Optimal Envy-Free Pricing with Metric Substitutability

Ning Chen*
University of Washington
Seattle, WA
ning@cs.washington.edu

Arpita Ghosh
Yahoo! Research
Santa Clara, CA
arpita@yahoo-inc.com

Sergei Vassilvitskii
Yahoo! Research
New York, NY
sergei@yahoo-inc.com

ABSTRACT

We study the envy-free pricing problem faced by a profit maximizing seller when there is *metric substitutability* among the items — consumer i's value for item j is $v_i - c_{i,j}$, and the substitution costs, $\{c_{i,j}\}$, form a metric. Our model is motivated from the observation that sellers often sell the same product at different prices in different locations, and rational consumers optimize the tradeoff between prices and substitution costs. While the general envy-free pricing problem is hard to approximate, the addition of metric substitutability constraints allows us to solve the problem exactly in polynomial time by reducing it to an instance of weighted independent set on a perfect graph.

When the substitution costs do not form a metric, even in cases when a $(1+\epsilon)$-approximate triangle inequality holds, the problem becomes NP-hard. Our results show that triangle inequality is the exact sharp threshold for the problem of going from "tractable" to "hard".

We then turn our attention to the multi-unit demand case, where consumers request multiple copies of the item. This problem has an interesting paradoxical non-monotonicity: The optimal revenue the seller can extract can actually decrease when consumers' demands increase. We show that in this case the revenue maximization problem becomes APX-hard and give an $O(\log D)$ approximation algorithm, where D is the ratio of the largest to smallest demand. We extend these techniques to the more general case of arbitrary non-decreasing value functions, and give an $O(\log^3 D)$ approximation algorithm.

Categories and Subject Descriptors

F.2 [**Analysis of algorithms and problem complexity**]

General Terms

Theory, Algorithms

*Part of the work was done while visiting Yahoo! Research.

1. INTRODUCTION

It is common to see a seller pricing the same product differently at different locations. The latest iPod costs $299 in the US, but is £199(≈ $392) in the U.K. Similarly, the online retailer Amazon prices some book titles very differently on amazon.com and amazon.co.uk, even though international shipping is possible. Even within the same geographic area, one can often find the same item sold for different prices at different locations. For example, a vendor will frequently set the price of gas differently at different gas stations. In the near future, a consumer interested in buying, say gas, will be able to use her cell phone to instantaneously find out the prices at the different locations and the distance from her current location to the different points of sale. As a result, rational consumers will be able to optimize the following tradeoff — buy the item they need at a nearby location or pay a transportation cost to buy the item at a more distant location. Assuming that consumers are rational in this way, how should a seller price items at different locations to maximize the total revenue? This question is the focus of this paper.

In our model each consumer lies on some node of an edge-weighted graph $G = (V, E)$, and has a value v_i, which is the maximum price she is willing to pay for the product. The weight of an edge (i, j) in the graph is the commute cost from i to j. Then, given the prices at all locations, each consumer will choose the location that maximizes her utility, *i.e.*, the difference between her value for the product and the total expense (price plus commute cost). We are interested in the problem facing a profit-maximizing seller — choose a price for each location to maximize the total revenue, *i.e.*, the sum of prices paid by consumers for the items (note that the commute costs spent by consumers are not collected by the seller).

This problem falls into the setting of the unit-demand envy-free pricing problem, introduced by Guruswami et al. [13]. There is a single seller with m items and there are n consumers with values for each of these items. Given a vector of prices for the items, each consumer buys an item which maximizes her utility. The goal of the seller is to set prices for the items to maximize the total revenue. Guruswami et al. [13] showed that the problem in general is APX-hard, and provided an $O(\log n)$-approximation algorithm. Briest [3] recently showed that given appropriate complexity assumptions, the problem can not be approximated within $O(\log^\epsilon n)$ for some $\epsilon > 0$.

Our problem is a special case of the envy-free pricing problem, where there is *metric substitutability* among the items: A consumer's values for different items are related in terms of the metric space defined by the shortest path distance in G. That is, consumer i's value for the item at location j is $v_i - c_{i,j}$, where $c_{i,j}$ is the shortest path distance between her location and location j in G. In this formulation the distances (*i.e.*, substitutability) between items, $\{c_{i,j}\}$, form a metric space. As we will see below, this simple change drastically changes the complexity of the problem — while the envy-free pricing problem is NP-hard, the problem with metric substitutability can be solved exactly in polynomial time.

1.1 Results and Algorithmic Ideas

Our main result is the following:

THEOREM 1.1. *The profit-maximizing envy-free pricing problem with unit demand and metric substitutability can be solved exactly in polynomial time.*

The tradeoff in revenue maximization arises since consumers are not constrained to buy at their own locations: The seller must decide whether it is more profitable to charge a higher price at a particular location, thereby extracting more revenue from nearby high-valued nodes who do not have a cheaper option, or charge a lower price and make up the difference in volume (since more consumers would now be able to afford the item). What makes the problem complicated in the general setting is that the relationship between prices on different items can be quite complex, and potentially depends on the (arbitrary) preferences of the individual consumers. In our setting, however, the fact that all of the consumers' values are related through a metric space implies a *transitivity* condition — if consumer i prefers (to buy the item at) location j and consumer j prefers location k, then i prefers k as well. This transitivity allows us to construct a directed acyclic graph (DAG) on the set of nodes with the following property: If a node is being charged its full value, full revenue can not be extracted from any of its descendants in this DAG. The DAG structure and transitivity capture the special structure of our problem and are both crucial in deriving the algorithm.

We call a node which pays its full value in a pricing solution a *price-setter*. The DAG defined above says which subsets of nodes can simultaneously be price-setters. The revenue maximization problem then reduces to finding the optimal subset of price-setters. To that end, we transform the DAG into a node-weighted graph H with the property that the optimal revenue is exactly the weight of the maximum weighted independent set in H.

While the independent set problem is usually hard, it can be solved in polynomial time in perfect graphs by the seminal result of Grötschel et al. [12]. We prove, using the structure of the DAG, that H is a perfect graph by showing that H is a Berge graph and applying the Strong Perfect Graph Theorem [8], which states that a graph is perfect if and only if it is a Berge graph. Putting these together yields the desired result.

The fact that the underlying substitution costs, c, satisfy triangle inequality is crucial in obtaining the result above. In fact, in cases when the items do not lie in a metric space, or even when the space satisfies a $(1+\epsilon)$-approximate triangle inequality (*i.e.*, $c_{i,k} \le (1+\epsilon)(c_{i,j} + c_{j,k})$), for any given $\epsilon > 0$, the problem becomes NP-hard (Theorem 6.1). Given appropriate complexity assumptions, the problem is even hard to approximate to a ratio better than $O(\log^{\epsilon} n)$ for some $\epsilon > 0$ (Theorem 6.2). Our result shows that tractability of the envy-free pricing problem depends crucially on the existence of the underlying metric space. This gives an interesting natural instance for which triangle inequality is the exact sharp threshold of going from "tractable" to "hard".

Multi-Unit Demand Setting. We then turn our attention to the multi-unit demand case, where in addition to a value v_i, consumer i also has a demand d_i, *i.e.*, she needs d_i copies of the item (such as multiple gallons of gas). In this setting, when a consumer buys from a different location, she pays the commute cost only once, regardless of her demand. That is, consumer i's value for buying d_i copies of the item at location j is $d_i v_i - c_{i,j}$.

The multi-unit demand problem has a surprising non-monotonicity, which we call the *Demand Paradox*: While one would expect the optimal revenue to increase monotonically with increasing demand, this is not true (Example 7.1). That is, an increase in demand can actually decrease the optimal revenue extractable. This is reminiscent of a similar "Braess-style" paradox observed in selfish routing [14] and in "hiring-a-team" markets [7].

The Demand Paradox is one of the reasons why the multi-unit demand setting is technically quite different from the unit demand setting, and the algorithm established above does not produce an optimum solution. To understand why, note that the solution we are looking for is a per-unit price vector, but the effective per-unit costs for consumers are no longer related through a metric space — each consumer's effective per-unit cost is scaled by her individual demand, which differs across consumers. Indeed, we show that the revenue maximization problem is APX-hard in the multi-unit demand setting, even when all demands are no more than 5 (Theorem 7.1).

We present an $O(\log D)$-approximation algorithm to maximize revenue for this problem, where D is the ratio of the largest to smallest demand (Theorem 7.2). We first show that when all of the demands are identical, the problem has a polynomial time solution. We then bound the loss in the paradox described above — although increasing demand may decrease total revenue, it will decrease it by at most a factor of D. These two facts together yield the desired result.

Finally, we study the more general setting where consumers can have different marginal values for each copy of the item they obtain. Specifically, each consumer now has a non-decreasing value function $f_i(x_i)$, which gives her value as a function of the number of copies x_i of the item she owns. For this more general setting, we give an $O(\log^3 D)$-approximation algorithm to maximize revenue (Theorem 7.3). Our algorithm uses the above $O(\log D)$-approximation algorithm as a subroutine; the proof proceeds by analyzing the effect of the change of values and demands on the total revenue.

1.2 Related Work

Envy-free pricing captures the notion of fairness of equilibrium pricing in economics (for related work, see, e.g., [10, 11] and the references within), and has recently received much attention in computer science [1, 13, 4, 9, 2, 3, 5, 6].

The problem of revenue maximization with envy-free pricing was initiated by Guruswami et al. [13]. In [13], it was shown that the problem of computing the optimal vector of prices, assuming unit demand and no structure on the valuations, is APX-hard. An $O(\log n)$-approximation algorithm was also provided [13]. Briest [3] recently showed that given appropriate complexity assumptions (the hardness of the balanced bipartite independent set problem in constant degree graphs or refuting random 3CNF formulas), the envy-free pricing problem can not be approximated within $O(\log^{\epsilon} n)$ for some $\epsilon > 0$. For the multi-unit demand setting, Briest [3] showed that the problem is hard to approximate within a ratio of $O(n^{\epsilon})$ for some ϵ, unless $NP \subseteq \bigcap_{\epsilon>0} BPTIME(2^{n^{\epsilon}})$.

When consumers desire a fixed subset of items (*i.e.*, consumers are single-minded), a logarithmic approximation algorithm was derived in [13] and an almost tight lower bound was provided by Demaine et al. [9]. A few special cases of single-minded demand, such as the tollbooth problem where consumers desire paths in a graph, were studied in [13, 4]. Balcan and Blum [2] studied the graph vertex pricing problem where each consumer requests the two endpoints of an edge in a given graph and the goal is to set prices on vertices to maximize the total revenue. The graph vertex pricing problem has a similar flavor to our model, for which a 4-approximation algorithm was given in [2].

Other pricing schemes (min-buying, max-buying, or rank-buying, where consumers buy an item with the smallest price, highest price, or highest ranking according to their preference) were studied in [15, 1, 5], where different algorithmic and lower bounds results were given.

2. PRELIMINARIES

Our model is the following. A single seller with unlimited supply sells an item on an underlying undirected, edge-weighted, graph $G = (V, E)$. We assume the weights of edges are positive. For any pair $i, j \in V$, the commute cost $c_{i,j}$ between i and j is the weight of the shortest path from i to j in G. Note that the costs satisfy $c_{i,i} = 0$ for any $i \in V$ and the triangle inequality, *i.e.*, $c_{i,j} \le c_{j,k} + c_{k,i}$ for any $i, j, k \in V$.

At every node $i \in V$, there is a user[1] with value v_i — this is the maximum value that the user at node i, which we abbreviate as user i, is willing to pay for the item. We suppose that each user is only interested in one copy of the item, *i.e.*, has unit demand. User i can either buy an item at i, or at some other location j in the graph, in which case she incurs an additional cost of $c_{i,j}$.

[1]We generalize our results to multiple users at each node in Section 5.

The seller determines a price p_j for each location $j \in V$ (note that p_j is the price *at* location j, and not (necessarily) the price paid by user j). If user i buys an item from location j, we say i is a *winner* and her utility is $v_i - (p_j + c_{i,j})$. In this case, $r_i = p_j$ is the revenue that the seller obtains from user i (note that the commute cost $c_{i,j}$ spent by user i is *not* collected by the seller). If i does not buy from any location, her utility is 0 and correspondingly $r_i = 0$. Given the price vector on all nodes, each user i decides which location to buy the item rationally, *i.e.*, to maximize her utility.

The total revenue to the seller is defined by $\sum_{i\in V} r_i$. The question we study is the following: Given the underlying graph G and value v_i for each user, set a price p_j for each node $j \in V$, such that the total revenue is maximized, given that users' choices of locations are rational.

3. CHARACTERIZATIONS OF THE OPTIMAL PRICING SOLUTION

We now make some fundamental observations about the structure of the problem. Given a price vector p, we say user $i \in V$ *prefers* location j if $p_j + c_{i,j} \le p_k + c_{i,k}$ for any $k \in V$.

PROPOSITION 3.1 (TRANSITIVITY). *Given any price vector p, if user i prefers location j and user j prefers location k, then i also prefers location k.*

PROOF. Since j prefers location k, $p_j \ge p_k + c_{j,k}$. By triangle inequality, we have

$$p_k + c_{i,k} \le p_k + c_{i,j} + c_{j,k} \le p_j + c_{i,j} \le p_i,$$

since i prefers location j; so i also prefers location k. □

Before we consider the optimization problem, we should first fix a rule for breaking ties. Assume user i prefers both location j and k, *i.e.* $p_j + c_{i,j} = p_k + c_{i,k} \le p_\ell + c_{i,\ell}$ for any $\ell \in V$. In this case, which node should i pick to buy the item, j or k? In this work, we assume i chooses a location with the smallest commute cost (if $c_{i,j} = c_{i,k}$, i chooses one arbitrarily). This tie-breaking rule is consistent with our common knowledge that given the same utility, users do not waste more time on commuting.

PROPOSITION 3.2. *Let p be an optimal price vector. Then every winner buys an item from her own location. Therefore, for any winner i, $p_i \le v_i$.*

PROOF. Assume to the contrary that user i buys an item from location j. By the tie-breaking rule we discussed above, we know $p_j + c_{i,j} < p_i$. We define another price vector p' as follows: let $p'_i = p_j + c_{i,j}$ and $p'_k = p_k$ for any $k \ne i$. Given price vector p', for any $k \ne i$, assume user k prefers location ℓ under price vector p. Note that

$$p_j + c_{k,j} \le p_j + c_{k,i} + c_{i,j} < p_i + c_{k,i}$$

which implies $\ell \ne i$. Hence, we have

$$p'_\ell + c_{k,\ell} = p_\ell + c_{k,\ell} \le p_j + c_{k,j} \le p_j + c_{k,i} + c_{i,j} = p'_i + c_{k,i}.$$

Thus, k still prefers ℓ under price vector p', which implies the revenue obtained from k does not decrease with price vector p'. For user i, observe that $p'_i \le p'_k + c_{i,k}$ for any $k \in V$. Therefore, user i buys an item from its own location and contributes a revenue of p'_i, which is larger than its contribution of p_j under price vector p, a contradiction. □

DEFINITION 3.1. *Given an optimal price vector p, we say a node i is a* price-setter *if $p_i = v_i$.*

The notion of price-setters will be central to many of the arguments. We begin by showing that all prices are determined by price-setters.

PROPOSITION 3.3. *Let p be an optimal price vector. If user i is a winner and $p_i < v_i$, there is a price-setter j such that $p_i = p_j + c_{i,j}$. In this case, we say the price p_i is* determined *by p_j.*

PROOF. Assume to the contrary that there is a winner i such that $p_i < v_i$ and no such price-setter exists. We recursively define a subset $S \subseteq V$ as follows: Initially $S = \{i\}$. For any $j \in S$, if there is $k \in V \setminus S$ such that $p_j = p_k + c_{j,k}$, then let $S \leftarrow S \cup \{k\}$.

We argue that for any winner $j \in S$, $p_j < v_j$. By Proposition 3.2, we know $p_j \le v_j$. Thus, it suffices to show the inequality is strict. Assume otherwise, there is such a $j \in S$ and $p_j = v_j$, *i.e.* j is a price-setter. Consider the sequence $i = i_1, i_2, \ldots, i_m = j$ of adding node j into S. By the construction of S, we have

$$\begin{aligned} p_i &= p_{i_1} = p_{i_2} + c_{i_1,i_2} \\ &= p_{i_3} + c_{i_2,i_3} + c_{i_1,i_2} \\ &\vdots \\ &= p_{i_m} + \sum_{j=1}^{m-1} c_{i_j,i_{j+1}} \\ &\ge p_j + c_{i,j} \end{aligned}$$

On the other hand, since i is a winner, by Proposition 3.2, we know $p_i \le p_j + c_{i,j}$. Thus, $p_i = p_j + c_{i,j}$, which contradicts our assumption.

For any winner $j \in S$, we claim that j does not prefer any location in $V \setminus S$. Suppose instead that j prefers $k \in V \setminus S$, which implies $p_j \ge p_k + c_{j,k}$. But since j is a winner, we know $p_j \le p_k + c_{j,k}$, which implies that equality holds. By construction of S, k should be put into S as well, a contradiction.

Let $\epsilon > 0$ be sufficiently small so that $p_j + \epsilon < v_j$ for any winner $j \in S$. We define a new price vector p' by $p'_j = p_j + \epsilon$ for any $j \in S$ and $p'_j = p_j$ otherwise. For any winner $k \in V \setminus S$ under price vector p, since we only increase prices from p to p', k still prefers its own location and pays the same amount under price vector p'. For any winner $j \in S$, by making ϵ sufficiently small, j still does not prefer any nodes not in S. Since the price of all nodes in S is increased by ϵ, we know j still prefers its own location under price vector p'. But now the contribution of j to the revenue is ϵ more than its contribution under price vector p, which contradicts the optimality of p. □

Hence, any optimal solution can be characterized in terms of the set of price-setters in that solution.

4. THE MAIN ALGORITHM

In this section, we describe our algorithm for implicitly computing prices to maximize the seller's revenue, leading to the proof of Theorem 1.1. To begin with, we use the idea of price-setters to construct a directed acyclic graph on the nodes of G with the property that if a node is chosen as a price-setter, all of its descendants are not price-setters. We then construct a new graph H (Section 4.2) from this graph to help choose the optimal subset of price-setters. In Section 4.3, we show that computing the optimal price vector of G is equivalent to computing the maximum weighted independent set of H. Finally we will show that the latter can be solved in polynomial time by proving that H is a perfect graph (Section 4.4).

4.1 Ancestors and Descendants

For each $i \in V$, define

$$D(i) = \{j \in V \mid v_i + c_{j,i} < v_j\}$$

to be the set of "descendants" of i, *i.e.* the set of users which would prefer to buy the item from location i (at a price $p_i = v_i$) rather than at their own location (at a price $p_j = v_j$). We also define

$$A(i) = \{j \in V \mid v_j + c_{i,j} < v_i\} = \{j \in V \mid i \in D(j)\}$$

to be the set of "ancestors" of i. That is, $A(i)$ is the set of locations that user i prefers to buy (at the price equal to the value of those users) rather than at her own location (at a price $p_i = v_i$). Let $n_i = |A(i)|$.

The intuition behind these definitions is characterized by the following simple lemmas.

LEMMA 4.1. *Let p be an optimal price vector. If $i \in V$ is a winner and price-setter, i.e. $p_i = v_i$, then any user $j \in A(i)$ does not win and every user $j \in D(i)$ wins.*

PROOF. For any $j \in A(i)$, we have $v_j + c_{i,j} < v_i$. If j wins, by Proposition 3.2, we know $p_j \le v_j$. Thus, $p_j + c_{i,j} \le v_j + c_{i,j} < v_i = p_i$, and user i would prefer location j rather than its own location, a contradiction.

For any $j \in D(i)$, we have $p_i + c_{j,i} = v_i + c_{j,i} < v_j$. Thus, node j always wants to buy the item (at least from location i with a positive utility of $v_j - (p_i + c_{j,i})$). □

LEMMA 4.2. *For any $i, j, k \in V$, if $j \in A(i)$ and $k \in A(j)$, then $k \in A(i)$.*

PROOF. By the conditions, we have $v_j + c_{i,j} < v_i$ and $v_k + c_{j,k} < v_j$. Thus, $v_k + c_{i,k} \le v_k + c_{i,j} + c_{j,k} < v_j + c_{i,j} < v_i$, which implies that $k \in A(i)$. □

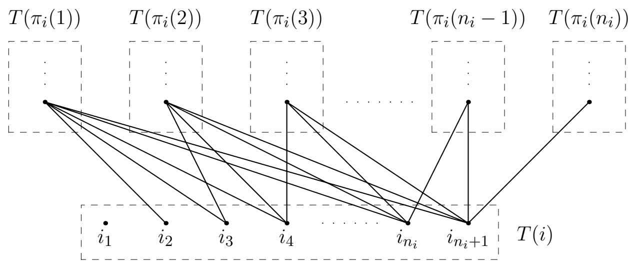

Figure 1: Construction of graph H.

The definitions of $A(i)$ and $D(i)$ allow us to rewrite G as a directed graph $G' = (V, E')$, where the direction is from "ancestors" to "descendants". That is $(j, i) \in E'$ if $j \in A(i)$. Note that the above lemma implies that G' is acyclic. This acyclic structure is helpful to understand the construction and proofs below.

4.2 Construction of Graph H

For each node $i \in V$, define an ordering π_i on all nodes in $A(i)$ according to the non-decreasing order of $v_j + c_{i,j}$. That is, for any $1 \le j < k \le n_i$, $v_{\pi_i(j)} + c_{i,\pi_i(j)} \le v_{\pi_i(k)} + c_{i,\pi_i(k)}$ (ties are broken arbitrarily), where $n_i = |A(i)|$.

We construct an undirected node-weighted graph H from G as follows. For each node $i \in V$, we create $n_i + 1$ isolated vertices $T(i) = \{i_1, i_2, \ldots, i_{n_i+1}\}$ in H. Let $T = \bigcup_{i \in V} T(i)$ be the set of vertices of H. The weight of each vertex in $T(i)$ is defined by

$$\begin{aligned}
w(i_1) &= v_{\pi_i(1)} + c_{i,\pi_i(1)} \\
w(i_2) &= \left(v_{\pi_i(2)} + c_{i,\pi_i(2)}\right) - \left(v_{\pi_i(1)} + c_{i,\pi_i(1)}\right) \\
&\vdots \\
w(i_{n_i}) &= \left(v_{\pi_i(n_i)} + c_{i,\pi_i(n_i)}\right) - \left(v_{\pi_i(n_i-1)} + c_{i,\pi_i(n_i-1)}\right) \\
w(i_{n_i+1}) &= v_i - \left(v_{\pi_i(n_i)} + c_{i,\pi_i(n_i)}\right)
\end{aligned}$$

Note that by the definition of $A(i)$ and π_i, all weights defined above are non-negative. Further, it is easy to see that $\sum_{k \in T_i} w(k) = v_i$.

For each $i_\alpha \in T(i)$, $\alpha = 1, \ldots, n_i$, we connect i_α to all vertices in $T(k)$ for any $k \in A(i)$ preceding $\pi_i(\alpha)$ in the ordering π_i. For the last vertex i_{n_i+1}, we connect it to all vertices in $T(k)$ for all $k \in A(i)$, as shown in Figure 1. That is, i_2 is connected to all vertices in $T(\pi_i(1))$, i_3 is connected to all vertices in $T(\pi_i(1))$ and $T(\pi_i(2))$, etc. It can be seen that for any $\alpha < \beta$, the set of neighbors of i_α is a subset of the neighbors of i_β.

Graph H has the following transitive property.

LEMMA 4.3. *For any node $i, j, k \in V$, where $j \in A(i)$ and $k \in A(j)$, if j_β is connected to i_α, then k_γ is connected to i_α as well, for any $k_\gamma \in T(k)$.*

PROOF. Since $j \in A(i)$ and $k \in A(j)$, we know $k \in A(i)$ by Lemma 4.2 and $\pi_i^{-1}(k) < \pi_i^{-1}(j)$. Since j_β is connected to i_α, we know i_α connects to all vertices preceding j in the ordering of π_i, which includes node k. Hence, k_γ is connected to i_α. $\square$

4.3 Connection between H and the Optimal Pricing Solution

In this subsection, we establish the connection between the maximum independent set of H and the optimal pricing solution of G, which plays the key role of proving Theorem 1.1.

LEMMA 4.4. *The value of the maximum weighted independent set on H is equal to the revenue obtained from the optimal price vector on G.*

The intuition behind the lemma is the following: Depending on which ancestors of node i are included as winners in addition to i, the amount of revenue that can be extracted from i changes. The construction of H, however, ensures that the contribution of i to the total revenue is equal to the contribution of $T(i)$ to the independent set solution. We prove this formally below.

PROOF. *(Prices $\Rightarrow$ IS)* First we show how to construct an independent set solution of H from an optimal pricing solution of G. Let $p_1, \ldots, p_n$ be an optimal price vector of G and $r_1, \ldots, r_n$ be the payment of each user.

Consider any winner i, by Proposition 3.2, we know $r_i = p_i$. If $r_i = v_i$, *i.e.*, i is a price-setter, let $T'(i) = T(i)$. Otherwise, assume p_i is determined according to p_j, *i.e.*, j is a price-setter and $p_i = v_j + c_{i,j} = p_j + c_{i,j}$ (by Proposition 3.3, we know such a j exists). Note that $v_j + c_{i,j} = p_i < v_i$, thus $j \in A(i)$. If there are multiple such price-setters, let j be the one with the smallest index in the ordering π_i. Let $\ell \in A(i)$ be the smallest index such that $\sum_{\alpha=1}^{\ell} w(i_\alpha) = v_j + c_{i,j} = p_i$ (by the construction of H, it can be seen that such an index always exists). In this case, let $T'(i) = \{i_1, \ldots, i_\ell\} \subseteq T(i)$.

For any user i that does not win, define $T'(i) = \emptyset$. We claim that $T' = \bigcup_{i \in V} T'(i)$ defines an independent set on H. Otherwise, there exist $i, k \in V$ and $i_\alpha \in T'(i), k_\gamma \in T'(k)$ such that i_α and k_γ are connected. Assume without loss of generality that $k \in A(i)$. If i is a price-setter, *i.e.*, $p_i = v_i$, then $T'(i) = T(i)$. By Lemma 4.1, we know no users in $A(i)$ are winners, which implies that $T'(k) = \emptyset$, a contradiction. If i is not a price-setter, let $j \in V$ be the price-setter that is used to determine price p_i and $T'(i)$. Similarly, we know $j \in A(i)$ and $T'(j) = T(j)$. There are two possible cases.

`Case 1.` $\pi_i^{-1}(k) > \pi_i^{-1}(j)$. In this case, by the construction of H, there is no edge between $T'(i)$ and $T'(j)$. Each vertex in $T'(i)$ only connects to those vertices in $T'(\ell)$ where $\pi_i^{-1}(\ell) < \pi_i^{-1}(j)$ by construction. In other words, no vertex in $T'(i)$ is connected to $T'(k)$, *i.e.*, i_α and k_γ can not be connected.

`Case 2.` $\pi_i^{-1}(k) < \pi_i^{-1}(j)$. Note that $T'(k) \neq \emptyset$, thus user k is a winner. If k itself is a price-setter, *i.e.*, $p_k = v_k$, then

$$p_k + c_{i,k} = v_k + c_{i,k} \le v_j + c_{i,j} = p_j + c_{i,j} = p_i$$

where the inequality follows from $\pi_i^{-1}(k) < \pi_i^{-1}(j)$. If it is a strict inequality, then user i prefers location k rather than her own location, which contradicts Proposition 3.2. If it is an equality, then both j and k are price-setters of p_i, but it contradicts the smallest index of choosing j when defining $T'(i)$. If k is not a price-setter, we can get a similar contradiction by considering the node that determines p_k and comparing it with node j.

Therefore, T' is an independent set of H and

$$\sum_{k\in T'} w(k) = \sum_{i\in V}\sum_{j\in T'(i)} w(j) = \sum_{i\in V} r_i.$$

Hence, we have shown that for any optimal price vector of G, the revenue is equal to the weight of an independent set of H. □

PROOF. *(IS ⇒ Prices)* In the other direction, let T' be a maximum weighted independent set solution on H. Consider any node $i \in V$; let $T'(i) = T' \cap T(i)$. The basic observation is that if $i_\alpha \in T'(i)$, then $i_\beta \in T'(i)$ for any $\beta \le \alpha$ by the construction of H, since T' is a maximum weighted independent set on H. Define $p_i = \sum_{j\in T'(i)} w(j)$. Note that $p_i \le v_i$, since $\sum_{k\in T(i)} w(k) = v_i$, and $p_i = v_i$ only when $T'(i) = T(i)$. If $T'(i) = \emptyset$, define $p_i = \infty$.

We first consider any node $i \in V$ where $p_i < v_i$. Assume $T'(i) = \{i_1, \ldots, i_\alpha\} \subset T(i)$. Observe that $i_{\alpha+1} \notin T'(i)$ because there is a vertex j_β such that $j_\beta \in T'$ which connects to $i_{\alpha+1}$. However, j_β is not connected to any vertices in $T'(i)$. The only possible such node j is from the "ancestor" set of i, *i.e.*, $j \in A(i)$. We claim that $p_j = v_j$. Otherwise, by the same argument, there is a vertex $k_\gamma \in T'$ such that $k \in A(j)$ (and thus, $k \in A(i)$). By the construction of H, however, it can be seen that i_α connects to k_γ, a contradiction to the independent set solution. Hence, $p_j = v_j$, and by construction of H, $p_i = p_j + c_{i,j}$.

We claim that we can obtain a revenue of p_i from each user i with $T'(i) \neq \emptyset$. Consider any such a node i, and let $k = \arg\min p_j + c_{i,j}$ (if there are multiple choices, let k be the one such that no node in $A(k)$ is a winner). If $p_i \le p_k + c_{i,k}$, i prefers its own location at price p_i. Otherwise, if $p_i > p_k + c_{i,k}$, we consider two cases.

`Case 1.` $p_i = v_i$. By the definition of p_i, we know $T'(i) = T(i)$, *i.e.*, all vertices in $T(i)$ are in the independent set solution T'. If $p_k = v_k$, then $T'(k) = T(k)$ and $v_i > v_k + c_{i,k}$, which implies $k \in A(i)$. However, by the construction of H, $T(i)$ and $T(k)$ can not be both in T', a contradiction. If $p_k < v_k$, by the above argument, we know there is $\ell \in A(k)$ such that $p_\ell = v_\ell$ and $p_k = p_\ell + c_{k,\ell}$. Hence,

$$p_i > p_k + c_{i,k} = p_\ell + c_{k,\ell} + c_{i,k} \ge p_\ell + c_{i,\ell}$$

which contradicts the selection of k.

`Case 2.` $p_i < v_i$. By the above argument, we know there is $j \in A(i)$ such that $p_j = v_j$ and $p_i = p_j + c_{i,j}$. If $p_k = v_k$, then $k \in A(i)$, since $v_k + c_{i,k} = p_k + c_{i,k} < p_i \le v_i$. Furthermore, we know $\pi_i^{-1}(k) < \pi_i^{-1}(j)$, since $v_k + c_{i,k} < p_i = p_j + c_{i,j} = v_j + c_{i,j}$, which implies there is vertex in $T'(i)$ that connects all vertices in $T'(k)$, a contradiction. If $p_k < v_k$, similarly, there is $\ell \in A(k)$ such that $p_\ell = v_\ell$ and $p_k = p_\ell + c_{k,\ell}$. Again, $\ell \in A(i)$ since $v_\ell + c_{i,\ell} \le p_\ell + c_{i,k} + c_{k,\ell} = p_k + c_{i,k} < p_i \le v_i$,

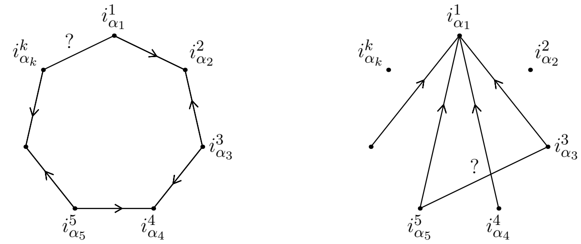

Figure 2: Proof of C_n free and co-C_n free.

and $\pi_i^{-1}(\ell) < \pi_i^{-1}(j)$, since $v_\ell + c_{i,\ell} < p_i = p_j + c_{i,j} = v_j + c_{i,j}$. Hence, we can get the same contradiction.

Therefore, the value of the maximum weight independent set of H is equal to the revenue obtained from a pricing solution of G. □

4.4 H is a Perfect Graph

Let C_n be a cycle composed of n nodes, and co-C_n be the complement of C_n. A subgraph G' of a graph G is said to be *induced* if, for any pair of vertices i and j of G', (i, j) is an edge of G' if and only if (i, j) is an edge of G. A graph is called a *Berge* graph if it does not contain C_n and co-C_n as an induced subgraph, for $n = 5, 7, 9, \ldots$.

LEMMA 4.5. *H is a Berge graph.*

PROOF. For the purpose of this proof, we define a direction on edges in H as follows: For any edge (j_β, i_α) in H, where $j_\beta \in T(j)$ and $i_\alpha \in T(i)$, assume $j \in A(i)$. Let the direction be $j_\beta \to i_\alpha$. That is, the direction is similar to G' defined in Section 4.1, *i.e.*, from "ancestors" to "descendants".

(*Cycles*). First we show that no induced subgraph of H is a cycle C_n for $n = 5, 7, 9, \cdots$. Suppose there is such a cycle on a subset of vertices $\{i^1_{\alpha_1}, i^2_{\alpha_2}, \ldots, i^k_{\alpha_k}\}$, where $k \ge 5$ is odd. Assume without loss of generality that we have a direction $i^1_{\alpha_1} \to i^2_{\alpha_2}$. Now consider the link between $i^2_{\alpha_2}$ and $i^3_{\alpha_3}$. We claim the direction must be $i^3_{\alpha_3} \to i^2_{\alpha_2}$. Otherwise, since $i^1 \in A(i^2)$ and $i^2 \in A(i^3)$, then there must also be an edge between $i^1_{\alpha_1}$ and $i^3_{\alpha_3}$ by Lemma 4.3. Proceeding this way, as Figure 2 shows, there is a contradiction for the direction of the link between $i^k_{\alpha_k}$ and $i^1_{\alpha_1}$, since k is odd and at least 5.

(*Co-cycles*). Next we show that no induced subgraph of H is co-C_n, for $n = 5, 7, 9, \cdots$. Note that $C_5 =$ co-C_5, thus it suffices to consider the case where $n \ge 7$. Suppose there is such a subgraph on a subset of vertices $\{i^1_{\alpha_1}, i^2_{\alpha_2}, \ldots, i^k_{\alpha_k}\}$, where the vertices are numbered so that the missing edges are $(i^1_{\alpha_1}, i^2_{\alpha_2}), \ldots, (i^{k-1}_{\alpha_{k-1}}, i^k_{\alpha_k}), (i^k_{\alpha_k}, i^1_{\alpha_1})$. Without loss of generality, assume that we have a direction $i^3_{\alpha_3} \to i^1_{\alpha_1}$. The direction between $i^4_{\alpha_4}$ and $i^1_{\alpha_1}$ must also be $i^4_{\alpha_4} \to i^1_{\alpha_1}$, since otherwise by Lemma 4.3 there must be an edge from $i^3_{\alpha_3}$ to $i^4_{\alpha_4}$. The same argument applies to all vertices in $i^4_{\alpha_4}, \ldots, i^{k-1}_{\alpha_{k-1}}$, as Figure 2 shows. That is, if vertex $i^1_{\alpha_1}$ has one incoming edge, all edges incident to $i^1_{\alpha_1}$ must be incoming as well. The same argument applies to all other vertices, *i.e.*, all edges incident to a vertex point to the same direction. This gives us a contradiction: consider for example $i^3_{\alpha_3}$ and $i^5_{\alpha_5}$. Since we have direction $i^3_{\alpha_3} \to i^1_{\alpha_1}$ and $i^5_{\alpha_5} \to i^1_{\alpha_1}$, all edges incident to *both* $i^3_{\alpha_3}$ and $i^5_{\alpha_5}$ must be directed outward. This leads to a contradiction for the edge between $i^3_{\alpha_3}$ and $i^5_{\alpha_5}$. Thus H cannot contain co-C_n as an induced subgraph for $n \geq 7$.

Therefore we have shown that H can not contain C_n or co-C_n as an induced subgraph for any $n = 5, 7, 9, \cdots$, so that H is a Berge graph. $\square$

A graph is said to be *perfect* if the chromatic number (*i.e.*, the least number of colors needed to color the graph) of every induced subgraph equals the clique number of that subgraph. By the seminal Strong Perfect Graph Theorem [8], which states that a graph is a perfect graph if and only if it is a Berge graph, we have the following corollary:

Corollary 4.1. *H is a perfect graph.*

We know the maximum weighted independent set problem can be solved in polynomial time on perfect graphs [12]. Combining all of these, we know the optimal price vector on G can be computed in polynomial time, which completes the proof of Theorem 1.1.

5. ALGORITHMIC EXTENSIONS

Our algorithm continues to work for a few more general settings. One generalization is that the seller incurs a fixed cost Δ to produce each copy of the item. This extension can be solved in polynomial time — remove all users with value less than Δ and define the value of remaining users to be $v_i - \Delta$. The optimal revenue obtained from the resulting instance is equal to that from the original input.

Another important generalization is when there are multiple users with (possibly) different values at each node: Given a graph $G = (V, E)$, for each node $i \in V$, there is a set A_i of users, with values $v_{i1}, \ldots, v_{ia_i}$, where $a_i = |A_i|$. The commute costs depend only on the locations, and not on the users. (We defer the proof to the full version of the paper.)

Theorem 5.1. *Given a graph $G = (V, E)$ and a set of users A_i at each node $i \in V$, the revenue maximization problem can be solved in polynomial time as well.*

6. BEYOND THE TRIANGLE INEQUALITY

Triangle inequality in the distance space c is crucial in obtaining our polynomial time algorithm above. As we will see below, even a minor relaxation of this condition makes the problem NP-hard. Before presenting our hardness results, we comment that an $O(\log n)$-approximation algorithm is well known for the general unit demand envy-free pricing [13]. In our setting, a simple optimal fixed-price algorithm gives an $O(\log n)$-approximation as well.

Theorem 6.1. *The revenue maximization problem is NP-hard when the substitution costs $c_{i,j}$ do not satisfy triangle inequality, even if they satisfy a $(1+\epsilon)$-approximate triangle inequality for any given $\epsilon > 0$.*

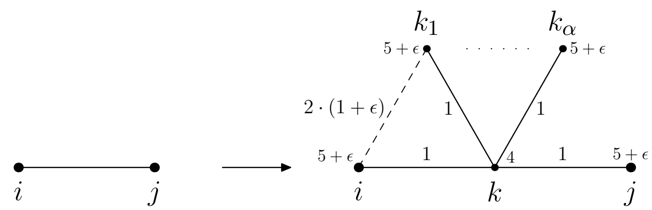

Figure 3: Construction of graph G'.

Proof. The reduction is similar to that established in [13]. We reduce from vertex cover: Given a graph $G = (V, E)$, we are asked if there is a subset $S \subseteq V$ of size k such that S covers all edges in E. Let $n = |V|$ and $m = |E|$. Let N be the input size of G and $\alpha = 4N/\epsilon$.

For any $\epsilon > 0$, we construct a graph $G' = (V', E')$ from G by adding an extra node on each edge in E. That is, for any $(i, j) \in E$, we add a node k and replace edge (i, j) by two edges (i, k) and (k, j). For each $i \in V$ (corresponding to a node in G), let its value $v_i = 5 + \epsilon$. For each $k \in E$ (corresponding to an edge in G), let its value $v_k = 4$. Further, we add a set of extra nodes $T_k = \{k_1, \ldots, k_\alpha\}$ with value $v_{k_\ell} = 5 + \epsilon$ each, and connect them to k. Denote the resulting graph by G'. For any edge $(i, j) \in E'$, define $c_{i,j} = 1$. For any $(i, j) \notin E'$, let its commute cost $c_{i,j}$ be $(1 + \epsilon)$ times the shortest path distance between i and j in G'. (For instance, $c_{i,k_1} = 2 \cdot (1 + \epsilon)$ as Figure 3 shows.) It can be seen the distance space, $\{c_{i,j}\}$, defined above satisfies $(1 + \epsilon)$-approximate triangle inequality.

If G has a vertex cover solution S, we define a pricing solution of G' as follows: Let $p_i = 3$ for each $i \in S$, $p_i = 5+\epsilon$ for each $V \setminus S$, $p_k = \infty$ for each edge $k \in E$ and $p_{k_\ell} = 5 + \epsilon$ for each $k_\ell \in T_k$. By the commute costs defined above, every node in V will buy an item from its own location, which induces a revenue of $3\cdot|S|+(5+\epsilon)\cdot(n-|S|) = (5+\epsilon)n-(2+\epsilon)|S|$. For each edge $k \in E$, since S is a vertex cover of G, there is a neighbor i of k such that $p_i = 3$. Thus, k can buy an item from i, which contributes 3 to the total revenue. In addition, each node $k_\ell \in T_k$ wins from its own location, which induces a revenue of $5 + \epsilon$. Therefore, the total revenue is $(5 + \epsilon)m\alpha + 3m + (5 + \epsilon)n - (2 + \epsilon)|S|$.

On the other hand, consider any optimal pricing solution p of G'. Assume without loss of generality that $p_i \leq 5 + \epsilon$, for any $i \in V'$. In other words, all nodes in V and T_k, for $k \in E$, are winners. Note that a trivial solution where set price to be $5 + \epsilon$ at all nodes gives a total revenue of $\Delta \triangleq (5 + \epsilon)(m\alpha + n)$. Thus, the revenue generated by p is at least Δ. We first argue that all edges in E must be winners. Otherwise, assume there is $k \in E$ such that k does not buy an item. Consider a neighbor $k_1 \in T_k$ of k. We know $p_{k_1} > 3$. It can be seen that by reducing the price of k_1 to 3 (this will not affect any other nodes except k and k_1), k becomes a winner and the total revenue will be increased by at least $2 \cdot 3 - (5 + \epsilon) > 0$, where the second term $5 + \epsilon$ is the upper bound of the revenue that can be extracted from k_1, which contradicts to the optimality of p. Secondly, it is safe to assume $p_k > 4$ for any $k \in E$. This is because, if $p_k \leq 4$ for some $k \in E$, then the total revenue we extract from T_k is at most 5α. In this case, even if we can extract full revenue from all other nodes, the total revenue is at most $(5+\epsilon)(m-1)\alpha + 5\alpha + 4m + (5+\epsilon)n < \Delta$, a contradiction. Therefore, each edge $k \in E$ wins an item from one of its neighbors. If $k \in E$ wins from a neighbor $k_\ell \in T_k$, then $p_{k_\ell} \leq 3$. In this case, by define $p_{k_\ell} = 5+\epsilon$ and $p_i = 3$, where $i \in V$ is a neighbor of k in V, k wins form i and the total revenue will not decrease. Therefore, without loss of generality, we can assume each $k \in E$ wins from one of its neighbors in V. Let

$$S = \{i \in V \mid \exists\, k \in E \text{ s.t. } k \text{ wins from } i\}$$

By the above argument, we know S is a vertex cover solution of G and $p_i \leq 3$ for each $i \in S$. Thus, the total revenue is at most $(5+\epsilon)m\alpha + 3m + (5+\epsilon)n - (2+\epsilon)|S|$.

Therefore, computing an optimal pricing solution is equivalent to finding an optimal vertex cover, which yields the desired NP-hardness result. $\Box$

Given appropriate complexity assumptions, we can show the problem is hard to approximate within a ratio of $O(\log^\epsilon n)$ for some $\epsilon > 0$ without triangle inequality. (We defer the proof to the full version of the paper.)

THEOREM 6.2. *Without triangle inequality, the problem is hard to approximate within a ratio of $O(\log^\epsilon n)$ for some $\epsilon > 0$, given the hardness of the balanced bipartite independent set problem in constant-degree graphs or refuting random 3CNF formulas.*

7. MULTI-UNIT DEMAND

We next consider the multi-unit demand setting, where users can have positive marginal value for more than one item. Specifically, let $G = (V, E)$ be a graph with a user at each location and commute costs $c_{i,j}$ as before. Each user $i \in V$ now has a demand $d_i \geq 1$, which is the maximum number of items she is interested in. Her value for x copies, $x = 0, 1, \ldots, d_i$, is now characterized by a *value function* $f_i : \{0, 1, \ldots, d_i\} \to \mathbb{R}^+$, where $f_i(x)$ is the maximum value that i is willing to pay for x items. We assume that the value functions satisfy $f_i(0) = 0$ and are non-decreasing for each $i \in V$.

In the multi-unit demand setting, when a user buys items from a different location, we assume the commute cost is paid only once, *i.e.*, it is independent of the number of items bought from that location[2]. Hence, when user i buys x items from location j at price p_j, her utility is $f_i(x) - xp_j - c_{i,j}$. Given a price vector, each user decides how many copies to buy from each location rationally, *i.e.*, to maximize her utility. It is easy to see that for any pricing solution p and user $i \in V$, we can assume that i either does not buy any item or buys items from only one location. Again, given the underlying graph and value functions, the goal is to decide a per-unit price for each location to maximize the total revenue.

[2]When the commute cost is paid once per unit, the problem can be solved efficiently for linear value functions (*i.e.*, constant marginal value per unit), by a simple modification of the algorithm described above. This is quite different from the setting we are considering, as we will see in Theorem 7.1.

There are some important differences between the unit demand setting and the multi-unit demand setting. Specifically, with multi-unit demands, Proposition 3.1 and therefore Proposition 3.2 do not hold, *i.e.*, it is no longer true that in an optimal price vector, every user who buys an item buys it from her own location. Also note that the number of items that i buys for a given price vector need not match her demand, *i.e.*, when user i buys x copies from location j, we can have $x < d_i$ if the marginal value of the $(x+1)$-th copy, $f_i(x+1) - f_i(x)$, is smaller than the per-unit price p_j at location j. Of course, when the marginal values are constant, which corresponds to a linear value function $f_i(x) = x \cdot v_i$ (where v_i, as in the unit demand setting, is the maximum value that i is willing to pay for one unit) for $0 \leq x \leq d_i$, each user buys either nothing or d_i copies for any price vector.

Another unusual property the multi-unit demand setting possesses is a paradoxical non-monotonicity of the optimal revenue w.r.t the demands: When the demands of some users increase, the revenue generated by the optimal solution can decrease, even when all value functions are linear.

EXAMPLE 7.1 (DEMAND PARADOX). *There are three nodes i, j, k, with costs $c_{i,j} = 50$, $c_{i,k} = 100$ and $c_{j,k} = 50$. For user i, let $d_i = 10$, $v_i = 10$; for user j, let $d_j = 10$, $v_j = 6$; and for user k, let $d_k = 100$, $v_j = 1$. In addition, we add many isolated nodes (with demand 1 and value 20 each) and connect them to i with cost 0. Note that when the number of these isolated nodes is large enough, the price at i is always at least 20. In the optimal solution, $p_j = 6$ and $p_k = 1$. User i buys from location j with revenue $10 \cdot 6 = 60$ and j buys from its own location with revenue $10 \cdot 6 = 60$. However, when the demand of i is increased to 20, the optimal solution is $p_j = 3.5$ and $p_k = 1$, and the revenue obtained from i and j is $30 \cdot 3.5 = 105 < 120$. (The revenue from other users is unchanged.)*

The following result shows that the problem with multi-unit demand, unlike the unit demand case, is hard to solve. (We defer the proof to the full version of the paper.)

THEOREM 7.1. *In the multi-unit demand setting, the revenue maximization problem is APX-hard even when all value functions are linear and demands are at most 5.*

Now we will study approximation algorithms for the multi-unit demand problem.

7.1 Linear Value Functions

We first give an approximation algorithm for the linear case, *i.e.*, $f_i(x) = x \cdot v_i$. We will define an instance of the input by $G(v, d)$, where $G = (V, E)$ refers to the underlying weighted graph, v refers to the vector of per-unit values of users in V (this is well-defined since f_i's are linear), and d is the vector of demands for these users. We will use $OPT(G(v,d))$ to denote the optimal revenue for this instance, and $R(G(v,d),p)$ to denote the revenue obtained from the instance $G(v,d)$ with a price vector p.

ALGORITHM: *Given an input $G(v,d)$, let D be value of the largest demand in d (more precisely, we can define D to be the ratio of the largest to smallest demand). For each $\alpha = 1,2,\ldots,\log(D)$, define $S^\alpha = \{i \in V \mid 2^{\alpha-1} \le d_i < 2^\alpha\}$. For each $\alpha = 1,2,\ldots,\log(D)$, define a new instance $G^\alpha(v^\alpha,d^\alpha)$ from the underlying graph G, where only users in S^α are available and the modified values v^α and demands d^α are as follows: for $i \in S^\alpha$, the per-unit value is v_i and demand is 2^α. Solve the instances $G^\alpha(v^\alpha,d^\alpha)$ for all α, and let the corresponding optimal price vector be p^α. Return the price vector given by $\max_\alpha R(G(v,d),p^\alpha)$.*

We will prove the following theorem.

THEOREM 7.2. *The above algorithm runs in polynomial time and gives an $O(\log D)$-approximation to the revenue maximization problem, where D is the upper bound of demands.*

The claim about the polynomial running time of the algorithm follows immediately from the following lemma.

LEMMA 7.1. *When all demands are same, the optimal pricing problem with multi-unit demand can be solved in polynomial time.*

PROOF. Let $G(v,d)$ be a given instance where $d_i = d$ for all users. Construct another instance G' from G with values $v'_i = d \cdot v_i$ and $d'_i = 1$. By the algorithm established in the previous section, we know the optimal price vector p' for G' can be computed in polynomial time. Consider a solution $p = p'/d$ (i.e. $p_j = p'_j/d$ for any location j) for G. If i wins (from location i, by Proposition 3.2) in G', we know (i) $v'_i \ge p'_i \Rightarrow v_i \ge p_i$; and (ii) $v'_i - p'_i \ge v'_i - p'_j - c_{i,j}$ for any $j \in V$, which implies $d \cdot (v_i - p_i) \ge d \cdot (v_i - p_j) - c_{i,j}$, *i.e.*, user i still wants to buy all items from its own location in G. So $R(G,p) \ge R(G',p')$. Conversely, let p^* be an optimal price vector for $G(v,d)$. Let $q^* = d \cdot p^*$ (*i.e.*, $q^*_i = d \cdot p^*_i$ for any $i \in V$) be a pricing solution defined on G'. If i wins from location j in G, we know (i) $d \cdot (v_i - p^*_j) - c_{i,j} \ge 0 \Rightarrow v'_i - q^*_j - c_{i,j} \ge 0$, and (ii) $d \cdot (v_i - p^*_j) - c_{i,j} \ge d \cdot (v_i - p^*_k) - c_{i,k}$ for any $k \in V$, which implies that $v'_i - q^*_j - c_{i,j} \ge v'_i - q^*_k - c_{i,k}$, *i.e.*, user i still wants to buy the item from location j in G'. Again, $R(G',q^*) \ge R(G,p^*)$.

Therefore, $R(G',q^*) \ge R(G,p^*) \ge R(G,p) \ge R(G',p')$. By the optimality of p' on G', we know all inequalities are tight. Hence, the price vector p defined above is optimal for G. $\square$

In the remainder of this subsection, we analyze the approximation ratio of the algorithm. Consider any given instance $G(v,d)$, and let $d_{\min} = \min_{i\in V} d_i$ and $d_{\max} = \max_{i\in V} d_i$. Let $\lambda = \frac{d_{\max}}{d_{\min}}$. Define a modified instance $G(v,\mathbf{d}_{\max})$ from $G(v,d)$, with the same value vector v but with demand $d_{\max}$ for each user.

First we prove two lemmas that will be used in the proof.

LEMMA 7.2. *$R(G(v,d),p) \ge \frac{1}{\lambda} OPT(G(v,\mathbf{d}_{\max}))$, where p is the optimal solution of $G(v,\mathbf{d}_{\max})$.*

PROOF. First we show that when prices p are used with demands d, each user continues to buy from her own location. Let S be the set of winners in the instance $G(v,\mathbf{d}_{\max})$ given solution p. Note that since all demands are equal, every winner buys at its own location. Therefore, for all $i \in S$, $p_i \le v_i$, so that i can still afford the item at its own location in the instance $G(v,d)$. Further,

$$d_{\max} \cdot (v_i - p_i) \ge d_{\max} \cdot (v_i - p_j) - c_{i,j}$$

for any location $j \in S$. Hence,

$$p_i \le p_j + \frac{c_{ij}}{d_{\max}} \le p_j + \frac{c_{ij}}{d_i} \Rightarrow d_i \cdot (v_i - p_i) \ge d_i \cdot (v_i - p_j) - c_{i,j}.$$

Hence, i still prefers to buy from its own location in the instance $G(v,d)$. Therefore the revenue extracted by price vector p is

$$\begin{aligned} R(G(v,d),p) &\ge \sum_{i\in S} d_i \cdot p_i \\ &\ge \frac{d_{\min}}{d_{\max}} \sum_{i\in S} d_{\max} \cdot p_i \\ &= \frac{1}{\lambda} OPT(G(v,\mathbf{d}_{\max})). \end{aligned}$$

$\square$

Next we relate the optimal revenues with demand vectors d and $\mathbf{d}_{\max}$. Note that the non-monotonicity in Example 7.1 implies that the factor $1/\lambda$ in the statement below cannot be increased to one.

LEMMA 7.3. *$OPT(G(v,\mathbf{d}_{\max})) \ge \frac{1}{\lambda} OPT(G(v,d))$.*

PROOF. Let p^* be the optimal solution for $G(v,d)$ and S be the set of winners. Let $\pi(i)$ denote the location where i buys the items for any $i \in S$. Note that $\pi(i)$ need not be equal to i, but i will always buy all of its items from one location.

Consider the price vector $q = \frac{p^*}{\lambda}$ for instance $G(v,\mathbf{d}_{\max})$. Since $\lambda \ge 1$, for any $i \in S$, we have $q_{\pi(i)} \le p^*_{\pi(i)} \le v_i$ which implies

$$d_{\max} \cdot \left(v_i - q_{\pi(i)}\right) - c_{i,\pi(i)} \ge d_i \cdot \left(v_i - p^*_{\pi(i)}\right) - c_{i,\pi(i)} \ge 0.$$

That is, i can still afford to buy at location $\pi(i)$ for the instance $G(v,\mathbf{d}_{\max})$ given solution q.

Further, for any $j \in V$, we have

$$d_i \cdot \left(v_i - p^*_{\pi(i)}\right) - c_{i,\pi(i)} \ge d_i \cdot \left(v_i - p^*_j\right) - c_{i,j} \qquad (1)$$

which implies that

$$\lambda q_{\pi(i)} - \lambda q_j = p^*_{\pi(i)} - p^*_j \le \frac{c_{i,j} - c_{i,\pi(i)}}{d_i}.$$

There are two cases.

- If $c_{i,j} \geq c_{i,\pi(i)}$, then
$$q_{\pi(i)} - q_j \leq \frac{c_{i,j} - c_{i,\pi(i)}}{\lambda d_i} = \frac{c_{i,j} - c_{i,\pi(i)}}{\frac{d_{\max}}{d_{\min}} d_i} \leq \frac{c_{i,j} - c_{i,\pi(i)}}{d_{\max}}$$
which implies that
$$d_{\max} \cdot \left(v_i - q_{\pi(i)}\right) - c_{i,\pi(i)} \geq d_{\max} \cdot (v_i - q_j) - c_{i,j}.$$
That is, i will still prefer to buy from location $\pi(i)$.
- If $c_{i,j} < c_{i,\pi(i)}$, according to (1), we know $p_j^* \geq p_{\pi(i)}^*$. In this case, even if i prefers to buy from location j, the price it pays is $q_j \geq q_{\pi(i)}$.

Hence, in both cases, the revenue we obtain from i is at least $d_{\max} \cdot q_{\pi(i)}$.

Therefore, the revenue extracted with demands $\mathbf{d}_{\max}$ and price vector q is
$$\begin{aligned} R(G(v, \mathbf{d}_{\max}), q) &\geq \sum_{i \in S} d_{\max} \cdot q_{\pi(i)} \\ &= \sum_{i \in S} d_{\max} \cdot p_{\pi(i)}^* \frac{d_{\min}}{d_{\max}} \\ &= \sum_{i \in S} d_{\min} \cdot p_{\pi(i)}^* \\ &\geq \frac{1}{\lambda} OPT(G(v, d)) \end{aligned}$$
and since $OPT\left(G(v, \mathbf{d}_{\max})\right) \geq R\left(G(v, \mathbf{d}_{\max}), q\right)$, we are done. $\square$

Combining Lemmas 7.2 and 7.3, we obtain the following result.

PROPOSITION 7.1. *$R(G(v, d), p) \geq \frac{1}{\lambda^2} OPT(G(v, d))$, where p is the optimal solution of $G(v, \mathbf{d}_{\max})$.*

We are now ready to finish the proof of Theorem 7.2.

PROOF OF THEOREM 7.2. The claim about polynomial running time follows from Lemma 7.1. The claim about the approximation ratio follows from the result above, from the following sequence of inequalities. (Recall that p^α is the optimal price vector for the instance $G^\alpha(v^\alpha, d^\alpha)$, where $d_i^\alpha = 2^\alpha$.)
$$\begin{aligned} OPT(G(v, d)) &\leq \sum_{\alpha=1}^{\log D} OPT(G^\alpha(v, d)) \\ &\leq 4 \sum_{\alpha=1}^{\log D} R(G^\alpha(v, d), p^\alpha) \\ &\leq (4 \log D) \cdot \max_\alpha R(G^\alpha(v, d), p^\alpha) \\ &\leq (4 \log D) \cdot R(G(v, d), p^\alpha) \end{aligned}$$
where the first inequality follows by using the price vector corresponding to $OPT(G(v, d))$ for each $G^\alpha(v, d)$, and the second inequality is exactly the result in Proposition 7.1. $\square$

7.2 General Value Functions

In this subsection we will describe an $O(\log^3 D)$-approximation algorithm for general value functions. The algorithm is simple: Given an instance G, for any user i, let $v_i = \frac{f_i(d_i)}{d_i}$, where d_i is the (maximum) demand of i. Construct an instance $G(v, d)$ from G with linear value functions where the demand and per-unit value of each user i are d_i and v_i, respectively. We compute an approximately optimal price vector for $OPT(G(v, d))$ by Theorem 7.2 and return it as the solution for G. (We defer the proof to the full version of the paper.)

THEOREM 7.3. *The above algorithm gives an $O(\log^3 D)$-approximation to the revenue maximization problem for general value functions.*

8. ACKNOWLEDGEMENTS

We thank Andrei Broder, Anna Karlin, Ravi Kumar, Mohammad Mahdian and David Reiley for helpful discussions. We are especially grateful to Anna Karlin and Ravi Kumar for their insightful suggestions, which much improved the paper. We also thank Jason Hartline for pointing out some related work.

9. REFERENCES

[1] G. Aggarwal, T. Feder, R. Motwani, A. Zhu, *Algorithms for Multi-Product Pricing*, ICALP 2004, 72-83.

[2] M. F. Balcan, A. Blum, *Approximation Algorithms and Online Mechanisms for Item Pricing*, EC 2006, 29-35.

[3] P. Briest, *Towards Hardness of Envy-Free Pricing*, ECCC, 2006.

[4] P. Briest, P. Krysta, *Single-Minded Unlimited Supply Pricing on Sparse Instances*, SODA 2006, 1093-1102.

[5] P. Briest, P. Krysta, *Buying Cheap is Expensive: Hardness of Non-Parametric Multi-Product Pricing*, SODA 2007, 716-725.

[6] S. Chawla, J. D. Hartline, R. D. Kleinberg, *Algorithmic Pricing via Virtual Valuations*, EC 2007, 243-251.

[7] N. Chen, A. R. Karlin, *Cheap Labor Can Be Expensive*, SODA 2007, 707-715.

[8] M. Chudnovsky, N. Robertson, P. Seymour, R. Thomas, *The Strong Perfect Graph Theorem*, Annals of Mathematics, V.164, 51-229, 2006.

[9] E. D. Demaine, M. T. Hajiaghayi, U. Feige, M. R. Salavatipour, *Combination Can be Hard: Approximability of the Unique Coverage Problem*, SODA 2006, 162-171.

[10] X. Deng, C. H. Papadimitriou, S. Safra, *On the Complexity of Price Equilibria*, JCSS, V.67(2), 311-324, 2003.

[11] N. R. Devanur, C. H. Papadimitriou, A. Saberi, V. V. Vazirani, *Market Equilibrium via a Primal-Dual-Type Algorithm*, to appear in JACM.

[12] M. Grötschel, L. Lovász, A. Schrijver, *The Ellipsoid Method and its Consequences in Combinatorial Optimization*, Combinatorica, V.1, 169-197, 1981.

[13] V. Guruswami, J. D. Hartline, A. R. Karlin, D. Kempe, C. Kenyon, F. McSherry, *On Profit-Maximizing Envy-Free Pricing*, SODA 2005, 1164-1173.

[14] T. Roughgarden, É. Tardos, *How Bad is Selfish Routing?*, JACM, V.49(2), 236-259, 2002.

[15] P. Rusmevichientong, B. Van Roy, P. W. Glynn, *A Non-Parametric Approach to Multi-Product Pricing*, Operations Research, V.54(1), 82-98, 2006.