Welfare and Profit Maximization with Production Costs

Avrim Blum* Anupam Gupta* Yishay Mansour† Ankit Sharma*

Abstract— Combinatorial Auctions are a central problem in Algorithmic Mechanism Design: pricing and allocating goods to buyers with complex preferences in order to maximize some desired objective (e.g., social welfare, revenue, or profit). The problem has been well-studied in the case of limited supply (one copy of each item), and in the case of digital goods (the seller can produce additional copies at no cost). Yet in the case of resources—oil, labor, computing cycles, etc.—neither of these abstractions is just right: additional supplies of these resources can be found, but at increasing difficulty (marginal cost) as resources are depleted.

In this work, we initiate the study of the algorithmic mechanism design problem of combinatorial pricing under increasing marginal cost. The goal is to sell these goods to buyers with unknown and arbitrary combinatorial valuation functions to maximize either the social welfare, or the seller's profit; specifically we focus on the setting of *posted item prices* with buyers arriving online. We give algorithms that achieve *constant factor* approximations for a class of natural cost functions—linear, low-degree polynomial, logarithmic—and that give logarithmic approximations for more general increasing marginal cost functions (along with a necessary additive loss). We show that these bounds are essentially best possible for these settings.

1. Introduction

Combinatorial Auctions are a central problem in Algorithmic Mechanism Design: pricing and allocating goods to buyers with complex preferences in order to maximize some desired objective (e.g., social welfare, revenue, or profit). This problem is typically studied in one of two extreme cases – the case of *limited supply* (one copy of each item) or the case of *unlimited supply* (the seller can produce additional copies at no cost). For the case of limited supply, there are strong negative results (see [7] and the references therein) unless one makes additional assumptions on the buyers' valuations (e.g., submodularity [12], [15], [11], [10]). In contrast, for unlimited supply, which is characteristic of digital goods, maximizing social welfare is trivial by giving all the items away for free, and for revenue maximization, good bounds can be achieved for general buyers [8], [5]. However, in the case of resources—whether oil or computing cycles or food or attention span—the unlimited-supply case is too optimistic and the limited-supply case too pessimistic. More often than not, additional sources can be found, but at higher cost. Indeed, the classical market equilibrium in economic theory assumes that as prices rise, supply increases while demand decreases (giving a unique price which clears the market). Such a supply curve corresponds to a cost of obtaining goods that increases with the number of items desired.

In this work, we initiate the study of this setting where additional resources can be found, but at *increasing marginal cost*, for the algorithmic mechanism design problem of combinatorial pricing.[1] That is, a seller has n goods, and for each good i there is a non-decreasing marginal cost function $c_i()$, capturing the fact that additional units of this good can be obtained, but at an increasing difficulty to the seller per unit. We specifically focus on the most challenging setting of posted item prices[2] in the face of an unknown series of buyers, with unknown and arbitrary combinatorial valuation functions, who arrive online. That is, the seller (e.g., a supermarket) must assign prices to each of n goods, then a buyer arrives with some arbitrary combinatorial valuation function and purchases the bundle maximizing her own quasilinear utility (valuation minus price). After the buyer has made her purchase, the seller may adjust prices, then the next buyer arrives, and so on. In this setting, the seller *cannot* ask a buyer to submit her utility function, cannot run VCG, and cannot charge an admission fee to enter the store. We consider two natural goals – maximizing social welfare (the sum of buyers' valuations on bundles purchased, minus the costs incurred by the seller for obtaining these items)[3] and maximizing profit (the total amount paid by the buyers, minus the costs incurred by the seller). Our main result is that using appropriate algorithms we can in fact perform nearly as well as in the much easier setting of digital goods for a wide range of cost curves.

A second scenario where our results are applicable is in the context of network routing with congestion. Specifically, we would like to maximize the sum of valuations of routed connections (each user has some pair of terminals and a

*Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213. This work was supported in part by the National Science Foundation under grants CCF-0830540 and CCF-1101215, as well as by the MSR-CMU Center for Computational Thinking.

†School of Computer Science, Tel Aviv University. This research was supported in part by the Google Inter-university center for Electronic Markets and Auctions, by a grant from the Israel Science Foundation, by a grant from United States-Israel Binational Science Foundation (BSF), and by a grant from the Israeli Ministry of Science (MoS).

[1] One could also study decreasing marginal costs, though we point out that modeling decreasing marginal costs in a non-Bayesian adversarial setting poses difficulties. For example, there are situations where any algorithm can make positive profit only by initially going into deficit, at which point the adversary could send in no more buyers.

[2] By virtue of posted pricing, our mechanisms are inherently incentive compatible.

[3] Social welfare is a natural objective if we view the seller as a resource allocator within a company, and buyers as various units in the company needing resources.

valuation on receiving a connection) minus the congestion cost of routing them. The congestion cost can reflect either the energy required to support the traffic on the network or the cost of additional infrastructure needed to maintain the quality of service under increased load. [1] indicate that energy curves for processors exhibit dis-economies of scale i.e. energy expenditure is super-linear as a function of processor speed. Such a scenario is captured by our model of increasing marginal cost, which for network routing would mean increasing marginal congestion costs.

We will sometimes refer to marginal cost to the seller of the k^{th} copy of an item as the production cost of that item.

1.1. Our Results and Techniques

We focus on two goals: maximizing social welfare, and maximizing profit. Social welfare is the total valuation of the buyers for their bundles purchased, minus the cost to the seller of all items sold. That is, it is the total utility of all players including the seller. Note that because the production costs are not flat, even to maximize social welfare, one cannot simply sell items at their production costs; one must sell at a price higher than the production cost[4]. This is in order to ensure that items reach the buyers who want them (approximately) most. The second goal is to maximize profit, i.e., the sum of prices charged for items sold minus their costs to the seller.

For a wide range of reasonable cost functions (linear, low-degree polynomial, logarithmic), we present a pricing algorithm that achieves a social welfare within a *constant* factor of the optimal social welfare allocation minus a necessary additive loss. This holds for buyers with arbitrary combinatorial valuation functions. Furthermore, the algorithm is quite 'natural' and reasonable: we price the k^{th} copy of any good at the production cost of the $2k^{th}$ copy[5]. This algorithm, that we call *twice-the-index*, appears in Section 3.

We then consider general increasing cost functions, on which twice-the-index may not give good guarantees. For these, we present the *smoothing algorithm* in Section 4 that gives a logarithmic approximation (again with a necessary additive loss) for maximizing social welfare, generalizing results of [6] that hold in the limited-supply case. The main technique here is to find a "smooth" price curve that does not make sharp jumps, but that tries to stay within a constant factor of the cost function. For the case of *convex* cost functions, the result yields a logarithmic approximation to the social welfare, as long as the production cost for the first logarithmically many copies of all the items is small compared to the optimum welfare. (This result is essentially the best possible, with both logarithmic losses unavoidable.)

Our deterministic online algorithms for welfare-maximization can be converted into randomized algorithms for *profit* maximization, at a potential further loss of a logarithmic factor in the approximation. This conversion is an extension of a result of [4] for the case of single-minded buyers; the details of this conversion are given in the full version.

1.2. Related work

There is a huge body of literature on combinatorial auctions and pricing algorithms: we refer the reader to [7], [13] and the references therein—in particular, note [6], [15], [12], [11], [9], [14]. The setting of combinatorial auctions has been considered both in Bayesian (stochastic),; there is a large body of work in the Bayesian (stochastic) setting, where the buyers' valuations are assumed to come from a known prior distribution, and non-Bayesian (adversarial) settings. Our work focuses on the non-Bayesian or adversarial setting.

The algorithms of [9] give truthful mechanisms that achieve constant approximations to social welfare for $\Omega(\log n)$ copies of each item (see also [2]) in the *offline* setting. For the *online* setting, [6] give posted-price welfare-maximizing algorithms for combinatorial auctions in the limited supply setting—the approximation guarantees they give are logarithmic (when there are $\Omega(\log n)$ copies of each item) or worse (when there are fewer copies); their results are (nearly) tight for the online limited-supply setting. The smoothing algorithm presented in Section 4 generalizes the results of [6] to more general increasing costs and not just 0-∞ costs (i.e. the limited supply case).

The work of [4] shows how to convert deterministic (or some special kind of randomized) online mechanisms for allocation problems into (randomized) posted-pricing schemes that achieve $(\rho + \log V_{max})$-fraction of the optimal profit possible, where the online algorithm is ρ-competitive for the allocation problem and V_{max} is the maximum valuation of any agent over the set of items. We extend their analysis to convert our social welfare maximizing algorithms to profit maximizing algorithms. The details of this conversion are given in the full version.

2. Model, Notation, and Definitions

We consider the following setting. A seller is selling a set $\mathcal{I} = \{1, \ldots, n\}$ of n items to a sequence $\mathcal{B}$ of m buyers who arrive one at a time. The seller can obtain (or produce) additional copies of each item but at increasing (or at least non-decreasing) production cost; specifically, let $c_i(k)$ denote the production cost to the seller for the kth copy of item i. For each item i, let $C_i(k)$ be the cumulative cost for the first k copies—i.e., $C_i(k) = \sum_{k' \leq k} c_i(k')$. Let $c_i^{\mathsf{inv}}(p)$ be the number of copies of item i available before the production cost exceeds p; in case $c_i(\cdot)$ is invertible, it follows that $c_i^{\mathsf{inv}}(p) = c_i^{-1}(p)$.

Before each buyer arrives, the seller may mark up the costs to determine a sales price π_i for each item i. Every

[4]See Appendix A.2.1 for an example.

[5]For illustrative examples showing why some closely related algorithms *fail*, see Appendix A.1.

buyer b has some (unknown to the seller) valuation function $v_b : 2^{\mathcal{I}} \to \mathbb{R}$ over possible bundles of items, and purchases the utility-maximizing bundle for herself at the current prices. That is, buyer b purchases the set S maximizing $v_b(S) - \sum_{i \in S} \pi_i$. After a buyer finishes purchasing her desired set, the seller may then readjust prices, and then the next buyer arrives, and so on.

For any particular sequence of buyers, let opt be the allocation that maximizes the social welfare. Clearly, the social welfare achieved under opt, denoted by $W(\mathsf{opt})$, is an upper bound on both the maximum social welfare and maximum profit achievable by any online algorithm.

For any algorithm alg, $W(\mathsf{alg})$ shall denote the social welfare attained through the algorithm. The algorithm shall determined a pricing scheme for the seller and $\pi_i(k)$ shall denote the sales price charged for the k^{th} copy of item $i \in \mathcal{I}$. While this could in principle depend on other items sold, for all our algorithms it will depend only on k and the cost-curve for the item. x_i shall denote the total number of copies of item i sold by the algorithm, and P_i^f shall denote the price of the first *unsold* copy of item i—i.e., $P_i^f = \pi_i(x_i + 1)$.

We shall denote the *total production cost* suffered by the algorithm by $C(\mathsf{alg})$ and and the *revenue* made by $P(\mathsf{alg})$. profit_i shall denote the *profit* made by the algorithm from the sales of item i. The total profit made by the algorithm is $\sum_{i \in \mathcal{I}} \mathsf{profit}_i = P(\mathsf{alg}) - C(\mathsf{alg})$.

Since x_i are the total number of copies sold by the algorithm alg for item i, therefore, $C(\mathsf{alg}) = \sum_{i \in \mathcal{I}} \sum_{k=1}^{x_i} c_i(k)$, $P(\mathsf{alg}) = \sum_{i \in \mathcal{I}} \sum_{k=1}^{x_i} \pi_i(k)$ and $\mathsf{profit}_i = \sum_{k=1}^{x_i} \pi_i(k) - \sum_{k=1}^{x_i} c_i(k)$.

The total valuation of buyers on their allocated bundles under alg is denoted by $V(\mathsf{alg}) = \sum_{b \in \mathcal{B}} v_b(\mathsf{alg}(b))$ where $\mathsf{alg}(b)$ denotes the set of items bought by buyer b from the algorithm alg. The *social welfare* made by the algorithm $W(\mathsf{alg})$ is $V(\mathsf{alg}) - C(\mathsf{alg})$.

For opt, the welfare-maximizing allocation, λ_i denotes the number of copies of item i allocated in opt. $C(\mathsf{opt})$, $V(\mathsf{opt})$ and $W(\mathsf{opt})$ are defined analogously.

2.1. A Structural Lemma

A basic challenge for maximizing social welfare in the presence of increasing production costs is that if one charges too little, then items may be purchased by an initial sequence of buyers whose valuations are too low to generate much social welfare, until the production cost has jumped to a point where only very costly items remain that are out of reach of the subsequent high valuation buyers. On the other hand, if one charges too much, then one loses the opportunity to make certain sales. This problem is compounded by the fact that buyers may have very different combinatorial preferences—one does not want to "run out" of cheap copies of one item for buyers who may have high valuation on large sets containing that item. In the following sections, we describe two pricing algorithms for addressing these issues

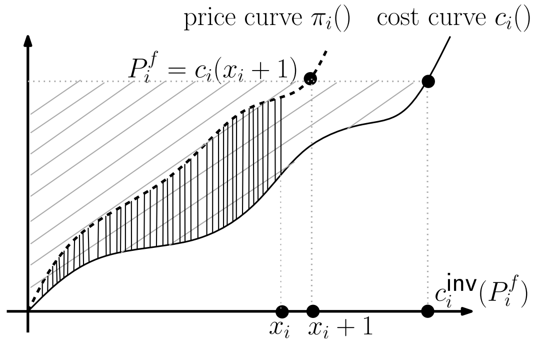

Figure 1. Structural Lemma: if the lightly shaded area is bounded by a small multiple of the doubly shaded area, then we get good social welfare. x_i is the last sold copy of the item and $x_i + 1$ is the first unsold copy. The lower continuous curve is the cost curve while the upper dashed curve is the price curve.

and achieving good social welfare guarantees. In order to analyze the pricing algorithms, we first prove a key structural lemma regarding pricing under increasing production costs; this lemma will be used for all our subsequent analyses.

Lemma 2.1. *For a pricing algorithm alg with non-decreasing price functions π_i suppose there exists some $\alpha \geq 1$ and $\beta \geq 0$ such that for every allowed set of values of the final prices P_i^f,*

$$\sum_{i \in \mathcal{I}} \sum_{k=1}^{c_i^{\mathsf{inv}}(P_i^f)} (P_i^f - c_i(k)) \leq \alpha \sum_{i \in \mathcal{I}} \mathit{profit}_i + \beta \ , \quad (1)$$

then on every instance of buyers

$$W(\mathit{alg}) \geq \frac{1}{\alpha}(W(\mathit{opt}) - \beta) \ .$$

The term $\sum_{k=1}^{c_i^{\mathsf{inv}}(P_i^f)} (P_i^f - c_i(k))$ denotes the maximum possible social welfare which can be achieved by buyers who have valuation P_i^f for item i and zero for everything else. To see this note that (i) $P_i^f - c_i(k)$ is the contribution to social welfare if the k^{th} of item i is allocated to such a buyer and, (ii) the contribution $P_i^f - c_i(k)$ remains non-negative as long as $P_i^f \geq c_i(k)$ which is true only for $k \leq c_i^{\mathsf{inv}}(P_i^f)$. The theorem says that if for every possible set of final prices, we can bound such a social welfare summed over items by the profit generated by the algorithm, then for every sequence of buyers the algorithm gets a good social welfare compared to the optimum.

Graphically, as shown in Figure 1, $\sum_{k=1}^{c_i^{\mathsf{inv}}(P_i^f)} (P_i^f - c_i(k))$ is the area between the production curve $c_i()$ and the dotted line parallel to x-axis, marked by $P_i^f = c_i(x_i + 1)$, (the lightly shaded area) while profit_i is the region between the price curve and production curve (the doubly shaded area).

Proof of Lemma 2.1 : When buyer $b \in \mathcal{B}$ arrives, let $x_i^{(b)}$ be the number of copies of item i sold before b comes in. Hence, the price b sees for item i would be $\pi_i(x_i^{(b)} + 1)$; for brevity we denote this $q_b(i)$, and for a set $S \subseteq \mathcal{I}$, $q_b(S) := \sum_{i \in S} q_b(i)$. The utility of a set S for buyer b therefore is $v_b(S) - q_b(S)$. Since each buyer buys the set that maximizes her utility, hence in particular it implies that the set $\mathsf{alg}(b)$ which buyer b bought from alg must be giving her at least as much utility as the set opt allocated to her i.e.

$$v_b(S_b) - q_b(S_b) \geq v_b(S_b^*) - q_b(S_b^*) \ .$$

Summing over all buyers, we get

$$\textstyle\sum_{b\in\mathcal{B}}(v_b(S_b) - q_b(S_b)) \geq \sum_{b\in\mathcal{B}}(v_b(S_b^*) - q_b(S_b^*)) \ .$$

Adding and subtracting $C(\mathsf{alg})$ and $C(\mathsf{opt})$ on the left hand and right hand sides respectively, we get

$$\begin{aligned}&\big(\textstyle\sum_{b\in\mathcal{B}} v_b(S_b) - C(\mathsf{alg})\big) - \big(\sum_{b\in\mathcal{B}} q_b(S_b) - C(\mathsf{alg})\big)\\ \geq&\big(\textstyle\sum_{b\in\mathcal{B}} v_b(S_b^*) - C(\mathsf{opt})\big) - \big(\sum_{b\in\mathcal{B}} q_b(S_b^*) - C(\mathsf{opt})\big) \ .\end{aligned}$$

Identifying the term $\sum_{b\in\mathcal{B}} v_b(S_b) - C(\mathsf{alg})$ with $W(\mathsf{alg})$, the term $\sum_{b\in\mathcal{B}} q_b(S_b) - C(\mathsf{alg})$ with $\sum_{i\in\mathcal{I}}\mathsf{profit}_i$ and the term $\sum_{b\in\mathcal{B}} v_b(S_b^*) - C(\mathsf{opt})$ with $W(\mathsf{opt})$ we get

$$W(\mathsf{alg}) - \sum_{i\in\mathcal{I}}\mathsf{profit}_i \geq W(\mathsf{opt}) - \big(\sum_{b\in\mathcal{B}} q_b(S_b^*) - C(\mathsf{opt})\big) \ . \tag{2}$$

Since prices are non-decreasing, hence the price faced by any buyer cannot be more than the final price of the various items. Therefore for each buyer b, $q_b(S_b^*) = \sum_{i\in S_b^*}\pi_i(x_i^{(b)}+1) \leq \sum_{i\in S_b^*}\pi_i(x_i+1) = \sum_{i\in S_b^*} P_i^f$. Hence, the term $\sum_{b\in\mathcal{B}} q_b(S_b^*)$ is at most $\sum_{b\in\mathcal{B}}\sum_{i\in\mathsf{opt}(b)} P_i^f = \sum_{i\in\mathcal{I}}(P_i^f\cdot\lambda_i)$ where recall that λ_i denotes the number of copies of item i allocated under opt. Moreover, since $C(\mathsf{opt}) = \sum_{i\in\mathcal{I}}\sum_{k=1}^{\lambda_i} c_i(k)$, we have

$$\begin{aligned}\sum_{b\in\mathcal{B}} q_b(S_b^*) - C(\mathsf{opt}) &\leq \textstyle\sum_{i\in\mathcal{I}}(P_i^f\cdot\lambda_i) - \sum_{i\in\mathcal{I}}\sum_{k=1}^{\lambda_i} c_i(k)\\ &= \textstyle\sum_{i\in\mathcal{I}}\sum_{k=1}^{\lambda_i}(P_i^f - c_i(k)) \ .\end{aligned} \tag{3}$$

The quantity $(P_i^f - c_i(k))$ is non-negative until $c_i(k) \leq P_i^f$, that is it is non negative for $k \leq c_i^{\mathsf{inv}}(P_i^f)$. Hence, we have $\sum_{b\in\mathcal{B}} q_b(S_b^*) - C(\mathsf{opt}) \leq \sum_{i\in\mathcal{I}}\sum_{k=1}^{\lambda_i}(P_i^f - c_i(k)) \leq \sum_{i\in\mathcal{I}}\sum_{k=1}^{c_i^{\mathsf{inv}}(P_i^f)}(P_i^f - c_i(k))$. Therefore using Equation (2) we get

$$\begin{aligned}W(\mathsf{alg}) - \sum_{i\in\mathcal{I}}\mathsf{profit}_i &\geq W(\mathsf{opt}) - \big(\sum_{b\in\mathcal{B}} q_b(S_b^*) - C(\mathsf{opt})\big)\\ &\geq W(\mathsf{opt}) - \textstyle\sum_{i\in\mathcal{I}}\sum_{k=1}^{c_i^{\mathsf{inv}}(P_i^f)}(P_i^f - c_i(k)) \ .\end{aligned}$$

If $\sum_{i\in\mathcal{I}}\sum_{k=1}^{c_i^{\mathsf{inv}}(P_i^f)}(P_i^f - c_i(k)) \leq \alpha\sum_{i\in\mathcal{I}}\mathsf{profit}_i + \beta$, then using above equation we get

$$\begin{aligned}W(\mathsf{alg}) - \textstyle\sum_{i\in\mathcal{I}}\mathsf{profit}_i \geq W(\mathsf{opt}) - (\alpha\sum_{i\in\mathcal{I}}\mathsf{profit}_i + \beta) \Rightarrow\\ W(\mathsf{alg}) + (\alpha-1)\sum_{i\in\mathcal{I}}\mathsf{profit}_i \geq W(\mathsf{opt}) - \beta \ .\end{aligned}$$

Finally using the social welfare generated by the algorithm is at least the profit made, i.e. $W(\mathsf{alg}) \geq \sum_{i\in\mathcal{I}}\mathsf{profit}_i$, we get the desired result $W(\mathsf{alg}) \geq (W(\mathsf{opt}) - \beta)/\alpha$ □

In Section A.1 we present a variant of the structural lemma that will be useful for the analysis of the smoothing algorithm presented in Section 4.2. In the following section, we give pricing strategies that satisfy Lemma 2.1 (or its variant) for suitable α, β.

3. Algorithm: Pricing at twice the index

The first two ideas for pricing items with production costs are perhaps to (a) sell at cost, or (b) sell at some constant times the cost; however, these schemes fail even for simple cost functions like linear and logarithmic production costs, respectively (Appendix A.1). In this section, we consider the next natural pricing scheme: *The price $\pi_i(k)$ of the k^{th} copy of an item is the production cost of the $(2k)^{th}$ copy.* I.e.,

$$\pi_i(k) := c_i(2k).$$

There is nothing special about pricing at *twice* the index, other factors would work as well, just giving slightly different bounds. We shall analyze this algorithm for function classes including polynomial $c_i(x) = x^d$ and logarithmic $c_i(x) = \ln(1+x)$. Since these functions are strictly increasing and hence invertible, hence we shall have $c_i^{\mathsf{inv}}(c_i(x)) = x$ for all $x \geq 0$. To analyze this algorithm, we shall use the result of Lemma 2.1.

Define $A_i(x_i) := \sum_{k=1}^{c_i^{\mathsf{inv}}(P_i^f)}(P_i^f - c_i(k))$. To apply Lemma 2.1, we will show that $\forall x_i \geq 0, A_i(x_i) \leq \alpha\cdot\mathsf{profit}_i(x_i) + \beta_i$ and thereby get $\sum_{i\in\mathcal{I}} A_i(x_i) \leq \alpha\sum_{i\in\mathcal{I}}\mathsf{profit}_i(x_i) + \beta$ where $\beta = \sum_{i\in\mathcal{I}}\beta_i$.

Since the price $\pi_i(k)$ of the k^{th} copy is $c_i(2\,k)$, hence the profit made from the sales of such of a copy is $c_i(2\,k) - c_i(k)$. Further, since x_i copies of item i have been sold, therefore, $P_i^f = c_i(2\,(x_i+1))$ and hence $c^{\mathsf{inv}}(P_i^f) = 2\,x_i + 1$. Therefore, when pricing at twice the index, we have $A_i(x_i) = \sum_{k=1}^{2(x_i+1)}(c_i(2(x_i+1)) - c_i(k))$ and $\mathsf{profit}_i(x_i) = \sum_{k=1}^{x_i}(c_i(2k) - c_i(k))$.

3.1. *Performance on some cost functions*

We now show that for some "well-behaved" classes of functions, we get $A_i(x) \leq \alpha\cdot\mathsf{profit}_i(x) + \beta_i$; the β_i term will usually depend on the production cost of the first few copies of the items—hence we will guarantee that we get a multiplicative α-fraction of the welfare if we ignore the production cost of the first few copies.

- Linear production costs: $c_i(x) = a_i x + b_i$ for some constant $a_i, b_i \geq 0$, then we have $A_i(x) = a_i(x+1)(2x+1)$, and $\mathsf{profit}_i(x) = \frac{1}{2}a_i x(x+1)$, and hence $A_i(x) \leq 6\,\mathsf{profit}_i + a_i$. Lemma 2.1 implies that

  $$\begin{aligned}W(\mathsf{alg}) &\geq \tfrac{1}{6}\big(W(\mathsf{opt}) - \textstyle\sum_{i\in\mathcal{I}} a_i\big)\\ &= \tfrac{1}{6}\big(W(\mathsf{opt}) - \textstyle\sum_{i\in\mathcal{I}}(c_i(2) - c_i(1))\big) \ .\end{aligned}$$

  This result, with suitably modified guarantees, can easily be extended to the case where the actual production cost lies between *two linear curves* whose slopes are within a constant factor of each other.

- Polynomial production costs: $c_i(x) = a_i x^d$ for $d > 1$. Then $A_i(x) \leq a_i \frac{d}{d+1}(2(x+1))^{d+1}$, whereas $\mathsf{profit}_i(x) \geq a_i \frac{1}{d+1}(2^d - 1)\, x^{d+1}$, so some algebra implies that $A_i(x) \leq 12\, d\, \mathsf{profit}_i(x) + 2^{d+1}(d+2)^{d+1} a_i$. Hence

  $$W(\mathsf{alg}) \geq \tfrac{1}{12\, d}\left(W(\mathsf{opt}) - 2\,(d+2)^{d+1} \textstyle\sum_{i \in \mathcal{I}} c_i(2)\right) \ .$$

  Such a bound also holds for $c_i(x)$ being a polynomial of degree at most d with positive coefficients. The additive loss of $2^{O(d\ \log(d))}$ should be compared to the lower bound of $\Omega(2^d/d)$ in Corollary A.3
- Logarithmic production costs: $c_i(x) = \ln(1+x)$. By algebra, $A_i(x) \leq (2x+3)$, and $\mathsf{profit}_i(x) \geq \ln(\frac{3}{2})\, x$, so $A_i(x) \leq \frac{2}{\ln(3/2)} \mathsf{profit}_i(x) + 3$, and Lemma 2.1 implies

  $$W(\mathsf{alg}) \geq \tfrac{\ln(3/2)}{2}\left(W(\mathsf{opt}) - 3|\mathcal{I}|\right) \ .$$

In the full version of the paper, we show that in the guarantees given above, gains in the multiplicative factor can be made while trading-off commensurate losses in the additive loss terms. Further, in Appendix A.4, we show that without any other information (like an estimate on the optimum welfare), no deterministic algorithm can give a purely multiplicative loss. Finally, in Appendix A.3, we show that twice-the-index fails for some increasing cost functions—e.g., when the production cost is zero for B items and ∞ thereafter. Hence, in the next section, we give an algorithm for increasing production costs.

4. General Increasing Cost Functions

In this section, we present an algorithm that applies to general increasing cost functions, giving a logarithmic approximation minus an additive term that depends on the cost function (Theorem 4.4)[6]. For the case of *convex* cost functions, the analysis allows us to give a more explicit form for the additive loss (Corollary 4.6). In fact for the case of convex production costs, we get a multiplicative logarithmic approximation to $W(\mathsf{opt})$ as long as the production cost of the first few logarithmic copies of all the items is small compared to $W(\mathsf{opt})$. For the limited-supply setting, if we have $\Omega(\log nm)$ copies of each item, the additive loss is zero and the algorithm gets a logarithmic fraction of the optimal social welfare like in [6].

4.1. Intuition

Ideally, we would like to set prices which are sufficiently far above the cost curve (so that we generate a large social welfare), yet not be too far above it (else the high prices may result in no sales, causing a large additive loss). Hence, we run into problems when the cost curve increases sharply—and the intuitive goal is to create a price curve which smooths out these sharp changes in the cost curve while staying "close" to it.

The smoothing algorithm takes the cost curve, and creates a price function which is a monotone step function: copies of the item are grouped into intervals, with all copies in an interval having the same price. We call these intervals "price intervals". The algorithm creates the price curve from *right to left*. If we think of ℓ_i as the effective number of copies of item i and Z as the highest price, then the ℓ_i^{th} copy is priced first at price Z through creation of the price interval $[\lfloor \frac{2}{3}\, \ell_i \rfloor, \infty)$[7], with items in this interval priced at Z; subsequently, price intervals are created progressively moving leftwards until we have priced the first copy. At each point, we use the intuition from above: if the price is much higher than the cost, we set the price for the new interval such that the price-cost gap is slashed by a factor of 2, else we set the price to maintain a sufficient gap from the cost.

4.2. The smoothing algorithm

Before we give the algorithm (in Figure 2), let us give some definitions; we urge the impatient reader to jump to Section 4.3 to get a quick rough feel of the algorithm. We assume that the cost of the first copy of every item is 0 i.e. $\forall i, c_i(1) = 0$[8]. Define U_{max}, the maximum welfare any single buyer can achieve:

$$U_{max}(\mathcal{I}, \mathcal{B}) = \max\nolimits_{b \in \mathcal{B}} \max\nolimits_{T \subseteq \mathcal{I}} \left(v_b(T) - \textstyle\sum_{i \in T} c_i(1)\right), \tag{4}$$

which equals $\max_{b \in \mathcal{B}} \max_{T \subseteq \mathcal{I}}\ v_b(T)$ because $c_i(1) = 0$. Note that the optimal social welfare, $W(\mathsf{opt})$, lies between U_{max} and $m \cdot U_{max}$. The smoothing algorithm requires a parameter Z which satisfies $Z \in (U_{max}, U_{max}/\epsilon]$. [9]

Define $\ell_i = \min\{c_i^{\mathsf{inv}}(Z), m\}$ and $B_i = \lceil 12 \log(4n\ell_i/\epsilon) \rceil$. At a high level, think of ℓ_i as being the "effective number" of copies of item i available, and B_i as the "number of different price levels" we create in our price curve. Define $c_i^{\mathsf{inv}t}(p)$ to be the "truncated" value $\min\{c_i^{\mathsf{inv}}(p), c_i^{\mathsf{inv}}(Z), m\}$, it is the maximum number of copies of item i that opt can allocate before the production cost exceeds p (Corollary A.1). Note that using $c_i^{\mathsf{inv}t}$ (as opposed to using c_i^{inv}) is a technicality; one can imagine $c_i^{\mathsf{inv}t} \approx c_i^{\mathsf{inv}}$ for a first read. Define $\mathsf{width}_i(p) := \lfloor \frac{c_i^{\mathsf{inv}t}(p)}{B_i} \rfloor$; this function will determine the number of copies we group together in a price interval. We assume that

$$\ell_i \geq B_i \geq 12; \tag{5}$$

see the full version for why this is without loss of generality.

[6]An alternative *bi-criteria* guarantee can be achieved through a simple discretization of the cost function that allows us to reduce to the case of step functions and use the algorithm of [6]. The resulting pricing scheme gives a logarithmic approximation against the optimum on a higher cost curve. Details are in the full version.

[7]We abuse notation slightly by denoting the integer interval $\{r, r+1, \ldots, s-1\}$ as the half-open real interval $[r, s)$.

[8] In the paper's full version we show that this is without loss of generality.

[9]We can remove this assumption at a further loss of $O(\log W(\mathsf{opt})\, (\log\log W(\mathsf{opt}))^2)$ in the approximation guarantee [5].

```
1:  for all $x \geq \lfloor \frac{2}{3}\,\ell_i \rfloor$, **set** $\pi_i(x) := Z$
2:  **set** $x \leftarrow \lfloor \frac{2}{3}\,\ell_i \rfloor$
3:  **while** $x > 1$ **do**
4:    **if** $\mathsf{width}_i(\pi_i(x)) \geq 1$ **then**
5:      **set** $x' \leftarrow \max\{x - \mathsf{width}_i(\pi_i(x)), 1\}$
6:      **set** $\Delta = \begin{cases} \frac{\pi_i(x) - c_i(x)}{2} & \text{if } \pi_i(x) \geq 3\,c_i(x) \\ \frac{c_i(x)}{2} & \text{otherwise} \end{cases}$
7:      for all $y \in [x', x)$ , **set** $\pi_i(y) := c_i(x) + \Delta$
8:      **set** $x \leftarrow x'$
9:    **else**
10:     for all $y \in [1, x)$, **set** $\pi_i(y) := \pi_i(x)$
11:     **set** $x \leftarrow 1$
```

Figure 2. Smoothing algorithm

Let $\pi_i : \mathbb{Z}_+ \rightarrow \mathbb{R}_+$ be the price function and let $\mathcal{J}_i$ denote the set of price intervals for item i, and with $z_i = |\mathcal{J}_i|$. We refer to the q^{th} interval of item i as J_{iq}, with J_{i1} being the price interval that contains the first copy of item i, and $J_{iz_i} = [\lfloor \frac{2}{3}\,\ell_i \rfloor, \infty)$. Let $\pi_i(J_{iq})$ be the price of the copies in the interval J_{iq}. Depending on the production curve, two consecutive price intervals may have the same price. Also, we will formally state later that the prices we generate are non-decreasing, and always stay above the production cost for all copies less than ℓ_i.

4.3. The main ideas

Smoothing: Step 6 ensures a smooth price curve: if the price is more than thrice the production cost, we slash the gap between the price and production cost by two else we allow the price to stay at a sufficient gap from the cost.

Price Interval Size : The idea of the analysis is to show that whenever the number of copies sold moves from a lower price interval to a higher one, the social welfare generated by selling copies at the lower price is enough to be competitive against opt, even if we sell no further copies at the higher price. Consequently, the size of a price interval J_{iq} must depend on the price of items in the next interval $J_{i\,q+1}$. It turns out that to get a multiplicative approximation factor of $O(B_i)$, if the price of copies in $J_{i\,q+1}$ were P, it suffices to set the width of $J_{i\,q}$ to be $\lfloor \frac{c_i^{\mathsf{invt}}(P)}{B_i} \rfloor = \mathsf{width}_i(P)$. Here is a simple special case that illustrates why: suppose only item i was being sold and we sold all copies from J_{iq} but no copies from interval $J_{i\,q+1}$. We would like to apply Lemma 4.4. The final price P_i^f in that case is $P = \pi_i(J_{i\,q+1})$. Staring at the left hand side of Equation (1), we see that it is at most $P \cdot c_i^{\mathsf{invt}}(P)$. Since we sold all the copies in price interval J_{iq}, we sold at least $|J_{iq}| = \lfloor \frac{c_i^{\mathsf{invt}}(P)}{B_i} \rfloor$ many copies, each at profit at least $P/6$ (something we will prove later). Hence on the right hand side of Equation (1), the term profit_i is at least $P \cdot \lfloor \frac{c_i^{\mathsf{invt}}(P)}{B_i} \rfloor / 6$. Putting $\alpha = O(B_i)$ we satisfy Equation (1) and thereby get an $O(B_i)$ approximation. Since the width of J_{iq} depends on the price of $J_{i\,q+1}$, it is natural that our pricing algorithm creates price intervals from *right to left*.

Termination: The algorithm terminates in one of two ways: either while creating such appropriately sized price intervals, we hit the first copy (i.e., $x' \leftarrow 1$ in Step 5, and then the loop condition fails in Step 3) or the price p of some price interval is low enough that $p < c_i(B_i)$, which implies $c_i^{\mathsf{invt}}(p) < B_i$ (the proof of implication appears later) and therefore $\mathsf{width}_i(p) = \lfloor \frac{c_i^{\mathsf{invt}}(p)}{B_i} \rfloor < 1$: this causes $x \leftarrow 1$ in Step 11. In the latter case, the price has become low enough that we can simply group all remaining copies into the lowest priced interval J_{i1} at price p. The subsequent analysis will often have to separately consider these two cases: whether $x \leftarrow 1$ is achieved in Step 5 or in Step 11.

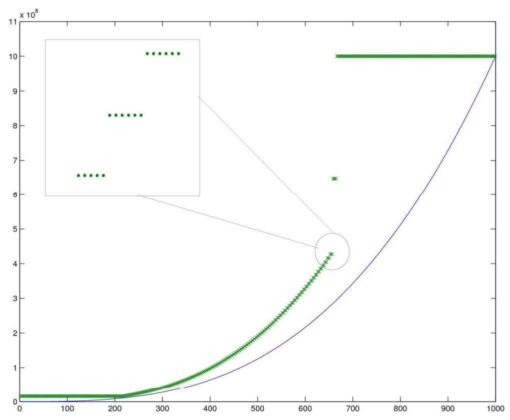

Figure 3. The figure shows the pricing curve drawn by the smoothing algorithm for the production curve $c_i(x) = x^3$. The lower line is the production curve. The upper thicker line is the pricing curve. We can observe that the price curve is flat towards the extreme right; this flat region contains the right-most price interval. Towards the extreme left the price curve *appears* to be a smooth curve. The inset shows the individual price intervals.

4.4. The Analysis

Let us call an interval $J_{iq} = [r, s)$ to be *full-sized* if its width equals $\mathsf{width}_i(\pi_i(s))$. Note that the right-most interval J_{iz_i} is not full sized since it semi-infinite. Further, the left-most interval J_{i1} *may not* be full-sized either because the algorithm ran out of copies, or the price became too low so that all remaining unpriced copies were bunched together. We first show that if we sell at least $|J_{i1}| + |J_{i2}|$ copies of item i, i.e., we have sold at least one full-sized interval, we get a good approximation factor for the reasons we discussed in Section 4.3. This is proved in Lemma 4.2.

Then we consider the case when the number of items sold is less than $|J_{i1}| + |J_{i2}|$: in this case we cannot show a good multiplicative loss. Instead, we show that the price of items in the first two intervals is small in this case, which bounds the additive loss. This is proved in Lemma 4.5. Finally, our main result Theorem 4.4 follows from these two lemmas.

All the intervals except the leftmost J_{i1} and rightmost J_{iz_i} ones are created in a similar fashion; intervals J_{i1} and J_{iz_i} have to treated as special cases at several points in the analysis. Also, the analysis which follows from this point onwards up till (and not including) Theorem 4.4 is *per item*. Hence the subscript i in the terms involved is irrelevant from the point of analysis and is present only to maintain uniformity in presentation.

To begin, we state some useful properties of the prices and widths of the intervals (proof is in the full version).

Lemma 4.1 (Prices and Widths)**.** *The following facts about interval prices hold for the intervals in $\mathcal{J}_i$ for any non-decreasing cost curve:*

a. For any $J_{iq} = [r, s)$ such that $q \neq z_i$, $\pi_i(J_{iq}) \geq \frac{3}{2}\, c_i(s)$. Hence, $\pi_i(x) \geq \frac{3}{2}\, c_i(x)$ for $x \in J_{iq}$.

a'. If the cost curve is convex, $\pi_i(\lfloor \frac{2}{3}\,\ell_i \rfloor) \geq \frac{3}{2}\, c_i(\lfloor \frac{2}{3}\,\ell_i \rfloor)$.

b. For consecutive J_{iq} and $J_{i\,q+1}$ and $q \neq z_i - 1$, we have $\pi_i(J_{iq}) \leq \pi_i(J_{i\,q+1}) \leq 2\,\pi_i(J_{iq})$. If the cost curve is convex the claim also holds for $q = z_i - 1$.

c. All price intervals $J_{iq} = [r, s)$ ($q \notin \{1, z_i\}$) have $|J_{iq}| = \mathsf{width}_i(\pi_i(s)) = \mathsf{width}_i(\pi_i(J_{i\,q+1}))$.

Lemma 4.1(a) states that the price of any copy is sufficiently far from the production cost of that copy. Lemma 4.1(a') states the same claim about the left end of the right-most interval J_{iz_i} in case the cost curve is in addition convex. Lemma 4.1(b) states the price of copies in the interval $J_{i\,q+1}$ is higher than that of $J_{i\,q}$, but not too far from it. Lemma 4.1(d) states that all price intervals except possibly the left-most and the right-most are full-sized. Armed with these facts, we first show that if "many" copies of item i are sold, then we are in good shape. The other case where "few" copies are sold, is dealt with subsequently.

4.4.1. The Case of Many Copies.: Suppose we sell *all* copies in some interval J_{iq} for $q > 1$: then we get that the profit made from that interval alone gives us a good approximation.

Lemma 4.2. *If the number of sold copies x_i of item i is at least $|J_{i1}| + |J_{i2}|$, then $P_i^f \cdot c_i^{\mathsf{invt}}(P_i^f) \leq 12 B_i \cdot \mathsf{profit}_i$, where $\mathsf{profit}_i := \sum_{k=1}^{x_i} (\pi_i(k) - c_i(k))$.*

Proof: Let q be the largest integer such that $J_{iq} = [r, s)$ is completely sold out; hence $q \in [2, z_i)$. The final price is $P_i^f = \pi_i(J_{i\,q+1}) = \pi_i(s)$. We want to show we make a reasonable profit from the sales of copies in J_{iq}. From Lemma 4.1(c), there are $\mathsf{width}_i(\pi_i(s))$ many copies in J_{iq}. For each of these copies $k \in [r, s)$, the profit is $\pi_i(k) - c_i(k) \geq \pi_i(k) - c_i(s)$, because costs are non-decreasing.

However, by Step 7 of the pricing algorithm, for all $k \in J_{iq}$, $\pi_i(k) = c_i(s) + \Delta$, where Δ is determined by Step 6.

- Either $\pi_i(s) \geq 3c_i(s)$, $\Delta = \frac{1}{2}\,(\pi_i(s) - c_i(s)) \geq \frac{1}{3}\,\pi_i(s)$,
- Or $\pi_i(s) < 3c_i(s)$, $\Delta = c_i(s)/2 > \frac{1}{6}\,\pi_i(s)$.

So, we make a profit of at least $\pi_i(s)/6$ from each of the $\mathsf{width}_i(\pi_i(s)) = \lfloor \frac{c_i^{\mathsf{invt}}(\pi_i(s))}{B_i} \rfloor$ many copies in J_{iq}:

$$\mathsf{profit}_i \geq \frac{\pi_i(s)}{6} \cdot \lfloor \frac{c_i^{\mathsf{invt}}(\pi_i(s))}{B_i} \rfloor \geq \frac{\pi_i(s) \cdot c_i^{\mathsf{invt}}(\pi_i(s))}{12\, B_i},$$

where the last inequality is because $\lfloor t \rfloor \geq t/2$ for $t \geq 1$. Plugging in $P_i^f = \pi_i(s)$ completes the proof. ■

4.4.2. The Case of Few Copies: Now suppose item i is such that the number of copies we sell either lies within the left-most interval J_{i1}, or only covers a small fraction of the second interval J_{i2}: the argument given above does not hold in that case. However we can show the following result.

Lemma 4.3. *If the number of sold copies x_i of item i is less than $|J_{i1}| + |J_{i2}|$ then $\pi_i(P_i^f) \cdot c_i^{\mathsf{invt}}(P_i^f) \leq \pi_i(J_{i2}) \cdot c_i^{\mathsf{invt}}(\pi_i(J_{i2}))$.*

Proof: Since we end up selling less than $|J_{i1}| + |J_{i2}|$ copies, hence the final price P_i^f is at most $\max\{\pi_i(J_{i1}), \pi_i(J_{i2})\}$ which is $\pi_i(J_{i2})$ since Lemma 4.1(b) tell us that $\pi_i(J_{i1}) \leq \pi_i(J_{i2})$. Hence, $P_i^f \cdot c_i^{\mathsf{invt}}(P_i^f) \leq \pi_i(J_{i2}) \cdot c_i^{\mathsf{invt}}(\pi_i(J_{i2}))$ ($c_i^{\mathsf{invt}}(p)$ is non-decreasing function of p). ■

4.4.3. Finishing the Analysis: Lemma 4.2 and Lemma 4.3 together give us the main result of this section.

Theorem 4.4. *The social welfare $W(\mathsf{alg})$ achieved by the smoothing algorithm on a non-decreasing cost curve given an estimate $Z \in (U_{max}, U_{max}/\epsilon]$ satisfies*

$$W(\mathsf{alg}) \geq \frac{W(\mathsf{opt}) - \sum_{i \in \mathcal{I}} \pi_i(J_{i2}) \cdot c_i^{\mathsf{invt}}(\pi_i(J_{i2}))}{12\, \max_{i \in \mathcal{I}} B_i},$$

where $B_i := \lceil 12 \log(4n\ell_i/\epsilon) \rceil$, and $\ell_i := \min\{c_i^{\mathsf{inv}}(Z), m\}$.

Proof: For each item i, depending on the number of copies sold, either Lemma 4.2 or Lemma 4.3 applies, which implies that for each $i \in \mathcal{I}$,

$$P_i^f \cdot c_i^{\mathsf{invt}}(P_i^f) \leq 12\, B_i \cdot \mathsf{profit}_i + \pi_i(J_{i2}) \cdot c_i^{\mathsf{invt}}(\pi_i(J_{i2})).$$

Summing over all items i, we get

$$\sum_{i \in \mathcal{I}} P_i^f \cdot c_i^{\mathsf{invt}}(P_i^f) \leq 12 \max_{i \in \mathcal{I}} B_i \cdot \mathsf{profit}_i + \sum_{i \in \mathcal{I}} \pi_i(J_{i2}) \cdot c_i^{\mathsf{invt}}(\pi_i(J_{i2})).$$

Applying Corollary A.1, we get

$$W(\mathsf{alg}) \geq \frac{W(\mathsf{opt}) - \sum_{i \in \mathcal{I}} \pi_i(J_{i2}) \cdot c_i^{\mathsf{invt}}(\pi_i(J_{i2}))}{12 \max_{i \in \mathcal{I}} B_i},$$

which completes the proof. ■

4.5. Convex cost curves

The analysis in this section holds only for convex cost curves. The crucial lemma which we will prove in this section is:

Lemma 4.5. *For a convex cost curve, $\pi_i(J_{i2}) \cdot c^{\mathsf{invt}}(\pi_i(J_{i2})) \leq \max\{B_i\, c_i(B_i), \frac{\epsilon\, Z}{2\, n}\}$.*

which will suffice to prove the following result.

Corollary 4.6. *The social welfare $W(\mathsf{alg})$ achieved by the smoothing algorithm on a non-decreasing convex cost curve given an estimate $Z \in (U_{max}, U_{max}/\epsilon]$ satisfies*

$$W(\mathsf{alg}) \geq \frac{W(\mathsf{opt})/2 - \sum_{i\in\mathcal{I}} B_i \cdot c_i(B_i)}{12\ \max_{i\in\mathcal{I}} B_i},$$

where $B_i := \lceil 12\log(4n\ell_i/\epsilon)\rceil$, and $\ell_i := \min\{c_i^{\mathsf{inv}}(Z), m\}$.

Proof: Putting Theorem 4.4 and Lemma 4.5 together,

$$W(\mathsf{alg}) \geq \frac{W(\mathsf{opt}) - \epsilon\, Z/2 - \sum_{i\in\mathcal{I}} B_i \cdot c_i(B_i)}{12\ \max_{i\in\mathcal{I}} B_i}.$$

Using $\epsilon Z \leq U_{max} \leq W(\mathsf{opt})$, we get the desired result. ■

We now need to prove Lemma 4.5. The pricing algorithm terminates when it has priced all the copies, i.e. x is set to 1 and the `if` condition in Step 3 becomes false. x can be set to 1 either in Step 8 (preceded by x' being set to 1 in Step 5) or in Step 11. We consider these two cases separately.

- Algorithm terminates through Step 11: Lemma 4.8 proves that $c_i^{\mathsf{invt}}(\pi_i(J_{i2})) \cdot \pi_i(J_{i2}) < B_i \cdot c_i(B_i)$.
- Algorithm terminates through Step 5: Lemma 4.12 proves that $c_i^{\mathsf{invt}}(\pi_i(J_{i2})) \cdot \pi_i(J_{i2}) < \frac{\epsilon Z}{2n}$.

Proof of Lemma 4.5 : The algorithm terminates either through Step 5 or Step 11 and Lemma 4.12 and Lemma 4.8 together indicate that $\pi_i(J_{i2}) \cdot c^{\mathsf{invt}}(\pi_i(J_{i2})) \leq \max\{B_i\, c_i(B_i), \frac{\epsilon Z}{2n}\}$. □

Before proving Lemma 4.8 and Lemma 4.12, we state and prove the following lemma that characterizes the circumstances under which the algorithm terminates in either condition.

Lemma 4.7. *The pricing algorithm terminates through Step 11 if and only if $\pi_i(J_{i2}) < c_i(B_i)$.*

Proof: Let $J_{i2} = [s, r)$. We first prove that if $\pi_i(J_{i2}) < c_i(B_i)$, then the algorithm terminates in Step 11. If $\pi_i(s) = \pi_i(J_{i2}) < c_i(B_i)$, then it implies that $c_i^{\mathsf{inv}}(\pi_i(s)) < B_i$, and by definition of $c^{\mathsf{invt}}()$, $c_i^{\mathsf{invt}}(\pi_i(s)) < B_i$ which implies that $\mathsf{width}_i(\pi_i(s)) = \lfloor \frac{c_i^{\mathsf{inv}}(\pi_i(s))}{B_i} \rfloor = 0$. Hence, right after creation of J_{i2}, when the algorithm checks for the `if` condition on point s in Step 4, it shall evaluate to false and therefore, the algorithm shall terminate through Step 11.

To prove the other direction, if the algorithm terminates through Step 11, then it must be the case that the `if` condition in Step 4 evaluated to false for some x. Further, x must be the left-end point of J_{i2}. This is because once the `if` condition evaluates to false, the algorithm jumps to Step 11 and creates a single price interval containing all copies that have not been priced yet and it includes the first copy and hence, this price interval must be J_{i1}. So x must be the left-end point of the price interval just after J_{i1}, i.e. J_{i2}.

Now, $\mathsf{width}_i(\pi_i(x))$= $\mathsf{width}_i(\pi_i(J_{i2}))$= $\lfloor \frac{c_i^{\mathsf{invt}}(\pi_i(J_{i2}))}{B_i} \rfloor < 1$ implies that $\frac{c_i^{\mathsf{invt}}\pi_i(J_{i2}))}{B_i} < 1$ and so $c_i^{\mathsf{invt}}(\pi_i(J_{i2})) < B_i$. By definition of $c_i^{\mathsf{invt}}()$, this implies that $\min\{c_i^{\mathsf{inv}}(\pi_i(J_{i2}), \ell_i\} < B_i$. Since by Equation (5), $\ell_i \geq B_i$, it must be the case $c_i^{\mathsf{inv}}(\pi_i(J_{i2})) < B_i$, which by definition of $c_i^{\mathsf{inv}}()$ can occur only if $\pi_i(J_{i2}) < c_i(B_i)$. ■

We now prove Lemma 4.8 and Lemma 4.12 that treat the two conditions under which the algorithm can terminate.

Algorithm terminates through Step 11: The proof that price of J_{i2} is small follows almost immediately in this case.

Lemma 4.8. *If the algorithm terminated through Step 11 then $\pi_i(J_{i2}) \cdot c_i^{\mathsf{invt}}(\pi_i(J_{i2})) < c_i(B_i)\, B_i$.*

Proof: If the algorithm terminated through Step 11, then Lemma 4.7 implies that $\pi_i(J_{i2}) < c_i(B_i)$. By definition of $c_i^{\mathsf{invt}}()$, this implies that $c_i^{\mathsf{invt}}(\pi_i(J_{i2})) < B_i$ and hence we get the result. ■

Algorithm terminates through Step 5: We will prove that price of the interval J_{i2} is 'small' by showing that relative to the price of right-most interval J_{iz_i}, the prices for the subsequently created intervals on its left, have been slashed sufficiently often. For item i, label a copy x *close* if $\pi_i(x) < 3\, c_i(x)$, else label it as *far*. Depending on which of r and s are close or far, mark a price interval $J_{iq} = [r, s)$ as one of $\{(C, C), (F, C), (C, F), (F, F)\}$. Note that the right-most interval J_{iz_i} is not marked since it is semi-infinite. The following lemma indicates that in case prices are 'far' from the production cost, the algorithm slashes the prices exponentially.

Lemma 4.9. *If a contiguous sequence of price intervals $J_{iq}, J_{i\,q+1}, \cdots, J_{i\,q+t-1}$ are all marked (F, F) and $J_{i\,q+t}$ is marked (F, C), then $\pi_i(J_{iq}) \leq (\frac{2}{3})^t\, \pi_i(J_{i\,q+t})$.*

Lemma 4.10 states that if we ever have a price interval that is marked (F, C), there are 'many' price intervals to the left of that interval. Lemma 4.11 states that there are 'many' intervals to the left of the right-most interval J_{iz_i}.

Lemma 4.10. *Consider an interval $J_{iq} = [r, s)$ with $q \neq z_i$ that is marked (F, C). If the algorithm terminated through Step 5, then there are at least $B_i/4$ intervals $J_{iq'}$ with $q' < q$. In particular, J_{iq} cannot be the first price interval i.e. $q \neq 1$.*

Lemma 4.11. *If the algorithm terminated through Step 5, then there are at least $B_i/3$ intervals J_{iq} with $q < z_i$.*

Lemma 4.12. *If the algorithm terminated through Step 5 then $\pi_i(J_{i2}) \cdot c_i^{\mathsf{invt}}(\pi_i(J_{i2})) < \frac{\epsilon Z}{2n}$.*

Proof: The interval J_{i1} can be marked either (F, C) or (F, F), since $c_i(1) = 0$ while $\pi_i(1) > 0$ (Observation A.5). By Lemma 4.10, J_{i1} cannot be marked (F, C). Hence, the only case left is when J_{i1} is marked (F, F). Let q be the smallest value, if one exists, such that J_{iq} is marked (F, C); note that $q > B_i/4$ by Lemma 4.10, and in particular $q > 2$. If no such (F, C) interval exists, set $q \leftarrow z_i$.

By definition of J_{iq}, all intervals between J_{i1} and J_{iq} are marked (F, F). Depending on whether $q \neq z_i$ or $q = z_i$, Lemma 4.10 or Lemma 4.11 respectively imply there are at least $B_i/4$ of these intervals. By Lemma 4.9, $\pi_i(J_{i1}) \leq (\frac{2}{3})^{B_i/4}\, \pi_i(J_{iq}) \leq \frac{\pi_i(J_{iq})}{4n\ell_i/\epsilon}$, since $B_i = \lceil 12 \log(4n\ell_i/\epsilon) \rceil$.

Moreover, by Lemma 4.1(b), $\pi_i(J_{i2}) \leq 2 \cdot \pi_i(J_{i1}) \leq \frac{2\pi_i(J_{iq})}{4n\ell_i/\epsilon}$. By definition of $c^{\mathsf{invt}}()$, $c^{\mathsf{invt}}(\pi_i(J_{i2})) \leq \ell_i$; this gives $\pi_i(J_{i2}) \cdot c^{\mathsf{invt}}(\pi_i(J_{i2})) \leq \frac{2\pi_i(J_{iq})}{4n\ell_i/\epsilon} \cdot \ell_i \leq \frac{\epsilon Z}{2n}$. ■

The smoothing algorithm can give purely multiplicative guarantees as long as the cost of the first $O(\log n)$ copies of the items is small compared to $W(\mathsf{opt})$. As an example, suppose the cost functions are $c_i(k) = 0$ for $k \leq d \log n$, and $c_i(k) = \infty$ for $k > d \log n$ for some constant d. Then $\ell_i \leq d \log n$, and $B_i = O(\log n/\epsilon)$. So for d large enough constant, $B_i \cdot c(B_i) = 0$, and we get an $O(\log n)$ approximation to the social welfare, as in [6]. (This is best possible for online algorithms [3].)

5. Profit Maximization

In this section we show how to combine an online algorithm for social welfare maximization in the presence of increasing costs (such as those in previous sections) with an algorithm for a single-buyer profit maximization (such as the algorithm in [5]) to yield an algorithm with strong profit guarantees for any sequence of buyers under increasing costs. This result builds on work of [4] for the case of single-minded buyers. Specifically, suppose we are given access to two algorithms:

- a deterministic social-welfare maximizing algorithm A, which given production cost curves $\{c_i\}_{i \in \mathcal{I}}$, outputs pricing schemes $\{\pi_i(\cdot)\}_{i \in \mathcal{I}}$ such that on any sequence σ of buyers, gives the guarantee
$$\rho \cdot W(A(\sigma)) + \beta \geq W(\mathsf{opt}(\sigma)) \qquad (6)$$
- and, a randomized single-buyer profit maximization algorithm B, which outputs a non-negative price vector τ for items $i \in \mathcal{I}$ and gives the guarantee that for any buyer b, with valuation $v_b()$,
$$\mu \cdot \mathbb{E}_\tau[\textstyle\sum_{i \in S_\tau} \tau_i] + \kappa \geq \max_{s \subseteq \mathcal{I}} v_b(s) \qquad (7)$$
where S_τ is the set of items bought by the buyer b when the price vector τ is presented and so $\sum_{i \in S_\tau} \tau_i$ is the resultant profit generated from buyer b.

Note that $\max_{s \subseteq \mathcal{I}} v_b(s)$ is an upper bound on maximum profit that can be generated from buyer b. Further algorithm B operates in a world with zero production costs, and may take as input a parameter T such that $\max_{S \subseteq \mathcal{I}} v_b(S) < T$; the parameters μ, κ may be functions of T.[10] We further assume that for every item $i \in \mathcal{I}$ and $k \in \mathbb{N}$, the price $\pi_i(k)$ set by algorithm A is at least as much as the production cost of that copy of the item. The main result of this section is

[10] [5] give such a single-buyer profit maximization algorithm, a slight variant of which has $\kappa = T/(2mn)$ and $\mu = O(\log(mn))$. The algorithm picks a uniform price on a geometric scale for all items. It can be combined with either of the social welfare maximizing algorithms in this paper to give a $O(\log(mn))$-profit maximizing algorithm+additive loss.

Theorem 5.1. *Given a (ρ, β)-social welfare maximization algorithm A (satisfying Equation (6)) and a (μ, κ)-single buyer profit maximization algorithm B (satisfying Equation (7)), we can construct a randomized profit-maximizing algorithm C whose expected profit over any sequence σ of buyers is at least $(W(\mathsf{opt}(\sigma)) - O(\beta + \kappa \cdot |\sigma|))/O(\rho + \mu)$.*

Proof sketch: Algorithm C maintains a copy of algorithm A running in the background and keeps updating A's state with the sets buyers are buying. For each buyer j, with probability 1/2, algorithm C presents the price vector as specified by the current state of A and with probability 1/2, adds a random price vector, generated using B, to the price vector specified by A. The analysis, building on work of [4], is deferred to the full version.

References

[1] M. Andrews, S. Antonakopoulos, and L. Zhang, "Minimum-cost network design with (dis)economies of scale," in *51st FOCS*, 2010.

[2] A. Archer, C. Papadimitriou, K. Talwar, and É. Tardos, "An approximate truthful mechanism for combinatorial auctions with single parameter agents," *Internet Math.*, vol. 1, no. 2, 2004.

[3] B. Awerbuch, Y. Azar, and S. Plotkin, "Throughput-competitive on-line routing," in *FOCS*, 1993.

[4] B. Awerbuch, Y. Azar, and A. Meyerson, "Reducing truth-telling online mechanisms to online optimization," in *35th STOC*, 2003.

[5] M.-F. Balcan, A. Blum, and Y. Mansour, "Item pricing for revenue maximization," in *9th EC*, 2008.

[6] Y. Bartal, R. Gonen, and N. Nisan, "Incentive compatible multi unit combinatorial auctions," in *9th TARK*, 2003.

[7] L. Blumrosen and N. Nisan, "Combinatorial auctions," in *Algorithmic Game Theory*. Cambridge University Press, 2007.

[8] P. Briest, M. Hoefer, and P. Krysta., "Stackelberg network pricing games," in *25th STACS*, 2008.

[9] P. Briest, P. Krysta, and B. Vöcking, "Approximation techniques for utilitarian mechanism design," in *STOC*, 2005.

[10] T. Chakraborty, Z. Huang, and S. Khanna, "Dynamic and non-uniform pricing strategies for revenue maximization," in *FOCS*, 2009.

[11] S. Dobzinski, "Two randomized mechanisms for combinatorial auctions," in *APPROX/RANDOM*, 2007.

[12] S. Dobzinski, N. Nisan, and M. Schapira, "Approximation algorithms for combinatorial auctions with complement-free bidders," *Math. of OR*, vol. 35, no. 1.

[13] J. Hartline and A. Karlin, "Profit maximization in mechanism design," in *Algorithmic Game Theory*. Cambridge University Press, 2007.

[14] R. Lavi and C. Swamy, "Truthful and near-optimal mechanism design via linear programming," in *46th FOCS*, 2005.

[15] B. Lehmann, D. Lehmann, and N. Nisan, "Combinatorial auctions with decreasing marginal utilities," *Games and Economic Behavior*, 2006.

Appendix

1. *Variant of Structural Lemma*

We now prove a variant of the structural theorem. Define $c_i^{\mathsf{invt}}(p) = \min\{c_i^{\mathsf{inv}}(p), m, c_i^{\mathsf{inv}}(U_{max})\}$ where m is the number of buyers and for a given set of buyers $\mathcal{B}$ and items $\mathcal{I}$, $U_{max} = \max_{b\in\mathcal{B}} \max_{T\subseteq\mathcal{I}} \left(v_b(T) - \sum_{i\in T} c_i(1)\right)$ is the maximum welfare any single buyer can achieve.

Corollary A.1. *For a pricing algorithm* alg *with non-decreasing price functions* π_i *suppose there exists some* $\alpha \geq 1$ *and* $\beta \geq 0$ *such that for every allowed set of values of the final prices* P_i^f*,*

$$\sum_{i\in\mathcal{I}} \sum_{k=1}^{c_i^{\mathsf{invt}}(P_i^f)} (P_i^f - c_i(k)) \leq \alpha \sum_{i\in\mathcal{I}} \mathsf{profit}_i + \beta \ , \quad (8)$$

then on every instance of buyers $W(\mathsf{alg}) \geq \frac{1}{\alpha}(W(\mathsf{opt}) - \beta)$ *.*

Proof Sketch: Note that in the proof of Lemma 2.1 just after Equation (3), we argued that $\lambda_i \leq c_i^{\mathsf{inv}}(P_i^f)$. Instead of summing all the way to $c_i^{\mathsf{inv}}(P_i^f)$, we could stop the summation at $\min\{c_i^{\mathsf{inv}}(P_i^f), m, c_i^{\mathsf{inv}}(U_{max})\}$. Indeed, this is because

- $\lambda_i \leq m$: each buyer wants at most one copy of each item, so at most m copies of item i can be allocated in the optimal solution.
- $\lambda_i \leq c_i^{\mathsf{inv}}(U_{max})$: each copy beyond $c_i^{\mathsf{inv}}(U_{max})$ has cost strictly greater than U_{max}; allocation of any such copy can only decrease the social welfare.

□

2. *Some 'natural' pricing schemes*

We give some natural pricing schemes and instances where they fail to achieve good social welfare.

2.1. Pricing at Cost: While the algorithm of pricing *at cost* (i.e., setting $\pi(k) = c(k)$) gives an optimal welfare for the unlimited supply setting (where production costs are zero), it is not a good algorithm even for "simple" cost curves. E.g., for a single item with linear costs $c(k) = k$, consider a sequence of m buyers with the i^{th} buyer having value i for $i \in \{1, \ldots, m\}$, followed by m buyers with value m each. Pricing at cost will sell to the first m buyers and give zero welfare for them, after which the production cost will be too high to sell any further copies. In contrast, the optimal solution is to sell to the second set of m buyers with welfare $m^2 - \frac{m(m+1)}{2} = \Omega(m^2)$.

2.2. Pricing at Twice the Cost: Another natural algorithm is to price at twice (or any fixed multiple) of the *cost* of each item. However, while this can be shown to perform well for linear and low-degree polynomial cost functions, it performs poorly for the case of logarithmic costs. Indeed, consider a single item with production cost $c(x) = \log x$, and suppose we price the i^{th} item at cost $\pi(i) = 2\log i$. Suppose the first m buyers have valuations $2\log 1, 2\log 2, \ldots, 2\log m$ respectively, and are followed by m^2 buyers with valuation $2\log m = \log m^2$. The algorithm would sell to the first m buyers, getting a social welfare of $\sum_{i=1}^{m}(2\log i - \log i) = O(m\log m)$, after which the cost would be too high for the remaining buyers. In contrast, optimum would sell to the last m^2 buyers, and get a social welfare of $\sum_{i=1}^{m^2}(\log m^2 - \log i) = \Omega(m^2)$.

3. *Pricing at Twice the Index*

Here is an example where twice-the-index algorithm fails to produce good social welfare—e.g., consider the limited supply-like setting where $c(k) = 0$ for $k \leq B$, and $c(k) = V$ for $k > B$. Consider sending in B buyers with valuation zero, followed by B buyers with valuation $V - \varepsilon$. Twice-the-index prices the first $B/2$ copies at zero, and the rest at V, whence we get zero welfare, whereas the optimal welfare of $B(V - \varepsilon)$ is achieved by selling to just the later B buyers.

4. *The Necessity of Additive Loss*

If we do not have an estimates for $W(\mathsf{opt})$, we give a trade-off between the additive and multiplicative loss (even for a single item), for any algorithm where the prices are at least the production cost.

Lemma A.2. *For any deterministic pricing algorithm (in a single item setting) acting on production costs* $c()$ *and that price copies at at least their production cost, to give the guarantee* $W(\mathsf{alg}) \geq (W(\mathsf{opt}) - \Delta)/\alpha$*, it is necessary that* $\alpha \geq \frac{c(2)-c(1)}{\Delta} - 1$.

Proof: Let $\pi(1) = c(1) + \gamma$. Note that $\gamma \leq \Delta$ because otherwise a buyer sent in with valuation $c(1) + \gamma - \varepsilon$ would buy nothing and hence $W(\mathsf{alg}) = 0$ while $W(\mathsf{opt}) = \gamma - \epsilon$ and therefore $W(\mathsf{alg}) \geq (W(\mathsf{opt}) - \Delta)/\alpha$ would be false.

Now consider a sequence of two buyers, the first with valuation $c(1) + \gamma$ and the second with valuation $c(2) - \varepsilon$. The first buyer will buy the first copy. Since the price of second copy is at least $c(2)$, hence the second buyer won't buy. Hence, $W(\mathsf{alg}) = \gamma$ while $W(\mathsf{opt}) = c(2) - c(1) - \varepsilon$. In such a scenario, for the guarantee to hold we require that $\gamma \geq (c(2) - c(1) - \varepsilon - \Delta)/\alpha$ which implies that $\gamma\,\alpha + \Delta \geq c(2) - c(1) - \varepsilon$. Noting that $\gamma \leq \Delta$ and that the inequality needs to hold for any $\varepsilon \geq 0$, the claim follows. ■

Corollary A.3. *For production curves* $c(x) = x^d$*, for* $\alpha = 4\,d$*,* $\Delta = \Omega(2^d/d)$*.*

5. *Some observations and results for Section 4*

Observation A.4. $\mathsf{width}_i(p)$ *is non-decreasing in* p*.*

Observation A.5. *Assuming the parameter* $Z > 0$*, for every copy* x*, the price set by the algorithm,* $\pi_i(x) > 0$*.*

Proposition A.6 (The left-most interval)**.** *The following facts hold for the left-most interval* $J_{i1} = [1, s)$*:*

a. *If the procedure terminated through Step 5, then* $|J_{i1}| \leq \mathsf{width}_i(\pi_i(s)) = \mathsf{width}_i(\pi_i(J_{i\,2}))$.
b. *If the procedure terminated through Step 11, then* $\pi_i(J_{i1}) = \pi_i(J_{i2})$.