Algorithms for Multi-Product Pricing

Gagan Aggarwal* Tomás Feder† Rajeev Motwani‡ An Zhu§

Abstract

In the information age, the availability of data on consumer profiles has opened new possibilities for companies to increase their revenue via data mining techniques. One approach has been to strategically set prices of various products, taking into account the profiles of consumers. We study algorithms for the multi-product pricing problem, where given consumer preferences among products, budgets, and the costs of production, the goal is to set prices of multiple products from a single company, so as to maximize the overall revenue of the company. We present approximation algorithms as well as negative results for several variants of the multi-product pricing problem, modeling different purchasing patterns and market assumptions.

*Computer Science Department, Stanford University, Stanford, CA 94305. Supported in part by a SGF fellowship from Stanford and NSF Grant EIA-0137761. E-mail gagan@cs.stanford.edu

†268 Waverley St., Palo Alto, CA 94301. E-mail: tomas@theory.stanford.edu

‡Computer Science Department, Stanford University, Stanford, CA 94305. Supported in part by NSF Grant IIS-0118173 and EIA-0137761, an Okawa Foundation Research Grant, and grants from Microsoft and Veritas. E-mail rajeev@cs.stanford.edu

§Computer Science Department, Stanford University, Stanford, CA 94305. Supported in part by a GRPW fellowship from Bell Labs, Lucent Technologies, and NSF Grant EIA-0137761. E-mail anzhu@cs.stanford.edu

1 Introduction

Through interaction with online consumers, e-commerce websites can gather data reflecting consumer preferences. Such data allows significant revenue increases through strategic price setting via sophisticated analytical tools. While the airline and hotel industry were the traditional users of revenue management [6, 11], corporations in other verticals in retail and manufacturing have recently started employing intelligent pricing strategies to boost their bottom lines. For instance, Dell quotes different prices to different market segments for the same product, enabling Dell to increase its market share and profitability [7]. Other documented examples include Ford [2] and ShopKo Stores [8].

Motivated by the availability of such data, Rusmevichientong, Van Roy, and Glynn [9, 10] formulated the non-parametric multi-product pricing problem. In multi-product pricing, given demands and production costs, the problem is to determine product pricing that maximize overall revenue or profit. When faced with choice between multiple substitutable products, consumers may be indifferent between two or more products and substitute one product for another. The problem of modeling substitutability among products and determining optimal pricing in that context remains a challenging open problem in this research area. As noted by Rusmevichientong et al., in order to capture substitutability, in most models, consumer-demand functions are generally assumed to take on specific parametric forms. However, these parametric forms may not reflect the true demand function accurately. We adopt the non-parametric approach proposed by Rusmevichientong et al.

The non-parametric approach employs large quantities of consumer data in order to reduce reliance on rigid parametric models, where the consumer data is obtained from Internet sites. A concrete example is the General Motors' Auto Choice Advisor website [3]. This website is set up to advise potential purchasers of cars (of all brands) on products that meet their requirements. The website asks the users various questions about their preferences with respect to cars, such as model, budget, primary use, etc., and then recommends some cars that satisfies their requirements. Thus, GM has access to large quantities of data that reflect consumer preferences. This data can be used by GM to optimize the pricing of their own vehicles. Based on this scenario, Rusmevichientong et al. [9, 10] introduce the non-parametric multi-product pricing model described next.

Consider the problem of pricing n different products $\mathcal{P}$ indexed by $A = \{1, \ldots, n\}$, given a set of data samples, $(R_1, B_1), \ldots, (R_m, B_m)$, each associated with one of m different consumers that have visited an e-commerce website. For each j, let $R_j = \pi_j(A)$ be the preference ordering of the jth consumer over all n products, where the lower-ranked products have higher preference. Further, let $B_j = \{b_{ij}, \forall 1 \leq i \leq n\}$ denote the sequence of budgets b_{ij} of the jth consumer for product i. The budget is the maximum amount the consumer is willing to pay for a product. We assume consumers are rational, i.e., the order of their budgets for various products obeys their preference ordering. Each sample (R_j, B_j) represents all available information about the jth consumer. We further assume that, given that the products are priced at $P = \{P_1, \ldots, P_n\}$, the jth consumer will purchase the lowest-ranked product in their preference list which she can afford, i.e., they will buy product k, where $k = \arg\min_{B_{ij} \geq P_i} R_j(i)$. This model of consumer behavior is termed *Rank-Buying*. In addition, we may assume that there is a *Price-Ladder (PL)* constraint, i.e., the prices of the products must obey a fixed linear order. Such a constraint can usually be deduced from market studies/policies and manufacturing costs. Assuming each consumer buys only one product, the objective is to set the product prices so as to maximize the revenue.

Rusmevichientong et al. [9, 10] study this basic model and present various heuristics. However, they leave open the problem of designing algorithms with provable performance guarantees. In this paper, we present a PTAS for this problem. In addition, we consider some interesting variants of the original model:

1. Given the prices, suppose the consumer will buy the most expensive (least expensive) product that she can afford. These variants, called the *Max-Buying* (respectively, *Min-Buying*) models, were proposed in Rusmevichientong et al [9, 10]. Assuming a price ladder, the Min-Buying model has been shown to be solvable in polynomial time using dynamic programming [9, 10].

2. In another variant, called *No-Price-Ladder (NPL)*, we do not assume a price ladder on the products.

3. In yet another variant, we are given costs of production, and the goal is to maximize the profit rather than the revenue.

4. We also consider the model where there are a limited number of copies of each product – inventory/production constraints might lead to such a condition.

We present algorithms and negative results for various combinations of these variations on the original model. Our results are summarized in the following table. The first column lists the particular model being considered in terms of price ladder or no price ladder, the buying behavior, and limited versus unlimited copies of products. The second column gives upper and lower bounds on the approximation bounds achievable in polynomial time. The last column gives certain extended scenarios where the algorithm remains applicable.

Model	Upper [Lower] Bounds	Extensions
PL & Rank-Buying	PTAS	Max-Buying instead of Rank-Buying
PL & Max-Buying & Limited-Copies	4	Consumers arrive online
NPL & Max-Buying	1.59 [16/15]	Maximize Profit instead of Revenue
NPL & Min-Buying	$\log m$ $[1+\varepsilon]$	Upper bound holds for all models

The rest of the paper is organized as follows. In Section 2, we present the PTAS and the 4-approximation algorithm for the Rank-Buying with Price-Ladder problem and related models. In Section 3, we discuss the No-Price-Ladder model. We present the 1.59-approximation algorithm and the hardness result for the Max-Buying with No-Price-Ladder problem, and the hardness result for the case of Min-Buying with No-Price-Ladder. Section 4 presents the $O(\log m)$-approximation algorithm that works for all models and discusses some related issues. Finally, we conclude with some open problems in Section 5.

2 Rank/Max Buying with Price-Ladder

We first show that when there is a Price-Ladder constraint, the Rank-Buying model can be reduced to the Max-Buying model[1].

[1]Note that the reduction is valid only in this setting, and may not work for other extensions and variations in this paper.

Lemma 1 *We can transform an instance I in the Rank-Buying model to an equivalent instance I' in the Max-Buying model.*

Proof Sketch: In the instance I, consider a consumer j and any two products i and i' such that $P_i \leq P_{i'}$ and $R_j(i) < R_j(i')$, where $R_j(i)$ denotes the position of product i in j's preferences list, with the most preferred product placed first. For such a pair of products, if the consumer could afford both products, Max-Buying would lead the consumer to buy i', while Rank-Buying would let her buy i, a product which is cheaper and more preferred. In order to reduce Rank-Buying to Max-Buying, we would like to eliminate all such product pairs without altering the solution to the instance. We note that for such product pairs i, i', since the budgets are assumed to be in the same order as the rank, the budget for i is higher than the budget for i'. Since $P_i \leq P_{i'}$, the consumer can afford i, whenever she can afford i', and since the consumer buys by rank, she would buy i rather than i'. Thus, we see that the consumer never buys i'. So we can reduce the budget $b_{i'j}$ to 0, without affecting the outcome in the Rank-Buying model. By repeating this for every product pair with conflicting rank and price orders, we can create an equivalent instance I' in which the rank(budget) order conforms to the price order for each consumer. Consequently, Max-Buying gives the same outcome as Rank-Buying on this new instance I', which in turn gives the same outcome as Rank-Buying on the original instance I. $\square$

We now present a PTAS for the Max-Buying model, which along with the above transformation, will give us a PTAS for the Rank-Buying model. We begin by noting that given any solution assigning prices to products, we can transform the solution to one in which the prices are taken from the set of distinct budget values, without decreasing the revenue generated by the solution. This general observation holds for all models studied in this paper.

Assume that the products are listed in the order of decreasing prices (as specified by the Price-Ladder), i.e., $P_1 \geq P_2 \geq \cdots \geq P_n$. We first relax the problem in two ways:

1. Let $\mathcal{B} = \max_{i,j} B_{ij}$. We discretize the search space of possible prices into values of the form $p_i = \mathcal{B}/s^i$, where $s > 1$ will be chosen later. (Note that P_i denotes the price of product i, while p_i denotes the value $\mathcal{B}/s^i$.)

2. We relax the constraint that a consumer can purchase at most one product. Instead, we allow the consumer to buy multiple products. However, if j buys a product at price p, then she is not allowed to buy any other product with price p', where $p \leq p' < p \times s^k$, where the integer k is chosen later.

Consider the modified instance. By the first relaxation, we lose a factor of s; by the second relaxation, we lose a factor of $1 + \frac{1}{s^k} + \frac{1}{s^{2k}} + \frac{1}{s^{3k}} + \cdots \leq \frac{1}{1-s^{-k}}$. Combining the two error factors gives a factor of $\frac{s}{1-s^{-k}}$, which is minimized at $s = (k+1)^{1/k}$, where it equals $(k+1)^{\frac{1}{k}}(1+\frac{1}{k}) = 1 + \frac{\log k}{k}(1+o(1))$. This approximation factor is 4 for $k = 1$ and can be made $1+\varepsilon$, for any constant ε, by taking a suitably large k.

We show how to obtain the optimal solution to this modified problem by dynamic programming. Define $F(i, x_i, x_{i-1}, x_{i-2}, \ldots, x_{i-k+1})$ to be the maximum revenue generated by only considering products with prices ranging from $p_0 = \mathcal{B}$ to p_i, with x_j being the last product with price set to p_j or higher. Note that in order to respect the price ladder, x_j must precede x_{j+1} or be equal to it in the price ladder. To compute $F(i+1, x_{i+1}, x_i, \ldots, x_{i-k+2})$, we enumerate through each choice of

x_{i-k+1} which is at most x_{i-k+2}. Let $C(x_{i+1}, x_i, \ldots, x_{i-k+2}, x_{i-k+1})$ denote the number of consumers that satisfy the following:

- For $i-k+2 \leq j \leq i$, if prices of all products $x_{j-1}+1, \ldots, x_j$ are set to p_j, the consumer will not be able to afford any of these products.
- If the price of products $x_i+1, \ldots, x_{i+1}$ was set to p_{i+1}, the consumer will be able to afford at least one of these products.

Let $G(i+1, x_{i+1}, \ldots, x_{i-k+2}, x_{i-k+1}) = F(i, x_i, \ldots, x_{i-k+1}) + p_{i+1} \times C(x_{i+1}, x_i, \ldots, x_{i-k+1})$. In other words, $G(i+1, x_{i+1}, \ldots, x_{i-k+2}, x_{i-k+1})$ is the sum of $F(i, x_i, \ldots, x_{i-k+1})$ and the payoff generated by consumers who buy products at price p_{i+1}, while ensuring that they haven't bought products of price p_j, for $i-k+2 \leq j \leq i$. We obtain the following recurrence:

$$F(i+1, x_{i+1}, \ldots, x_{i-k+2}) = \max_{x_{i-k+1} \leq x_{i-k+2}} G(i+1, x_{i+1}, \ldots, x_{i-k+2}, x_{i-k+1}).$$

This leads to the following theorem.

Theorem 1 *The Max-Buying with Price-Ladder problem has a PTAS. It can be approximated within a factor of* $(k+1)^{\frac{1}{k}}(1+\frac{1}{k}) = 1 + \frac{\log k}{k}(1+o(1))$ *in time* $\binom{n+k-1}{k} n^2 m^2 = O(n^{k+2} m^2)$ *and space* $O(n^{k+1} m)$. *With an additional approximation factor of* $1 + \frac{1}{\mathrm{poly}(n)}$, *the complexity can be improved to time* $O(n^{k+1} \log n)$ *and space* $O(n^k \log n)$.

Proof Sketch: The optimal solution will be the best value of $F(l, \ldots)$, where p_l is the smallest price considered. The number of choices for the arguments of $F(i, x_i, x_{i-1}, \ldots, x_{i-k+1})$ is nm for the value of i and $\binom{n+k-1}{k}$ for the x_j's. Note that there are at most $O(nm)$ distinct B_{ij}'s, and it is easy to see that there exists an optimal solution where the set of distinct prices is a subset of the set of distinct budget values. Each such value of F requires $O(nm)$ time and $O(1)$ space, giving the stated bounds. If we restrict the smallest allowed price to $\frac{\mathcal{B}}{n\mathrm{poly}(n)}$, we incur an additional $1 + \frac{1}{\mathrm{poly}(n)}$ approximation factor, but reduce the number of choices for i from nm to $O(\log n)$, giving the better time and space bounds. $\square$

In Appendix A we present another algorithm which requires only linear space, but has a worse (super-polynomial) time bound.

We now consider the Max-Buying with Price-Ladder model with the additional constraint that there are only N_i copies of the ith product. Since the number of copies of the products are limited, we need to redefine the optimal revenue generated by a setting of prices. First we assume that among the available products when a consumer arrives, she will buy the most expensive, still-available product that she can afford. We define the maximum revenue for a setting of prices as the payoff generated by the best order of arrival of consumers (or the best allocation of the limited copies of products to consumers in the offline setting). In the realistic online case of consumers arriving in any order, the payoff generated is within a factor 2 of the best possible allocation as shown by the following lemma.

Lemma 2 *Let OPT denote the revenue generated by the best allocation order. If* $\mathcal{R}(\pi)$ *denotes the revenue generated by the arrival order* π *of consumers, then* $\mathcal{R}(\pi) \geq \frac{1}{2} OPT$.

Proof Sketch: Let $A(i)$ (respectively, $B(i)$) be the set of consumers in the optimal (respectively, π) allocation buying product i at price P_i. Consider the values of i for which $|A(i)| > |B(i)|$. Since some copies of product i are still left over at the end of the π-allocation, the consumers in $A(i) - B(i)$ must all have bought some product with price at least P_i. So we charge the extra revenue generated by product i under optimal allocation to the revenue generated by these consumers under allocation π. Since each consumer only gets charged at most once, the extra revenue generated by optimal allocation is no more than the total revenue generated by π, and hence the lemma. □

Theorem 2 *The Limited-Copies variant can be approximated within a factor of 4, yielding an 8-approximation algorithm for the online case.*

Proof Sketch: We use the same relaxation techniques as in the PTAS above. We set $k = 1$, which leads to the relaxation where a consumer is allowed to buy one product at every price p_i. We set up a recursion for $F(i, x_i)$ (we enumerate over all possible values of $x_{i-1} \leq x_i$, adding the payoff from setting the price of products $x_{i-1}+1, \ldots, x_i$ to p_i to the optimal payoff $F(i-1, x_{i-1})$ from lower prices, and take the maximum over all these values to get $F(i, x_i)$). This gives us an approximation factor of 4. The reason higher values of k do not work lies in the difficulty of recursion in the dynamic programming. If we try to set up a recursion for $F(i, x_i, x_{i-1})$ instead of $F(i, x_i)$, the values x_i and x_{i-1} are not enough, in the limited copies scenario, to capture the information about which set of consumers will buy products with price p_i. □

The above PTAS can be extended to take competitors' products into account in the Rank-Buying model. In addition to the company's own products $A = \{1, \ldots, n\}$, this extension incorporates products $\overline{A} = \{n+1, \ldots, N\}$ produced by competing companies. Each consumer has a budget for each of the N products and a total ranking order on all the products. In addition, we assume that the prices of the competitors' products $P_{n+1}, \ldots, P_N$ are fixed and known. The consumer behaviour in case of ties is to buy the highest-ranked products that she can afford. If a consumer chooses a competitor's product, then the company receives no revenue from that consumer. The objective is to maximize the revenue generated by the company's products. We can reduce any instance of the problem involving competitors' products to an instance of the original problem involving only the company's products. This reduction works as follows. For each consumer j, we find the highest-ranked competing product, say C_j, that she can afford. Since consumer j is not going to buy any product ranked lower than C_j in j's preference list, we need to consider only those products of the company that are ranked higher than C_j (preferred more than C_j). If j can afford any of these products, by Rank-Buying she will purchase one of them and contribute to the revenue of the company. Thus, it is necessary and sufficient to modify j's preference list so as to include only those products of the company that are ranked higher than C_j. This model assumes that the competitors do not respond to the company's pricing decisions. The detailed proof of the following lemma can be found in [9].

Lemma 3 *The Max/Rank-Buying model can be generalized to include competitors' products.*

3 The No-Price-Ladder model

We now study a model where the prices of products are not constrained to follow any particular ordering. We first study the Max-Buying with No-Price-Ladder problem and give a 1.59-approximation algorithm for the problem. We also show that it is NP-hard to approximate this problem within $16/15$. Then, we discuss the Min-Buying with No-Price-Ladder model. The Min-Buying with Price-Ladder problem can be solved optimally by using dynamic programming [10]. However, removing the Price-Ladder constraint makes the problem hard.

3.1 An Algorithm for Max-Buying with No-Price-Ladder

The unordered nature of prices renders the previous approach ineffective. Instead, we use linear programming and rounding techniques to get a 1.59-approximation algorithm for this variant. We will also show how to derandomize this algorithm. Consider the following linear program:

$$
\begin{aligned}
\text{Maximize :}\quad & \textstyle\sum_p p \times y_{jp} & \\
\text{subject to :}\quad & \textstyle\sum_p x_{ip} = 1 & (1)\\
& y_{jp} \le \textstyle\sum_{p \le B_{ij}} x_{ip} & (2)\\
& \textstyle\sum_p y_{jp} \le 1 & (3)\\
& 0 \le x_{ip} \le 1 & \\
& 0 \le y_{jp} \le 1 &
\end{aligned}
$$

Here x_{ip} indicates that product i is assigned price p and y_{jp} indicates that consumer j buys an product at price p. Clearly if x_{ip} and y_{jp} take only values in $\{0,1\}$, the objective function is exactly the revenue we are maximizing. Instead relax the constraints on x_{ip} and y_{jp} to allow fractional solutions. We round the fractional optimal solution to obtain an integer solution by assigning product i price p with probability x_{ip}.

Theorem 3 *The Max-Buying with No-Price-Ladder problem can be approximated in polynomial time within a factor* $\frac{e}{e-1} < 1.59$.

Proof Sketch: We introduce some notation first. Let $s_{jp} = \sum_{p \le B_{ij}} x_{ip}$ be the total amount of (fractional) products priced at p which consumer j can afford. Let $t_{jp} = \sum_{p \le p'} s_{jp'}$ be the total amount of products priced p or higher which consumer j can afford to buy. Let $z_{jp} = \sum_{p \le p'} y_{jp'}$, the total amount of products consumer j buys at a price p or higher.

We thus have the following relations: $y_{jp} \le s_{jp}$ and $z_{jp} \le t_{jp}$. Then, the probability that consumer j buys at price at least p is $q_{jp} = 1 - \prod_i (1 - \sum_{p \le p' \le B_{ij}} x_{ip'})$. Then $q \ge 1 - (1 - \frac{t_{jp}}{n})^n > 1 - e^{-t_{jp}} \ge \min(t_{jp}, 1)(1 - e^{-1})$; recall the definition of t_{jp}.

We now look at the optimal fractional solution. Consumer j buys at price at least p with fractional value exactly $r_{jp} = \min(t_{jp}, 1)$. This implies that q_{jp} majorizes $(1 - e^{-1}) r_{jp}$. Thus, we get that the expected value of our rounded solution is at least a $(1 - e^{-1})$ fraction of that of the optimal fractional solution, thus giving us an approximation factor of 1.59.

The algorithm can be derandomized by replacing any two $0 < x_{ip} < 1$ and $0 < x_{ip'} < 1$ by $x_{ip} + \varepsilon$ and $x_{ip'} - \varepsilon$, for a suitable ε. Since the expected payoff is a linear function of ε, either any positive ε makes the payoff nondecreasing, or else any negative ε ensures the payoff does not decrease. We may select such an ε with the appropriate sign to obtain $x_{ip} = 0$ or $x_{ip'} = 0$. Repeatedly performing this transformation ensures $x_{ip} = 0$ or $x_{ip} = 1$ for all x_{ip}. □

We remark that this linear programming formulation can be extended to the model of maximizing profit instead of revenue. For each product i, let $c(i)$ denote the fixed manufacturing cost. We simply change the definition of x_{ip} to be product i is assigned profit p, and y_{jp} to be consumer j buys a product with profit p. We substitute inequality 2 in our linear programming formulation with the following: $y_{jp} \leq \sum_{p:p+c(i)\leq B_{ij}} x_{ip}$. The rest of the analysis goes through as before.

We now turn to the hardness result for the Max-Buying with No-Price-Ladder model.

3.2 Negative Result for Max-Buying with No-Price-Ladder

Consider a special case of this problem where each consumer j specifies a set S_j of products she is interested in. In addition, her budget $b_{ij} = B_j$ for $i \in S_j$, and $b_{ij} = 0$ otherwise. Also, $B_j \in \{a, b\}$, for all j. We call this the *Uniform-two-budget* problem. We show below that even this special case of the Max-Buying with No-Price-Ladder problem is MAX SNP-hard.

Theorem 4 *The Uniform-two-budget problem with Max-Buying and No-Price-Ladder cannot be approximated within* $16/15$ *unless* $P = NP$*. There exists a polynomial time algorithm to approximate it within* $1/0.78$*.*

Proof Sketch: If there are only two distinct budget values $a > b$, then the only prices used in an optimal solution are a and b. A consumer with budget a will always spend b, and may or may not spend the additional $a - b$. For every product i, consider a boolean variable x_i with $x_i = 1$ if i has price a and $x_i = 0$ if i has price b. Since we are considering the Max-Buying setting, a consumer j with budget a will pay the additional $a - b$ if the disjunction of x_i for the products in S_j holds, while a consumer j with budget b will pay b if the disjunction of the $\overline{x_i}$ for the products in S_j holds. The problem is thus an instance of MAX-SAT with disjunctions of positive literals having weight $a - b$ and disjunctions of negative literals having weight b. Since the MAX-SAT problem can be approximated within $1/0.78$ [1], this gives us an algorithm for solving the Uniform-two-budget case with an approximation factor of $1/0.78$.

For the hardness result, we reduce MAX-3SAT, which is hard to approximate within $\frac{8}{7} - \varepsilon$ for any $\varepsilon > 0$ [4], to our problem. Consider an instance of MAX-3SAT. Replace clauses that have both positive and negative literals by two clauses that have only positive or only negative literals, such that the original clause is satisfied if both the new clauses are satisfied. For example, the clause $x \vee y \vee \overline{z}$ is replaced by clauses $x \vee y \vee t$ and $\overline{t} \vee \overline{z}$. Since the number of clauses at most doubles, the modified problem is hard to approximate within $\frac{16}{15} - \varepsilon$. Now we reduce this modified instance to an instance of our problem. Let n be the number of clauses. We create consumers corresponding to every clause, and the literals in each clause correspond to the products of interest to the consumers (i.e. S_j). For a product, setting the corresponding variable to be *true* (respectively, *false*) corresponds to setting the price to a (respectively, b). For the positive clauses, there is one consumer with $B_j = a = n$, while for the negative clauses, we have n identical consumers, each with budget $B_j = b = 1$. Let n_a be the number of consumers with budget a. A price setting where n_{ab} of the n_a consumers with budget a spend $b = 1$ has a contribution of n_{ab} from these consumers to the total payoff. Since setting the price of all products to $a = n$ leads to a payoff of $n \times n_a$, and $n_{ab} \leq n_a$, in the optimal solution, the fraction of the total payoff contributed by these n_{ab} consumers is at most $1/n$, which is negligible. Thus any $16/15-\varepsilon$ approximate solution for this multi-product pricing instance leads to a $16/15-\varepsilon$ approximate solution for the MAX-SAT instance. Thus, the Uniform-two-budget case of the Max-Buying with No-Price-Ladder problem cannot be approximated within $16/15$ unless $P = NP$. □

We now give a hardness result for the Min-Buying with No-Price-Ladder problem.

3.3 Min-Buying with No-Price-Ladder

We consider the Uniform-two-budget case of the problem where each consumer j specifies a set S_j of products she is interested in, and her budget $b_{ij} = B_j$ for $i \in S_j$, and $b_{ij} = 0$ otherwise. Also, $B_j \in \{a,b\}$, for all j.

Theorem 5 *The Uniform-two-budget case of the Min-Buying with No-Price-ladder problem is NP-hard to approximate within* $1+\varepsilon$*, for some constant* $\varepsilon > 0$*.*

Proof Sketch: We do a reduction from the following MAX-CSP. Consider the MAX-SAT problem consisting of the following two types of clauses: conjunction of only positive literals and disjunction of only negative literals. We first show that this MAX-CSP is MAX SNP-hard. For this, we will show that it is NP-hard. Then, the results from Khanna, Sudan, and Williamson [5] will imply that this MAX-CSP problem is MAX SNP-hard. We now show that even if the conjunctions contain only singleton x_i's and the disjunctions are of the form $\overline{x_i} \vee \overline{x_j}$ with two literals, the MAX-CSP problem is NP-hard. We first note that there exists an optimal solution to this MAX-CSP which maximizes the number of x_i set to 1 while satisfying all $\overline{x_i} \vee \overline{x_j}$ clauses (If a solution does not satisfy all disjunctive clauses, we can convert it into a equally good or better solution having all the disjunctive clause satisfied by negating either of the literals for an unsatisfied disjunctive clause). Now, if we have a vertex i for each x_i and view the disjunctions $\overline{x_i} \vee \overline{x_j}$ as edges (i, j), then this MAX-CSP is equivalent to the the maximum independent set problem, which is NP-hard to solve.

Given an instance of this MAX-CSP, we reduce it to an instance of the Min-Buying with No-Price-Ladder problem as follows. There are two distinct budget values $a > b$ with $a = 2b$. Corresponding to each variable x_i, we have a product i. For a product, setting the price to a (respectively, b) corresponds to setting the corresponding variable to be *true* (respectively, *false*). Corresponding to a conjunctive clause, we have a consumer with budget a interested in the products appearing in the clause. Similarly, corresponding to a disjunctive clause, we have a consumer with budget b interested in the products appearing in the clause. Since we are in the Min-Buying setting, a consumer with budget a will always pay b, and will pay the an additional b if the conjunction of the x_i for the products it is interested in holds. A consumer with budget b will pay b if the disjunction of the $\overline{x_i}$ for the products it is interested in holds. If the maximum number of satisfiable clauses is at least a constant fraction of the total number of clauses, then the MAX SNP-hardness of the MAX-CSP implies MAX SNP-hardness of this case of the pricing problem. □

4 General Algorithms for All Models

In this section, we present an algorithm and hardness result applicable to all six models – {Rank-Buying, Max-Buying, Min-Buying} with {Price-Ladder, No-Price-Ladder}. Recall that the number of products is n while the number of consumers is m.

Theorem 6 *Consider the Rank/Max/Min-Buying problem with Price-Ladder/No-Price-Ladder. An algorithm which is allowed to assign only k distinct prices to the products cannot have an approximation ratio better than* $\frac{H_m}{k}$ *with respect to an unrestricted optimal solution.*

Proof Sketch: To show this lower bound, we create an instance with $n = m$. Consider a situation where a consumer j is interested only in product j and has budget m/j for it and 0 for all other products. In the optimal solution, product i is priced m/i. Thus, in the optimal solution, consumer j spends his budget m/j, and the total payoff is mH_m. Now consider a solution which assigns only k distinct prices to the products. Products priced at any single price m/j can be afforded by at most the first j consumers, thus giving a payoff of $\leq m$ for that price. Thus the total payoff with k distinct prices is at most km. This gives the H_m/k lower bound. □

When $k = 1$, the above bound is tight. Let $B_j^{max} = \max_i B_{ij}$. Assume that the consumers are ordered by their maximum budgets, B_j^{max}, in decreasing order. If we set the prices of all products to B_j^{max}, then the first j consumers will be able to afford some product they are interested in, and pay price B_j^{max} for it (irrespective of the policy governing consumer behavior in case of ties). Thus, the payoff generated will be jB_j^{max}. If $j' = \arg\max_j jB_j^{max}$, then we set the price of all products to be $B_{j'}^{max}$. For this single price algorithm, we get the following theorem.

Theorem 7 *The revenue generated by the single price algorithm is an* H_m*-approximation to the optimal revenue.*

Proof Sketch: We note that the revenue generated by an optimal solution $OPT \leq \sum_j B_j^{max}$, since each consumer j spends at most B_j^{max}, her maximum budget over all products. Let $\mathcal{R}$ be the revenue generated by the single price algorithm. Then, $\mathcal{R} = j'B_{j'}^{max} \geq jB_j^{max}$ for all j. Thus, $OPT \leq \sum_j B_j^{max} \leq \sum_j \mathcal{R}/j = \mathcal{R}H_m$. □

5 Conclusion and Open Problems

In this paper, we studied the non-parametric multi-product pricing problem. We presented a PTAS for the realistic Rank-Buying with Price-Ladder model, thus providing a useful pricing tool to companies with access to customer profiles. We also present approximation algorithms and complexity analysis for various extensions and variants of the model.

Many problems still remain open. The complexity of the Rank-Buying with Price-Ladder problem is unresolved. It will be interesting to extend the results for profit maximization to the Rank-Buying with Price-Ladder problem. One can also study other extensions of the multi-product pricing problem. One possible variation is a model where a consumer's budget for a product represents its valuation (utility) for the product, and consumer behaviour in case of multiple affordable products is to buy the product which gives her the maximum gain (gain for a product is defined as the difference between the valuation and the price). Another useful extension might be to consider a *budget range* for each consumer – each consumer has a lower as well as upper bound on the amount they are willing to spend to buy a product. Along with Min-Buying, such a model might be a good reflection of consumer behavior in case of ties.

References

[1] T. Asano, and D. Williamson. "Improved approximation algorithms for MAX SAT." In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 96–115, 2000.

[2] P. Coy. "The Power of Smart Pricing." *Business Week*, April 10, 2000

[3] General Motors. *The Auto Choice Advisor web site.* `http://www.autochoiceadvisor.com`.

[4] J. Håastad. "Some Optimal Inapproximability results." In *Proceedings of the 28th ACM Symposium on the Theory of Computing*, pp. 1–10, 1997.

[5] S. Khanna, M. Sudan, and D. Williamson. "A Complete Classification of the Approximability of Maximization Problems Derived from Boolean Constraint Satisfaction." In *Proceedings of the 29th ACM Symposium on the Theory of Computing*, pp. 11–20, 1997.

[6] J. I. McGill, and G. Van Ryzin. "Revenue Management: Research Overview and Prospects." *Transportation Science*, vol. 33, no. 2, pp. 233–256, 1999.

[7] G. McWilliams. "Lean Machine: How Dell Fine Tunes Its PC Pricing to Gain Edge in a Slow Market." *Wall Street Journal*, June 8, 2001.

[8] A. Merrick. "Priced to Move: Retailers Try to Get Leg Up on Markdowns With New Software." *Wall Street Journal*, August 7, 2001

[9] P. Rusmevichientong. "A Non-Parametric Approach to Multi-Product Pricing: Theory and Application." PhD Thesis, 2003.

[10] P. Rusmevichientong, B. Van Roy, and P. Glynn. "A Non-Parametric Approach to Multi-Product Pricing." Submitted to *Operations Research*, 2002.

[11] L. R. Weatherford, and S. E. Bodily. "A Taxonomy and Research Overview of Perishable-Asset Revenue Management: Yield Management, Overbooking, and Pricing." *Management Science*, vol. 40, no. 5, pp. 831-844, 1992.

A Alternate Algorithm for Price-Ladder with Rank/Max-Pricing

Theorem 8 *For the Max-Buying with Price-Ladder model, an approximation ratio of* $1+\varepsilon(1+o(1))$ *can be achieved in time* $O(n^{\frac{1}{\varepsilon}\log\log\log n})$ *and space* $O(n)$ *for any* $\varepsilon > 0$ *by choosing* $\frac{\log n}{\varepsilon}$ *distinct prices.*

Proof Sketch: We modify the instance so that every product be of interest to at most one consumer. In the original instance, if m_i consumers are interested in the same product, we create m_i products for each of the m_i consumers in the new instance, so that the m_i products in decreasing price correspond to the consumers in order of increasing budget. Thus, an optimal solution to the new instance will assign the same price to all these m_i products. Hence, an optimal solution to the new instance can easily be translated to a solution for the original one.

We may assume that the prices used are of the form $r/(1+\mu)^i$, where r is the highest price, and no smaller than $\delta r/n$ costing an approximation factor of $1+(\mu+\delta)(1+o(1))$, and giving at most $k = \frac{\log n}{\mu}(1+o(1))$ possible prices.

We may assume that if s is the highest income possible with a single price, then every chosen price gives income at least $\delta s/k$, costing an approximation factor of $1+\delta$. If we choose the number of consumers that will pay a price to be of the form $(1+\delta)^i$, then we lose another approximation factor of $1+\delta$, and there are at most $\frac{\log k}{\delta}(1+o(1))$ possible choices for the number of consumers, for a given price.

The number of choices of how many consumers will pay which price is thus $R = (\frac{\log k}{\delta})^k = (\frac{\log\log n - \log\mu}{\delta})^{\frac{\log n}{\mu}} = n^{\frac{1}{\mu}(\log(\log\log n - \log\mu) - \log\delta)}$. If we choose $\mu + 3\delta = \mu(1+o(1))$ and $\mu = \varepsilon(1+o(1))$, then the approximation factor is $1+\varepsilon(1+o(1))$ and $R = n^{\frac{1}{\varepsilon}\log\log\log n}$.

Once we have chosen how many consumers will pay which price, the solution chooses certain consumers to pay these prices. We can greedily start with the highest priced product and select the chosen number of consumers to pay the chosen price. If the solution that gives the desired payoff skips an product q between prices w_i and w_{i+1}, but the consumer t could have paid price w_{i+1} for q but was paying price w_{i+j} for another product, then we can assign price w_{i+1} to q and remove the assigned price w_{i+1} to some q', so that q' can be given price w_{i+2}, and so on until some product is given price w_{i+j} to compensate for the fact that t is no longer purchasing an product of price w_{i+j}. This justifies the greedy assignment of prices to products to achieve the desired choice out of R, and completes the proof. $\square$