Near-Optimal Pricing in Near-Linear Time*

Jason D. Hartline[1] and Vladlen Koltun[2]

[1] Microsoft Research, Mountain View, CA, 94043.
hartline@microsoft.com.
[2] Computer Science Division, University of California, Berkeley, CA 94720-1776.
vladlen@cs.berkeley.edu.

Abstract. We present efficient approximation algorithms for a number of problems that call for computing the prices that maximize the revenue of the seller on a set of items. Algorithms for such problems enable the design of auctions and related pricing mechanisms [3]. In light of the fact that the problems we address are APX-hard in general [5], we design near-linear and near-cubic time approximation schemes under the assumption that the number of distinct items for sale is constant.

1 Introduction

Imagine a software provider that is about to release a new product in *student*, *standard*, and *professional* editions. How should the editions be priced? Naturally, high prices lead to more revenue per sale, whereas lower prices lead to more sales. There is also interplay between the prices on the different versions, e.g., a consumer willing to pay a high price for the professional version might be lured away by a bargain price on the standard version. *Given consumer preferences over a set of items for sale, how can the items be priced to give the optimal profit?*

Consumer preferences are rich in combinatorial structure and this leads to much of the difficulty in price optimization. General *combinatorial* preferences might have aspects of both *substitutability* and *complementarity*. As an example, a vacationer might like either a plane ticket to Hawaii together with a beachfront hotel, or a ticket to Paris with a hotel on Ile St. Louis. We address the pure complements case (i.e., *single-minded* consumers that want a set bundle of products for a specific price) and the pure substitutes case (i.e., *unit-demand* consumers that want exactly one unit of any of the offered products and have different valuations on the products). In the above software pricing scenario, it is natural to assume that consumers are unit-demand. See the following sections for definitions.

These variants of the pricing problem are considered by Guruswami et al. [5] who show that even in simple special cases, these pricing problems are APX-hard. (That is, there is no polynomial time approximation scheme given standard complexity assumptions.) They give a logarithmic factor approximation algorithm

* Work on this paper by the second author has been supported by NSF Grant CCR-01-21555.

and leave the problem of obtaining a better approximation factor open. Unfortunately, the logarithmic approximation does not yield any insight on questions like the pricing problem faced by the software company in our opening example; it would recommend selling all editions of the software at the same, albeit optimally chosen, price. Obviously, this defeats the purpose of making different editions of the software and is thus inadequate.

In this paper we consider the above pricing problems under the assumption that the number of distinct items for sale is constant. As is illustrated by the software pricing example, this assumption is pertinent to many real-life scenarios. A brute-force exact algorithm exists, but is exponential in the number of items for sale and generally impractical. We present approximation schemes with significantly superior running times. As the APX-hardness of the problems indicates, simplifying assumptions such as this are necessary for the achievement of such results.

In the *unlimited supply* case in which the seller is able to sell any number of units of each item, we give near-linear time approximation schemes for both the pure substitutes problem and the pure complements problem. Specifically, for unit-demand consumers a $(1+\varepsilon)$-approximation is achieved in time $O\left(n \log\left(\frac{1}{\varepsilon}\log\frac{n}{\varepsilon}\right) + \left(\frac{1}{\varepsilon}\log\frac{n}{\varepsilon}\right)^{m(m+1)}\right)$ for n consumers, $m = O(1)$ items, and an arbitrarily small $\varepsilon > 0$. For single-minded consumers a $(1+\varepsilon)$-approximation is achieved in time $O\left(\left(n + \left(\frac{1}{\varepsilon}\log\frac{n}{\varepsilon}\right)^{m}\right)\log\left(\frac{1}{\varepsilon}\log\frac{n}{\varepsilon}\right)\right)$.

The more general *limited supply* case requires more care. In particular, we demand that the computed prices satisfy an explicit fairness criterion called *envy-freedom* [5] that makes sure that the prices are such that no item is oversold.[3] For unit-demand consumers we give a $(1+\varepsilon)$-approximation algorithm for the limited supply (envy-free) pricing problem that runs in time $O(n^3 \log^m_{1+\varepsilon} n)$.

Part of the motivation for considering these pricing problems comes from auction mechanism design problems. In mechanism design it is pointedly assumed that the seller does not know the consumer preferences in advance. Instead, an auction must compute payments that encourage the consumers to reveal their preferences. Intuitively, however, understanding how to optimally price items given *known* preferences is necessary for the more difficult problem of running an auction when the preferences are unknown in advance. For the unlimited supply unit-demand scenario, Goldberg and Hartline [3] made this connection concrete by giving a reduction from the game theoretic auction design problem to the algorithmic price optimization problem. They left the problem of computing optimal or approximately optimal prices from known consumer preferences in polynomial time open. This is one of the questions addressed by our work.

This paper is organized as follows. In Sections 2 and 3 we describe the unlimited supply algorithms for the pure substitutes and pure complements problems, respectively. In Section 4 we formally define envy-freedom, give background material, and present the approximation algorithm for the limited supply pure

[3] Envy-freedom is implicit in the definition of unlimited supply pricing problem since we require that the identical units of each item be sold at the same price and that each consumer pick their desired items after the prices are set.

substitutes problem. We conclude in Section 5 with a discussion of the difficulty in generalizing our approach to combinatorial preferences that contain both substitutes and complements.

2 Unlimited Supply, Unit-Demand Consumers

We assume that the seller has m distinct items for sale, each available in *unlimited supply*. There are n consumers each of whom wishes to purchase at most one item (i.e., there is *unit-demand*). We define a consumer's *valuation* for an item as the value assigned by the consumer to obtaining one unit of the item. For consumer i and item j, let v_{ij} denote this valuation. Given a price p_j for item j, consumer i's *utility* for this item is $u_{ij} = v_{ij} - p_j$. We assume that a consumer's only goal is to maximize this utility. Therefore, given a pricing of all items, $\mathbf{p} = (p_1, \ldots, p_m)$, consumer i will choose one of the items j such that $v_{ij} - p_j \geq v_{ij'} - p_{j'}$ for all $j' \neq j$, or no item if $v_{ij} < p_j$ for all j.

The pricing problem we address asks for computing, given the consumer valuations, a pricing that is *seller optimal*, i.e., the price vector that maximizes the sum of the prices of all the units of items sold, also called the seller's *profit*. For a given price vector $\mathbf{p}$, let $\text{Profit}_{\mathbf{p}}$ be the profit obtained from $\mathbf{p}$. Let $\bar{\mathbf{p}}$ be the price vector with the maximum profit and set $\text{OPT} = \text{Profit}_{\bar{\mathbf{p}}}$. The following assumption is natural in many context (See Section 4) and insures that $\text{Profit}_{\mathbf{p}}$ is well defined: a consumer indifferent between several items will choose following the discression of the pricing algorithm. For our goal of unlimited supply profit maximization we assume that indifferent consumers consumers choose the item with the higher price.

This problem was first posed in [3]. Both Guruswami et al. [5] and Aggarwal et al. [1] give logarithmic approximations and APX-hardness proofs that hold even for the special case where $v_{ij} \in \{0, 1, 2\}$. For the case where the number of distinct items for sale is constant we get the following result.

Theorem 1. *For any $\varepsilon < 1$, an envy-free pricing that gives profit at least $\text{OPT}/(1+\varepsilon)$ can be computed in time $O(n \log(Mm^2) + m^{2m+1} M^{m+1})$, where $M = O(m^m \log_{1+\varepsilon}^m \frac{n}{\varepsilon})$. For constant m, the running time is*

$$O\left(n \log\left(\frac{1}{\varepsilon} \log \frac{n}{\varepsilon}\right) + \left(\frac{1}{\varepsilon} \log \frac{n}{\varepsilon}\right)^{m(m+1)}\right).$$

In brief synopsis, the analysis proceeds as follows. We first show that there is an approximately optimal price vector $\tilde{\mathbf{p}}$ that satisfies $\text{Profit}_{\tilde{\mathbf{p}}} \geq (1-\delta)\,\text{OPT}$ with the property that for all j, $\tilde{p}_j \in [\frac{\delta h}{n}, h]$ where $h = \max_{i,j} v_{ij}$ is the highest valuation of any user for any item. We then show that we can cover the space of price vectors $[\frac{\delta h}{n}, h]^m$ with a "small" set of price vectors $\mathcal{W}$, such that there exists $\mathbf{p} \in \mathcal{W}$ with $\text{Profit}_{\mathbf{p}} \geq \text{Profit}_{\tilde{\mathbf{p}}}/(1+\delta)$. A brute-force search over $\mathcal{W}$ can find the optimal such vector $\mathbf{p} \in \mathcal{W}$ in time $O(nm\,|\mathcal{W}|)$ which is $O(n \log n)$ for constant ε and m. A more clever search yields the stated runtime $O(n \log\log n)$.

From a practical standpoint the brute-force approach may be more desirable due to its great simplicity of implementation.

Lemma 1. *There exists* $\tilde{\mathbf{p}}$ *with* $\tilde{p}_j \in [\frac{\delta h}{n}, h]^m$ *for all* j *and* $\text{Profit}_{\tilde{\mathbf{p}}} \geq (1-\delta)\,\text{OPT}$.

Proof. First, we can assume that $\bar{\mathbf{p}}$ satisfies $\bar{p}_j \leq h$ as setting a price above the highest valuation cannot increase the profit. Indeed, the revenue from an item priced this way can only be zero. Consider $\tilde{\mathbf{p}}$ with $\tilde{p}_j = \max(\bar{p}_j, \frac{\delta h}{n})$. Let J' be the set of items with $\tilde{p}_j = \bar{p}_j$ (all other items have price $\tilde{p}_j = \frac{\delta h}{n} > \bar{p}_j$). Any consumer that prefers an item from J' under pricing $\bar{\mathbf{p}}$ prefers the same item under $\tilde{\mathbf{p}}$. This is because we kept the price of this preferred item fixed and only raised prices of other items. On the other hand, the total profit from consumers that preferred items with $\bar{p}_j < \frac{\delta h}{n}$ is at most $\delta h \leq \delta\,\text{OPT}$. (Here we note that $\text{OPT} \geq h$, since for $h = v_{ij}$, setting the price of all items to h ensures that consumer i will purchase item j, yielding profit at least h.) Even if we assume no revenue from these consumers under $\tilde{\mathbf{p}}$, the profit is still $\text{Profit}_{\tilde{\mathbf{p}}} \geq (1-\delta)\,\text{OPT}$. □

We define below a grid of prices $\mathcal{W}$ parameterized by $\delta > 0$ to fill a region slightly larger than $[\frac{\delta h}{n}, h]^m$ such that there is a grid point $\mathbf{p} \in \mathcal{W}$ that gives a profit close to the optimal price vector $\tilde{\mathbf{p}}$ for the region $[\frac{\delta h}{n}, h]^m$. For integer $0 \leq i < \log_{1+\delta} \frac{n}{\delta}$ and $0 \leq k < (2+\delta)m$, let Z be the $\lceil \log_{1+\delta} \frac{n}{\delta} \rceil$ values of $Z_i = \frac{\delta h}{n}(1+\delta)^i$ on the interval $[\frac{\delta h}{n}, h)$ and let W_i be the $\lceil (2+\delta)m \rceil$ values of the form $Z_{i-1} + Z_{i-1}\frac{\delta k}{m}$ on the interval $[Z_{i-1}, Z_{i+1})$. Let $W = \bigcup_i W_i$ for i with $Z_i \in Z$. Define the sets $\mathcal{W} = W^m$ and $\mathcal{Z} = Z^m$. Let $M = |\mathcal{W}| = \left(\lceil (2+\delta)m \rceil \lceil \log_{1+\delta} \frac{n}{\delta} \rceil\right)^m$.

Lemma 2. *For any* $\tilde{\mathbf{p}} \in [\frac{\delta h}{n}, h]^m$*, there exists* $\mathbf{p} \in \mathcal{W}$ *such that* $\text{Profit}_{\mathbf{p}} \geq \text{Profit}_{\tilde{\mathbf{p}}}/(1+\delta)$.

Proof. Reindex the items so that $\tilde{p}_j \leq \tilde{p}_{j+1}$ for all j. For each j, let Z_{i_j} (resp., w_j) be the price obtained by rounding $\tilde{p}_j$ down to the nearest value in Z (resp., in W_{i_j}). Set $d_j = \frac{\delta}{m} Z_{i_j}$ and consider the price vector $\mathbf{p}$ defined with $p_j = w_j - j d_j \in W_{i_j}$. We show that $\mathbf{p} \in \mathcal{W}$ satisfies the statement of the lemma.

We claim that no consumer that prefers an item j under $\tilde{\mathbf{p}}$ would prefer an item $j' < j$ under pricing $\mathbf{p}$ (i.e., one with a lesser price). In going from prices $\tilde{\mathbf{p}}$ to $\mathbf{p}$, the increase $\tilde{p}_j - p_j$ in a consumer's utility for item j, is higher than the increase $\tilde{p}_{j'} - p_{j'}$ for item j'. Given $j' + 1 \leq j$ we have:

$$\begin{aligned} \tilde{p}_j &\geq w_j & \qquad \tilde{p}_{j'} &< w_{j'} + d_{j'} \\ \tilde{p}_j - p_j &\geq j d_j & \qquad \tilde{p}_{j'} - p_{j'} &< (j'+1) d_{j'}. \end{aligned}$$

Since $d_j \geq d_{j'}$ and therefore $j d_j \geq (j'+1) d_{j'}$, the desired inequality is established.

Since $p_j \geq \tilde{p}_j/(1+\delta)$ and no consumer changes preference to a cheaper item under $\mathbf{p}$, we have $\text{Profit}_{\mathbf{p}} \geq \text{Profit}_{\tilde{\mathbf{p}}}/(1+\delta)$. □

Combining Lemmas 1 and 2 and taking $\varepsilon = \Theta(\delta)$, we conclude that there exists $\mathbf{p} \in \mathcal{W}$ such that $\text{Profit}_{\mathbf{p}} \geq \text{Profit}_{\bar{\mathbf{p}}} / (1+\varepsilon)$.

It remains to show that the optimal price vector from $\mathcal{W}$ can be found efficiently. The trivial algorithm enumerates all $\mathbf{p} \in \mathcal{W}$ and computes the profit of each, which takes time $O(Mmn)$. Below is an improved algorithm that utilizes the assumption that $n \gg M$, which suggests the possibility of preprocessing the consumer valuations into a data structure that can be queried at each $\mathbf{p} \in \mathcal{W}$.

A given price vector $\mathbf{p}$ divides $\mathbb{R}^m$ (which we view as the space of consumer valuation vectors) into $m+1$ regions, each corresponding to how a consumer valuation relates to the given prices. Given $\mathbf{p}$, each consumer prefers one of m items or none. The boundaries of all the regions are delimited by $O(m^2)$ hyperplanes. For all M price vectors in $\mathcal{W}$ we have a total of $O(Mm^2)$ hyperplanes. We use the fact that an arrangement of K hyperplanes in $\mathbb{R}^m$ can be preprocessed in time $O(K^m / \log^m K)$, such that a point location query can be performed in time $O(\log K)$ [2].

Definition 1 (Unlimited Supply Pricing Algorithm).

1. *Preprocess the hyperplane arrangement defined by the M price points into a point location data structure. Associate a counter with each arrangement cell, initially set to zero. Analysis: $O((Mm^2)^m / \log^m(Mm^2))$ time.*
2. *For each consumer, query the point location data structure with the consumer's valuation vector. Increment the counter associated with the located cell. Analysis: $O(n \log(Mm^2))$ time.*
3. *For each $\mathbf{p} \in \mathcal{W}$, iterate over all arrangement cells. In each cell, multiply the value of the associated counter by the price yielded by the preferred item for that cell (if any), and add this to the profit associated with $\mathbf{p}$. Analysis: $O(Mm(Mm^2)^m)$ time.*
4. *Output $\mathbf{p} \in \mathcal{W}$ with highest profit.*

The total runtime of the unlimited supply pricing algorithm is $O(n \log(Mm^2) + Mm(Mm^2)^m)$. Taking m to be a constant, $M = O(\log^m_{1+\varepsilon} \frac{n}{\varepsilon})$ and we have a total runtime of $O(n \log\left(\frac{1}{\varepsilon} \log \frac{n}{\varepsilon}\right) + \left(\frac{1}{\varepsilon} \log \frac{n}{\varepsilon}\right)^{m(m+1)})$. This concludes the proof of Theorem 1.

3 Unlimited Supply, Single-Minded Consumers

As in the unit-demand case of the preceding section, we assume that the seller has m distinct items available for sale, each in *unlimited supply*. Each of the n consumers is interested in purchasing a particular bundle $S_i \subseteq \{1, \ldots, m\}$ of items. Denote consumer i's valuation for their desired bundle by v_i and let $h = \max_i v_i$. We assume that if the total cost of the items in S_i is at most v_i, the consumer will purchase all of S_i, otherwise the consumer will purchase nothing. Given the valuations v_i, we wish to compute a pricing that maximizes the seller's profit. Define $\text{Profit}_{\mathbf{p}}$, $\bar{\mathbf{p}}$, and OPT as in the previous section.

Guruswami et al. [5] give an algorithm for computing a logarithmic approximations to OPT and an APX-hardness proof that hold evens for the special case where $v_i = 1$ and $|S_i| \leq 2$. For the case where the number of distinct items for sale is constant we get the following result.

Theorem 2. *For any $\varepsilon < 1$, a pricing that gives profit at least* $\mathrm{OPT}/(1+\varepsilon)$ *can be computed in time $O((n + 2^m M)\log M)$, where $M = O(\log^m_{1+\varepsilon} \frac{n}{\varepsilon}))$. For constant m, the running time is*

$$O\left(\left(n + \left(\frac{1}{\varepsilon}\log\frac{n}{\varepsilon}\right)^m\right)\log\left(\frac{1}{\varepsilon}\log\frac{n}{\varepsilon}\right)\right).$$

Lemma 3. *There exists $\tilde{\mathbf{p}}$ with $\tilde{p}_j \in \{0 \cup [\frac{\delta h}{nm}, h]\}^m$ for all j and* $\mathrm{Profit}_{\tilde{\mathbf{p}}} \geq (1-\delta)\,\mathrm{OPT}$.

Proof. As in Lemma 1, we can assume that $\bar{\mathbf{p}}$ satisfies $\bar{p}_j \leq h$. Consider $\tilde{\mathbf{p}}$ with $\tilde{p}_j = 0$ if $\bar{p}_j \leq \frac{\delta h}{nm}$ and $\tilde{p}_j = \bar{p}_j$ otherwise. It is clear that the overall profit decreases by at most δh when prices shift from $\bar{\mathbf{p}}$ to $\tilde{\mathbf{p}}$. It is easy to see that $\mathrm{OPT} \geq h$ and thus $\mathrm{Profit}_{\tilde{\mathbf{p}}} \geq (1-\delta)\,\mathrm{OPT}$. □

For integer $0 \leq i < \lceil \log_{1+\delta} \frac{nm}{\delta} \rceil$, define Z to be the $\lceil \log_{1+\delta} \frac{nm}{\delta} \rceil$ values of $Z_i = \frac{\delta h}{nm}(1+\delta)^i$ on the interval $[\frac{\delta h}{nm}, h)$, augmented by the value 0. Define $\mathcal{Z} = Z^m$. Let $M = |\mathcal{Z}| = O(\log^m_{1+\delta}(\frac{nm}{\delta}))$.

Lemma 4. *For any $\tilde{\mathbf{p}} \in [\frac{\delta h}{nm}, h]^m$, there exists $\mathbf{p} \in \mathcal{Z}$ such that* $\mathrm{Profit}_{\mathbf{p}} \geq \mathrm{Profit}_{\tilde{\mathbf{p}}}/(1+\delta)$.

Proof. Let $\mathbf{p} \in \mathcal{Z}$ be the price vector obtained by taking the coordinates of $\tilde{\mathbf{p}}$ and rounding each of them down to the nearest value in Z. Since the price of any bundle under $\mathbf{p}$ is at least the price of this bundle under $\tilde{\mathbf{p}}$ divided by $1+\delta$, we have $\mathrm{Profit}_{\mathbf{p}} \geq \mathrm{Profit}_{\tilde{\mathbf{p}}}/(1+\delta)$. □

We conclude that by setting $\delta = \Theta(\varepsilon)$, there exists $\mathbf{p} \in \mathcal{Z}$ such that $\mathrm{Profit}_{\mathbf{p}} \geq \mathrm{Profit}_{\bar{\mathbf{p}}}/(1+\varepsilon)$. The following algorithm computes the optimal price vector from $\mathcal{Z}$.

Definition 2 (Single-Minded Consumers Pricing Algorithm).

1. *Compute the profit for each bundle and price vector, $(S, \mathbf{p}) \in 2^{\{1,\ldots,m\}} \times \mathcal{Z}$:*
   (a) *Compute total cost of S given $\mathbf{p}$: $\sum_{j\in S} p_j$.*
   (b) *For each S sort $\mathbf{p} \in \mathcal{Z}$ by total cost. Associate a counter for each price vector in this sorted array.*
   (c) *For each consumer i, consider the sorted array associated with S_i and increment the counter on the price vector in this array that has the highest total cost that is at most v_i.*
   (d) *The seller's profit for S and $\mathbf{p}$ is the sum of the counters for price vectors with total cost at least that of $\mathbf{p}$ on S.*

   Analysis: $O(n\log M + 2^m M \log M)$

2. *For each* $\mathbf{p} \in \mathcal{Z}$*, sum profits from all bundles. Analysis:* $O(2^m M)$
3. *Output optimal* $\mathbf{p} \in \mathcal{Z}$*.*

The total runtime of the unlimited supply pricing algorithm is $O((n+2^m M)\log M)$. For $m = O(1)$, the running time is $O\left(\left(n + \left(\frac{1}{\varepsilon}\log\frac{n}{\varepsilon}\right)^m\right)\log\left(\frac{1}{\varepsilon}\log\frac{n}{\varepsilon}\right)\right)$. This concludes the proof of Theorem 2.

4 Limited Supply, Unit-Demand Consumers

We now consider the limited supply case of the unit-demand pricing problem. In this case the seller can only offer a limited number of units of each of the m items available for sale. For limited supply pricing problems not all pricings lead to well defined outcomes. For example, an item might be priced so low that the demand for the item exceeds the supply. To avoid this problem, and to make our pricing algorithms useful in auction design problems, we restrict our attention to computing prices that are *envy-free*. We review envy-freedom as well as some concepts from Economics literature below.

4.1 Notation and Background

Index the items $1, \ldots, m$ and let J represent the multiset of items the seller is offering. There are n consumers each of whom wishes to purchase at most one item. Define a consumer's valuation for an item as the value assigned by the consumer to obtaining one unit of the item. For consumer i and item j, let v_{ij} denote this valuation and V the matrix of consumer valuations. Given a price p_j for item j, consumer i's utility for this item (at its price) is $u_{ij} = v_{ij} - p_j$. Given a pricing of all items, $\mathbf{p} = (p_1, \ldots, p_m)$, consumer i will choose one of the items that maximize their utility, i.e., an item j such that $v_{ij} - p_j \geq v_{ij'} - p_{j'}$ for all $j' \neq j$, or no item if $v_{ij} < p_j$ for all j. We say that a consumer is *happy* with an item they would choose in this way. Under a particular pricing, we refer to the set of items that a consumer is happy with as the *demand set*. A consumer's utility under a pricing, denoted u_i for consumer i, is the utility they would obtain for any item in this set.

A pricing $\mathbf{p}$ is *envy-free* if there is an assignment of items to consumers such that (a) no item is oversold, and (b) all consumers with positive utility under the pricing obtain an item in their demand set. Given the demand sets, computing such an assignment or determining that none exists is a simple bipartite matching problem.

The pricing problem we address is that of computing, when given the consumer valuations, the envy-free pricing that is seller optimal. This is the price vector that maximizes the seller's profit (i.e., the sum of the prices of all the units of items sold) subject to the condition that the pricing is envy-free.

Envy-free pricing for unit-demand problems is closely related to *Walrasian equilibria* [6]. When restricted to unit-demand problems Walrasian equilibria are precisely those pricings that are both envy-free and satisfy the additional condition that unsold items have price zero. Let $\mathrm{MWM}_V(J)$ denote the value of the maximum weighted matching on the bipartite graph induced by the consumers' valuations V on item multiset J. Gul and Stacchetti [4] give the following algorithm for computing the Walrasian equilibrium that gives the seller the maximum profit.

Definition 3 (Maximum Walrasian Prices Algorithm, MaxWalEq**).** *Sell item j at price $p_j = \mathrm{MWM}_V(J) - \mathrm{MWM}_V(J \setminus \{j\})$.*

It is easy to see that MaxWalEq can be computed in time $O(m^3n^3)$, since the initial maximum weighted matching solution $\mathrm{MWM}_V(J)$ can be computed in time $O(m^3n^3)$ and then $\mathrm{MWM}_V(J \setminus \{j\})$ can be computed for each $j \in \{1, \ldots, m\}$ that is matched in time $O(m^2n^2)$.

The difference between the seller optimal Walresian equilibrium (as computed by MaxWalEq) and the seller optimal envy-free prices is that in the former all unsold items must have price zero. This constrant results in a total seller profit for MaxWalEq that in general is lower than that obtained in the optimal envy-free pricing. In the special case that the optimal envy-free pricing happens to sell all available items then it gives a Walrasian equilibrum (since all unsold items have price zero). Thus, under the assumption that all items are sold in the optimal envy-free pricing, MaxWalEq computes this optimal pricing. We use this fact below to search for the optimal subset of items to sell, $J' \subseteq J$, using MaxWalEq to obtain prices for each subset.

4.2 The Algorithm

The following algorithm gives an exact optimal envy free pricing for the limited supply case. It runs in cubic time assuming that the total number of units for sale, m', is constant.

Definition 4 (Limited Supply Pricing Algorithm). *For each subset J' of the J items for sale, compute* MaxWalEq(J')*. Output the prices that give the highest profit.*

In the case that the number of units m' for sale is constant, then we make $2^{m'}$ calls to MaxWalEq giving a total runtime of $O(m^3n^32^{m'})$, which is cubic for constant m'.

We next give give a simple algorithm for computing approximately optimal within a factor of $(1+\varepsilon)$ envy-free prices when the number of distinct items for sale is a constant (but with an arbitrary number of units of each item). This algorithm runs in time $O(m^3n^3 \log_{1+\varepsilon}^m n)$. First a lemma.

Lemma 5. *Consider multisets J and J', such that $J' \subset J$. For item $j \in J'$, let p_j (resp., p'_j) be the price for j in* MaxWalEq(J) *(resp., in* MaxWalEq(J')*). Then $p'_j \geq p_j$.*

Proof. Given $J' \subset J$, we are arguing that $\mathrm{MWM}_V(J) - \mathrm{MWM}_V(J \setminus \{j\}) \leq \mathrm{MWM}_V(J') - \mathrm{MWM}_V(J' \setminus \{j\})$ for all $j \in J'$. We note in passing that this is equivalent to showing that $\mathrm{MWM}_V(\cdot)$ is a submodular function. We rearrange the statement to show that $\mathrm{MWM}_V(J) + \mathrm{MWM}_V(J' \setminus \{j\}) \leq \mathrm{MWM}_V(J') + \mathrm{MWM}_V(J \setminus \{j\})$. Letting $A = J \setminus \{j\}$ and $B = J'$ we have $A \cap B = J' \setminus \{j\}$ and $A \cup B = J$ making our goal to prove that $\mathrm{MWM}_V(A \cap B) + \mathrm{MWM}_V(A \cup B) \leq \mathrm{MWM}_V(A) + \mathrm{MWM}_V(B)$, the familiar definition of submodularity. We will show this for arbitrary sets A and B. We proceed by showing that given the matched edges of $\mathrm{MWM}_V(A \cap B)$ and $\mathrm{MWM}_V(A \cup B)$ we can assign them to either matchings of A or B. Of course $\mathrm{MWM}_V(A)$ will be at least the sum of the weights of edges assigned to the matching of A and $\mathrm{MWM}_V(B)$ at least that for B proving the result.

Consider putting the matchings $\mathrm{MWM}_V(A \cap B)$ and $\mathrm{MWM}_V(A \cup B)$ together; we get single edges, double edges, cycles, and paths. We now show how to use all the edges that make up these components to construct a matching of A and a matching of B. Single edges incident on $A \setminus B$ go in the matching of A, single edges incident on $B \setminus A$ go in the matching of B, and double edges incident on $A \cap B$ go one to each of A and B. A cycle can only have edges incident upon $\mathrm{MWM}_V(A \cap B)$; we assign its odd edges to the matching of A and even edges to the matching of B (say). As for paths, note that a path has every other edge from $\mathrm{MWM}_V(A \cap B)$ and $\mathrm{MWM}_V(A \cup B)$. Thus a path cannot have both endpoints at vertices representing items in $A \otimes B$ as it would then have even length requiring one of the endpoints to have an incident edge from $\mathrm{MWM}_V(A \cap B)$, a contradiction. Thus we can simply assign every other edge in each path to the matching of A or B making sure that if the path has an end point in $A \setminus B$ (resp., $B \setminus A$) then the incident edge on this vertex is assigned to the matching of A (resp., B). □

Definition 5 (General Approximate Pricing Algorithm, GAPA**).** *For each subset J' of the J items for sale with the property that the multiplicity of each distinct item is either 0 or a power of $(1+\varepsilon)$ (rounded down), compute* MaxWalEq(J')*. Output the prices that give the highest profit.*

We need only consider having at most n copies of each item. There are at most $\lfloor \log_{1+\varepsilon} n \rfloor + 1$ multiplicities as described in the definition. Considering all possible combinations of these multiplicities across the distinct items gives at most $(1 + \log_{1+\varepsilon} n)^m$ calls to MaxWalEq. This gives a total runtime of $O(n^3 \log^m_{1+\varepsilon} n)$, which is near-cubic for a constant m.

We now show that this algorithm computes a $(1+\varepsilon)$-approximation to the optimal envy-free pricing. Let J^* be the set of items sold by the optimal envy-free pricing. Let n_j be the number of items of type j sold by J^* and let p_j be their prices. The optimal envy-free profit is thus $\sum_j n_j p_j$. Round the multiplicities of J^* down to the nearest power of $(1+\varepsilon)$ (or to zero) and note that GAPA gets at least the profit of MaxWalEq on this smaller set of items. Let p'_j and n'_j be the resulting prices and number of items sold. We have $n'_j \geq n_j/(1+\varepsilon)$ and by Lemma 5, $p'_j > p_j$. Thus the profit of GAPA is at least $\sum_j n'_j p'_j \geq \sum_j n_j p_j/(1+\varepsilon) = \mathrm{OPT}/(1+\varepsilon)$.

5 Conclusion

In this paper we have shown that it is possible to approximate the optimal pricing in near-linear or near-cubic time for several natural pricing problems. One of the techniques we utilize is based on forming a small grid of prices that is guaranteed to contain an approximate optimum. Another technique is to guess how many of each item are sold in the optimal solution and use the seller optimal Walrasian equilibrium prices for this set of items.

It is our hope that these ideas can be extended to address the unlimited supply problem for general combinatorial consumers, described in [5]. There are several difficulties in proceeding in this direction. First, we have to show that a price vector in some smaller space, such as $(\{0\} \cup [\frac{\delta h}{n}, h])^m$ is close to optimal. This is not trivial, since when the items are complements, e.g., in the single-minded case, it is natural to round prices in $(0, \frac{\delta h}{n})$ down to zero; whereas when the items are substitutes, as in the unit-demand case, it is natural to round these prices up to $\frac{\delta h}{n}$. Second, a small grid has to be generated such that for any point in our reduced price space we can find a grid point that is almost as good. For single-minded consumers this task is simplified by the fact that each consumer only wants one particular product bundle, and rounding item prices down cannot cause them to switch to a drastically different set of products. For unit-demand consumers it is more complicated, and a locally linear grid has to be generated so that when we round down to a grid point we give up more for higher priced items, thus ensuring that a consumer only switches to a higher priced item. General combinatorial consumers, however, may want to switch from one set to a completely different set of items. These two sets may have different cardinality and consist of differently priced items. Further elaboration of our ideas seems necessary to address these challenges.

Another problem left for future work is the design of efficient algorithms for the limited supply case when the consumers are single-minded.

References

1. G. Aggarwal, T. Feder, R. Motwani, and A. Zhu. Algorithms for multi-product pricing. In *Proc. of ICALP*, 2004.
2. B. Chazelle and J. Friedman. Point location among hyperplanes and unidirectional ray-shooting. *Comput. Geom. Theory Appl.*, 4:53–62, 1994.
3. A. Goldberg and J. Hartline. Competitive Auctions for Multiple Digital Goods. In *Proc. 9th European Symposium on Algorithms*, pages 416–427. Springer–Verlag, 2001.
4. F. Gul and E. Stacchetti. Walrasian Equilibrium with Gross Substitutes. *Journal of Economic Theory*, 87:95–124, 1999.
5. V. Guruswami, J. Hartline, A. Karlin, D. Kempe, K. Kenyon, and F. McSherry. On Profit Maximizing Envy-free Pricing. In *Proc. 16th Symp. on Discrete Algorithms*. ACM/SIAM, 2005.
6. L. Walras. *Elements of Pure Economics*. Allen and Unwin, 1954.