set of demand from the buyers, how to allocate users to buyers, and how to price the transactions. This problem arises in online advertising, and is particularly relevant in advertising in social platforms like Facebook, LinkedIn and others where users are represented with many attributes, and advertisers are buyers with specific targets. This problem also arises more generally in selling data about online users, in a variety of data markets.

We introduce *arbitrage-free* pricing, that is, pricing that prevents buyers from acquiring a lower unit price for their true target by strategically choosing substitute targets and combining them suitably. We show that *uniform* pricing – pricing where all the targets have identical price – can be computed in polynomial time, and while this is arbitrage-free, it is also a logarithmic approximation to the maximum revenue arbitrage-free pricing solution. We also design a different arbitrage-free non-uniform pricing – pricing where different targets have different prices – solution which has the same guarantee as the arbitrage-free uniform pricing but is empirically more effective as we show through experiments. We also study more general versions of this problem and present hardness and approximation results.

KEYWORDS

Pricing; User Attribute; Arbitrage; Arbitrage-free; Revenue Maximization; Advertising; Data; Market; Algorithm

ACM Reference Format:
Chaolun Xia and S. Muthukrishnan. 2018. Arbitrage-free Pricing in User-based Markets. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 9 pages.

1 INTRODUCTION

User-based markets are a central part of the Internet Economy. In user data markets, many companies are selling opt-in email addresses. In online advertising markets, advertisers want to buy the impressions of users with specific sets of attributes, e.g. a luxury car company may prefer to show ads to rich users.

In such user-based markets, the core value of a user arises from her attributes. In TowerData, buyers can purchase the emails of users with specific demographics. Google AdWords, for example, the largest online ad network, allows advertisers to target users based on demographics and search terms. Ad markets run by online social networks, including Facebook, LinkedIn and Twitter, offer much finer targeting controls over user attributes with detailed information which is shared directly by users, inferred from user daily activities or purchased from third parties. This includes users' educational records, past and present employment experience, significant life events like changes in marital status or birth of a baby, etc. Twitter allows advertisers to target users by topics that the users are interested in.

In such markets, buyers can purchase their *target* users[1] through the query system provided by the market. Let q_i denote a simple selection query with conditions over user attributes, e.g. q_i ="Gender:1" returns all the male users. Let U_i be the set of all the users satisfying query q_i. A buyer can specify the users with a query and purchase them. Therefore, the market owner needs to solve a pricing problem – how to price all the queries from buyers that return users with different attributes? In this paper, we consider the following posted pricing model. Let p_i denote the price of query q_i, i.e. the price of any user $\mathbf{u}$ satisfying query q_i (i.e. $\mathbf{u} \in U_i$). A buyer needs to pay $n \cdot p_i$ if he purchases $n \in \{1, \ldots, |U_i|\}$ users in U_i. The practical need behind this pricing model is that a single target user can provide positive utility to the buyer. Moreover, we assume that it would not cause much trouble to the buyer if he gets additional users that do not satisfy his target query[2].

The above pricing model benefits from the *versioning* theory for pricing information goods [24]. This theory proposes that different buyers may use an information product in different ways, and the market should provide different versions for such a product at different prices. In user-based markets, a user with multiple attributes can potentially have different versions. For example, a user who is a programmer interested in cars, can be priced and sold as at least two versions, including as a user interested in cars as one version for car dealers and as a programmer as another version for IT companies on hiring. Versioning theory is needed in such user-based markets because a user may be retrieved by multiple queries if she has more than one attribute.

We point out that such a pricing model, however, may suffer **version-arbitrage** (see Definition 6). In user-based markets, version-arbitrage occurs if two queries q_i and $q_{i'}$ return similar user sets but p_i and $p_{i'}$ differ a lot. If version-arbitrage exists, a buyer who really wants q_i (or $q_{i'}$) might purchase $q_{i'}$ (or q_i) instead. Version-arbitrage is caused by the fact that a user with multiple attributes potentially satisfies many queries. We use an example to illustrate the version-arbitrage and its difference from *determinancy* arbitrage in [17].

Example 1. *Let q_1 ="Income > 100" and q_2 ="Income > 101" be two queries, i.e. q_1 (or q_2) returns all the users with income higher*

[1] "Buying a user" is short for buying, for example, the impression of a user in advertising markets.

[2] This assumption is obviously true in advertising markets, i.e. showing an ad to a non-target user will not decrease the sale. In data markets, it is true if the buyer does not want aggregate results.

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

than 100 *(or* 101*). If* $p_2 < p_1$*, the version-arbitrage exists: a savvy buyer who wants to buy a user satisfying* q_1 *will buy a user satisfying* q_2 *instead (assuming* $U_2 \neq \emptyset$*) because any user satisfying* q_2 *satisfies* q_1 *with probability 1. However, in non-trivial databases,* q_1 *(or* q_2*) does not determine* q_2 *(or* q_1*) [17].* □

Motivated by the discussion above, we study pricing queries with conditions over user attributes, and seek revenue-maximizing pricing for a given demand. In particular, we formulate *arbitrage-free* pricing problem in user-based markets and our contributions are:

- Any uniform pricing where all the queries have identical price is arbitrage-free. We show that the optimal uniform pricing can be computed in polynomial time. We also show that this is an $O(\log min\{\mathcal{U}, \sum_{j=1}^{\mathcal{B}} d_j\})$ approximation to the optimal, possibly non-uniform arbitrage-free pricing, where $\mathcal{U}$ is the total number of users and $\sum_{j=1}^{\mathcal{B}} d_j$ is total number of users requested by the buyers. Besides, we show that this approximation bound is tight for uniform pricing solutions.
- We design a different, efficient greedy algorithm to compute the arbitrage-free non-uniform pricing with the same $O(\log \min\{\mathcal{U}, \sum_{j=1}^{\mathcal{B}} d_j\})$ guarantee. But by experiments, we show that its revenue is significantly larger than that of the optimal uniform pricing.
- We consider a generalized setting where a buyer has a *minimal* demand on his target users. In previous setting, the allocation problem (given a pricing) is polynomial time solvable. In this setting, we prove that both the allocation problem and pricing problem are not only NP-hard, but also hard to approximate. We present an $O(D)$ approximate allocation algorithm where D is the largest minimal demand. Turning to the pricing problem, we present a polynomial algorithm, that – based on the approximate allocation – computes a uniform pricing and we show that it is an $O(D \log \min\{\mathcal{U}, \sum_{j=1}^{\mathcal{B}} d_j\})$ approximation to the optimal arbitrage-free pricing.

Due to space limitation, the missing proofs for some theories are available in the full technical report[3].

2 RELATED WORK

Revenue maximizing and envy-free pricing. Pricing is a well-studied area in Economics. In particular, the envy-free pricing [12, 13, 20] in Walrasian Equilibrium [23] is relevant to our work. In recent years, envy-free pricing is studied in various settings [4, 5, 9, 10, 14, 15]. [13] first addresses the computational issue of envy-free pricing. They show that the problem is NP-hard even for the two special cases where the buyers are either unit-demand or single-minded. For the latter case, the uniform pricing provides a logarithmic approximation in terms of the number of buyers and the number of items. In the unlimited supply setting, [1] uses a randomized single price to achieve expected revenue within a logarithmic factor of the total social welfare for buyers with general valuation functions. [2] proves that the envy-free pricing problem in a graph where items are edges is NP-hard, and provides a better approximation algorithm than [13] for sparse instances. [25] claims some equivalency between envy-free pricing and a pricing free of determinancy-arbitrage in data markets. However, envy-free pricing is essentially different from the arbitrage-free pricing in the user markets we considered, because not every (revenue-maximizing) envy-free pricing is a pricing free of version-arbitrage, and it is unclear how to convert an envy-free pricing to an arbitrage-free pricing while maximizing the revenue.

Arbitrage-free pricing in data markets. The issue of query-based price arbitrage has gained much attention in data markets [6, 17, 18, 21, 22, 25]. [17] introduces the notion *arbitrage-freeness* to query pricing. They define that a query q is *determined* by a set Q of queries on database instance $\mathcal{D}$ if the answer of q can be inferred from the answer of Q on $\mathcal{D}$. Determinancy-arbitrage occurs if q is determined by Q and the price of q is less than the total price of all the queries in Q. However, determinancy-arbitrage is foundamentally different from version-arbitrage, which can be easily seen in Example 1 and Definition 6.

Price arbitrage in OSN advertising markets. Price arbitrage has been discovered and exploited in OSN advertising markets. [8] first exploits price arbitrage in topic targeting, and [7] proposes a more complex arbitrage strategy through path combination. [26] shows through data analysis that arbitrage exists in both Facebook and LinkedIn ad markets, and proposes strategies to exploit arbitrage to benefit advertisers. Our work is partially motivated by these arbitrage strategies, and our arbitrage-free pricing can make these strategies infeasible in online advertising markets.

3 PRELIMINARIES

3.1 Pricing Model

The market provides N queries for buyers. Let q_i be a *query* where $i \in \{1, \ldots, N\}$. For example, q_i="Income>100k ∧ Gender:Female" returns all the female users with income higher than 100k. Different from a general database query, the notion *query* in this paper can be viewed as a simplified selection rule over user attributes. A user is said to satisfy q_i if she can be retrieved by q_i. For a buyer, we assume that there exists a query that represents his true target. The pricing is over queries, and the price p_i of query q_i is the **unit** price, i.e. the price per user retrieved by q_i. For example, if $p_i = \$2$, a buyer needs to pay \$10 for 5 users satisfying q_i.

3.2 Other Notations

$[n]$ denotes the integer set $\{1, \ldots, n\}$. Let U be the set of all the users, and we define that $\mathcal{U} \triangleq |U|$. Let $\mathbf{u} \in U$ be a user. $u^i = 1$ if $\mathbf{u}$ satisfies q_i where $i \in [N]$, otherwise $u^i = 0$. Let M be the quantity that $M \triangleq \sum_{i=1}^{N} |\{\mathbf{u}|u^i = 1, \mathbf{u} \in U\}|$. Let B be the set of buyers, and we define that $\mathcal{B} \triangleq |B|$. Buyer $j \in [\mathcal{B}]$ is denoted by a triplet (t_j, d_j, c_j) indicating that he wants to buy at most $d_j \in \mathbb{Z}_+$ users (as *demand*) satisfying q_{t_j} (as *target*) where $t_j \in [N]$, and $c_j \in \mathbb{R}_+$ is the *maximum cost* that he is willing to pay for each target user. $c_j \cdot d_j$ can be viewed as the budget constraint of buyer j.

Let $A = (A_1, \ldots, A_{\mathcal{B}})$ be an allocation of (indivisible) users to buyers where A_j is the set of users allocated to buyer j. We assume that any user can be either sold to one[4] buyer or unsold.

[3] http://paul.rutgers.edu/~cx28/papers/user_pricing_full.pdf

[4] In our full technical report (see footnote[3]), we also discuss a more general setting where a user can be sold to multiple buyers with limited times. The algorithms for that setting are almost the same, so are the corresponding analyses.

Definition 2. *An allocation A is feasible if the three constraints are all satisfied $\forall j \in [\mathcal{B}]$:*

- *Target Constraint:* $\forall \mathbf{u} \in A_j$, $u^{t_j} = 1$;
- *Demand Constraint:* $|A_j| \leq d_j$;
- *Uniqueness Constraint:* $\forall j' \neq j$, $A_{j'} \cap A_j = \emptyset$.

Let $\mathbf{P} = (p_1, \ldots, p_N)$ be the pricing function over the N queries.

Definition 3. *$(\mathbf{P}, A)$ is feasible if A is feasible and $\forall A_j \neq \emptyset$, $p_{t_j} \leq c_j$.*

Given $(\mathbf{P}, A)$[5], the revenue is $R(\mathbf{P}, A) \triangleq \sum_{j=1}^{\mathcal{B}} |A_j| \cdot p_{t_j}$. In this paper, we define and solve the following *pricing problem.*

Definition 4 (Pricing Problem). *Given (N, U, B) as the input, compute $\mathbf{P}$ and A such that $\mathbf{P}$ is arbitrage-free (see Definition 7) and $R(\mathbf{P}, A)$ is maximized.*

To solve the pricing problem, we need to define and solve the *allocation problem.*

Definition 5 (Allocation Problem). *Given $(N, U, B, \mathbf{P})$ as the input, compute $A = \text{argmax}_{A'} R(\mathbf{P}, A')$.*

Let $R(\mathbf{P}) \triangleq \max_A R(\mathbf{P}, A)$ be the optimal revenue of $\mathbf{P}$ (when the allocation is optimal).

3.3 Arbitrage-Free Pricing

In this part, we first formally introduce version-arbitrage and then define the arbitrage-free pricing. Note that, our arbitrage-free pricing is free of version-arbitrage, different from the arbitrage-free pricing in Koutris et al. [17] which is free of determinancy-arbitrage.

In user markets, version-arbitrage is the opportunity that a buyer is able to get a lower unit price (in expectation) of his target users by strategically choosing a substitute target (i.e. query) other than his true target. Assuming that any query is satisfied by at least one user, let $\pi(i|i')$ be the conditional probability that a user satisfies q_i if she satisfies $q_{i'}$:

$$\pi(i|i') = \frac{|\{\mathbf{u}|u^i = 1, u^{i'} = 1, \mathbf{u} \in U\}|}{|\{\mathbf{u}|u^{i'} = 1, \mathbf{u} \in U\}|}$$

We assume that a buyer, whose true target is q_i, has the *prior belief* that buying a user satisfying $q_{i'}$ is equivalent (in expectation) to buying a fraction $\pi(i|i')$ of a user satisfying q_i. Although this assumption is not necessarily practical in all the user-based markets due to unpredictable allocation rules, strategies based on this assumption (or similar ones) were studied for advertisers in online advertising markets [7, 8, 26]. Based on this assumption, we define the version-arbitrage as follows.

Definition 6 (Version-Arbitrage). *In a market with U, the pricing $\mathbf{P}$ contains* version-arbitrage *if $\exists i, i' \in [N]$, $p_{i'} < \pi(i|i') \cdot p_i$.*

It is easy to see that Example 1 is a special case where $\pi(1|2) = 1$ and $p_2 < p_1$, so arbitrage exists. Next, we define the arbitrage-free pricing.

Definition 7 (Arbitrage-free). *In a market with U, the pricing $\mathbf{P}$ is said to be* arbitrage-free *if $\forall i, i' \in [N]$, $p_{i'} \geq \pi(i|i') \cdot p_i$.*

[5]By default, we require $(\mathbf{P}, A)$ to be feasible and we will not explicitly mention this requirement later.

Notably, the arbitrage-free constraints are independent of buyers B, but only depends on U in the market. This is a desirable property that no matter whether buyers report their parameters truthfully [23] or not, the market can always make the pricing arbitrage-free.

4 UNIFORM PRICING

In this section, we first show that any uniform pricing is arbitrage-free. Then we show the optimal uniform pricing can be computed in polynomial time. Finally we prove that the optimal uniform pricing provides an $O(\log min\{\mathcal{U}, \sum_{j=1}^{\mathcal{B}} d_j\})$ guarantee to the optimal arbitrage-free pricing. A pricing is *uniform* if all its entries are identical, otherwise *non-uniform*. Let $\mathbf{p}_\xi$ denote the uniform pricing that $\forall i \in [N]$, $p_i = \xi$.

Proposition 1. *Any uniform pricing is arbitrage-free.*

It is easy to verify Proposition 1. Although we do not know whether to find the optimal arbitrage-free pricing is NP-hard or has polynomial time solutions, Proposition 1 provides a class of feasible solutions.

Theorem 2. *The optimal uniform pricing $\mathbf{p}_{\xi^*}$ can be computed in polynomial time $O(\mathcal{U}\mathcal{B}M + \mathcal{U}\mathcal{B}^2)$.*

To prove this theorem, it is enough to prove Lemma 3 and Corollary 5. Basically, Lemma 3 states that given any uniform pricing $\mathbf{p}_\xi$, the optimal allocation can be computed in polynomial time. Following Lemma 4, Corollary 5 shows that the optimal uniform price ξ^* must be one of the maximum costs of buyers.

Lemma 3. *The allocation problem for uniform pricing can be solved in polynomial time $O(\mathcal{U}M + \mathcal{U}\mathcal{B})$.*

Proof. We model the allocation problem when the pricing is $\mathbf{p}_\xi$ as a maxflow problem as follows. We introduce s and t as the source and sink respectively. We then introduce N nodes $y_1, \ldots, y_N$. Assuming all the users are indexed from 1 to $\mathcal{U}$ and $\mathbf{u}_k$ denotes the k-th user. For each $k \in [\mathcal{U}]$: (1) we introduce a node x_k and a directed edge from s to x_k with capacity 1; and (2) $\forall i \in [N]$, we introduce a directed edge from x_k to y_i with capacity 1 if $u_k^i = 1$. For each buyer j that $c_j \geq \xi$, introduce a node z_j, a directed edge from y_{t_j} to z_j and a directed edge from z_j to t with capacity d_j. It is easy to verify that the amount of the maximum flow f^* is the maximum number of sold users when the pricing is $\mathbf{p}_\xi$, thus producing the revenue $R(\mathbf{p}_\xi) = \xi \cdot f^*$. The allocation can also be easily inferred from the residual graph after the maxflow algorithm completes. We use the Ford-Fulkerson algorithm that runs in $O(|E| \cdot f^*)$. Since $f^* \leq \mathcal{U}$ and $|E| \leq M + 2\mathcal{B}$, the time complexity is $O(\mathcal{U}M + \mathcal{U}\mathcal{B})$. □

Next we show that it only needs to solve at most $\mathcal{B}$ allocation problems to compute the optimal uniform price ξ^* in Corollary 5. Before showing Corollary 5 that is specific to uniform pricing, we show a general result for any pricing in Lemma 4, which immediately implies Corollary 5 and will be used for proving Lemma 8 later.

Lemma 4. *$C \triangleq \{c_j | j \in [\mathcal{B}]\}$. For any $\mathbf{P}$ that $\exists p_i \notin C$, there exists $\mathbf{P}'$ (not necessarily arbitrage-free) that $R(\mathbf{P}') \geq R(\mathbf{P})$ and $\forall i \in [N], p_i' \in C$.*

Proof. W.l.o.g., we assume that the distinct values $\theta_1, \ldots, \theta_C$ in C are: $\theta_0 < \theta_1 < \ldots < \theta_C$ where $\theta_0 = 0$ is a dummy variable. Given such a $\mathbf{P}$, we construct the corresponding $\mathbf{P}'$ as follows. $\forall p_i \in C$, we still set $p'_i = p_i$; $\forall p_i > \theta_C$, we set p'_i to θ_C, and clearly no revenue is lost since no user can be sold at the price higher than θ_C; $\forall p_i \in (\theta_{k-1}, \theta_k)$, we set p'_i to θ_k, and no revenue is lost because any user who can be sold at p_i can be sold at θ_k. It is easy to verify that $R(\mathbf{P}') \geq R(\mathbf{P})$ and $\forall i \in [N], p'_i \in C$. □

Lemma 4 implies the fact that there exists a pricing (not necessarily arbitrage-free) with the optimal revenue and every entry in C. With similar proof (omitted), we have Corollary 5.

Corollary 5. *The optimal uniform price $\xi^* \in \{c_j | j \in [\mathcal{B}]\}$.*

Based on Lemma 3 and Corollary 5, the optimal uniform price ξ^* can be computed by Algorithm 1 that solves at most $\mathcal{B}$ allocation problems, which proves Theorem 2.

Algorithm 1 Optimal Uniform Pricing

Input: N, U and B
Output: ξ^*
1: $C \leftarrow \{c_j | j \in [\mathcal{B}]\}$
2: $\xi^* \leftarrow \text{argmax}_{\xi \in C} R(\mathbf{p}_\xi)$
3: **return** ξ^*

Theorem 6. *The optimal uniform pricing computed by Algorithm 1 is an $O(\log min\{\mathcal{U}, \sum_{j=1}^{\mathcal{B}} d_j\})$ approximation to the optimal arbitrage-free pricing.*

Proof. Let $C \triangleq \{c_j | j \in [\mathcal{B}]\}$. W.l.o.g., we assume that the distinct values $\theta_1, \ldots, \theta_C$ in C are $\theta_1 < \ldots < \theta_C$. Let O_k denote the maximum number of sold users when the uniform price is θ_k. It is true that $\forall k \in [C]$, $R(\mathbf{p}_{\theta_k}) = \theta_k O_k$. Besides, it is true that $O_1 \geq \ldots \geq O_C$, but the sequence of revenues $\theta_1 O_1, \ldots, \theta_C O_C$ is not necessarily monotone. Let $\mathbf{P}^*$ be the optimal pricing without the arbitrage-free constraint. $R(\mathbf{P}^*)$ is a trivial upper bound of the revenue of the optimal arbitrage-free pricing. To prove the theorem, we first prove Lemma 7 and Lemma 8.

Lemma 7. *In any feasible allocation for a pricing, the number of users sold at the price no less than θ_k is bounded by O_k, $\forall k \in [C]$.*

Proof. Assume there exists $\mathbf{P}$ for which we can find a feasible allocation A such that $\exists k \in [C]$, the number of sold users at prices no less than θ_k is larger than O_k. We can create an allocation A' from A as follows. For the users sold at the price no less than θ_k in A, we allocate them to the same buyers in A' at the price θ_k. For other users, we discard them. It is easy to verify that A' is feasible for the uniform price θ_k and $|A'| > O_k$, contradicting with that O_k is the maximum number of sold users for the uniform price θ_k. □

Lemma 8. $R(\mathbf{P}^*) \leq \theta_C O_C + \sum_{k=1}^{C-1} \theta_k (O_k - O_{k+1})$.

Proof. Let o_k denote the number of users sold at the price no less than θ_k in the optimal allocation for $\mathbf{P}^*$. Thus, $o_k - o_{k+1}$ is the number of users sold at the exact price θ_k. From Lemma 4, we know that every entry of $\mathbf{P}^*$ is in C, so its revenue can be formulated as:

$$\begin{aligned} R(\mathbf{P}^*) &= \theta_C o_C + \sum_{k=1}^{C-1} \theta_k (o_k - o_{k+1}) = \sum_{k=2}^{C} o_k(\theta_k - \theta_{k-1}) + o_1 \theta_1 \\ &\leq \sum_{k=2}^{C} O_k(\theta_k - \theta_{k-1}) + O_1 \theta_1 \qquad \text{(a)} \\ &= \theta_C O_C + \sum_{k=1}^{C-1} \theta_k (O_k - O_{k+1}) \end{aligned}$$

Inequality (a) is because (1) from Lemma 7, $\forall l \in [N]$, $o_k \leq O_k$ and (2) $\forall k \in \{2, \ldots, C\}$, $\theta_k \geq \theta_{k-1}$. □

Now we can prove Theorem 6. Since $R(\mathbf{p}_{\xi^*}) = \max\{\theta_1 O_1, \ldots, \theta_C O_C\}$, it is true that $\forall k \in [C]$, $\theta_k \leq \frac{R(\mathbf{p}_{\xi^*})}{O_k}$. Replacing θ_k by $\frac{R(\mathbf{p}_{\xi^*})}{O_k}$ in Lemma 8, we complete the proof as follows:

$$\begin{aligned} R(\mathbf{P}^*) &\leq R(\mathbf{p}_{\xi^*})\Big(1 + \sum_{k=1}^{C-1} \frac{O_k - O_{k+1}}{O_k}\Big) \\ &\leq R(\mathbf{p}_{\xi^*})\Big(1 + \sum_{k=1}^{C-1} (\frac{1}{1 + O_{k+1}} + \ldots + \frac{1}{O_k})\Big) = R(\mathbf{p}_{\xi^*})\Big(1 + \sum_{i=O_C+1}^{O_1} \frac{1}{i}\Big) \\ &\leq R(\mathbf{p}_{\xi^*})\Big(\sum_{i=1}^{O_C} \frac{1}{i} + \sum_{i=O_C+1}^{O_1} \frac{1}{i}\Big) = R(\mathbf{p}_{\xi^*}) \cdot H_{O_1} \\ &\leq R(\mathbf{p}_{\xi^*})(\ln O_1 + 1) \\ &\leq R(\mathbf{p}_{\xi^*})(\ln min\{\mathcal{U}, \textstyle\sum_{j=1}^{\mathcal{B}} d_j\} + 1) \end{aligned}$$

□

Proposition 9. *The $O(\log min\{\mathcal{U}, \sum_{j=1}^{\mathcal{B}} d_j\})$ revenue guarantee is tight for uniform pricing solutions.*

Proof. We show a worst case. $\forall i \in [N]$: (1) there are 2^i users that only satisfy q_i, and (2) exists buyer i represented as $(i, 2^i, 2^{-i})$. Thus $\mathcal{U} = \sum_{j=1}^{\mathcal{B}} d_j = 2^{N+1} - 1$. It is easy to see that the revenue of any uniform pricing is less than 2, but the revenue of the optimal arbitrage-free pricing is N (any pricing is arbitrage-free in this case). □

5 NON-UNIFORM PRICING

Based on uniform pricing, in this section we study arbitrage-free *non-uniform* pricing, which is more practical for real markets. We first devise a greedy algorithm to produce an arbitrage-free non-uniform pricing of which the revenue is guaranteed to be no less than the revenue of the optimal uniform pricing. In order to speed up the algorithm for large markets, we propose an approximate algorithm to solve the allocation problem efficiently for any pricing while preserving the same performance guarantee.

Definition 8. *Given $\mathbf{P}$, $\alpha_i \triangleq \max\{\pi(i'|i) \cdot p_{i'} | i' \in [N], i' \neq i\}$ and $\beta_i \triangleq \min\{\frac{p_{i'}}{\pi(i|i')} | i' \in [N], i' \neq i\}$. We call $[\alpha_i, \beta_i]$ the* arbitrage-free interval[6] *of p_i.*

[6] W.l.o.g. we assume that β_i always exists, i.e. $\{i' | i' \in [N], i' \neq i, \pi(i|i') > 0\} \neq \emptyset$, otherwise, the arbitrage-free interval of p_i becomes $[\alpha_i, \infty)$.

Proposition 10. *If* $\mathbf{P}$ *is arbitrage-free and only one entry is varied within its arbitrage-free interval, i.e. updating* p_i *into any value in* $[\alpha_i, \beta_i]$*, the resulting pricing is still arbitrage-free.*

Although Proposition 10 is straightforward to verify, it connects uniform pricing which is naturally arbitrage-free with arbitrage-free non-uniform pricing. Most importantly, Proposition 10 provides valid operations to update the pricing while keeping it arbitrage-free. It is easy to see that with proper preprocessing, α_i and β_i can be computed in $O(N)$ time. Based on Lemma 4 and Proposition 10, we have Corollary 11.

Corollary 11. $C_i(\mathbf{P}) \triangleq \{\alpha_i, \beta_i\} \cup \{c_j | j \in [\mathcal{B}], t_j = i, c_j \in [\alpha_i, \beta_i]\}$. *For any arbitrage-free pricing* $\mathbf{P}$ *that* $\exists i \in [N]$, $p_i \notin C_i(\mathbf{P})$, *there exists an arbitrage-free pricing* $\mathbf{P}'$ *that* $R(\mathbf{P}') \geq R(\mathbf{P})$ *and* $\forall i \in [N]$, $p'_i \in C_i(\mathbf{P}')$.

With Proposition 10, the proof of Corollary 11 is similar to the proof of Lemma 4, thus omitted. Corollary 11 reveals the desirable property of the optimal arbitrage-free non-uniform pricing, which implies that any algorithm only needs to search over $O(\mathcal{B})$ values (because $|C_i(\mathbf{P})| \leq \mathcal{B} + 2$) other than the entire real interval $[\alpha_i, \beta_i]$ for p_i. Based on this, we propose Algorithm 2 that iteratively updates the optimal uniform pricing to arbitrage-free non-uniform pricing. First, we show a polynomial subroutine to solve the allocation problem for a non-uniform pricing as follows.

Lemma 12. *For any non-uniform pricing, the optimal allocation can be computed in polynomial time* $O(\mathcal{U}M + \mathcal{U}\mathcal{B} + \mathcal{B}\log\mathcal{B})$.

Proof. We model the allocation problem for non-uniform pricing as a minimum cost maximum flow problem. The network is constructed as follows. We introduce s and t as the source and sink respectively. We introduce N nodes $y_1, \ldots, y_N$. Assuming all the users are indexed from 1 to $\mathcal{U}$ and $\mathbf{u}_k$ denotes the k-th user. For each $k \in [\mathcal{U}]$: (1) we introduce a node x_k and a directed edge from s to x_k with capacity 1 and cost 0; and (2) $\forall i \in [N]$, we introduce a directed edge from x_k to y_i with capacity 1 and cost 0 if $u_k^i = 1$. For each buyer j that $c_j \geq p_{t_j}$, introduce a node z_j, a directed edge from y_{t_j} to z_j with cost 0 and a directed edge from z_j to t with capacity d_j and a negative cost $-p_{t_j}$. We can verify that there is no directed cycle with negative cost.

Let $w(f) > 0$ be the absolute value of the minimum cost when the amount of the required s-t flow is f. Clearly, it is true that $w(f)$ is the maximum revenue when exactly f users are sold. Since the costs are all negative, $w(f)$ is maximized when $f = f^*$ where f^* is the amount of the maximum s-t flow.

Since only the edges from z_j to t are associated with non-zero costs, to find the augmenting path with the smallest cost is $O(|E|)$ (after sorting buyers by c_j) where $|E| \leq M + 2\mathcal{B}$. Since $f^* \leq \mathcal{U}$, the overall complexity is $O(\mathcal{U}M + \mathcal{U}\mathcal{B} + \mathcal{B}\log\mathcal{B})$. □

Now we present Algorithm 2. It starts with the optimal uniform pricing, then iteratively and greedily updates the pricing and finally outputs an arbitrage-free non-uniform pricing. In each iteration, the algorithm greedily updates the price of a query if the update increases the revenue. To update p_i, it picks up the new price from $C_i(\mathbf{P})$ that greedily maximizes the revenue. Let $\mathbf{P}_{-i,p}$ denote the resulting pricing when we only change the i-th entry of $\mathbf{P}$ to p.

Algorithm 2 Arbitrage-free Non-uniform Pricing

Input: N, U, B and $\mathbf{p}_{\xi^*}$
Output: An arbitrage-free non-uniform pricing $\mathbf{P}$

1: $\mathbf{P} \leftarrow \mathbf{p}_{\xi^*}$
2: $\forall i \in [N]$, compute α_i and β_i
3: **repeat**
4: **for** $i \in [N]$ **do**
5: $p^* = \text{argmax}_{p \in C_i(\mathbf{P})} R(\mathbf{P}_{-i,p})$
6: **if** $R(\mathbf{P}_{-i,p^*}) > R(\mathbf{P})$ **then**
7: $\mathbf{P} \leftarrow \mathbf{P}_{-i,p}$
8: $\forall i' \in [N]$, re-compute $\alpha_{i'}$ and $\beta_{i'}$
9: **end if**
10: **end for**
11: **until** no price changes
12: **return** $\mathbf{P}$.

Proposition 13. *Algorithm 2 always converges; in each iteration of Repeat-Loop (lines* 4-10*), it solves at most* $\mathcal{B} + 2N$ *allocation problems for non-uniform pricing.*

Proof. Since the revenue of any pricing is trivially bounded by $\sum_{j=1}^{\mathcal{B}} c_j d_j$ and after each iteration except the last one the revenue always increases, the algorithm will always converge. Since $|\bigcup_{i=1}^{N} C_i(\mathbf{P})| \leq \mathcal{B} + 2N$, there will be at most $\mathcal{B} + 2N$ allocation problems of non-uniform pricing in each iteration. □

Let T be the number of iterations, the overall complexity is $O(T(\mathcal{B}+N)(\mathcal{U}M+\mathcal{U}\mathcal{B}+\mathcal{B}\log\mathcal{B}))$. We leave the theoretical analysis of T for future work, however, we will see in experiments that Algorithm 2 converges very quickly, i.e. $T < 4$ on average. We next analyze its performance in Proposition 14. Experiments show that the arbitrage-free non-uniform pricing has significantly larger revenue in practice, however, the fact that the arbitrage-free interval of any entry is dynamic in each iteration makes it difficult to analyze the theoretical improvement of the arbitrage-free non-uniform pricing.

Proposition 14. *Assume that Algorithm 2 converges after* $T \geq 1$ *iterations (of Repeat-Loop, i.e. lines 4-10). Let* $\mathbf{P}^t$ *be the pricing computed after* $t \in [T]$ *iterations. The following statements are true:*

(a) $\forall t \in [T]$, $\mathbf{P}^t$ *is arbitrage-free.*
(b) $\forall t \in [T]$, $R(\mathbf{P}^t)$ *has* $O(\log min\{\mathcal{U}, \sum_{j=1}^{\mathcal{B}} d_j\})$ *performance guarantee to the optimal arbitrage-free pricing;*
(c) If $T > 1$, $\mathbf{P}^t$ *is non-uniform,* $\forall t \in [T]$.
(d) Unless breaking the arbitrage-free constraints, changing any entry of $\mathbf{P}^T$ *alone cannot increase the revenue.*

5.1 Faster Allocation

In Algorithm 2, the minimum cost maximum flow solver (described in the proof for Lemma 12) that computes the optimal allocation for a non-uniform pricing is called frequently, i.e. up to $\mathcal{B} + 2N$ times in each iteration. Although to our best knowledge, it is faster than most of general mincost maxflow solvers for integral flows and non-unit capacity (see details in the survey [19]), it still does not scale for large inputs. By generalizing the online bipartite matching [16], we propose an approximation as Algorithm 3 to the allocation

problem. Algorithm 3 greedily sells users satisfying queries with the highest price.

Algorithm 3 Efficient Approximate Allocation

Input: N, U, B and $\mathbf{P}$
Output: $\hat{A}$
1: $\forall j \in [\mathcal{B}], \hat{A}_j \leftarrow \emptyset$
2: Re-order all the buyers such that $p_{t_1} \geq \ldots \geq p_{t_\mathcal{B}}$
3: **for** $j \in [\mathcal{B}]$ **do**
4: **if** $c_j \geq p_{t_j}$ **then**
5: **while** $|\hat{A}_j| < d_j$ **and** $\exists \mathbf{u} \in U, u^{t_j} = 1$ **do**
6: $\hat{A}_j \leftarrow \hat{A}_j \cup \{\mathbf{u}\}$
7: $U \leftarrow U - \{\mathbf{u}\}$
8: **end while**
9: **end if**
10: **end for**
11: **return** $\hat{A}$.

Proposition 15. *For any pricing, Algorithm 3 produces an allocation in polynomial time $O(M + \mathcal{B} \log \mathcal{B})$.*

Algorithm 3 is much faster than the mincost maxflow solver in Lemma 12, i.e. by at least a factor $\frac{\mathcal{U}}{\log \mathcal{B}}$. In real user markets, the number of users to sell is significantly larger than the number of buyers, i.e. $\mathcal{U} \gg \mathcal{B}$, and $\mathcal{U}$ could be billions, so the improvement is significant. We next show the performance guarantee of Algorithm 3 in Lemma 16, which will be also used for proving Theorem 21 later.

Lemma 16. *Algorithm 3 is a 2-approximation to the allocation problem for any pricing.*

Proof. Let A^* be the optimal allocation for $\mathbf{P}$. Consider some $\mathbf{u}$ who is allocated to buyer j in A^* (i.e. $\mathbf{u} \in A^*_j$) by the mincost maxflow solver, but is allocated to a different buyer j' in $\hat{A}$ (i.e. $\mathbf{u} \in \hat{A}_{j'}$) or not allocated to any buyer in $\hat{A}$. There are three cases. Case 1: if $\mathbf{u}$ is finally unallocated in $\hat{A}$, it is true that $|\hat{A}_j| = d_j \geq |A^*_j|$ because if $|\hat{A}_j|$ was less than d_j, $\mathbf{u}$ must have been allocated to buyer j. Consider that $\mathbf{u}$ is allocated to buyer j' in $\hat{A}$. Case 2: if $p_{t_j} > p_{t_{j'}}$, it must be true that $|\hat{A}_j| = d_j \geq |A^*_j|$ for the same reason as Case 1. In Case 1 and 2, the revenue contributed by $\hat{A}_j$ is no less than A^*_j. Case 3: if $p_{t_j} \leq p_{t_{j'}}$, it is true that we might lose the revenue p_{t_j} because $\mathbf{u}$ is not allocated to j. In this case, however an equal or higher revenue $p_{t_{j'}}$ is produced. This implies that the total revenue loss $R(\mathbf{P}, A^*) - R(\mathbf{P}, \hat{A})$ is bounded by the total revenue produced $R(\mathbf{P}, \hat{A})$. Therefore we prove the lemma that $2R(\mathbf{P}, \hat{A}) \geq R(\mathbf{P}, A^*) = R(\mathbf{P})$. □

We construct Algorithm 2.1 with the faster allocation in Algorithm 3, to compute an arbitrage-free non-uniform pricing. Algorithm 2.1 is the same with Algorithm 2 except that we replace the optimal allocation (mincost maxflow solver $R(\mathbf{P}_{-i,p})$ in line 5) with the approximate allocation computed by Algorithm 3.

Theorem 17. *Let T be the number of iterations that Algorithm 2.1 needs to converge. Algorithm 2.1 runs in $O(T(\mathcal{B} + N)(M + \mathcal{B} \log \mathcal{B}))$ and has all the properties (a)-(d) claimed for Algorithm 2 in Proposition 14.*

6 A GENERALIZED SETTING: MINIMAL DEMAND

Previously, we assumed that any buyer only restricts the maximum number d_j of target users he will buy. In this section, we consider a general setting where a buyer also has a minimum demand. That is, buyer j now becomes $(t_j, \underline{d}_j, d_j, c_j)$ where $\underline{d}_j$ and d_j ($\underline{d}_j \leq d_j$) are the minimum and maximum number of target users that buyer j will buy respectively. In this setting, the *Demand Constraint* in Definition 2 for a feasible allocation A becomes: $|A_j| \in \{0, \underline{d}_j, \ldots, d_j\}$. This means that, for buyer j, we either allocate 0 or at least $\underline{d}_j$ users to him. Note that, the pricing problem in previous setting is indeed a special case that $\forall j, \underline{d}_j = 1$ of this setting.

This generalized setting is also practical in many user markets. For example, a business wants to trigger a cascade of promotion for its product in an online community. If the initial seed size is too small, the growth of cascade would be very slow or even not triggered [3, 11] at all. Therefore the business needs some guarantee of the seed size by specifying a large enough $\underline{d}_j$. Another example is that, a buyer wants to purchase the contacts of users with certain attributes for survey. If he is not able to reach enough number of target users, the survey results lack statistical significance. Therefore, he is not willing to buy any data set with size less than his minimum demand.

However, the minimum demand constraint makes the allocation problem and the pricing problem harder to solve, which can be seen in Theorem 18 and Corollary 19.

Theorem 18. *In the generalized setting, the allocation problem (even if the pricing is uniform at 1) is (1) NP-hard and (2) hard to approximate (unless P=NP) within $\mathcal{U}^{\frac{1}{2}-\epsilon}$ or $\mathcal{B}^{1-\epsilon}$, $\forall \epsilon > 0$.*

Corollary 19. *In the generalized setting, the pricing problem is both NP-hard and hard to approximate (unless P=NP) within $\mathcal{U}^{\frac{1}{2}-\epsilon}$ or $\mathcal{B}^{1-\epsilon}$, $\forall \epsilon > 0$.*

Next, we first propose an approximate algorithm for the allocation problem, and then based on this approximation, we propose another approximate algorithm for the pricing problem.

Algorithm 4 computes the approximate allocation in two steps, each of which produces a partial allocation. The *first partial allocation* process (lines 1-9), first re-orders all the buyers so that $p_{t_1}\underline{d}_1 \geq \ldots \geq p_{t_\mathcal{B}}\underline{d}_\mathcal{B}$. Then starting from $j = 1$, it allocates $\underline{d}_j$ target users (if enough) to buyer j, in sequence. After that, the *second partial allocation* process (lines 10-16) discards buyers who received no user in the first partial allocation. For each of the remaining buyers, it creates a dummy buyer without the minimum demand as $(t_j, d_j - \underline{d}_j, c_j)$. Then it calls Algorithm 3 with the remaining users, the dummy buyers and the same pricing as the input. Finally, the two partial allocations are merged as the whole approximate allocation.

Proposition 20. *In the generalized setting, Algorithm 4 computes an approximate allocation $\hat{A}$ for any pricing in polynomial time $O(M + \mathcal{B} \log \mathcal{B})$.*

Theorem 21. *Let $D \triangleq \max\{\underline{d}_j | j \in [\mathcal{B}]\}$. The approximate allocation $\hat{A}$ produced by Algorithm 4 is an $O(D)$-approximation to the optimal allocation.*

Algorithm 4 Approximate Allocation With Minimal Demand

Input: N, U, B and $\mathbf{P}$
Output: An approximate allocation $\hat{A}$

1: Re-order $\mathcal{B}$ buyers so that $p_{t_1}\underline{d}_1 \geq \ldots \geq p_{t_\mathcal{B}}\underline{d}_\mathcal{B}$
2: Let $\hat{A}^1$ and $\hat{A}^2$ be two empty allocations
3: **for** $j \in [\mathcal{B}]$ **do**
4: $\quad U_j \leftarrow \{\mathbf{u} | u^{t_j} = 1, \mathbf{u} \in U\}$
5: $\quad$ **if** $c_j \geq p_{t_j}$ **and** $|U_j| \geq \underline{d}_j$ **then**
6: $\quad\quad \hat{A}^1_j \leftarrow$ arbitrary $\underline{d}_j$ users from U_j
7: $\quad\quad U \leftarrow U - \hat{A}^1_j$
8: $\quad$ **end if**
9: **end for**
10: $B' \leftarrow \emptyset$
11: **for** $j \in [\mathcal{B}]$ **do**
12: $\quad$ **if** $|\hat{A}^1_j| = \underline{d}_j$ **and** $d_j \neq \underline{d}_j$ **then**
13: $\quad\quad B' \leftarrow B' \cup \{(t_j, d_j - \underline{d}_j, c_j)\}$
14: $\quad$ **end if**
15: **end for**
16: $\hat{A}^2 \leftarrow$ Call Algorithm 3 with N, U, B' and $\mathbf{P}$ as input
17: $\forall j \in [\mathcal{B}], \hat{A}_j \leftarrow \hat{A}^1_j \cup \hat{A}^2_j$
18: **return** $\hat{A}$

Proof. Let A^* be the optimal allocation. W.l.o.g. we assume that $\forall j \in [\mathcal{B}]$, the number of users satisfying q_{t_j} is no less than $\underline{d}_j$, otherwise we can remove buyer j. This theorem is implied by Lemma 22 and Lemma 16. In short, Lemma 22 implies that for every unit of revenue generated by the first partial allocation (lines 1-9), A^* can generate at most $D+1$ units of revenue. According to Lemma 16, it is true that for every unit of revenue generated by the second partial allocation (lines 10-16), A^* can generate at most 2 units of revenue. Therefore, we have $R(\mathbf{P}, \hat{A}) \geq \frac{R(\mathbf{P}, A^*)}{\max\{D+1, 2\}}$, which implies the theorem. We only need to prove Lemma 22. □

Lemma 22. *If buyer j is allocated with $\underline{d}_j$ target users in the first partial allocation, the loss of revenue (compared to A^*) is bounded by $p_{t_j}(\underline{d}_j)^2$.*

We next show that the $O(D)$-approximation is tight for Algorithm 4. Consider the following example. There are $d+1$ queries and d^2 users. $\forall i \in [d]$, all the $d-1$ users indexed between $(d-1)\cdot(i-1)+1$ and $(d-1)\cdot i$ (both inclusive) only satisfy q_i. The last d users, indexed between $d^2 - d + 1$ and d^2, satisfy all the queries. There are $d+1$ buyers, and buyer j is (j, d, d, ∞). Let the pricing be: $\forall i \in [d]$, $p_i = 1$ and $p_{d+1} = 1 + \epsilon$. Algorithm 4 will only allocate d users to buyer $d+1$ while all the other buyers are allocated with 0 user, thus the revenue is $d \cdot (1+\epsilon)$. However, the optimal solution that $\forall i \in [d]$, allocates buyer i with d users has revenue d^2.

With Theorem 21, we propose Algorithm 5 to produce a uniform pricing in this setting. Let $\hat{A}_\xi$ be the approximate allocation output by Algorithm 3 when the input uniform price is ξ. Similar to Algorithm 1, Algorithm 5 outputs the uniform price $\hat{\xi} = \text{argmax}_{\xi \in C} R(\mathbf{p}_\xi, \hat{A}_\xi)$ where $C = \{c_j | j \in [\mathcal{B}]\}$. We analyze it as follows.

Theorem 23. *Algorithm 5 runs in $O(\mathcal{B}M + \mathcal{B}^2 \log \mathcal{B})$ to compute a uniform pricing which is an $O(D \log \min\{\mathcal{U}, \sum_{j=1}^{\mathcal{B}} d_j\})$ approximation to the optimal arbitrage-free pricing in the generalized setting.*

Compared with the previous setting where buyers have no minimum demand, we do not find a polynomial time solution to compute the optimal uniform pricing in this setting. Because the allocation problem is NP-hard, the guarantee of the uniform pricing drops. However, as Algorithm 5 does not call any maxflow optimization, its time complexity is lower than that of Algorithm 1. We leave the arbitrage-free non-uniform pricing in this setting as future work.

7 EXPERIMENTS

In this section, we use synthetic data to evaluate Algorithms 1, 2, 2.1 and 3. To randomly generate instances of the pricing problem, we first set the values of $\mathcal{U}$, $\mathcal{B}$, N, m and c where m is the maximal number of queries that any user can satisfy and c is the upper bound of buyers' maximum costs. For each buyer $j \in [\mathcal{B}]$, t_j, d_j and c_j are independently and uniformly sampled from the integer sets $[N]$, $[\lfloor \frac{4\mathcal{U}}{\mathcal{B}} \rfloor]$ and $[c]$ respectively. For each user $\mathbf{u}_i \in U$, m_i is independently and uniformly sampled from the integer set $[m]$, and we randomly make her satisfy m_i distinct queries. We generate instances of the pricing problem of three different sizes: *small* size where $(\mathcal{U}, \mathcal{B}, N, m, c) = (100, 20, 10, 4, 5)$, *medium* size where $(\mathcal{U}, \mathcal{B}, N, m, c) = (1000, 100, 50, 20, 1000)$ and *large* size where $(\mathcal{U}, \mathcal{B}, N, m, c) = (10^6, 1000, 500, 200, 1000)$.

7.1 Optimal Arbitrage-free Pricing

In this part, we compare the optimal arbitrage-free pricing, the optimal uniform pricing computed by Algorithm 1 and the arbitrage-free non-uniform pricing by Algorithm 2. Since we haven't found any (even pseudo) polynomial algorithm to compute the optimal arbitrage-free pricing, we use an exponential algorithm with grid search and backtracking to compute the numerically optimal arbitrage-free pricing. Thus the experiment can be only conducted on instances of small size, and we randomly generate 1000 such instances.

Let $\mathbf{P}^*$ be the optimal arbitrage-free pricing. For each test case, we record two revenue ratios $r_1 = \frac{R(\mathbf{p}_{\xi^*})}{R(\mathbf{P}^*)}$ and $r_2 = \frac{R(\mathbf{P})}{R(\mathbf{P}^*)}$ where $\mathbf{p}_{\xi^*}$ is the optimal uniform pricing computed by Algorithm 1 and $\mathbf{P}$ is the arbitrage-free non-uniform pricing by Algorithm 2. The results are shown in Fig 1 where the x-axis denotes the interval of revenue ratios and y-axis denotes the proportion of the test cases of which the revenue ratios fall into the interval denoted by x. The red bar is for r_1 and the blue bar is for r_2. Among all the cases, $r_1 \in [0.732, 0.983]$ and $r_2 \in [0.796, 0.997]$. The mean values of r_1 and r_2 are 0.849 and 0.948 respectively. We observe that the actual revenue ratios of the optimal arbitrage-free uniform pricing and the non-uniform pricing are both significantly larger than the theoretical guarantee which is $1/(1 + \log \min\{\mathcal{U}, \sum_{j=1}^{\mathcal{B}} d_j\}) \approx 0.18$. In particular, the arbitrage-free non-uniform pricing is remarkably close to the optimal arbitrage-free pricing.

7.2 Approximate Allocation

In this part, we compare the approximation in Algorithm 3 with the optimal one (i.e. the mincost maxflow solver in Lemma 12) for the allocation problem. Note that in line 6, Algorithm 3 selects any user

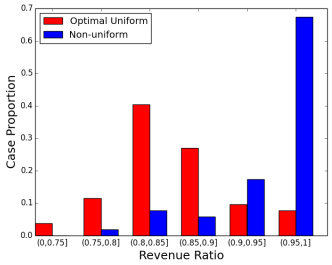

Figure 1: Uniform pricing vs. Non-uniform pricing (with baseline)

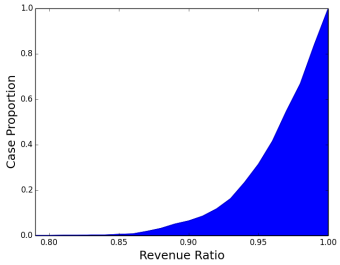

Figure 2: The optimal allocation vs. the approximate allocation

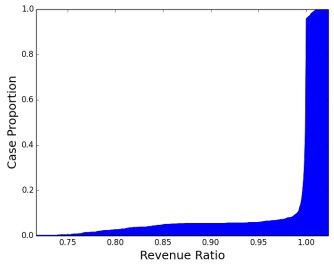

Figure 3: Compare the non-uniform pricings by Algorithm 2 and 2.1

$\mathbf{u}$ satisfying q_{t_j}. For experiments, we use a heuristic to select such a user: among all the users satisfying q_{t_j}, we select the one with the least number of other queries that she satisfies. With proper preprocessing, this heuristic does not increase the asymptotic time complexity of Algorithm 3.

The experiments are conducted on the instances of medium size. We first randomly generate 1000 cases of medium size and for each case we randomly generate a pricing vector $\mathbf{P}$ where $\forall i \in [N]$, p_i is independently and uniformly sampled from $[c]$. We define the revenue ratio as $\frac{R(\mathbf{P},\hat{A})}{R(\mathbf{P},A^*)}$ where $\hat{A}$ is the approximate allocation computed by Algorithm 3 and A^* is the optimal allocation computed by the mincost maxflow solver in Lemma 12. We plot the results in Fig 2 where x-axis denotes the ratio and y-axis denotes the accumulated proportion of cases of which the revenue ratios are less than or equal to x. Among all the 1000 cases, the least revenue ratio is 0.79; the ratios of 76.6% cases are at least 0.95; 16% cases reach the optimum, i.e. with ratio as 1. The average ratio is 0.968, much larger than the theoretical guarantee 0.5.

Next, we randomly generate 1000 cases of medium size, for each of which, we compute its arbitrage-free non-uniform pricing. Let $\mathbf{P}$ and $\mathbf{P}'$ be the pricing output by Algorithm 2 and 2.1 respectively. In order to compare $R(\mathbf{P})$ and $R(\mathbf{P}')$, we still call the mincost maxflow solver after $\mathbf{P}$ and $\mathbf{P}'$ are produced. We plot the results in Fig 3 where x-axis denotes the revenue ratio $\frac{R(\mathbf{P}')}{R(\mathbf{P})}$ and y-axis denotes the accumulated proportion of cases of which the revenue ratios are less than or equal to x. With the mean value 0.988, the ratios are in $[0.717, 1.023]$. Among the 1000 cases, we find that in 94.1% cases, the ratio is at least 0.95 and in 4.1% cases, the ratio is larger than 1 (because Algorithm 2 does not guarantee global optimum). Notably, Algorithm 2.1 is faster than Algorithm 2 by at least a factor $\frac{\mathcal{U}}{\log \mathcal{B}}$.

7.3 Uniform and Non-uniform Pricing

In this part, we first measure the convergence of Algorithm 2.1, and then compare the arbitrage-free non-uniform pricing by Algorithm 2.1 with the optimal uniform pricing by Algorithm 1. We randomly generate two datasets, 1000 cases of large size and 5000 cases of medium size.

Convergence. We observe that Algorithm 2.1 converges very quickly, as shown in Fig 4. Among all the all the 1000 instances of large size, it converges after 3.77 iterations on average, and in the worst case, 14 iterations. Among all the 5000 instances of medium size, it converges after 2.96 iterations on average, and in the worst case, 7 iterations.

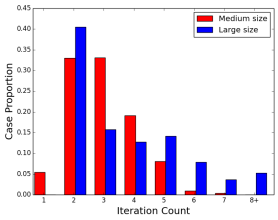

Figure 4: Convergence of Algorithm 2.1

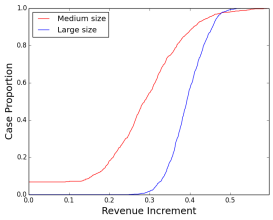

Figure 5: Uniform pricing vs. non-uniform pricing

Let R be the (approximate) revenue of the arbitrage-free non-uniform pricing output by Algorithm 2.1, and the relative revenue increment is calculated as $\frac{R}{R(\mathbf{p}_{\xi^*})} - 1$. We observe that R is significantly larger than $R(\mathbf{p}_{\xi^*})$, shown in Fig 5. For each curve, x-axis is the relative revenue increment and y-axis is the accumulated case proportion of which the relative revenue increment is less than or equal to x. For all the 1000 instances of large size, the increment is in $[0.25, 0.43]$, with 0.394 as the mean and 0.046 as the standard deviation. For all the 5000 instances of medium size, the increment is in $[0, 0.596]$, with 0.281 as the mean and 0.112 as the standard deviation. We conclude that the arbitrage-free non-uniform pricing significantly outperforms the optimal uniform pricing, typically in large markets.

8 CONCLUSION

In this paper, we addressed the pricing problem, in particular, revenue maximizing arbitrage-free pricing in user-based markets. We presented a variety of efficient algorithms for arbitrage-free pricing with provable approximation guarantees on their revenue, and hardness results for certain variations. We believe that there is a real need to study mechanisms for allocation and pricing of users based on multiple attributes as much of online user-based markets rely on such systems.

REFERENCES

[1] Maria-Florina Balcan, Avrim Blum, and Yishay Mansour. 2008. Item Pricing for Revenue Maximization. In *EC*.

[2] Patrick Briest and Piotr Krysta. 2006. Single-minded unlimited supply pricing on sparse instances. In *SODA*.

[3] Meeyoung Cha, Alan Mislove, Ben Adams, and Krishna P. Gummadi. 2008. Characterizing Social Cascades in Flickr. In *WOSN*.

[4] Ning Chen, Arpita Ghosh, and Sergei Vassilvitskii. 2008. Optimal Envy-free Pricing with Metric Substitutability. In *EC*.

[5] Maurice Cheung and Chaitanya Swamy. 2008. Approximation algorithms for single-minded envy-free profit-maximization problems with limited supply. In *FOCS*.

[6] Shaleen Deep and Paraschos Koutris. 2017. The Design of Arbitrage-Free Data Pricing Schemes. In *ICDT*.

[7] Milad Eftekhar, Nick Koudas, and Yashar Ganjali. 2015. Reaching a desired set of users via different paths: an online advertising technique on micro-blogging platforms.. In *EDBT*.

[8] Milad Eftekhar, Saravanan Thirumuruganathan, Gautam Das, and Nick Koudas. 2014. Price trade-offs in social media advertising. In *COSN*.

[9] Michal Feldman, Amos Fiat, Stefano Leonardi, and Piotr Sankowski. 2012. Revenue Maximizing Envy-free Multi-unit Auctions with Budgets. In *EC*.

[10] Amos Fiat and Amiram Wingarten. 2009. Envy, multi envy, and revenue maximization. In *WINE*.

[11] Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A. Zighed. 2013. Information Diffusion in Online Social Networks: A Survey. *SIGMOD Rec.* (2013).

[12] Faruk Gul and Ennio Stacchetti. 1999. Walrasian equilibrium with gross substitutes. *Journal of Economic theory* (1999).

[13] Venkatesan Guruswami, Jason D. Hartline, Anna R. Karlin, David Kempe, Claire Kenyon, and Frank McSherry. 2005. On Profit-maximizing Envy-free Pricing. In *SODA*.

[14] Jason Hartline and Qiqi Yan. 2011. Envy, Truth, and Profit. In *EC*.

[15] Sungjin Im, Pinyan Lu, and Yajun Wang. 2010. Envy-free Pricing with General Supply Constraints. In *WINE*.

[16] R. M. Karp, U. V. Vazirani, and V. V. Vazirani. 1990. An Optimal Algorithm for On-line Bipartite Matching. In *STOC*. 352–358.

[17] Paraschos Koutris, Prasang Upadhyaya, Magdalena Balazinska, Bill Howe, and Dan Suciu. 2012. Query-based Data Pricing. In *PODS*.

[18] Paraschos Koutris, Prasang Upadhyaya, Magdalena Balazinska, Bill Howe, and Dan Suciu. 2013. Toward Practical Query Pricing with QueryMarket. In *SIGMOD*.

[19] Péter Kovács. 2015. Minimum-cost flow algorithms: an experimental evaluation. *Optimization Methods and Software* (2015).

[20] Herman B Leonard. 1983. Elicitation of honest preferences for the assignment of individuals to positions. *The Journal of Political Economy* (1983).

[21] Chao Li and Gerome Miklau. 2012. Pricing Aggregate Queries in a Data Marketplace.. In *WebDB*.

[22] Bing-Rong Lin and Daniel Kifer. 2014. On Arbitrage-free Pricing for General Data Queries. *VLDB Endow.* (2014).

[23] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. 2007. *Algorithmic game theory*. Cambridge University Press.

[24] Carl Shapiro and Hal R Varian. 1998. Versioning: the smart way to sell information. *Harvard Business Review* (1998).

[25] Vasilis Syrgkanis and Johannes Gehrke. 2015. Pricing Queries Approximately Optimally. In *CoRR*.

[26] Chaolun Xia, Saikat Guha, and Shan Muthukrishnan. 2016. Targeting Algorithms for Online Social Advertising Markets. In *ASONAM*.